sensors

Volume 1

UAV or Drones for Remote Sensing Applications

Edited by
Felipe Gonzalez Toro and Antonios Tsourdos

Printed Edition of the Special Issue Published in *Sensors*

www.mdpi.com/journal/sensors

MDPI

UAV or Drones for Remote Sensing Applications

UAV or Drones for Remote Sensing Applications

Volume 1

Special Issue Editors

Felipe Gonzalez Toro
Antonios Tsourdos

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade

Special Issue Editors
Felipe Gonzalez Toro
Queensland University of Technology
Australia

Antonios Tsourdos
Cranfield University
UK

Editorial Office
MDPI
St. Alban-Anlage 66
Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Sensors* (ISSN 1424-8220) from 2017 to 2018 (available at: http://www.mdpi.com/journal/sensors/special_issues/UAV_drones_remote_sensing)

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name* **Year**, *Article Number*, Page Range.

Volume 1
ISBN 978-3-03897-091-0 (Pbk)
ISBN 978-3-03897-092-7 (PDF)

Volume 1–2
ISBN 978-3-03897-113-9 (Pbk)
ISBN 978-3-03897-114-6 (PDF)

Contents

About the Special Issue Editors

Felipe Gonzalez Toro, Associate Professor at the Science and Engineering Faculty, Queensland University of Technology (Australia), with a passion for innovation in the fields of aerial robotics and automation and remote sensing. He creates and uses aerial robots, drones or UAVs that possess a high level of cognition using efficient on-board computer algorithms and advanced optimization and game theory approaches that assist us to understand and improve our physical and natural world. Dr. Gonzalez leads the UAVs-based remote sensing research at QUT. As of 2017, he has published nearly 120 peer reviewed papers. To date, Dr. Gonzalez has been awarded $10.1M in chief investigator/partner investigator grants. This grant income represents a mixture of sole investigator funding, international, multidisciplinary collaborative grants and funding from industry. He is also a Chartered Professional Engineer, Engineers Australia—National Professional Engineers Register (NPER), a member of the Royal Aeronautical Society (RAeS), The IEEE, American Institute of Aeronautics and Astronautics (AIAA) and holder of a current Australian Private Pilot Licence (CASA PPL).

Antonios Tsourdos obtained a MEng on Electronic, Control and Systems Engineering, from the University of Sheffield (1995), an MSc on Systems Engineering from Cardiff University (1996) and a PhD on Nonlinear Robust Autopilot Design and Analysis from Cranfield University (1999). He joined the Cranfield University in 1999 as lecturer, was appointed Head of the Centre of Autonomous and Cyber-Physical Systems in 2007 and Professor of Autonomous Systems and Control in 2009 and Director of Research—Aerospace, Transport and Manufacturing in 2015. Professor Tsourdos was a member of the Team Stellar, the winning team for the UK MoD Grand Challenge (2008) and the IET Innovation Award (Category Team, 2009). Professor Tsourdos is an editorial board member of: Proceedings of the IMechE Part G: *Journal of Aerospace Engineering, IEEE Transactions of Aerospace and Electronic Systems, Aerospace Science & Technology, International Journal of Systems Science, Systems Science & Control Engineering* and the *International Journal of Aeronautical and Space Sciences*. Professor Tsourdos is Chair of the IFAC Technical Committee on Aerospace Control, a member of the IFAC Technical Committee on Networked Systems, Discrete Event and Hybrid Systems, and Intelligent Autonomous Vehicles. Professor Tsourdos is also a member of the AIAA Technical Committee on Guidance, Control and Navigation; AIAA Unmanned Systems Program Committee; IEEE Control System Society Technical Committee on Aerospace Control (TCAC) and IET Robotics & Mechatronics Executive Team.

Preface to "UAV or Drones for Remote Sensing Applications"

The rapid development and growth of unmanned aerial vehicles (UAVs) as a remote sensing platform, as well as advances in the miniaturization of instrumentation and data systems, have resulted in an increasing uptake of this technology in the environmental and remote sensing science communities. Although tough regulations across the globe may still limit the broader use of UAVs, their use in precision agriculture, ecology, atmospheric research, disaster response biosecurity, ecological and reef monitoring, forestry, fire monitoring, quick response measurements for emergency disaster, Earth science research, volcanic gas sampling, monitoring of gas pipelines, mining plumes, humanitarian observations and biological/chemo-sensing tasks continues to increase.

This Special Issue provides a forum for high-quality peer-reviewed papers that broaden the awareness and understanding of UAV developments, applications of UAVs for remote sensing, and associated developments in sensor technology, data processing and communications, and UAV system design and sensing capabilities.

This topic encompasses many algorithms and process flows and tools, including: robust vehicle detection in aerial images based on cascaded convolutional neural networks; a stereo dual-channel dynamic programming algorithm for UAV image stitching, as well as seamline determination based on PKGC segmentation for remote sensing image mosaicking; the implementation of an IMU-aided image stacking algorithm in digital cameras; the study of multispectral characteristics at different observation angles, rapid three-dimensional reconstruction for image sequence acquired from UAV cameras; comparisons of Riegl Ricopter UAV Lidar-derived canopy height and DBH with terrestrial Lidar; vision-based target finding and inspection of a ground target using a multirotor UAV system; a localization framework for real-time UAV autonomous landing using an on-ground deployed visual approach; curvature continuous and bounded path planning for fixed-wing UAVs; the calculation and identification of the aerodynamic parameters for small-scaled fixed-wing UAVs

Several wildfire and agricultural applications of UAVS including: deep learning-based wildfire identification in UAV imagery; postfire vegetation survey campaigns; secure utilization of beacons; and UAVS used in emergency response systems for building fire hazards; observing spring and fall phenology in a deciduous forest with aerial drone imagery; the design and testing of a UAV mapping system for agricultural field surveying; artificial neural network to predict vine water status spatial variability using multispectral information obtained from an unmanned aerial vehicle; automatic hotspot and sun glint detection in UAV multispectral images obtained via uncooled thermal camera calibration and optimization of the photogrammetry process for UAV applications in agriculture; olive yield forecast tool based on the tree canopy geometry using UAS imagery; spatial scale gap filling downscaling method for applications in precision agriculture; automatic co-registration algorithm to remove canopy shaded pixels in UAV-borne thermal images to improve the estimation of crop water stress on vineyards; methodologies for improving plant pest surveillance in vineyards and crops using UAV-based hyperspectral and spatial data; UAV-assisted dynamic clustering of wireless sensors and networks for crop health monitoring.

Several applications of UAVS in the fields of environment and conservation including the following: the automatic detection of pre-existing termite mounds through UAS and hyperspectral imagery; aerial mapping of forests affected by pathogens using UAVs; hyperspectral sensors and artificial i ntelligence; c oral r eef and coral bleaching monitoring; invasive grass and vegetation surveys in remote arid lands. UAVs are also utilized in many other applications: vicarious calibration of SUAS microbolometer temperature imagery for the estimation of radiometric land surface temperature; the documentation of hiking trails in alpine areas; the detection of nuclear sources by UAV teleoperation using a visuo-haptic augmented reality interface; the design of a UAV-embedded microphone array system for sound source localization in outdoor environments; the monitoring of concentrated solar power plants, accuracy analysis of a dam model from drone surveys, mobile sensing and actuation infrastructure, UAV-based frameworks for river hydromorphological characterization; online aerial terrain mapping for ground robot navigation.

Felipe Gonzalez Toro, Antonios Tsourdos
Special Issue Editors

sensors

MDPI

Article

Detection of Nuclear Sources by UAV Teleoperation Using a Visuo-Haptic Augmented Reality Interface

Jacopo Aleotti [1], Giorgio Micconi [1], Stefano Caselli [1], Giacomo Benassi [2], Nicola Zambelli [2], Manuele Bettelli [3] and Andrea Zappettini [3,*]

[1] Department of Engineering and Architecture, University of Parma, 43124 Parma, Italy; jacopo.aleotti@unipr.it (J.A.); micconi@ce.unipr.it (G.M.); caselli@ce.unipr.it (S.C.)
[2] due2lab s.r.l., 43121 Parma, Italy; benassi@due2lab.com (G.B.); zambelli@due2lab.com (N.Z.)
[3] IMEM-CNR, 43124 Parma, Italy; manuele.bettelli@imem.cnr.it
* Correspondence: andrea.zappettini@imem.cnr.it; Tel.: +39-0521-269-296

Received: 19 July 2017; Accepted: 15 September 2017; Published: 29 September 2017

Abstract: A visuo-haptic augmented reality (VHAR) interface is presented enabling an operator to teleoperate an unmanned aerial vehicle (UAV) equipped with a custom CdZnTe-based spectroscopic gamma-ray detector in outdoor environments. The task is to localize nuclear radiation sources, whose location is unknown to the user, without the close exposure of the operator. The developed detector also enables identification of the localized nuclear sources. The aim of the VHAR interface is to increase the situation awareness of the operator. The user teleoperates the UAV using a 3DOF haptic device that provides an attractive force feedback around the location of the most intense detected radiation source. Moreover, a fixed camera on the ground observes the environment where the UAV is flying. A 3D augmented reality scene is displayed on a computer screen accessible to the operator. Multiple types of graphical overlays are shown, including sensor data acquired by the nuclear radiation detector, a virtual cursor that tracks the UAV and geographical information, such as buildings. Experiments performed in a real environment are reported using an intense nuclear source.

Keywords: CdZnTe-based detector; nuclear radiation detector; haptic teleoperation; unmanned aerial vehicles

1. Introduction

UAVs in nuclear inspection tasks can be either teleoperated by a human operator, usually with standard remote controllers [1,2], or fly autonomously [3–9], following a predefined route or using an exploration behavior. In both cases, an expert human operator is required to oversee the entire operation, being aware of the state of the UAV, verifying sensor data acquisition and checking for potential obstacles or dangerous conditions related to the mission. The advantage of using human teleoperation is that an expert operator can focus on selected areas of the environment rather than calling for an exhaustive scan, thereby overcoming the problem of the limited duration of each flight afforded by onboard batteries. However, UAV teleoperation in critical tasks raises a fatigue issue: maintaining a high degree of situation awareness is challenging, as it imposes a high mental demand for the human operator. Hence, it is crucial to provide additional information to the pilot by exploiting multiple feedback channels. Potential applications are the localization and identification of radioactive materials in industrial plants (such as steel mills), construction, recycling factories and landfills.

This paper investigates visuo-haptic teleoperation of an unmanned aerial vehicle carrying a custom nuclear radiation detector for environmental monitoring. To maintain a high level of situation awareness, the VHAR interface provides visual feedback in real time and, simultaneously, force feedback through the haptic device to the operator. The use of visuo-haptic interfaces for UAV teleoperation has not been considered in previous works. In particular, a haptic rendering algorithm is

presented, based on impedance control, that provides an attractive force feedback around the location of the most intense detected radiation source. The purpose of the attractive force feedback is to keep the UAV close to the nuclear radiation source once a target is found. Preliminary experiments with users [10] have shown that mental load and difficulty in associating perception to sensor localization increase as the UAV flies farther from the operator. Hence, in order to provide a more adequate support for critical operations, visual feedback is added, in addition to haptic feedback, to convey coherent information to the operator.

Visual feedback is provided to the user on a computer screen using augmented reality. The video stream of a fixed camera on the ground, which observes the environment where the UAV is flying, is augmented with graphical overlays. In particular, a 3D histogram of the measured radiation intensity is displayed on top of the video stream to let the operator see the most recent measured intensity values, as well as the location of the maximum radiation found during the mission. A 2D virtual cursor is also displayed around the UAV that is computed from a vision-based tracking algorithm. Visual tracking not only facilitates the operator to detect the UAV on the image when flying at large distances, but it also improves the estimation of the UAV 3D pose and, therefore, the localization of the nuclear sources, compared to the accuracy that would result by using the UAV onboard sensors alone. Other elements such as buildings in close proximity to the UAV are retrieved from a geographic information system (GIS), registered using a semi-automatic approach and highlighted on the screen. Finally, simple bounding boxes of the building are used to generate geo-fences for the UAV, i.e., a virtual perimeter for collision avoidance.

Complete experiments in a real environment have been performed with an intense nuclear source as shown in Figure 1, under the supervision of the public environmental protection agency. Experiments have been performed by an expert operator due to the critical setup. The UAV was clearly visible to the operator, but the location of the radiating substance was not known to the operator in advance. Quantitative data have been collected such as the task completion time and the error between the location of the radiating substance, estimated by the operator, with respect to its actual location taking advantage of a video camera mounted on the UAV. Experiments show that a teleoperation approach that supports switching between heading-based and position to position control modes increases the position detection accuracy of the radio-active material with respect to a pure heading-based control mode [10,11]. Usability experiments, performed in a simulated environment, are also reported. Results indicate that adding visual feedback does not further improve position detection accuracy, but it increases the situation awareness of the operator and reduces mental workload.

Small multi-rotor unmanned aerial systems can obtain high spatial resolution maps of radiological contamination sources as pointed out in [2]. Several authors have investigated the application of UAVs for monitoring environmental radioactivity [1–5,7,9,12–14]. However, none of these works has considered the use of haptic teleoperation for nuclear radiation detection. Indeed, either standard remote controllers were adopted or pre-programmed flight missions were used for autonomous UAVs. A survey of networking aspects for small unmanned aerial systems is reported in [15].

In [12] a UAV-mounted biosensor system is described for environmental monitoring applications including radiation leakage detection. Okuyama et al. [9] developed an autonomous helicopter for measuring radiation data during a flight, with real-time data transmission including images to a monitoring ground station. Boudergui et al. [1] provided a preliminary evaluation of a teleoperated UAV equipped with a CdZnTe sensor and a gamma camera for nuclear and radiological risk characterization. However, the system was developed for indoor environments, whereas we focus on outdoor environments, which pose different problems in terms of UAV localization, as well as the operator's situational awareness. Unmanned aerial vehicles with a fixed wing, flying at high altitude and high speed have been presented in [5,13] for radiation detection in outdoor environments. In [7], a remotely-piloted UAV was proposed to measure hazardous gaseous sources. Martin et al. [3] presented a UAV for the radiological characterization of uranium mines. Sanada et al. [14] developed

an unmanned helicopter to monitor radiation at the Fukushima Dai-ichi nuclear power plant (FDNPP). Radioactive cesium deposition was successfully measured on the ground. In [4], an unmanned aerial system was presented that was capable of producing more accurate radiation distribution maps in the FDNPP with a resolution of more than 1 m. In other works [6,16], simulation results have been reported. In [16], a simulated UAV for imaging and radiation detection was developed using an autonomous helicopter. In [6], a simulation was developed with multiple UAVs for contour mapping of nuclear radiation with formation flight control.

Figure 1. UAV equipped with a CdZnTe gamma-ray detector in flying tests (top). The operator using the visuo-haptic user interface (bottom). The operator sits in front of a cloth that prevents him from seeing the nuclear source on the ground.

In previous works, augmented reality for unmanned aerial vehicles has been investigated mainly by using videos from onboard cameras. In [17], a UAV equipped with a video camera was used to generate an augmented reality environment for construction site monitoring that supported registration and visualization of 3D building models. In [18], a similar approach for real-time UAV video augmentation was presented with applications to disaster response. In [19], another augmented telepresence system was developed for large-scale environments by exploiting an omni-directional camera. Iwaneczko et al. [20] presented a heads-up display to be used in UAV ground control stations to improve the UAV manual control performance of the operator. In [21], a mixed reality environment was developed where a user could interactively control a UAV and visualize range data in real-time. The closest work to ours using a fixed ground camera was proposed by Zollmann et al. [22], where an augmented reality system was developed. However, the system was aimed at specifying waypoints for the UAV from a touchscreen and at checking for potential collisions with the surrounding environment.

So far, haptic teleoperation of aerial vehicles has been investigated exclusively for collision avoidance or to make the flight process easier [23–36]. Reyes et al. [23] developed a remotely-operated UAV for indoor and outdoor environments where force feedback is proportional to the translational speed and proximity to objects. In [24], a novel intuitive technique for UAV teleoperation was introduced where a repelling force feedback is generated proportional to the UAV's velocity. Lam et al. [25] investigated artificial force fields to generate haptic feedback in UAV teleoperation in simulated scenarios. In [26], an approach was presented for target identification and obstacle avoidance in indoor environments. A bilateral control system was developed for haptic teleoperation with force feedback, and a 3D map of the environment was built using computer vision. In [27], an intuitive teleoperation technique was presented, including force feedback, to safely operate a UAV by an

untrained user in cluttered environments. Masone et al. [28] proposed a method for semi-autonomous UAV path specification and correction where a human operator could modify the shape of the path of the UAV, while an autonomous algorithm ensured obstacle avoidance and generated force feedback. In [29,30], an admittance haptic control technique was introduced based on both the UAV position, which was tracked by an indoor visual sensor, and on the force exerted on the haptic device. Ruesch et al. [32] proposed a UAV haptic teleoperation method to overcome the workspace limitation of the haptic device and cover large distances.

CdZnTe detectors, thanks to their wide operative energy range, room temperature operation and good spectroscopic properties, have been used in many fields such as in medical applications [37], security [38], environmental control [39] and astrophysics [37]. Furthermore, CdZnTe detectors are of particular interest in UAV applications due to their high stopping power for high energy gamma radiation, robustness, low weight and low power consumption.

The paper is organized as follows. Section 2 provides an overview of the system. Section 3 describes the components of the visuo-haptic augmented reality interface. Section 4 presents the experimental results. Section 5 concludes the paper, discussing the results and providing suggestions for possible extensions.

2. Overview of the System

Figure 2 shows the overall architecture of the proposed system, which is based on the ROS framework. In nuclear source detection tasks, the UAV flies at a constant altitude and constant speed. A low velocity set point value is adopted since a high speed would affect georeferencing of the sensor data. The operator of the haptic device is in charge of guiding the UAV during the hazard search and localization phase by sending motion commands on the xy plane (parallel to the ground) while receiving visuo-haptic feedback. A fixed camera on the ground observes the UAV and provides visual feedback. The camera is connected to the ground station through a Gigabit Ethernet cable. Visual feedback is displayed on a computer screen while a 2D planar force feedback is conveyed through the haptic device connected to a ground station. A second human operator (supervisor), using a standard remote controller, is responsible for take off, landing and setting the altitude set point. The second operator can also take full control of the UAV at any time, as required by the Italian regulation for unmanned air vehicles, thus overriding haptic commands. Both operators have direct sight of the UAV. The main functions of the ground station are to process motion commands provided by the operator of the haptic device, to send commands to the UAV, to receive sensor data from the UAV, and to compute and render both visual and force feedback. Information received by the ground station consists of UAV telemetry data (including position, speed, height and battery charge), sent through the UAV radio link, and sensor data from the onboard gamma-ray detector (number of photon counts for each energy band in a fixed time frame), sent through a dedicated wireless link.

Assuming a planar environment, the gamma-ray detector will measure a maximum intensity when it is on the vertical of the radiating target. The UAV GPS coordinates will be assumed as the coordinates of the radiating part at the ground. The haptic device used in this work is a 3DOF Novint Falcon, with a resolution of about 400 dpi, a maximum force feedback capability of about 10 N and a range of motion of about 10 cm^3.

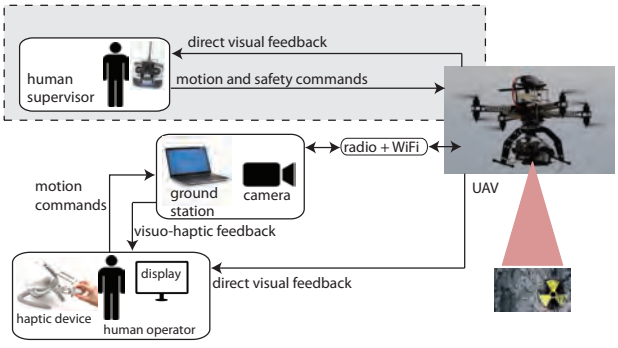


Figure 2. Overall architecture of the system.

2.1. UAV Platform

The UAV equipped with the CdZnTe gamma-ray detector is shown in Figure 3. The aerial vehicle is an octocopter in coaxial configuration, produced by Virtual Robotix Italia, with a gross payload up to 4 kg and a maximum flight time of about 10–15 min. The UAV transmits telemetry data to the ground station (868 MHz RF link). The UAV is built with a mixed Carbon and Aluminium structure. The size of the frame is within 550 mm (without propellers). The UAV is equipped with an MEMS accelerometer, gyro, magnetometer and GPS sensors. A VRBrain autopilot system is used (based on the ArduCopter firmware adapted by Virtual Robotix Italia), which comprises a 168-MHz ARM CortexM4F microcontroller with DSP and floating-point hardware acceleration. The autopilot system supports multiple flying modes such as loiter, return to home and guided mode.

Figure 3. UAV equipped with the CdZnTe gamma-ray detector (mounted at the bottom).

The CdZnTe gamma-ray detector is enclosed in a box and mounted on a two-axis brushless gimbal. An embedded system based on the Intel Galileo board reads sensor data from the gamma-ray detector and sends the data stream to the ground station through a dedicated long-range 5-GHz WiFi connection. Intel Galileo is a single core i586 platform (400 MHz, 256 MB RAM). The dedicated wireless data connection link avoids bandwidth contention on the UAV RF channel and does not affect the UAV autopilot system, which runs on a real-time operating system. Two external antennas are connected to the embedded platform allowing a WiFi communication range up to 170 m. The embedded system is powered by an external Li-Ion 10-Ah, 5-V battery pack.

2.2. CdZnTe Gamma-Ray Detector

The goal of the gamma-ray detector is to detect nuclear sources on the ground in a wide energy range to reveal the most dangerous contaminants that may be dispersed in the environment. The detector, designed and developed for this work, is lightweight (about 0.3 kg), and it has low power consumption. The measurable energy range is from 10 KeV–1.5 MeV so that all o the main nuclear isotopes can be detected. Radioactivity data measured by the detector are represented by a histogram of 2800 energy bands. Each bin i contains the number of counts $C[i]$ detected in a time frame $\triangle t = 2$ s. The count rate C for each energy band i varies according to the inverse square law:

$$C \propto \frac{1}{l^2} \tag{1}$$

where l is the distance to the nuclear source. The time required to transmit a full spectrum to the ground station, through the WiFi link, with all 2800 energy bands (including sensor reading and transmission) is about 2.2 s (0.45 Hz). The sensor features a large field of view. Since the higher the energy of the photon to be revealed, the larger the thickness of the detector that must be used, a 6 mm-thick CdZnTe (cadmium zinc telluride) detector was adopted. As one of the most important applications of the proposed system is the detection of nuclear sources considered dangerous for operators and workers in industrial plants, the detector was designed to measure nuclear sources whose average effective dose is 1 mSv/year at a 1-m distance. Indeed, 1 mSv/year is the dose limit set by the law for workers. Table 1 reports the typical number of counts per second measured by a $20 \times 20 \times 6$ mm detector at a 2-m distance from some nuclear sources. The values indicate that by using the proposed detector at about 2 m from the ground, it is possible to measure a number of counts per second that is sufficient for localizing nuclear sources that are relevant for workers' safety. The choice of a single carrier device was due to the fact that hole transport properties are not good enough to ensure optimal charge drift over several millimeters. Thus, the detector was fabricated by using a contact geometry known as "drift strip", which is known to ensure good energy resolution even for large volume detectors [40].

Table 1. Measured radioactivity from different nuclear sources by a $20 \times 20 \times 6$ mm detector at a 2-m distance.

Nuclear Source	Dose (mSv/year)	Source Activity (Bq)	Counts/s
Americium 241	1	1.6×10^8	1270
Cobalt 57	1	5.4×10^7	411
Cesium 137	1	8.2×10^6	159

Four $5 \times 6 \times 20$ mm^3 drift strip detectors were realized using CdZnTe material acquired from Redlen. Figure 4 shows the contact geometry of the anode and cathode contacts. The anode is realized with seven strips, the central being the collecting one, the adjacent strips are polarized in such a way to drift the carriers towards the collecting strip. Electroless gold contacts were used in order to obtain blocking contacts [41], and surface passivation was adopted to reduce the surface leakage current. Analog electronics were used to read out the generated signals, according to the scheme in Figure 5. The detector, the read out electronics, the batteries and the DC-DC circuit, which generates the high voltage supply, were stored in a box easily mounted on the UAV, as shown in Figure 3.

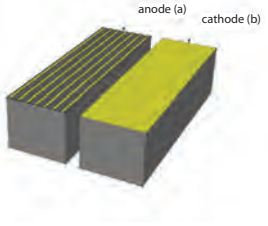


Figure 4. Drift strip gamma ray detector: anode (**a**) and cathode (**b**).

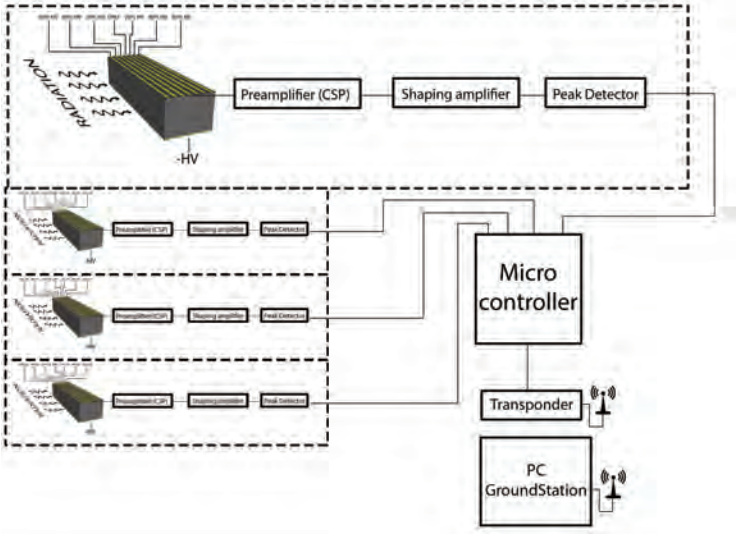


Figure 5. Read out electronic, with charge sensitive preamplifier (CSP), and data transmission scheme.

An automatic procedure is performed at the beginning of each flight for background radiation removal (the sensor does not have a built-in auto-calibration feature) assuming that the UAV take off location is far away from all of the nuclear sources of the environment. Indeed, once the UAV is airborne and hovering at the desired height, a set of radiation measurements is taken for some time frames $\triangle t_j$, $j = 1 \dots K$, then the intensity of the background radiation is set as $I_b = \max_{\triangle t_j} \sum_i C_b[i]$, i.e., the largest measured value of the sum of the counts over all energy bands $i = 1 \dots 2800$. Then, when the UAV is flying during the teleoperated mission, the current radiation intensity I in a time frame is georeferenced using the current estimated UAV pose, and it is used to update the position on the ground with the highest radiation that in turn is needed to generate force feedback (as explained in Section 3.1). The current radiation intensity is computed as the difference between the current measured value $\sum_i C_m[i]$ and the background radiation acquired at the beginning of the mission, i.e.,

$$I = \begin{cases} \sum_i C_m[i] - I_b - \gamma & \text{if } (\sum_i C_m[i] - I_b) > \gamma, \\ 0 & \text{otherwise ,} \end{cases} \tag{2}$$

where γ is a small threshold. The proposed approach for background radiation removal is more realistic than the one proposed in [10], which was adopted for the initial evaluation of the teleoperation system in experiments with simulated nuclear radiation sources.

3. Visuo-Haptic Augmented Reality Interface

3.1. Haptic Teleoperation

Two impedance control modes (Figure 6) have been developed where the operator sends motion commands (via the haptic device tool point) and receives force feedback limited to the xy plane. Since inspection tasks require precise position-controlled flights, in both teleoperation modes, there is no direct mapping from the haptic device to the velocity of the UAV. Moreover, the two teleoperation modes have been designed to map the limited workspace of the haptic device to an arbitrarily large UAV workspace. In the first teleoperation mode ($M_{heading}$), position information read by the haptic interface (namely, the x, y coordinates of the tool point) is used to compute the horizontal heading direction of the UAV [10,11]. In particular, the current displacement $\vec{p} = (p_x, p_y, p_z)$ of the tool point of the haptic device with respect to its center is converted to a waypoint $\vec{w_h} = (x, y, z)$ for the UAV in world coordinates as follows:

$$\vec{w_h} = \frac{\eta}{\|\vec{p}\|} \underbrace{\begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}}_{{}^{W}_{H}\mathbf{T}} \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} \tag{3}$$

$$\eta = \begin{cases} 0 & \text{if } \|\vec{p}\| < D \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $\vec{u} = (u_x, u_y, u_z)$ is the current position of the UAV in the world reference frame, α is a constant and ${}^{W}_{H}\mathbf{T}$ is the rotation matrix from the haptic coordinate system H to the world reference frame W. The world reference frame corresponds to the UAV pose at take off. A flat Earth model is assumed for the conversion between Cartesian and GPS coordinates. The UAV flies in guided mode following the current waypoint with yaw heading pointing towards the waypoint itself. Each waypoint $\vec{w_h}$ is not meant to be reached by the UAV as it is continuously updated and placed at a long distance from the current UAV position ($\alpha \gg 0$), thus resulting in a heading-based approach. In Equation (3), the altitude of the generated waypoint (z coordinate) does not depend on the displacement of the tool point of the haptic device p_z. Indeed, as explained above, the UAV altitude set point u_z is provided by the second operator via the standard remote controller or assumed as a default mission parameter. If the tool point of the haptic device is within a fixed range D from the center of the haptic reference frame, the waypoint is not updated ($\eta = 0$ in Equation (4)), and the UAV hovers.

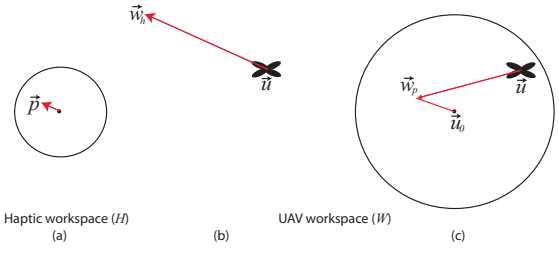


Figure 6. Diagram illustrating the two haptic teleoperation techniques. Haptic workspace on the x, y plane (**a**). Heading-based control technique (**b**); the current displacement $\vec{p}$ of the haptic device is used to compute a heading direction with respect to the current UAV position. Position to position control technique (**c**); $\vec{p}$ is used to compute a waypoint with respect to the center of operation $\vec{u_0}$.

The second teleoperation mode is a hybrid approach that allows the operator to switch between the heading-based technique ($M_{heading}$), discussed above, and a position to position control technique

($M_{position}$). Mode switching is achieved by pressing one of the buttons of the haptic device. The position to position control technique, inspired by the approach proposed in [27], enables a direct mapping between the haptic device to the UAV position around a fixed center of operation. In particular, in this second teleoperation mode, a waypoint $\vec{w}_p = (x, y, z)$ for the UAV in world coordinates is computed as follows:

$$\vec{w}_p = \begin{cases} \frac{\eta}{\|\vec{p}\|} {}^{W}_{H}\mathbf{T} \begin{bmatrix} \delta & 0 & 0 \\ 0 & \delta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} u_{0x} \\ u_{0y} \\ u_{0z} \end{bmatrix} & \text{if } M_{position}, \\ \\ \vec{w}_h & \text{if } M_{heading}. \end{cases} \tag{5}$$

where $\vec{u_0} = (u_{0x}, u_{0y}, u_{0z})$ is the center of operation, defined as the position of the UAV when $M_{position}$ is activated, and $\delta \ll \alpha$ is a constant. The idea is that the heading-based control technique can be used for long transition motions, e.g., when the UAV has to be moved to explore different regions of the environment, while the position to position control technique is helpful when the UAV explores an area in close proximity of a nuclear source. Although it is known that mode switching teleoperation can lead to instabilities, the proposed approach did not exhibit such a problem in any of the reported experiments (that included a single nuclear source), as well as in the simulated experiments, due to the low switching frequency (one to three mode switches per mission on average).

A force feedback $\vec{f}$ is provided to the operator, acting as a basin of attraction, to let the UAV fly close to the region where the radiation is maximal. Indeed, as the remotely-operated UAV travels away from the location of the most intense perceived radiation, the haptic device exerts a force in the horizontal direction towards the center of the haptic workspace. The force feedback is computed as:

$$\vec{f} = \frac{\zeta}{\|\vec{r} - \vec{u}\|} \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix}}_{{}^{H}_{W}\mathbf{T}} \begin{bmatrix} \beta & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_x - u_x \\ r_y - u_y \\ r_z - u_z \end{bmatrix} \tag{6}$$

$$\beta = \frac{f_{max}}{L^2} d^2(\vec{r}, \vec{u})_{xy} \tag{7}$$

$$\zeta = \begin{cases} 0 & \text{if } d(\vec{r}, \vec{u})_{xy} > L \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

where $\vec{r} = (r_x, r_y, r_z)$ is the estimated position of maximum radiation discovered so far by the UAV, which is updated periodically (with period $\triangle t$). Radiation intensity I is computed using Equation (2). If the latest measured radiation intensity is greater than the maximum radiation found so far, the latest radiation intensity is set as the current maximum radiation. The force feedback has a quadratic profile (Equation (7)), where $d(\vec{r}, \vec{u})_{xy}$ is the distance between the projection of the UAV on the ground and $\vec{r}$, f_{max} is the maximum exerted force and L is the radius of the basin of attraction. No force feedback is provided outside the basin of attraction ($\zeta = 0$ in Equation (8)). The provided force does not overpower the user who can also guide the UAV outside the basin of attraction if he/she desires. Moreover, the user can reset force feedback to zero by pressing a button on the tool point of the haptic device. This feature can be useful when multiple concentrated nuclear sources have to be explored.

3.2. Vision-Based UAV Detection

A fixed camera on the ground observes the UAV work space including the takeoff area. A takeoff platform, shown in Figure 7, includes a marker that is used to calibrate the reference frames.

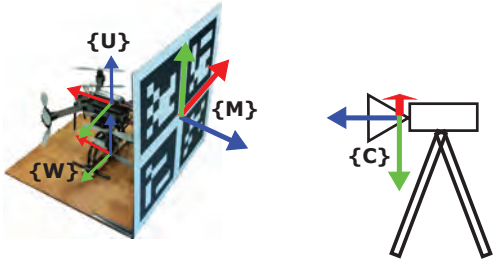


Figure 7. UAV takeoff platform with marker (80 × 80 cm) and reference frames.

The current 3D position ${}^{C}u^{t}_{ekf}$ of the UAV in the camera reference frame C is given by ${}^{C}u^{t}_{ekf} = {}^{C}_{M}T {}^{M}_{W}T \cdot {}^{W}u^{t}_{ekf}$, where ${}^{W}u^{t}_{ekf}$ is the current 3D position of the UAV in the world reference frame W (located at the base of the takeoff platform) provided by the autopilot extended Kalman filter (EKF). The EKF fuses data from the GPS sensor, IMU and barometric altimeter, and therefore, it is strongly affected by the position error of the GPS receiver. ${}^{M}_{W}T$ is a fixed transformation between W and the marker reference frame M, and ${}^{C}_{M}T$ is the transformation of the marker as seen from the camera reference frame.

The proposed approach for vision-based UAV detection, illustrated in Algorithm 1, exploits motion segmentation and SVM-classification using SURF local descriptors. The algorithm was developed on the OpenCV library. Main input data are the intrinsic camera parameters, the current image S from the camera, the UAV position estimated by the EKF in camera coordinates ${}^{C}u^{t}_{ekf}$, the projection of the UAV position on S at previous frame $\overline{u}^{t-1}_{ekf}$, a search window SW (defined by a center pixel and a size) and the last UAV predicted state $\overline{u}^{t-1}$ on S. The output is the current estimated position of the UAV in camera coordinates ${}^{C}u^{t}_{v}$. The estimated position of the UAV using visual detection is more accurate than the one provided by the autopilot EKF, and therefore, it also improves georeferencing of the nuclear source.

After projecting ${}^{C}u^{t}_{ekf}$ on the image plane (line 1), the foreground images $F(S)$, representing moving objects, are computed (line 2) using a background subtraction algorithm [42]. A dilation is then applied to $F(S)$ (line 3). An efficient search of the UAV is then performed in a sub-window SW of the dilated image $\overline{F}(S)$, centered at $\overline{u}^{t}_{ekf}$ (Lines 4–16). In particular, blobs are extracted from the sub-window, and then, the axis-aligned bounding box (AABB) Box_i of each blob is computed.

Afterwards, each box is evaluated by a binary bag-of-words SVM classifier, trained to detect the UAV in the image, using SURF features (Lines 9–16). The box with the highest positive score $BestBox$, if any, is selected as the candidate AABB that contains the UAV (this case is named SURF). The 3D position of the UAV in camera coordinates ${}^{C}u^{t}_{v}$ is estimated by back projection of the center pixel of $BestBox$ using the z coordinate of the orthogonal projection of ${}^{C}u^{t}_{ekf}$ on the optical ray (line 18). The current state $\overline{u}^{t}$ of the UAV on S is also updated using the $BestBox$ center (line 19). Finally, the size of SW is updated to the size of $BestBox$ enlarged by a constant value (line 20).

If a valid box is not found from the SVM classifier, the Box_i having the largest intersection with the search window and whose distance to the predicted position $\widehat{u}^{t}$ of the UAV does not exceed the threshold value Th (Lines 22–30) is selected (this case is named the closest box). Then, the box center is used to compute the 3D position of the UAV in camera coordinates (line 32) as above. Prediction (line 22) is computed using an alpha-beta filter. The size of SW is also incremented by a small value (line 34). If a valid box is still not found, prediction $\widehat{u}^{t}$ is used to compute the 3D position of the UAV in camera coordinates (line 36), and $\overline{u}^{t}$ is updated using the difference vector between the last two measured positions provided by the EKF (line 37) (this case is named prediction). Since the last case is valid for short-term prediction only, after a fixed time frame, the 3D position of the UAV is directly set as ${}^{C}u^{t}_{ekf}$.

Algorithm 1: Vision-based UAV detection.

Input: Intrinsic camera parameters

S: Image at t

${}^{C}u^{t}_{ekf}$:UAV position at t from EKF in camera frame

$\overline{u}^{t-1}_{ekf}$: UAV position from EKF at $t-1$ on S

SW: search window (center, size)

Th: displacement threshold between two frames

$\overline{u}^{t-1}$: last UAV predicted state on S

Output: ${}^{C}u^{t}_{v}$: estimated UAV position

$\overline{u}^{t}_{ekf} \leftarrow {}^{C}u^{t}_{ekf}$ projection on S
$F(S) \leftarrow$ Compute foreground image
$\overline{F}(S) \leftarrow F(S)$ dilation
$SW \leftarrow updateCenter(SW, \overline{u}^{t}_{ekf})$
$C \leftarrow$ Extract blobs in $SW(\overline{F}(S))$
for $C_i \in C$ **do**
    $Box_i \leftarrow$ Extract AABB of C_i
end for
$Score_{max} \leftarrow 0$
for each Box_i **do**
    $Score_i \leftarrow$ Classify Box_i with SVM
    if $Score_i \geq Score_{max}$ **then**
        $BestBox \leftarrow Box_i$
        $Score_{max} \leftarrow Score_i$
    end if
end for
if $Score_{max} > 0$ **then**
    ${}^{C}u^{t}_{v} \leftarrow Backproject(center(BestBox))$
    $\overline{u}^{t} \leftarrow center(BestBox)$
    $SW \leftarrow updateSize(SW, size(BestBox))$
else
    $\widetilde{u}^{t} \leftarrow predict(\overline{u}^{t-1})$ predicted position on S
    $Area_{max} \leftarrow 0$
    for each Box_i **do**
        $Area_i \leftarrow Box_i \cap SW$
        if $Area_i \geq Area_{max} \wedge$
        $\|center(Box_i) - \widetilde{u}\|^{t} \leq Th$ **then**
            $BestBox \leftarrow Box_i$
            $Area_{max} \leftarrow Area_i$
        end if
    end for
    if $Area_{max} > 0$ **then**
        ${}^{C}u^{t}_{v} \leftarrow Backproject(center(BestBox))$
        $\overline{u}^{t} \leftarrow center(BestBox)$
        $SW \leftarrow incrementSize(SW)$
    else
        ${}^{C}u^{t}_{v} \leftarrow Backproject(\widetilde{u}^{t})$
        $\overline{u}^{t} \leftarrow \overline{u}^{t-1} + \overline{u}^{t}_{ekf} - \overline{u}^{t-1}_{ekf}$
        $SW \leftarrow incrementSize(SW)$
    end if
end if

3.3. Visual Feedback Displayed in Augmented Reality

An augmented reality environment is generated and displayed on a computer screen to provide real-time visual feedback to the operator during the mission. 2D and 3D graphical overlays, called widgets, are rendered using the OpenGL library on top of the video stream acquired by the video camera. Geometric registration of the virtual camera is performed thanks to the calibration parameters obtained as explained in Section 3.2. An example of the augmented reality scene is shown in Figure 8.

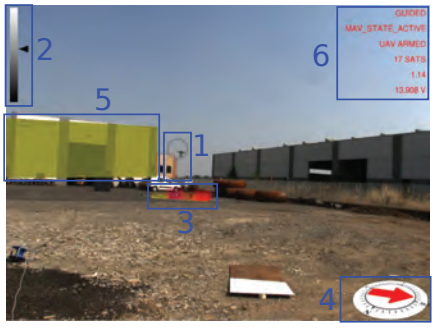


Figure 8. Augmented reality scene example with graphical overlays highlighted by blue boxes (for better visibility).

Widget 1 is a virtual cursor displayed as a ring around the UAV that is computed thanks to the vision-based UAV detection algorithm. The virtual cursor tracks the UAV, and therefore, it helps the operator to quickly identify its current position. A vertical line is also displayed to show the current projection of the UAV on the ground. The ring is displayed on a gray scale according to the distance of the UAV to the camera. This information is also displayed by a 2D cursor moving on a vertical bar (Widget 2, top left) that acts as a distance-color map. The maximum distance (white color) in the experiments was set to 50 m.

Widget 3 is a colored 3D histogram that represents the distribution of the georeferenced radiation intensity I on the ground. The histogram shows the bin of the maximum detected radiation intensity during the mission and the closest $k = 25$ bins to the current UAV position. The higher the bin, the higher the radiation intensity inside a cell of the grid. The highest bin has a fixed height, and all other bins are scaled in real time. The grid has a 1 m $\times$ 1 m cell size. The color of each bin also changes according to the radiation intensity value, ranging from green (low radiation intensity) to red (high radiation intensity).

The current UAV heading and magnetic north are displayed on a virtual compass (Widget 4). Geographical features in close proximity to the UAV workspace are retrieved from a query to the Open Street Map (OSM) database, registered in the augmented reality scene and highlighted on the screen (Widget 5) when the UAV flies close to them. Each building is displayed by computing an oriented bounding box from a 2D polyline provided by the geographical data. Buildings are also used to generate geo-fences for the UAV, i.e., a virtual perimeter for collision avoidance. Automatic registration only provides a coarse alignment of the geographical elements due to calibration errors and imprecisions in the OSM data. Hence, registration is manually refined during the calibration phase. More advanced approaches for automatic registration of buildings could be adopted [43–45]. However, these methods are beyond the scope of this work. Buildings are displayed using yellow color. Since OSM does not provide information about buildings' height, the height is fixed to a constant value. The last graphical overlay (Widget 6) provides textual information to the operator about mission-critical parameters including current flight mode, flight status, horizontal dilution of precision (HDOP) and number of satellites currently visible.

4. Experiments

4.1. Preliminary Evaluation

Initial evaluation of the CdZnTe detector was carried out in an indoor laboratory setup by measuring the number of counts at each energy band for low intensity nuclear sources. Figure 9 shows the spectra obtained with ^{57}Co and ^{137}Cs nuclear sources. The energy resolution is 4% at 122 keV and 2.9% at 662 keV, in spite of the mechanical vibration induced by the UAV.

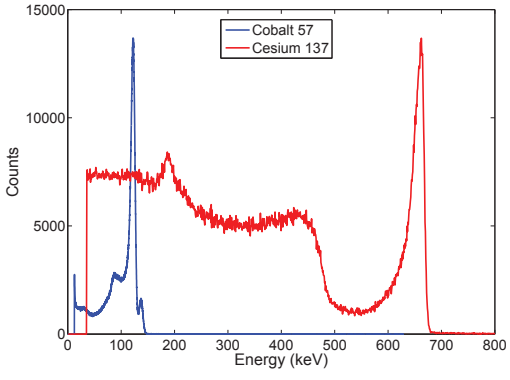


Figure 9. ^{137}Cs and ^{57}Co emission spectra as measured by the detector and the electronic mounted on the UAV in an indoor laboratory setup.

4.2. Evaluation of Nuclear Source Detection Accuracy in Real Environments

The visuo-haptic teleoperation system has been validated in an outdoor environment with a real nuclear source under the supervision of the public environmental protection agency. Experiments have been performed in daylight conditions with an intense ^{192}Ir nuclear source located in a service area of an industrial plant located in Gossolengo, Piacenza, Italy. Performing experiments with real nuclear sources in operational environments is a complex endeavor, requiring formal authorizations. Due to this procedural complexity, it took us about one year to set up the experiment reported in this section. The nuclear source was put into a vertical lead container placed on the ground to prevent lateral radiation emission (Figure 3). Due to the high-intensity source, the container was closed at the top by a 3 cm-thick lead plate. The operator was kept at a safety distance of about 30 m from the nuclear source. The UAV maintained a height from the ground ranging from 1.5–3 m.

A first experiment has been performed in heading-based teleoperation mode with prior knowledge of the location of the nuclear source, which was placed in direct sight of the operator. Figure 10 shows an image of the experiment, with the UAV close to the radiating source, as well as the corresponding total accumulated spectrum. The acquired energy spectra reveals all of the main emission peaks of ^{192}Ir, allowing a definitive identification of the emitting nuclear source. Figure 11 shows the total flight path of the UAV. The flight time of the experiment was about 7 min. In this experiment, the error between the actual position of the nuclear source and the location of the maximum measured radiation intensity was about 2.5 m (estimated using GPS coordinates).

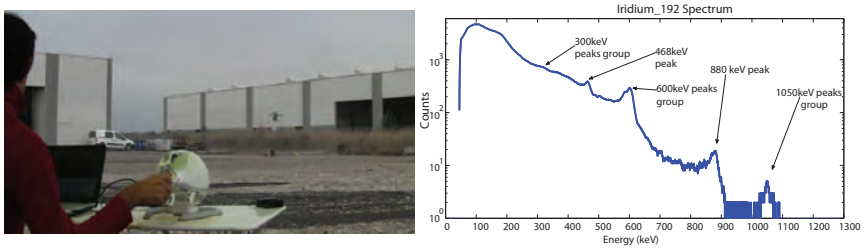


Figure 10. Detection of ^{192}Ir in a real environment with heading-based teleoperation mode (the operator has direct sight of the nuclear source): image during the exploration task (left). Total radiation spectrum obtained by summing up all measurements (right). A video of the experiment is available as Supplementary Material (video1).

Figure 11. Detection of ^{192}Ir in the real environment (the operator has direct sight of the nuclear source): flight path of the experiment shown in Figure 10; regions where radiation was measured by the CdZnTe detector (green circles with different sizes and shading) and the estimated location of the maximum radiation (apex of the yellow triangle).

To improve accuracy in measuring the error between the location of the nuclear radiation source, estimated by the operator of the haptic device, and its actual location, a video camera was installed on the UAV, pointing to the ground, to record each experiment. Indeed, the recorded videos have been used offline to measure the error with a significantly better accuracy than using the GPS sensor. A set of experiments (five repetitions) has been performed to compare the two teleoperation modes described in Section 3.1. The location of the nuclear source was changed in each experiment, and it was unknown to the operator. Indeed, the operator was sitting in front of a cloth that prevented him from seeing the nuclear source on the ground. The operator was still able to see the UAV during the task. The nuclear source was placed at the center of a target with concentric circles (0.5-m radius step size). Figure 12 shows images of the experiments. Again, each exploration experiment ended when the operator affirmed that the current position of maximum measured radiation was close enough to the nuclear source. On average, using the position to position teleoperation mode, the operator was able to detect the location of the nuclear source with a position error of 1.5 ± 0.5 m, while using the heading-based teleoperation mode, the position error was about 2.5 ± 0.5 m. The distance between the take-off position of the UAV and the location of the nuclear source was, on average, 20 m. The average flight time was 5 min. Experiments did not require any critical intervention by the operator of the remote controller.

Figure 12. Detection of ^{192}Ir (located at the center of the target within the lead container) in a real environment using the position to position teleoperation mode. Experimental setup (top), where the UAV is highlighted by the red circle. Image taken from the onboard camera (bottom) when the operator affirmed that the current position of the UAV returned the maximum measured radiation. In this set of experiments, the gimbal unit was not used to get a clearer view of the ground from the onboard camera. The low speed of the UAV ensured that the gamma-ray detector remained parallel to the ground.

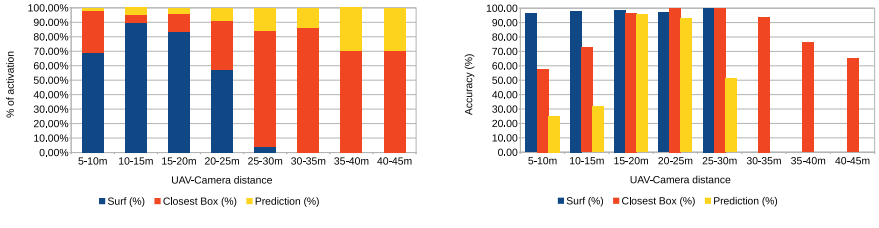
4.3. Evaluation of Visual Feedback

Images of the augmented reality environment are reported in Figure 13. The fixed ground camera used in the experiments (AVT Mako G125C) has a resolution of 1292 × 964 and a maximum frame rate of 30 fps. The camera was placed at approximately a 1.5-m height from the ground. The vision-based UAV detection algorithm takes, on average, 16.72 ms of execution time (59.8 Hz on an Intel Core i7-4720HQ). The training set (Figure 14) contained 4000 images (2000 negative samples and 2000 positive samples) acquired in a different environment (rural area). The SVM classifier, implemented by the OpenCV library, has been trained using bag-of-words with SURF features and a codebook of size 1000. The train_auto method was used, which implements a 10-fold cross-validation. The test set contained 3944 images taken from the industrial setting (Figure 13). The UAV detection algorithm was evaluated in a distance range between 5 and 45 m. Figure 15a shows the frequency of activation of the three UAV detection strategies (SURF, closest box and prediction) at different horizontal distances of the UAV to the camera. It can be noted that SURF-based estimation is mainly active when the UAV distance to the camera is under 25 m. Beyond that distance, the size of the UAV on the image reduces, so that SURF features are no longer usable; thus, closest box is mainly active. The activation frequency of the prediction-based estimation method also increases with the distance. Accuracy was computed by comparing the estimated UAV location on the image with a manually-annotated ground truth. A distance threshold of 15 pixels, estimated from an ROC curve, was used. The overall detection accuracy of the algorithm on the test set is around 90%. Figure 15b reports detailed detection results. The SURF method, when active, achieves an accuracy above 95%. The closest box method achieves an accuracy rate of about 60% for the short range (when it is rarely active) due to the fact that when the UAV is close to the camera, the search window includes spurious

blobs from the ground (e.g., moving dust or grass). Closest box also achieves a 60% accuracy rate at long distances, as background subtraction can fail to detect the UAV due to its small projection on the image plane. Although the accuracy rate of the prediction method, which is mainly active at long distances, is very low, it still provides an estimated position of the UAV on the image plane that is useful to the operator to identify the UAV, as confirmed by the user study reported in Section 4.4. A comparison between the vision-based UAV detection algorithm and the result provided by the autopilot EKF is shown in Figure 16. 3D registration of buildings required a manual correction. Indeed, the coarse registration of the geographical elements based on the OSM data has an average position error of 5 m and an average orientation error of 5°. An example of building registration is shown in Figure 17.

Figure 13. Example images of the augmented reality environment in an industrial setting.

Figure 14. Examples of the images used in the training set for vision-based UAV detection.



(**a**) (**b**)

Figure 15. Vision-based UAV detection performance. (**a**) Frequency of activation; (**b**) Accuracy.

Figure 16. Comparison of UAV position estimation methods: vision-based (left), EKF (onboard sensors) (right). A video of the experiment is available as Supplementary Material (video2).

Figure 17. Building visualization in augmented reality: result from coarse registration (left) and after manual correction (right).

4.4. Usability Experiments

The usability of the visuo-haptic interface has been evaluated in a simulated environment. The augmented reality environment (Figure 18) displays a video stream of the real workspace, and it includes a simulated 3D model of the UAV (animated using the ArduCopter SITL simulator), as well as a simulated nuclear source. A total of 10 participants was recruited for the evaluation. Each participant, after a training session, performed two tests in random order. In one case, the user performed the nuclear radiation detection task using the visuo-haptic interface. In the other case, graphical augmentation on the video stream was disabled, and the task was performed using only haptic feedback with additional information displayed on a side screen. In particular, the side screen displayed only the radiation spectrum and a 2D plot of the UAV trajectory with a mark indicating the current location of the maximum radiation found during the mission. The evaluation was carried out from the NASA-TLX questionnaire and the SPAM (situation present assessment method) [46]. Four categories of NASA-TLX have been considered, i.e., mental demand, performance, effort and frustration (rated for each task within a 100-points range with five-point steps). High values mean high mental demand, unsuccessful performance, great effort and high frustration or stress. Figure 19 shows the results in a star-like diagram. The VHAR interface received better scores for each parameter. One-way ANOVA analysis showed that the results were statistically significant for all parameters ($p \leq 0.05$).

Figure 18. Simulated environment developed for the usability tests. The UAV is a 3D model simulated using the ArduCopter SITL simulator.

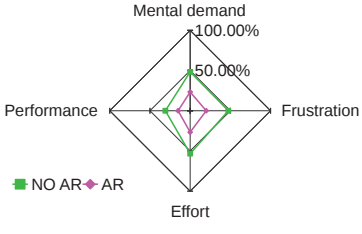


Figure 19. NASA-TLX questionnaire results. The AR line indicates average results of the VHAR interface; the NO AR line indicates average results without visual-feedback.

Participant were asked questions during the execution of the task according to the SPAM method. Questions were related to the current state of the task. In particular, three questions were asked: "What is the current heading of the UAV with respect to the magnetic north?", "How far is the UAV from the current location of maximum radiation?", "How far is the UAV from the camera?". Participants explicitly stated that the visual feedback of the VHAR interface was very useful to detect the nuclear radiation source and to determine distances. Indeed, statistically-significant results (Figure 20) were found for all of the parameters estimated by the users ($p \leq 0.05$). However, the results of the questionnaire indicated no significant differences in response time. Hence, it can be concluded that the main advantage of adding visual feedback is that it can improve the situation awareness of the task. Furthermore, the results indicate that adding visual feedback did not provide a significant improvement in position detection accuracy of the nuclear source.

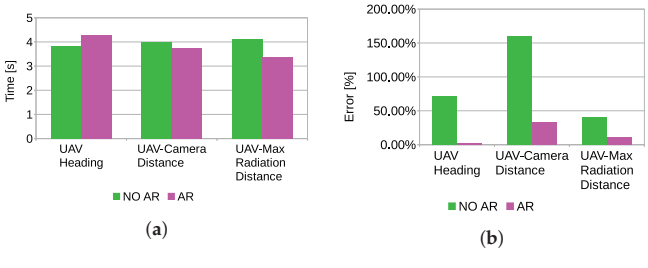


Figure 20. Situation present assessment method (SPAM) questionnaire results. (**a**) response time; (**b**) parameter estimation error.

5. Conclusions

This work presented a visuo-haptic augmented reality interface for UAV teleoperation with applications to the localization of nuclear sources in outdoor environments. A lightweight CdZnTe-based gamma-ray detector was developed and installed on a UAV. Two control modes were presented, which generate an attractive force feedback around the location of the most intense detected radiation, enabling efficient exploration of potentially dangerous areas. Experiments have been carried out in outdoor environments with a real radiation source, under the supervision of the public environmental protection agency. Visuo-haptic interfaces for UAV teleoperation in hazardous environments have not been considered in previous works. Results indicate that the proposed VHAR interface increases the situation awareness of the operator and reduces mental workload. Indeed, the real-time visuo-haptic feedback provides enough information to the user to oversee the entire mission. The augmented reality environment exploits a vision-based UAV detection algorithm that achieves a high detection rate. The main limitation of the VHAR interface is that the use of a single fixed camera on the ground reduces the available field of view of the environment. Therefore, in the proposed experiments, the UAV was clearly visible to the operator for safety reasons. Additional work is needed to achieve a fully-automatic registration of geographical data that are affected by large estimation errors. The developed prototype is available to third parties (http://www.imem.cnr.it/xdrone) such as agencies for environmental control, decommissioning companies and institutions with the responsibility of providing aid in the case of disasters and accidents involving nuclear or radiological materials.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/17/10/2234/s1.

Acknowledgments: This work has been developed within the project XDrone (Haptic teleoperation of UAV equipped with X-ray spectometer for detection and identification of radioactive materials in industrial plants) , Flagship Project "Factory of the Future", coordinated by the Italian National Research Council (CNR). We thank ARPA (Agenzia regionale per la protezione ambientale) Emilia-Romagna for providing supervision and support for the experimental evaluation and "Emiltest srl Controlli Non Distruttivi" for granting access to their facility in Gossolengo, Piacenza, Italy.

Author Contributions: Giorgio Micconi, Jacopo Aleotti and Stefano Caselli designed the VAHR interface. Giacomo Benassi, Nicola Zambelli, Manuele Bettelli and Andrea Zappettini designed the gamma-ray detector.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
VHAR	Visuo-Haptic Augmented Reality
CdZnTe	Cadmium Zinc Telluride

References

1. Boudergui, K.; Carrel, F.; Domenech, T.; Guenard, N.; Poli, J.P.; Ravet, A.; Schoepff, V.; Woo, R. Development of a drone equipped with optimized sensors for nuclear and radiological risk characterization. In Proceedings of the 2nd International Conference on Advancements in Nuclear Instrumentation Measurement Methods and their Applications (ANIMMA), Ghent, Belgium, 6–9 June 2011; pp. 1–9.
2. MacFarlane, J.; Payton, O.; Keatley, A.; Scott, G.; Pullin, H.; Crane, R.; Smilion, M.; Popescu, I.; Curlea, V.; Scott, T. Lightweight aerial vehicles for monitoring assessment and mapping of radiation anomalies. *J. Environ. Radioact.* **2014**, *136*, 127–130.
3. Martin, P.; Payton, O.; Fardoulis, J.; Richards, D.; Scott, T. The use of unmanned aerial systems for the mapping of legacy uranium mines. *J. Environ. Radioact.* **2015**, *143*, 135–140.
4. Martin, P.; Payton, O.; Fardoulis, J.; Richards, D.; Yamashiki, Y.; Scott, T. Low altitude unmanned aerial vehicle for characterising remediation effectiveness following the FDNPP accident. *J. Environ. Radioact.* **2016**, *151 Pt 1*, 58–63.

5. Pöllänen, R.; Toivonen, H.; Peräjärvi, K.; Karhunen, T.; Ilander, T.; Lehtinen, J.; Rintala, K.; Katajainen, T.; Niemelä, J.; Juusela, M. Radiation surveillance using an unmanned aerial vehicle. *Appl. Radiat. Isotopes* **2009**, *67*, 340–344.
6. Han, J.; Xu, Y.; Di, L.; Chen, Y. Low-cost Multi-UAV Technologies for Contour Mapping of Nuclear Radiation Field. *J. Intell. Robot. Syst.* **2013**, *70*, 401–410.
7. Neumann, P.; Bartholmai, M.; Schiller, J.; Wiggerich, B.; Manolov, M. Micro-drone for the characterization and self-optimizing search of hazardous gaseous substance sources: A new approach to determine wind speed and direction. In Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE), Phoenix, AZ, USA, 15–16 October 2010; pp. 1–6.
8. Newaz, A.A.R.; Jeong, S.; Lee, H.; Ryu, H.; Chong, N.Y.; Mason, M.T. Fast radiation mapping and multiple source localization using topographic contour map and incremental density estimation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
9. Okuyama, S.; Torii, T.; Suzuki, A.; Shibuya, M.; Miyazaki, N. A Remote Radiation Monitoring System Using an Autonomous Unmanned Helicopter for Nuclear Emergencies. *J. Nucl. Sci. Technol.* **2008**, *45*, 414–416.
10. Micconi, G.; Aleotti, J.; Caselli, S. Evaluation of a Haptic Interface for UAV Teleoperation in Detection of Radiation Sources. In Proceedings of the 18th IEEE Mediterranean Electrotechnical Conference (MELECON), Lemesos, Cyprus, 18–20 April 2016.
11. Micconi, G.; Aleotti, J.; Caselli, S.; Benassi, G.; Zambelli, N.; Zappettini, A. Haptic Guided UAV for Detection of Radiation Sources in Outdoor Environments. In Proceedings of the 3rd RED-UAS 2015 Workshop on Research, Education and Development of Unmanned Aerial Systems, Cancun, Mexico, 23–25 November 2015.
12. Lu, Y.; Macias, D.; Dean, Z.S.; Kreger, N.R.; Wong, P.K. A UAV-Mounted Whole Cell Biosensor System for Environmental Monitoring Applications. *IEEE Trans. NanoBiosci.* **2015**, *14*, 811–817.
13. Kurvinen, K.; Smolander, P.; Pöllänen, R.; Kuukankorpi, S.; Kettunen, M.; Lyytinen, J. Design of a radiation surveillance unit for an unmanned aerial vehicle. *J. Environ. Radioact.* **2005**, *81*, 1–10.
14. Sanada, Y.; Torii, T. Aerial radiation monitoring around the Fukushima Dai-ichi nuclear power plant using an unmanned helicopter. *J. Environ. Radioact.* **2015**, *139*, 294–299.
15. Frew, E.W.; Brown, T.X. Networking Issues for Small Unmanned Aircraft Systems. *J. Intell. Robot. Syst.* **2009**, *54*, 21–37.
16. Towler, J.; Krawiec, B.; Kochersberger, K. Radiation Mapping in Post-Disaster Environments Using an Autonomous Helicopter. *Remote Sens.* **2012**, *4*, 1995–2015.
17. Zollmann, S.; Hoppe, C.; Kluckner, S.; Poglitsch, C.; Bischof, H.; Reitmayr, G. Augmented Reality for Construction Site Monitoring and Documentation. *Proc. IEEE* **2014**, *102*, 137–154.
18. Sun, M.; Dong, N.; Jiang, C.; Ren, X.; Liu, L. Real-Time MUAV Video Augmentation with Geo-information for Remote Monitoring. In Proceedings of the Fifth International Conference on Geo-Information Technologies for Natural Disaster Management (GiT4NDM), Mississauga, ON, Canada, 9–11 October 2013; pp. 114–118.
19. Okura, F.; Kanbara, M.; Yokoya, N. Augmented telepresence using autopilot airship and omni-directional camera. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Seoul, Korea, 13–16 October 2010; pp. 259–260.
20. Iwaneczko, P.; Jędrasiak, K.; Nawrat, A. Augmented Reality in UAVs Applications. In *Innovative Simulation Systems*; Nawrat, A., Jędrasiak, K., Eds.; Springer: Cham, Switzerland, 2016; pp. 77–86.
21. Ai, Z.; Livingston, M.A.; Moskowitz, I.S. Real-time unmanned aerial vehicle 3D environment exploration in a mixed reality environment. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016.
22. Zollmann, S.; Hoppe, C.; Langlotz, T.; Reitmayr, G. FlyAR: Augmented Reality Supported Micro Aerial Vehicle Navigation. *IEEE Trans. Visualization Comput. Graph.* **2014**, *20*, 560–568.
23. Reyes, S.; Romero, H.; Salazar, S.; Lozano, R.; Santos, O. Outdoor haptic teleoperation of a hexarotor UAV. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 972–979.
24. Kanso, A.; Elhajj, I.H.; Shammas, E.; Asmar, D. Enhanced teleoperation of UAVs with haptic feedback. In Proceedings of the IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, Korea, 7–11 July 2015; pp. 305–310.

25. Lam, T.; Boschloo, H.; Mulder, M.; van Paassen, M. Artificial Force Field for Haptic Feedback in UAV Teleoperation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *39*, 1316–1330.
26. Carloni, R.; Lippiello, V.; D'Auria, M.; Fumagalli, M.; Mersha, A.; Stramigioli, S.; Siciliano, B. Robot Vision: Obstacle-Avoidance Techniques for Unmanned Aerial Vehicles. *IEEE Robot. Autom. Mag.* **2013**, *20*, 22–31.
27. Omari, S.; Hua, M.D.; Ducard, G.; Hamel, T. Bilateral haptic teleoperation of VTOL UAVs. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 2393–2399.
28. Masone, C.; Giordano, P.; Bulthoff, H.; Franchi, A. Semi-autonomous trajectory generation for mobile robots with integral haptic shared control. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6468–6475.
29. Hou, X.; Mahony, R.; Schill, F. Comparative Study of Haptic Interfaces for Bilateral Teleoperation of VTOL Aerial Robots. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *46*, 1352–1363.
30. Hou, X.; Mahony, R.; Schill, F. Representation of vehicle dynamics in haptic teleoperation of aerial robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 1485–1491.
31. Son, H.I.; Kim, J.; Chuang, L.; Franchi, A.; Giordano, P.; Lee, D.; Bulthoff, H. An evaluation of haptic cues on the tele-operator's perceptual awareness of multiple UAVs' environments. In Proceedings of the IEEE World Haptics Conference (WHC), Istanbul, Turkey, 21–24 June 2011; pp. 149–154.
32. Ruesch, A.; Mersha, A.; Stramigioli, S.; Carloni, R. Kinetic scrolling-based position mapping for haptic teleoperation of unmanned aerial vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 3116–3121.
33. Stramigioli, S.; Mahony, R.; Corke, P. A novel approach to haptic tele-operation of aerial robot vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 5302–5308.
34. Mersha, A.; Stramigioli, S.; Carloni, R. On Bilateral Teleoperation of Aerial Robots. *IEEE Trans. Robot.* **2014**, *30*, 258–274.
35. Mersha, A.; Hou, X.; Mahony, R.; Stramigioli, S.; Corke, P.; Carloni, R. Intercontinental haptic teleoperation of a flying vehicle: A step towards real-time applications. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 4951–4957.
36. Stegagno, P.; Basile, M.; Bulthoff, H.; Franchi, A. A semi-autonomous UAV platform for indoor remote operation with visual and haptic feedback. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3862–3869.
37. Del Sordo, S.; Abbene, L.; Caroli, E.; Mancini, A.; Zappettini, A.; Ubertini, P. Progress in the development of CdTe and CdZnTe semiconductor radiation detectors for astrophysical and medical applications. *Sensors* **2009**, *9*, 3491–3526.
38. Camarda, G.; Bolotnikov, A.; Cui, Y.; Hossain, A.; Kohman, K.; James, R. CdZnTe room-temperature semiconductor gamma-ray detector for national-security applications. In Proceedings of the 2007 IEEE Long Island Systems, Applications and Technology Conference, LISAT, Farmingdale, NY, USA, 4 May 2007; pp. 107–114.
39. Kowatari, M.; Kubota, T.; Shibahara, Y.; Fujii, T.; Fukutani, S.; Takamiya, K.; Mizuno, S.; Yamana, H. Application of a CZT detector to in situ environmental radioactivity measurement in the Fukushima area. *Radiat. Prot. Dosim.* **2015**, *167*, 348–352.
40. Kuvvetli, I.; Budtz-Jørgensen, C.; Zappettini, A.; Zambelli, N.; Benassi, G.; Kalemci, E.; Caroli, E.; Stephen, J.; Auricchio, N. A 3D CZT high resolution detector for X- and gamma-ray astronomy. *SPIE Int. Soc. Opt. Eng.* **2014**, *9154*, doi:10.1117/12.2055119.
41. Marchini, L.; Zappettini, A.; Gombia, E.; Mosca, R.; Lanata, M.; Pavesi, M. Study of surface treatment effects on the metal-CdZnTe interface. *IEEE Trans. Nucl. Sci.* **2009**, *56*, 1823–1826.
42. Zivkovic, Z.; van der Heijden, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognit. Lett.* **2006**, *27*, 773–780.
43. Sourimant, G.; Morin, L.; Bouatouch, K. GPS, GIS and Video Registration for Building Reconstruction. In Proceedings of the IEEE International Conference on Image Processing, San Antonio, TX, USA, 16 September–19 October 2007; Volume 6, pp. 401–404.

44. Karlekar, J.; Zhou, S.Z.; Nakayama, Y.; Lu, W.; Loh, Z.C.; Hii, D. Model-based localization and drift-free user tracking for outdoor augmented reality. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Singapore, 19–23 July 2010; pp. 1178–1183.
45. Min, S.; Lei, L.; Wei, H.; Xiang, R. Interactive registration for Augmented Reality GIS. In Proceedings of the International Conference on Computer Vision in Remote Sensing (CVRS), Xiamen, China, 16–18 December 2012; pp. 246–251.
46. Durso, F.T.; Dattel, A.R. SPAM: The real-time assessment of SA. In *A Cognitive Approach to Situation Awareness: Theory and Application*; Ashgate Publishing: Farnham, UK, 2004; pp. 137–154.

sensors

MDPI

Article

Implementation of an IMU Aided Image Stacking Algorithm in a Digital Camera for Unmanned Aerial Vehicles

Ahmad Audi [1,*], Marc Pierrot-Deseilligny [2], Christophe Meynard [1] and Christian Thom [1]

[1] Université Paris-Est, IGN, LaSTIG, LOEMI, 73 Avenue de Paris, 94160 Saint-Mandé, France; Christophe.Meynard@ign.fr (C.M.); Christian.Thom@ign.fr (C.T.)

[2] Université Paris-Est, IGN, LaSTIG, LOEMI, ENSG, 6-8 Avenue Blaise Pascal, 77420 Champs-sur-Marne, France; Marc.Pierrot-Deseilligny@ensg.eu

* Correspondence: Ahmad.Audi@ign.fr

Received: 30 May 2017; Accepted: 10 July 2017; Published: 18 July 2017

Abstract: Images acquired with a long exposure time using a camera embedded on UAVs (Unmanned Aerial Vehicles) exhibit motion blur due to the erratic movements of the UAV. The aim of the present work is to be able to acquire several images with a short exposure time and use an image processing algorithm to produce a stacked image with an equivalent long exposure time. Our method is based on the feature point image registration technique. The algorithm is implemented on the light-weight IGN (Institut national de l'information géographique) camera, which has an IMU (Inertial Measurement Unit) sensor and an SoC (System on Chip)/FPGA (Field-Programmable Gate Array). To obtain the correct parameters for the resampling of the images, the proposed method accurately estimates the geometrical transformation between the first and the N-th images. Feature points are detected in the first image using the FAST (Features from Accelerated Segment Test) detector, then homologous points on other images are obtained by template matching using an initial position benefiting greatly from the presence of the IMU sensor. The SoC/FPGA in the camera is used to speed up some parts of the algorithm in order to achieve real-time performance as our ultimate objective is to exclusively write the resulting image to save bandwidth on the storage device. The paper includes a detailed description of the implemented algorithm, resource usage summary, resulting processing time, resulting images and block diagrams of the described architecture. The resulting stacked image obtained for real surveys does not seem visually impaired. An interesting by-product of this algorithm is the 3D rotation estimated by a photogrammetric method between poses, which can be used to recalibrate in real time the gyrometers of the IMU. Timing results demonstrate that the image resampling part of this algorithm is the most demanding processing task and should also be accelerated in the FPGA in future work.

Keywords: UAVs; stacking; inertial measurement unit; image processing; photogrammetry; real-time; co-design

1. Introduction

Over the last decade, Unmanned Aerial Vehicles (UAVs) have been widely used for civil applications [1], especially for environmental surveys [2,3]. Their growing popularity has led the LOEMI (Laboratoire d'Opto-électronique, de Métrologie et d'Instrumentation) team of Institut national de l'information géographique (IGN)/LaSTIG (Laboratoire des Sciences et Technologies de l'Information Géographique) to design and produce an ultra-light smart digital aerial camera, more suitable for photogrammetric applications than consumer cameras.

This camera was originally used to exploit photogrammetric and metrological surveys using UAVs while some research works were already underway, such as [4,5]. The next targeted applications

involve shots under cloudy conditions, night-time surveys and narrow spectral bandwidth imagery, which usually imply a long exposure time. Images acquired with a long exposure time using a camera embedded on UAVs may exhibit motion blur due to the erratic movements of the UAV. This paper presents an image stacking algorithm that can provide an accurate blur-free composite image of photogrammetric quality with an equivalent long exposure time. An initial presentation of this work has been done in [6].

In this context, the main part of our algorithm is based on feature points image registration [7–9]. This technique consists of feature detection, feature matching, transformation estimation, image transformation and resampling. The ultimate objective of our work is to achieve real-time processing of the algorithm, leading to only the resulting image needing to be saved and therefore ensuring a better overall frame rate. The inertial measurement unit mounted inside the camera can gives the relative orientations of the camera during shoots, quantifying an initial amount of motion between poses. This consequently speeds up the expensive computing of matching feature points considerably, which is one of the most time-consuming tasks, making our stacking algorithm more appropriate for real-time performance. Images are acquired with a high frame rate, which implies minor deformation between images.

Two approaches are introduced in this paper. For the first approach, the UAV is assumed to be quasi-stationary during exposures; only camera attitude variations are considered; and only image deformation due to camera rotation is corrected. One interesting by-product of this method is a better estimation of variation in the orientation of the camera between shots, which can be used to recalibrate the IMU gyrometers. Other sensors, such as LiDAR sensors on the UAV, can then benefit from the resulting better quality of the IMU data. The second approach uses the 2D planar homography transformation supposing that not only the 3D rotation is taken into account on camera movement, but also the translation when the scene is nearly planar.

Our algorithm works completely automatically, whether offline or online. Some tests have been completed on several types of UAVs, including the Copter 1B UAV equipped with the IGN camera to assess the accuracy of the method, and some of the results are presented.

2. Related Works

This work relates to two fields: image stacking and visual-inertial fusion techniques.

The stacking technique creates a single image by combining many images that may be superimposed. One stacking method is HDR (High Dynamic Range) imaging, which combines multiple images with different exposure levels in order to obtain a single image with a large dynamic range [10–12]. These works assume that the images are superimposable; differing from our method on this basis.

Many visual-inertial fusion techniques have been proposed in the literature. We hereby present some of them. The authors of [13] suggests using the inertial measurement sensors to measure the camera's acceleration and angular velocity during exposure, which can then be used to estimate the blur function and then deconvolve the blurred image to achieve sharp results. Their method only works for very small movements, and many cited limitations, such as sensor accuracy and noise, constrain their usage. In the same context, the IMU is used in [14] during aerial shooting by a camera embedded on an aircraft to eliminate blur and provide a deblurred image. The authors' objective is close to ours, but their method only covers very small movements. Their results show that images with more than seven pixels in motion blur record minor improvements. Other works relate to the same objective using deblurring algorithms without IMU data: [15–17].

Some studies in literature expand on the combination of IMU and computer vision for tracking purposes. For example, a vision-assisted algorithm is combined to improve positioning and to track the performance of mobile devices for SLAM (Simultaneous Localization And Mapping) applications [18]. Parts of this approach seem similar to ours. It uses IMU gyrometer measurements to estimate rotation angles to improve the accuracy of trajectory estimation, but they also use reference nodes as control

points identified in the vision processing phase to correct the propagation of IMU errors. Therefore, this method is not optimal if no reference nodes are available along the trajectory. The authors of [19–21] use a combination of inertial sensors and a camera for motion tracking in augmented reality applications. Most fusion sensor works use inertial sensors to aid computer vision methods, making the system more accurate and more robust.

Some parts of our work have also been explored in alternative domains. For example, a real-time video stabilization algorithm for unmanned aerial vehicles is proposed in [22], but this method excludes any aided sensors, such as the IMU, to improve the accuracy of motion estimation or to reduce the tracking time for features in consecutive frames.

3. Materials and Methods

3.1. Hardware

The system presented in Figure 1 consists of a UAV equipped with an IGN lightweight photogrammetric camera [23]. The camera used to implement our algorithm is shown in Figure 2.

Figure 1. Institut national de l'information géographique (IGN) camera with the Copter 1B UAV developed by Wikhydro.

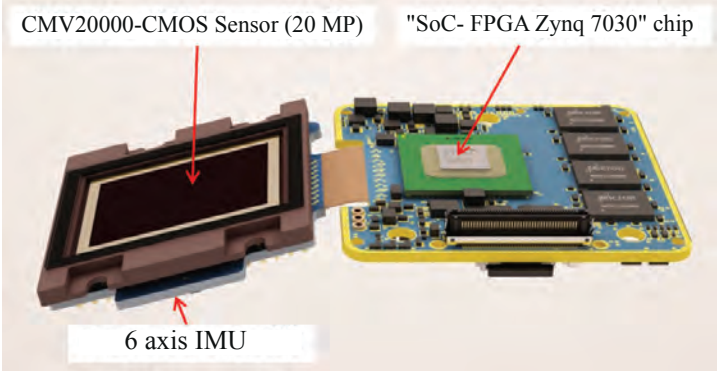


Figure 2. The CMOS sensor with the inertial sensors and the embedded intelligence.

This camera was developed at the French National Mapping Agency (IGN) by the LOEMI team. It is based on a global shutter CMOS 20MP sensor (CMOSIS CMV20000) from CMOSIS, Anvers, Belgium, and a Zynq 7030 SoC/FPGA from Xilinx, San Jose, CA, USA, allowing some image processing algorithms to be implemented in hardware, making them real-time compliant. It has both embedded intelligence, in the form of two ARM cortex A9 CPUs running a Linux OS and application software, and a programmable logic subsystem originally used to acquire image data from the sensor. A standard

35-mm optical lens is used for this purpose. The camera prototype we used was fitted with a Bayer sensor and has a resolution of 20 MP (5120 × 3840). Our work aims to be used for a hyperspectral imaging system consisting of several monochrome IGN cameras; thus, we use the 5-MP (2560 × 1920) grayscale images obtained from the original pixel-stream by averaging each green pair of the Bayer pattern.Recently, LOEMI team engineers integrated a GNSS (Global Navigation Satellite System) receiver, which can be used in future works to take into account UAV translation during shots.

The ICM-20608-G IMU sensor from InvenSense used in the camera is a low-cost MEMS (Micro-Electro-Mechanical System) device. It measures angular speed around the three axes with low latency and at a high frequency. The main disadvantage of MEMS technology is the reduced performance in terms of accuracy and stability: the slow evolving bias is the main type of error that affects measurements, as it accumulates over time due to the integration of angular velocity. We will show that it can be estimated by sensor fusion and then eliminated from the subsystem. Laboratory thermal calibration is used to eliminate the temperature effects caused by the dissipation of heat inside the camera (75 °C). The IMU sensor is tightly fixed inside the camera to ensure that both sensors experience the same 3D rotation as shown in Figure 3. We use the IMU subsystem to obtain the estimation of the absolute 3D rotation of each pose (R^n_{imu}) acquired by the CMOS sensor.

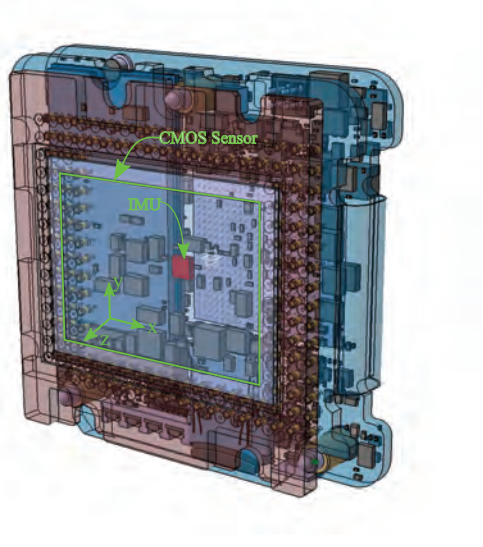


Figure 3. The IMU and the CMOS sensor inside the IGN camera.

3.2. Calibration of the Camera

In photogrammetric processes, cameras should be calibrated to take accurate 3D geometric measurements from their images. This calibration involves estimating their internal orientation. This is obtained using the camera self-calibration bundle adjustment method via MicMac, a free open-source photogrammetric software developed by IGN [24]. The method used in MicMac is based on SIFT points and the matching relationship in stereo images. It is applied to each pair of multiple view images taken using an un-calibrated camera. We use the radial standard distortion model [25] in our work.

3.3. Images and Data Processing

This algorithm is implemented and executed in the camera and is intended to benefit from the FPGA acceleration. It was initially developed on a desktop PC.

3.3.1. Algorithm

The architecture of the algorithm is illustrated in Figure 4.

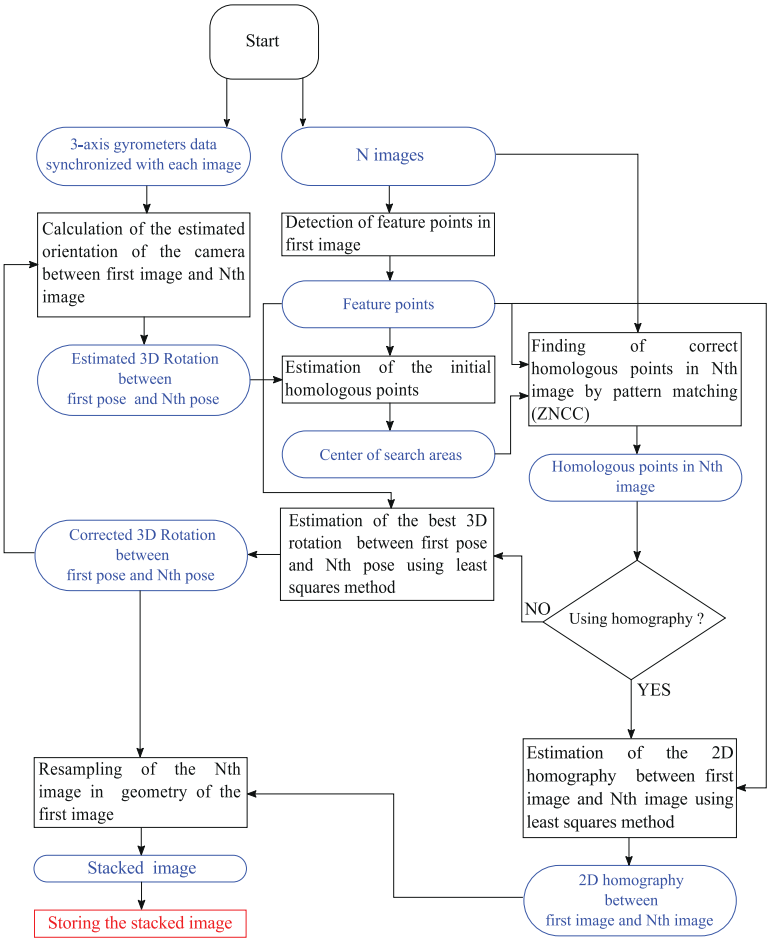


Figure 4. Architecture of the algorithm.

- Camera attitude is computed using measurements from the IMU subsystem.
- Acquisition of a burst of 10 images by the camera, saved in RAM at the maximum frame rate of the CMOS (30 images/s); the time difference between successive images is 33 ms.
- Detection of a reasonable number of feature points in the first image. This part is implemented in hardware.
- For the next sequential images:

    1. In this image, the predicted positions of feature points are computed using IMU orientation to speed up the next step.
    2. Template matching is used to determine the accurate homologous position of these feature points.

3. (a) 3D rotation between the two poses is estimated using the least squares method with outliers elimination. This information may possibly be used as feedback in the IMU subsystem to improve its overall quality by the real-time determination of sensor drifts.
   (b) 2D homography transformation between the two images is estimated using the least squares method with outliers' elimination.
4. The image is resampled in the geometry of the first image.
5. Stack it.

- Save the stacked image.

3.3.2. Features Detection

The detection of features is a fundamental task in computer vision applications, such as motion detection, object recognition and tracking, 3D reconstruction, visual odometry, etc. A feature can be defined as a pattern or distinct structure with special properties found in an image. Most of the time, these features are used in further processing steps. In our work, the feature detection algorithm was mainly selected on the basis of its ability to find a reasonable number of good features in a single image (leading by correlation to unambiguous matches in other images) and by the possibility for easy implementation in hardware. After analyzing the main existing feature detectors [26–28], we selected the FAST detector algorithm developed by [29].

This detector is many times faster in software than other existing detectors [30], proving that it is conceptually simple, leading to the fact that it can be easily implemented in hardware. In our application, we will not consider the stability of the detector because feature points will only be detected in the first image and then searched for by template matching in other images. Another advantage is that FAST requires far less resources than other detectors and, in parallel, offers sufficient performance for our application.

On the other hand, the FAST detector is dependent, as many other detectors, on a threshold, which must be set according to the image noise level, especially as FAST has been shown to be sensitive to high-level noise [26,27]. However, in our camera, the noise level is reduced to be constant in RMS with respect to gray levels by using the non-linear LUT (Look-Up Tables), which translate the 12 bits of pixel data into 8 bits. We can then use a threshold value independent of image brightness level and content.

3.3.3. Features Selection Approach

The authors of [31] show that FAST tends to detect a high number of feature points in high textured zones, but not in low textured zones. This means that the feature points may be concentrated only in high textured zones or in a specific zone of the image. This non-uniform distribution of points makes the estimation of transformation (rotation or homography) between images less accurate [32]. To limit the high number of feature points in high textured zones and ensure, if possible, that even low textured structure zones may obtain feature points when using a low threshold value, we use a grid-adapted feature selection approach that can maintain a compromise between achieving a more uniform distribution of feature points over the image and implementing this approach in hardware in a simple way.

This approach involves partitioning the image into blocks and only retaining the first feature found by the detector in each block. It is easy to implement in hardware and can improve the uniformity of the distribution of the features over the image as shown in Figure 5: we can see that no feature points are detected in very low textured zones, but a grid-like pattern is observed in high textured zones where a high number of potential feature points are shown.

Figure 5. Uniform distribution of features over the image. The threshold intensity value of the detector is set to 7. This image is acquired with a low exposure time, but brightened artificially to clearly show the feature points. The size of the grid-adapted features is set to 100 × 100 blocks.

3.3.4. The FAST Algorithm

Features from the FAST detector are based on a 7 × 7 pixels neighborhood centered on the candidate pixel p. The size (7 × 7) of the window is chosen according to the original implementation of the FAST detector. The original FAST-N algorithm compares the intensity of a feature candidate p with each pixel on a 16-pixel Bresenham circle surrounding p. The candidate pixel is detected as a feature point if N contiguous pixels on the Bresenham circle with the radii r all brighter or all darker than p by a threshold t. Originally, N was chosen to be 12 as shown in Figure 6.

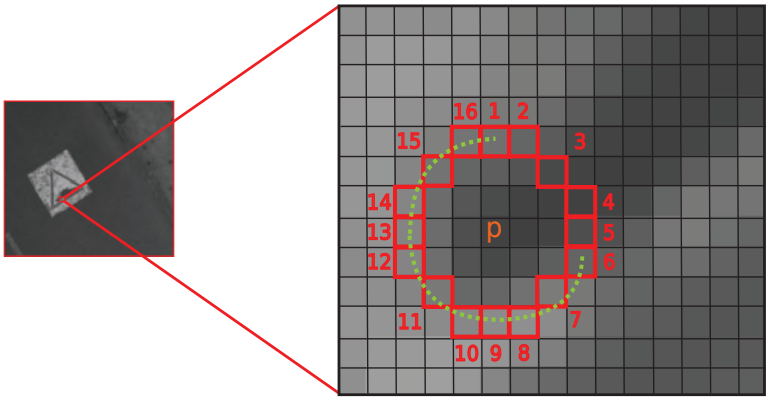


Figure 6. Features from Accelerated Segment Test (FAST)-12 image feature detector illustration.

Hardware Implementation

Processing high resolution images is considerably expensive in terms of computing. The SoC/FPGA integrated into the IGN lightweight photogrammetric camera can provide a suitable solution to speed up this processing. Thanks to this SoC that combines dual ARM cores and an FPGA, hardware/software can be co-designed allowing some processing-intensive tasks to be executed using the programmable logic array and leaving the CPUs free to be used for other tasks. Several hardware implementations have been proposed for FAST. For example, the authors of [33] not only test if candidate pixels are considered as feature points, but also compute a corner score for every feature detected for use in another module called NMS (Non-Maximal Suppression) to reduce the occurrence of adjacent features. Here, we present a simpler hardware implementation of FAST aiming to detect good feature points for the next processing stages with the addition of the selection of grid-adapted features.

Hardware Architecture

The design has the following characteristics:

- Image dimensions and threshold configured by software.
- Input pixels format: 8-bit grayscale value.
- Input rates are constant at 1 pixel/clock. For now, the image is read from the memory for debugging reasons. We could improve the design to process the stream of pixels directly from the sensor to achieve real-time performance.
- The output (x,y) coordinates of the feature points (2×16 bits) are written to memory.

The design is divided into two modules: the FAST detector and the selector. The FAST detector module computes a validity flag for each pixel under investigation. The selector module picks the first valid feature detected by the FAST detector module in each block of the grid-adapted image.

The "FAST detector" module is divided into two modules: the "(7×7) window" and the "corner test".

We need simultaneous access to the 16 pixels located on the Bresenham circle and the candidate centered pixel, requiring a 7×7 window containing the 17 pixels under investigation. For this purpose, we used 6 arrays as FIFO buffer whose depth is equal to the width of the image. Furthermore, we used (7×7) 8-bit registers to preserve the intensity values of the window pixels to be processed by the "corner test" module.

The "corner test" module only computes the 12 contiguous pixels with an intensity value greater than the candidate pixel intensity value plus the threshold, excluding those with an intensity value less than the candidate pixel intensity value minus the threshold. This is sufficient to obtain a reasonable number of good feature points. In the first clock cycle, the intensity value of each pixel located on the Bresenham circle is compared with the intensity value of the centered candidate pixel plus the threshold. The output of these 16 comparisons forms a 16-bit wide logic sequence of '0' and '1': '1' if the intensity value of the pixel in question is greater than the constant and '0' in all other cases. In the next clock cycle, the 16-bit sequence is simultaneously compared with the 16 rotations of 12 contiguous '1' and 4 contiguous '0'. The output (validation signal) of the "corner test" module is '1' if this sequence matches one of the 16 combinations and '0' in all other cases. Therefore, the output coordinates are delayed by two-clocks.

The "selector" module uses a flag register to deactivate the other valid features in each block. At the beginning of each grid row, the flag register is initialized to *'0'*. This module also controls the end of image by testing row and column input data. At the end of the image (last block), it sets the column and row output data to *'0'* and the feature valid output to *'1'*, so we can detect the end of the features list.

The Global Design

The global design presented in Figure 7 was implemented and tested on the camera Xilinx/Zynq. Our solution uses the standard DDR RAM memory for storing both the image and coordinates of feature points. The AXI bus was used to connect the processor system and memory: the AXI bus provides direct access to the DDR memory. The AXIS transmits stream data between modules in master/slave modes. As a first step, to verify the design of this algorithm, the image in the RAM is read by the first DMA engine and transmitted to the "grid-based FAST detector" module. The resulting feature points coordinates are written back in the memory by the second DMA engine. The timing result of the global design execution is 0.024 s as shown in Table 2; this is the time taken to transfer 2560 × 1920 pixels from the RAM. In the final design, the process will be applied directly to the stream of pixels from the sensor.

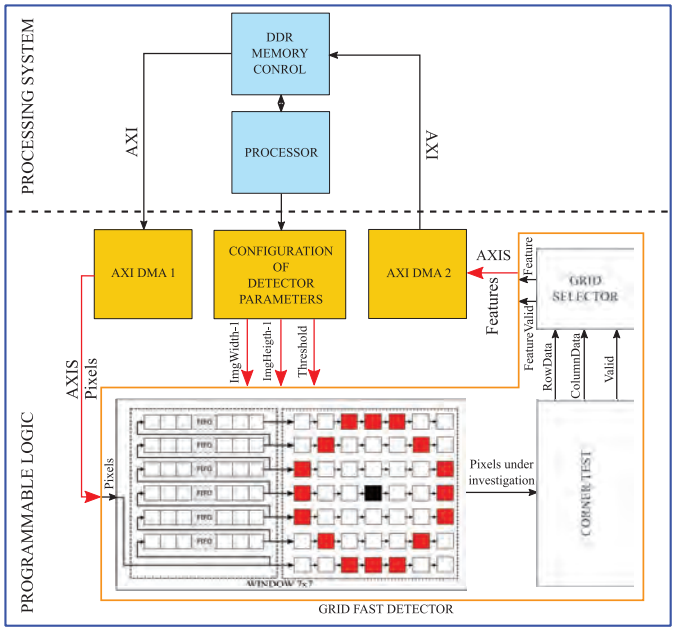


Figure 7. The global design of the grid-based fast detector implementation.

Resources Usage for the FAST Detector Design

Table 1 illustrates the number of resources used in the FPGA for the implemented design. Compared with other proposed hardware implementations, our design uses far fewer FPGA resources. As an example, the architecture presented in [33] uses about 10-times more look-up tables because they not only test if the candidate pixel is a feature point, but also calculate a corner quality score to use it in another module in order to select the feature point with the maximum score from other adjacent feature points.

Table 1. Resources usage of the FAST detector design in Xilinx/Zynq-7030 FPGA.

Resource	Used	Available	Utilization (%)
Slice LUTs	232	78,600	0.30
Slice of 4 registers	472	157,200	0.30
Memory (blocks of 36 KB)	6	265	2.26
Clocking	1	32	3.13

3.4. Estimation of the Predicted Positions of Homologous Points in Images

Using the internal orientation parameters produced by the optical calibration process and the 3D rotations between poses from the IMU, we can compute the predicted positions of homologous points of detected features in other images as illustrated in Figure 8.

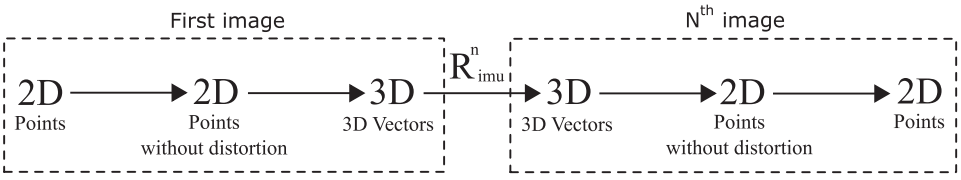


Figure 8. The various steps required to obtain the approximate initial homologous point in another image. R^n_{imu} is the 3D rotation obtained by IMU measurements.

Let us consider that $P^1_i(u,v)$ is a 2D point in the plane of the first image. First, the geometric distortion is removed, then $P^1_i(u,v)$ is transformed to a 3D vector $\overrightarrow{V^1_i}(x,y,z)$ in the camera coordinates system; the origin of this system is at the optical center of the camera. After applying the inverse of the 3D rotation (R^n_{imu}) to $\overrightarrow{V^1_i}(x,y,z)$, we obtain a new vector $\overrightarrow{V^n_i}(x',y',z')$ that will be projected into the plane of the N-th image to obtain the corresponding 2D point $P^n_i(u',v')$ after applying the geometric distortion. The accuracy of the estimated 3D camera rotation (R^n_{imu}) depends on the accuracy of the IMU. In fact, the main limitation of low-cost MEMS IMUs is the drift due to accumulated uncorrected bias. Subsequently, this rotation (R^n_{imu}) only provides an approximate position for homologous points in other images as shown in Figure 9.

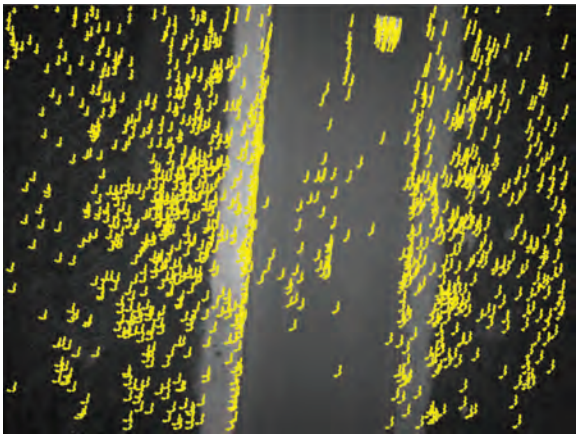


Figure 9. Positions of the features predicted by the IMU for all images.

We consider this predicted position as the center of a search area in which the correct position should be found by template matching as illustrated in Figure 10. This step brings the algorithm closer to real-time performance.

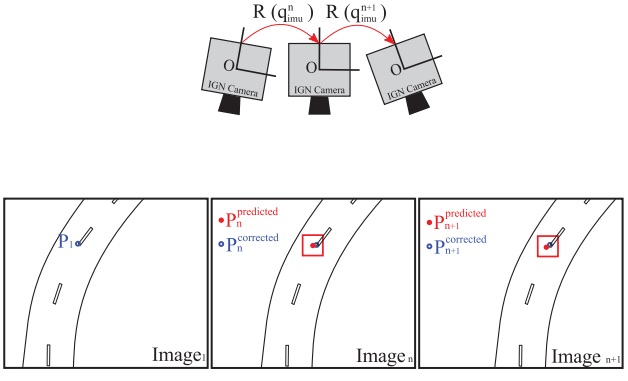


Figure 10. Rotation between three images.

3.5. Template Matching with ZNCC

Once the centers of search areas in other images have been estimated, the candidate homologous points can be found precisely by template matching as shown in Figures 11 and 12. Due to the fact that the deformations between images are small, the correlation scores are high (close to the maximum correlation value (=1), so we perform here a simple filtering to keep only the pairs with a correlation score higher than 0.85 in order to eliminate possible outliers. We chose ZNCC (Zero mean Normalized Cross-Correlation) [34] because it is a simple and reliable method that can be used to measure the similarity between image patches and is not influenced by variation in brightness and contrast [35]. Whereas ZNCC does not perform well with significant rotation and change of scale between two images, one of our assumptions for this work is that rotation between images is small.

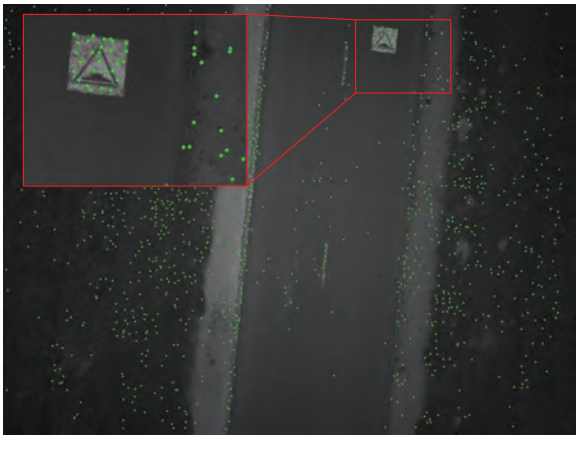


Figure 11. Feature points detected using the FAST algorithm in the first image; the threshold chosen here is 7. The number of grid blocks is defined as 100 × 100; image resolution: 2560 × 1920 pixels.

The feature point window and 7×7 FAST detector window should have the same size so as to obtain an accurate measurement of the degree of similarity between image patches through the correlation process. The size of search areas depends on the accuracy/bias of IMU gyrometers, which means that the size of search areas should be chosen to be large enough (11×11 pixels) between the first and second images to accommodate prediction errors. This size can then be reduced for all other images after eliminating bias. This correlation could be implemented in hardware to enable real-time processing [36].

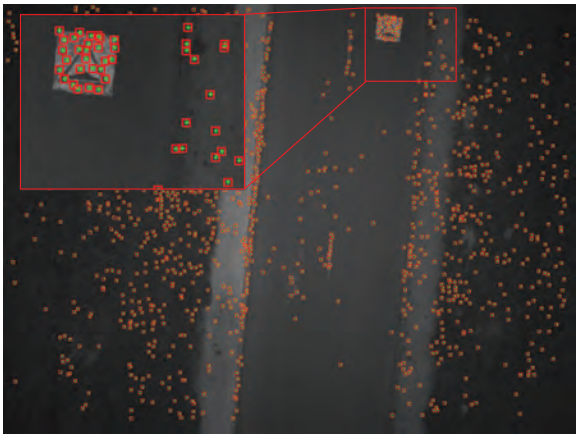


Figure 12. Search areas obtained by the IMU gyrometers between the first image and the 10th image. The green points represent the homologous points obtained by template matching. The size of each search area is 11×11 pixels; image resolution: 2560×1920 pixels.

The Sub-Pixel Correlation

One of drawbacks of ZNCC is that its resolution is limited to one pixel, i.e., the peak location coordinates have integer values. Matching precision can, however, be improved directly by interpolating the cross-correlation surface to a higher spatial resolution in order to locate the correlation peak with sub-pixel accuracy [37]. The sub-pixel location can be approximated by fitting a two-dimensional model through the correlation coefficients around the peak location. Several models are proposed in the research community such as parabola fitting and Gaussian fitting [38–40]. In our work, we choose the simple parabolic curve fitting model that computes the sub-pixel peak location separately for the two orthogonal directions where the curve reaches its maximum. Results presented in Figure 15 demonstrate that with this simple model, the sub-pixel approach improves the precision and accuracy of ZNCC-based image matching, eliminating an average of 0.15 residual pixels^2 with respect to the precision of the pixel-level matching, so it is not then necessary to use another complex fitting model. The rounding of uniformly-distributed random variables caused error in the estimation of the variance of 1/12 pixels^2. Since we have 2 independent variables x and y, the overall error will see its variance increased by 1/6 pixels^2, that is 0.17. We see that our method corrects nearly all of this noise.

3.6. Estimation of Transformation between Images

Now that we have good homologous pairs, the next step is to establish the best estimation of a geometric relation that maps pixel coordinates from one image to another. Two geometric transformations can be established here, 3D rotation between poses and 2D homography between images. Which of these two transformations is selected depends on the real movement of the camera.

3D rotation is used when the UAV is quasi-stationary and only attitude variations are considered, and it is beneficial for improving the robustness of the system because it only has three degrees of freedom. This approach is not optimal if the real movement of the camera is not a pure rotation. Moreover, in this case, it is not possible to recalibrate the IMU gyrometers directly from this transformation.

2D homography is used when the camera movement consists of both rotation and translation. This transformation has eight degrees of freedom, which means that it can partially absorb the translation movement of the camera. On the other hand, this approach is more sensitive to errors and does not work perfectly when the scene is not planar.

3.6.1. Estimation of 3D Rotation Camera Using the Least Squares Method

Once given a set of 3D vectors $\overrightarrow{V_i^1}(x, y, z)$ and corresponding set of 3D vectors $\overrightarrow{V_i^n}(x', y', z')$, the 3D rotation R_{img}^n is estimated using the least squares method, which minimizes the errors between estimated 3D vectors $\overrightarrow{\hat{V}_i^n}$ and renormalized 3D vectors $\overrightarrow{V_{i,corr}^n}$ resulting from the correlation. The IMU can be used to obtain an initial solution, and then, only one iteration is needed to be close enough to the true solution; several more iterations are then used to automatically eliminate outliers due to false correlations.

The relative 3D rotation between R_{img}^n and R_{imu}^n is used to estimate, in real time, most of the drift of the IMU bias and then improve the accuracy of the estimation of the consecutive 3D rotation R_{img}^{n+1}.

3.6.2. Estimation of 2D Homography Using the Least Squares Method

With the 2D planar homography model, geometrical transformation can be calculated easily using the least squares method, with the 2D points obtained by the FAST detector in the first image and their homologous points found by template matching in the N-th image. Outliers with errors greater than three pixels are automatically eliminated in order to obtain a robust estimation of the best fitting homography.

3.7. Resampling Images and Generating the Stacked Image

We used bilinear interpolation to find the resampled value of each pixel from the first image in the N-th images. The stacked intensity values are then saved with a coefficient that can be used to encode the values of the stacked image on 8 bits instead of 12 bits and optionally increase the brightness of the image. This stage is the most computationally-intensive task in the algorithm, as shown in Table 2. However, the process of mapping each pixel from the first image to the other images can be largely accelerated by using a bilinear approximation. This approach consists of dividing the first image into M $\times$ N blocks and computing for each block the bilinear function that matches the real transformation on the corners of the block, then using the nearest neighbor resampling method to resample each pixel in other images. Thus, we significantly reduce the number of arithmetic operations and memory access use. This method is very effective in software, and we currently aim to implement it in hardware because it uses less resources (especially DSP resources) and requires less memory access.

Table 2. (1) Initial geometrical transformation with bilinear interpolation; (2) accelerated geometrical transformation using the bilinear function method with bilinear interpolation; (3) accelerated geometrical transformation using the bilinear function method with nearest-neighbor resampling.

	Feature Detection (402 Points)	Matching (402 Points) & Estimation of 3D Rotation	Resampling			
			Rotation			Homography
			(1)	(2)	(3)	(1)
PC (Soft)	0.21	0.08	4.72	0.94	0.66	4.00
PC (Soft/2 Threads)			2.46	0.42	0.38	2.06
Camera (Soft)	1.64	0.83	36.94	7.24	5.28	25.17
Camera (Hard)	0.02					
Camera (Soft/2 Threads)			18.48	3.81	2.65	12.57

4. Results

We performed a few experiments to quantify the accuracy and performance of our algorithm. We used the hardware described in Section 3. Several datasets were acquired by the IGN photogrammetric camera, some of which are shown here. Each dataset consists of a sequence of ten images acquired at the maximum frame rate of the CMOS (30 images/s) by transmitting a wireless signal to trigger image acquisition when the UAV is hovering. The value of exposure time of acquisition is set to 1 ms to be closer to hyperspectral images with a low dynamic range.

4.1. Timing Results (in Seconds)

All tests were run on two platforms: firstly, a HPZ210 Workstation with an Intel Xeon 3.30-GHz processor and 8.0 GB of RAM; secondly, the photogrammetric IGN camera with 666-MHz dual-core ARM Cortex-A9 processors.

4.2. Accuracy of IMU

Figure 13 shows the 3D rotations respectively experienced by the IMU and estimated by the photogrammetric method. We chose here a dataset in which the bias of IMU drifts is significant.

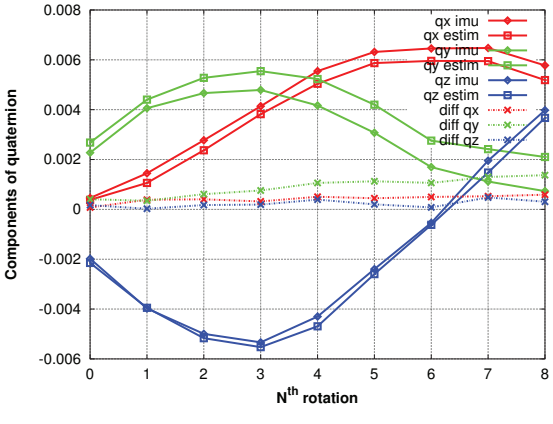
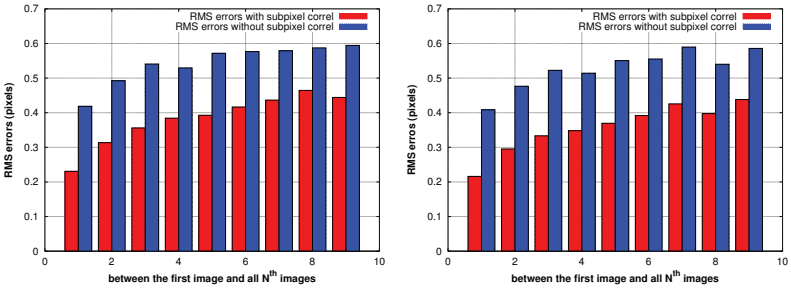


Figure 13. The two 3D rotations of the camera between poses.

4.3. Accuracy of Transformation Estimation

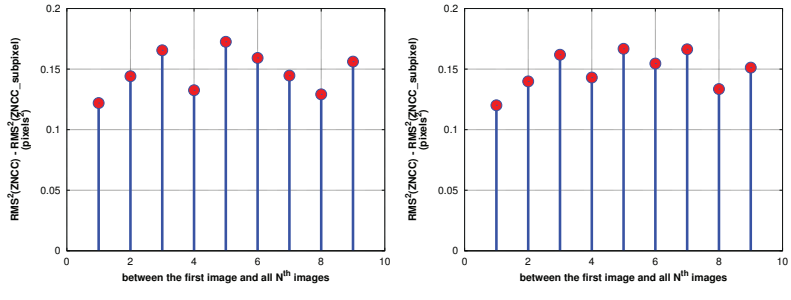
4.3.1. Case of Pure Rotation Camera Movement

Figures 14–16 illustrate a case in which the UAV has experienced a pure rotation movement. In order to explain the augmentation of RMS errors over images, we separate the calculation of errors between grass regions and road regions.



(**a**) *RMS*errors for 3D rotation. (**b**) *RMS* errors for 2D homography.

Figure 14. Residuals of the estimation of the 3D rotation and 2D homography.



(**a**) 3D rotation. (**b**) 2D homography

Figure 15. RMS^2 difference between ZNCC without sub-pixel and ZNCC with sub-pixel accuracy.

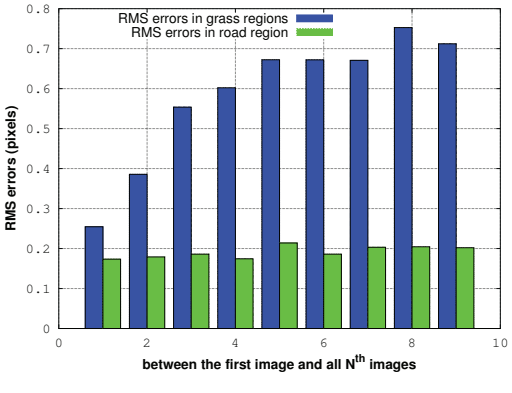
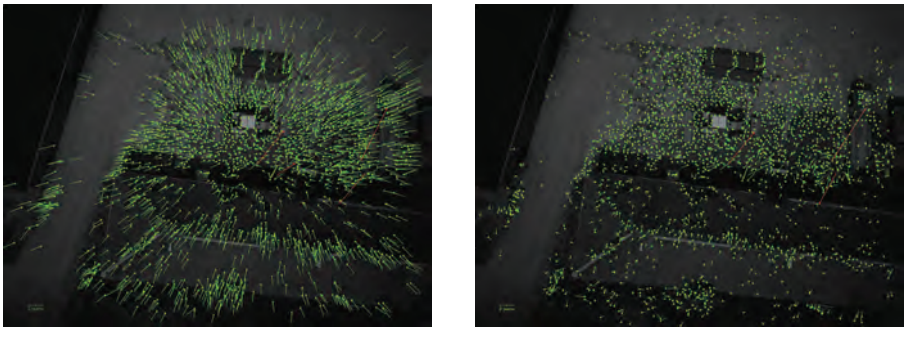


Figure 16. Residuals of the 3D rotation estimation between the first image and all other images in grass and road areas.

4.3.2. Case of Non-pure Rotation Camera Movement

Figures 17 and 18 illustrate a case in which the UAV has suffered a Z-translation movement caused by a free-fall.



(**a**) The case of 3D rotation. (**b**) The case of 2D homography

Figure 17. Residues obtained using the transformation estimation between the first image and the 10th image in the case of non-pure rotation mouvement of the camera. Red arrows represent the outliers.

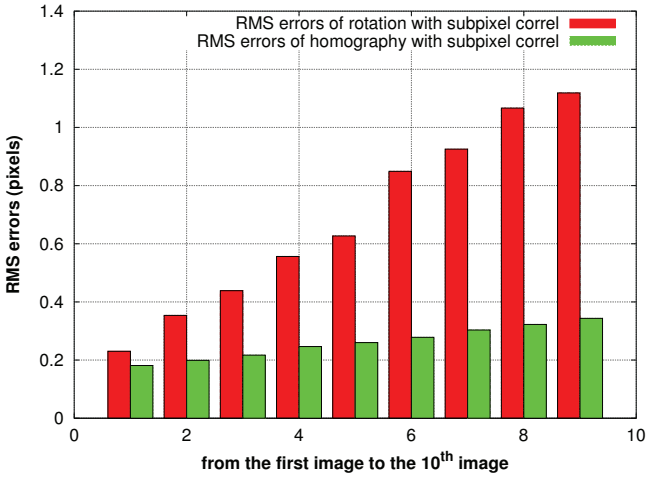


Figure 18. Residuals of the estimation of 3D rotation and 2D homography between the first image and all other images in the case of UAV translation.

4.4. Quality of Resulting Image

Figure 19 represents the final stacked image resulting from our algorithm. The red part shows the rotation experienced by the camera between the first and last image. It is possible here to use a brightness factor to illuminate the image.

Figure 19. The stacked image. The red part is the non-common part between the first image and the 10th image.

5. Discussion

5.1. Timing Results

Table 2 shows that the resampling part of the algorithm is the most consuming task for the two types of transformations. Benefiting from the presence of dual-core processors in the camera, we use the OpenMP library to accelerate the image resampling part; the number of threads was set to two; the processing time of this part was reduced by half. The acceleration factor when using the nearest-neighbor resampling method is seven. This part of the process should be accelerated even further with the help of the programmable logic (FPGA) to achieve real-time performance.

5.2. Accuracy of IMU

Our results show that the errors of components increase slowly and linearly over time, implying that the bias is constant over the sequence duration. This bias can be then estimated by computing the 3D rotation linking the attitude estimated by the photogrammetric method and the one obtained by the IMU. This rotation can then be used to obtain a better initial rotation for the following images, leading to the reduction of the size of the research areas and then of the correlation time, or to correct the biases' values in the IMU subsystem; which could be beneficial to other sensors (i.e., LiDAR) on the UAV.

5.3. Accuracy of Estimation of Transformations

5.3.1. Case of Pure Rotation Movement of the Camera

For the two transformations, the RMS errors are less than 0.5 pixels for all of the images that are processed. Our results presented in Figure 14 show also a clear improvement of accuracy when we use the sub-pixel correlation for both transformations. Figure 15 demonstrates that the difference in variance between sub-pixel correlation and non-sub-pixel correlation almost equals $(2/12)$, corresponding to the variance in the uniform distribution; which means that the sub-pixel

correlation works as expected. Additionally, the estimation of the homography has less residuals than the rotation, consistent with the fact that former has more degrees of freedom.

In both cases, the RMS values of errors increase linearly over images. Figure 16 shows that this is due to the movement of the grass next to the road: the residuals in the road region are constant and significantly lower. This indicates also that the noise of our process is only of the order of 0.2-pixel RMS.

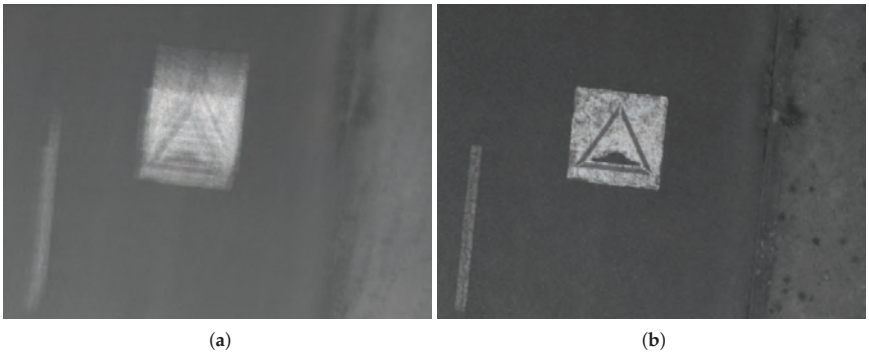
5.3.2. Case of Non-pure Rotation Movement of the Camera

In this case, the estimation of the rotation with our photogrammetric method does not work optimally because the optical center is moving; thus, the residual errors are bigger than expected. They converge to the center of the image as shown in Figure 17a because the UAV is experiencing a free-fall. These residuals, which are not compensated by the rotation, can be absorbed by the 2D homography, which includes a scale factor: Figure 18 demonstrates that the residual errors of the estimated homography are constant over images and much smaller than those of the 3D rotation. However, as expected, they are more important in zones where the surface gets away from the average plan in the scene such as the rooftop shown in Figure 17b. This means that for these parts of the image, the stacked image will be slightly blurred.

Thanks to the RMS error, we have a control quality factor that helps our processing to determine which type of transformation should be used to obtain the best results.

5.4. Quality of Resulting Image

Figure 19 demonstrates visually that no motion blur mars the stacked image. Figure 20 shows that our method can be used to eliminate the motion blur caused by erratic movement of the camera. Images can be improved to a point where the blur is almost undetectable even if the movement of the camera is complex (Figure 9), where the trajectories of predicted features are not the same over the entire image. Our method can be applied for images that exceed an angular motion blur of much more than several pixels unlike de-convolution methods [14] that have strict limits on the amount of angular motion.



(**a**) (**b**)

Figure 20. Example illustrating two subsections of two images. (**a**) is obtained by the average of the ten images; (**b**) is the stacked image. The angular motion between images is 1.2 degrees corresponding to 59 pixels.

6. Conclusions and Future Works

An image processing stacking algorithm has been presented, which consists of several steps. First, feature points are detected in the first image and matched using ZNCC correlation with the help of IMU measurements; then, 3D rotation (and/or 2D homography) between sequential images

is estimated using a photogrammetric method, which leads to a better accuracy. This geometrical transformation (3D rotation or 2D homography) is then used to resample images, and 3D rotation in particular can be used to recalibrate the IMU gyrometers by estimating their bias. This algorithm benefits from the presence of an FPGA to speed up some tasks, like detecting feature points. The proposed method was tested with the IGN camera embedded on an UAV, and the experimental results show the efficiency and accuracy of this method when the UAV is quasi-stationary. The IMU data processing part is already real time and can be used for other applications.

In the future, we will implement a real-time version by acceleration of the image resampling part in the FPGA, since this is the most demanding part of processing. We will refine all parameters to obtain good results in less time. We will also integrate the translation of the UAV in our model using GNSS/IMU data, which may later be used to correct translation deformation. Other sensors like LiDAR or other synchronized cameras can benefit from the resulting better quality of the IMU data. In fact, this methodology is intended to be used on an airborne prototype imaging system with different narrow spectral channels composed of 10 synchronized IGN cameras, which will make it possible to obtain images with an equivalent long exposure time.

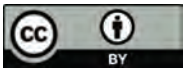
Author Contributions: This work is subject of a PhD thesis of Ahmad Audi, supervised by Marc Pierrot-Deseilligny and Christian Thom with the technical support of Christophe Meynard.

Conflicts of Interest: The authors declare no conflict of interest.

1. Boucher, P. Domesticating the Drone: The Demilitarisation of Unmanned Aircraft for Civil Markets. *Sci. Eng. Ethics* **2015**, *21*, 1393–1412.
2. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97.
3. Ham, Y.; Han, K.K.; Lin, J.J.; Golparvar-Fard, M. Visual monitoring of civil infrastructure systems via camera-equipped Unmanned Aerial Vehicles (UAVs): A review of related works. *Vis. Eng.* **2016**, *4*, 1.
4. Daakir, M.; Pierrot-Deseilligny, M.; Bosser, P.; Pichard, F.; Thom, C.; Rabot, Y. Study of lever-arm effect using embedded photogrammetry and on-board gps receiver on UAV for metrological mapping purpose and proposal of a free ground measurements calibration procedure. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XL-3/W4*, 65–70.
5. Tournadre, V.; Pierrot-Deseilligny, M.; Faure, P.H. Uav linear photogrammetry. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 327–333.
6. Audi, A.; Pierrot Deseilligny, M.; Meynard, C.; Thom, C. Implementation of real-time functionalities in the hardware of an intelligent ultra-light camera specialized for aerial photography. In *GeoUAV-ISPRS Geospatial Week*; ISPRS: La Grande Motte, France, 2015.
7. Le Moigne, J.; Netanyahu, N.S.; Eastman, R.D. (Eds.) *Image Registration for Remote Sensing*; Cambridge University Press: Cambridge, NY, USA, 2010.
8. Lakshmi, K.D.; Vaithiyanathan, V. Study of Feature based Image Registration Algorithms for Navigation of Unmanned Aerial Vehicles. *Indian J. Sci. Technol.* **2015**, *8*, doi:10.17485/ijst/2015/v8i22/79315.
9. Patel, P.; Shah, V. Image Registration Techniques: A Comprehensive Survey. *Int. J. Innov. Res. Dev.* **2014**, *3*, Available online: http://www.ijird.com/index.php/ijird/article/view/47537 (accessed on 13 July 2017).
10. Mann, S.; Picard, R.W. On Being 'undigital' with Digital Cameras: Extending Dynamic Range By Combining Differently Exposed Pictures. In Proceedings of the IS&T's 48th Annual Conference, Washington, DC, USA, 7–11 May 1995; pp.442–448.
11. Debevec, P.E.; Malik, J. Recovering high dynamic range radiance maps from photographs. In Proceedings of the Siggraph: 24th International ACM Conference on Computer Graphics & INteractive Techniques: Visual Proceedings, Los Angeles, CA, USA, 3–8 August 1997; pp. 369–378.
12. Robertson, M.A.; Borman, S.; Stevenson, R.L. Dynamic Range Improvement Through Multiple Exposures. In Proceedings of the 1999 International Conference on Image Processing, Kobe, Japan, 24–28 October 1999; pp. 159–163.

13. Joshi, N.; Kang, S.B.; Zitnick, C.L.; Szeliski, R. Image Deblurring Using Inertial Measurement Sensors. *ACM Trans. Gr.* **2010**, *29*, 1–9.
14. Shah, C.A.; Schickler, W. Automated blur detection and removal in airborne imaging systems using IMU data. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *XXXIX-B1*, 321–323.
15. Sieberth, T.; Wackrow, R.; Chandler, J.H. Automatic isolation of blurred images from uav image sequences. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-1/W2*, 361–366.
16. Sieberth, T.; Wackrow, R.; Chandler, J.H. Uav image blur—Its influence and ways to correct it. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-1/W4*, 33–39.
17. Klein, G.; Drummond, T. Tightly Integrated Sensor Fusion for Robust Visual Tracking. In Proceedings of the British Machine Vision Conference (BMVC'02), Cardiff, UK, 2–5 September 2002; pp. 787–796.
18. Tsai, F.; Chang, H.; Su, A.Y.S. Combining MEMS-based IMU data and vision-based trajectory estimation. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2014**, *XL-4*, 267–271.
19. Hol, J.D.; Schön, T.B.; Gustafsson, F.; Slycke, P.J. Sensor Fusion for Augmented Reality. In *2006 9th International Conference on Information Fusion*; IEEE Service Center: Florence, Italy, 2006.
20. You, S.; Neumann, U. Fusion of vision and gyro tracking for robust augmented reality registration. In Proceedings of the IEEE Virtual Reality 2001 Conference, Yokohama, Japan, 13–17 March 2001; pp. 71–78.
21. Caarls, J.; Jonker, P.; Persa, S. Sensor Fusion for Augmented Reality. In Proceedings of the Ambient Intelligence: First European Symposium, EUSAI 2003, Veldhoven, The Netherlands, 3–4 November 2003; Aarts, E., Collier, R.W., van Loenen, E., de Ruyter, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 160–176.
22. Wang, Y.; Hou, Z.; Leman, K.; Chang, R. Real-Time Video Stabilization for Unmanned Aerial Vehicles. In Proceedings of the MVA2011 IAPR Conference on Machine Vision Applications, Nara, Japan, 13–15 June 2011.
23. Martin, O.; Meynard, C.; Pierrot Deseilligny, M.; Souchon, J.P.; Thom, C. Réalisation d'une caméra photogrammétrique ultralégère et de haute résolution. In Proceedings of the colloque drones et moyens légers aéroportés d'observation, Montpellier, France, 24–26 June 2014.
24. Pierrot Deseilligny, M.; Clery, I. APERO, an open source bundle adjusment software for automatic calibration and orientation of set of images. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *XXXVIII-5/W16*, 269–276.
25. Hartley, R.I.; Kang, S.B. Parameter-free Radial Distortion Correction with Centre of Distortion Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1309–1321.
26. Pena, M. *A Comparative Study of Three Image Matching Algorithms: Sift, Surf, and Fast*; BiblioBazaar: Logan, UT, USA, 2012.
27. Le, X.; Gonzalez, R. Pattern-based corner detection algorithm. In Proceedings of the 2009 6th International Symposium on Image and Signal Processing and Analysis, ISPA 2009, Salzburg, Austria, 16–18 September 2009; pp. 238–243.
28. El-gayar, M.; Soliman, H.; meky, N. A comparative study of image low level feature extraction algorithms. *Egypt. Inform. J.* **2013**, *14*, 175–181.
29. Rosten, E.; Drummond, T. Fusing Points and Lines for High Performance Tracking. In Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV '05), Beijing, China, 15–21 October 2005; IEEE Computer Society: Washington, DC, USA, 2005; Volume 2, pp. 1508–1515.
30. Van der Wal, G.; Zhang, D.; Kandaswamy, I.; Marakowitz, J.; Kaighn, K.; Zhang, J.; Chai, S. FPGA Acceleration for Feature Based Processing Applications. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Boston, MA, USA, 7–12 June 2015.
31. Schauwecker, K.; Klette, R.; Zell, A. A new feature detector and stereo matching method for accurate high-performance sparse stereo matching. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 5171–5176.
32. Behrens, A.; Röllinger, H. Analysis of Feature Point Distributions for Fast Image Mosaicking Algorithms. *Acta Polytech. J. Adv. Eng.* **2010**, *50*, 12–18.
33. Kraft, M.; Schmidt, A.; Kasinski, A.J. High-Speed Image Feature Detection Using FPGA Implementation of Fast Algorithm. In Proceedings of the Third International Conference on Computer Vision Theory and Applications (VISAPP 2008), Funchal, Portugal, 22–25 January 2008; pp. 174–179.

34. Rao,Y.R.; Prathapani, N.; Nagabhooshanam, E. Application of normalized cross correlation to image registration. *Int. J. Res. Eng. Tech.* **2014**, *3*, 12–16.
35. Lewis, J. Fast normalized cross-correlation. *Vis. Interface* **1995**, *10*, 120–123.
36. Tabib, W.; Charnmanee, G.; Koenig, D. *An Hardware Efficient Normalized Cross Correlation FPGA Coprocessor*; An Astroteam Production: Pittsburgh, PA, USA, 2013.
37. Althof, R.J.; Wind, M.G.J.; Dobbins, J.T. A rapid and automatic image registration algorithm with subpixel accuracy. *IEEE Trans. Med. Imaging* **1997**, *16*, 308–316.
38. Huang, H.; Dabiri, D.; Gharib, M. On errors of digital particle image velocimetry. *Meas. Sci. Technol.* **1997**, *8*, 1427.
39. Shimizu, M.; Okutomi, M. An analysis of sub-pixel estimation error on area-based image matching. In Proceedings of the 2002 14th International Conference on Digital Signal Processing, DSP 2002, Santorini, Greece, 1–3 July 2002; Volume 2, pp. 1239–1242.
40. Nobach, H.; Honkanen, M. Two-dimensional Gaussian regression for sub-pixel displacement estimation in particle image velocimetry or particle position estimation in particle tracking velocimetry. *Exp. Fluids* **2005**, *38*, 511–515.



sensors

MDPI

Article

Devising Mobile Sensing and Actuation Infrastructure with Drones

Mungyu Bae, Seungho Yoo, Jongtack Jung, Seongjoon Park, Kangho Kim, Joon Yeop Lee and Hwangnam Kim *

School of Electrical Engineering, Korea University, Seoul 02841, Korea; nardyen@korea.ac.kr (M.B.); pen0423@korea.ac.kr (S.Y.); skylover89@korea.ac.kr (J.J.); psj900918@korea.ac.kr (S.P.); mybalam2k@korea.ac.kr (K.K.); charon7@korea.ac.kr (J.Y.L.)

* Correspondence: hnkim@korea.ac.kr; Tel.: +82-2-3290-4821

Received: 15 December 2017; Accepted: 13 February 2018; Published: 19 February 2018

Abstract: Vast applications and services have been enabled as the number of mobile or sensing devices with communication capabilities has grown. However, managing the devices, integrating networks or combining services across different networks has become a new problem since each network is not directly connected via back-end core networks or servers. The issue is and has been discussed especially in wireless sensor and actuator networks (WSAN). In such systems, sensors and actuators are tightly coupled, so when an independent WSAN needs to collaborate with other networks, it is difficult to adequately combine them into an integrated infrastructure. In this paper, we propose drone-as-a-gateway (DaaG), which uses drones as mobile gateways to interconnect isolated networks or combine independent services. Our system contains features that focus on the service being provided in the order of importance, different from an adaptive simple mobile sink system or delay-tolerant system. Our simulation results have shown that the proposed system is able to activate actuators in the order of importance of the service, which uses separate sensors' data, and it consumes almost the same time in comparison with other path-planning algorithms. Moreover, we have implemented DaaG and presented results in a field test to show that it can enable large-scale on-demand deployment of sensing and actuation infrastructure or the Internet of Things (IoT).

Keywords: wireless sensor and actuator networks; drone network; connectivity; architectures and applications for the Internet of Things; opportunistic and delay-tolerant networks

1. Introduction

Wireless sensor and actuator networks (WSAN) [1] consist of sensors and actuators that are usually supposed to have at least one wireless communication module to communicate with the sink node. The administrator can control the system by collecting the data from sensors, analyzing the data to generate actuation commands and sending them to the actuators. However, it is hardly feasible to combine independent and isolated WSANs into a new additional service network since sensors are tightly coupled with actuators. This tight coupling may result in the following: the actuation should correspond to the sensor data, and the actuator can affect the sensor data after the actuation, so that the actuator and the sensor may have a high spatial correlation. Furthermore, tight coupling is inevitable in traditional WSAN because sensing devices cannot freely create a network of heterogeneous devices due to their limited capabilities of communication and processing. Even if the technical aspect is excluded, many vendors operate both sensors and actuators together since they require appropriate sensing data for their service.

On the other hand, a novel network paradigm, called the Internet of Things (IoT), has emerged recently. The paradigm should facilitate the communication of various existing networks with each other to collect data or deploy new services, so that general users are able to access any

network with their mobile devices much more easily. According to the Ericsson mobility report [2], the number of non-cellular-based machine-to-machine (M2M) devices will dramatically increase from 6.4 billion–17.4 billion from 2017–2023. Moreover, [3] notes that the IoT will be powered by a trillion sensors in 2020, and the IoT could provide an economic impact of 14 trillion by 2020. Following these statistical reports, WSAN could be one of the core infrastructures to implement the IoT.

Additionally, new services in the paradigm demand separation between the sensor network and the actuator network [4–7], which means that they decide the requirements for the service-related actuator or run it with sensing data obtained from different sensor networks. IoT service providers can increase the profitability of the service by using public sensor data [8,9], rather than directly constructing sensor networks themselves. To satisfy this idea, using a mobile sink node, it is easy to collect public sensors' data on the IoT service providers part since the public sensor network does not have a sink node, but only broadcasts its sensing data. In the case of a disaster situation, mobile nodes in the disaster area need to be changed from normal mode to emergency mode, in which nodes broadcast their data because cellular communication may be unavailable. Assume a fire has broken out in a building. The sensor and actuator networks may be destroyed in the vicinity of the fire ignition point, and so, some part of them becomes isolated from the external network, such as the Internet, even if they are still alive. Thus, they cannot send any information to firefighters.

Based on these observations, we can realize that there needs to be a system that can improvise a networking infrastructure among independent devices, systems, networks and even services in order to dynamically merge sensing and actuation infrastructure into an integrated service network and consequently enhance existing services [10–12] or forthcoming IoT services. To enable this concept, drones can be employed to implement such a system. The mobility and degree of freedom (DoF) of drones enables an instant deployment of networks into regions that are difficult to reach by humans or ground-based vehicles. Already, some research works [13–18] have tried to adopt the unmanned aerial vehicle (UAV) as the mobile sink node, but these research works have focused on only implementing the drones in the sensor network and have not considered the advanced operation of the drones, such as dynamic path-planning and constructing a network among the drones.

Consequently, in this paper, we propose a new platform of a UAV or drone, drone-as-a-gateway (DaaG), which can connect the sensor and the actuator networks on demand. In contrast with other research works, we have focused on 'which sensor needs to be focused on in terms of service', 'which actuator has to be firstly visited by the drone to reduce the main service delay' and 'which data transmission method is to be used in the given environment', namely improvement of the interoperability of the drones to connect to a separate network. We provide DaaG with the intelligence that dynamically determines the flying path for drones and to generate messages to the actuators to improve the interoperability of the DaaG. In the previous firefighting example, when DaaG is deployed in the same situation, it can directly collect sensor data, which can be used to efficiently conduct search-and-rescue and firefighting missions, and send command messages to actuators. We have implemented the proposed system with multi-rotor-based drones and then showed via simulations and experiments that DaaG can connect independent network systems and devices as a gateway and also can enable large-scale on-demand deployment of sensing and actuation infrastructure for a WSAN or IoT network as a mobile sink.

The rest of the paper is organized as follows. In Section 2, we explain the structural and functional design of DaaG. Section 3 shows the simulation results by which our system can reduce the main service delay. We present an empirical experiment and evaluate this result to verify that DaaG works well in Section 4, and finally, we conclude our paper with Section 5.

2. Designing DaaG

2.1. Design Consideration

The principal features that have to consider to implement DaaG are enumerated as follows:

1. Connectivity: This enables the sensor network and actuator network to share their data when they are physically separated. Additionally, it acts as a data ferry, to deliver the data between the sensor and the actuator network, or a routing node in a multi-hop network, represented as Figure 1a, to forward the data between them. The most distinctive feature of the two methods is the mobility of the drones during the network provisioning service. In the data ferry method, represented as Figure 1b, the drones continuously travel between the sensors and the actuators. In the multi-hop network method, the drones hover and relay data between the sensors and the actuators.
2. Automated system: It can self-determine its flight plan to efficiently collect sensor data and appropriately decide the actuation.
3. Smartness: It should be smart enough to translate sensor data and actuation control between the sensor and actuator network since the latter cannot directly interpret raw sensor data that the former delivers and vice versa.

2.2. Structure of DaaG

Figure 2 shows the structure of DaaG, which has three components: embedded controller, sensors and drone controller. The embedded controller is a core component of the DaaG platform to provide dynamic connectivity and smartness to the separate networks. The controller has three sub-components: task management, drone flight control and communication modules.

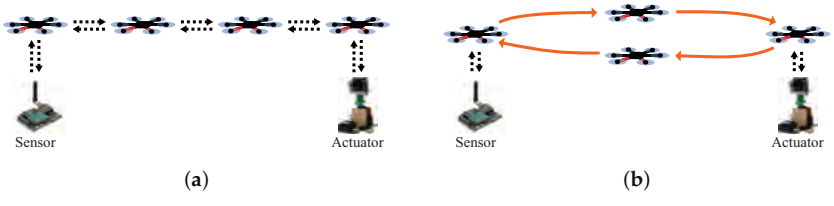


Figure 1. Two scenarios for the data transfer from the sensor. (**a**) multi-hop network method. (**b**) Data ferry method.

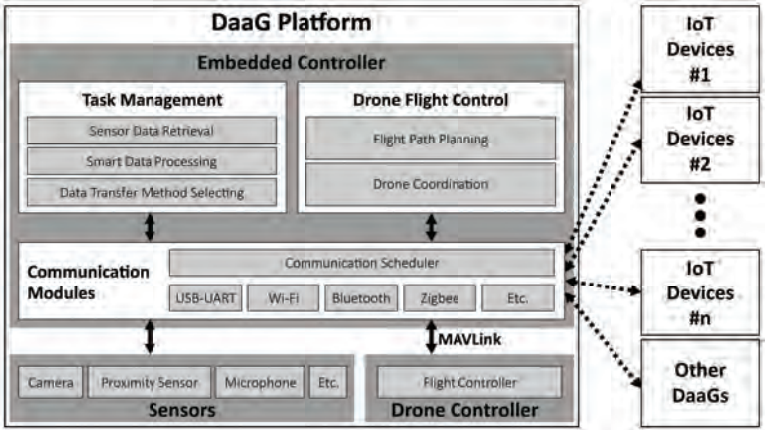


Figure 2. Structure of DaaG.

2.2.1. Task Management Module

The task management module has three functionalities; sensor data retrieval, smart data processing and data transfer method selection. A sensor data retrieval function collects the data from the sensors through communication modules. These data are basically stored in the storage module. A smart data processing function uses these data in storage to generate the actuator command message. To generate the message, DaaG firstly classifies problematic sensors, which have to be monitored, and calculates the coefficient value between problematic sensors and actuators. Then, it generates the command messages for actuators to efficiently monitor problematic sensors. This component guarantees the smartness in the features. A data transfer method selecting function considers the status of the sensor network and the actuator network. Transferring data at a long range can be carried out in two ways with drones: multi-hop network and data ferry. This function selects one method that is suitable for sensing and actuating in the network environment. This component guarantees the connectivity of the features.

2.2.2. Drone Flight Control Modules

There are two functionalities in the drone flight control module: flight path-planning and drone coordination. A flight path-planning function is used to plan the flight course to reduce the main service delay. The details of this function are described in Section 2.3.3. The DaaG flight maneuver requires the ability to change the path and destination during the flight, but some commercial drone controllers do not have this feature, called the off-board controller. A drone coordination function is important for the case where mission areas may overlap [19–21]. The module basically controls the deployment drones and prevents drone collisions, which is important in using the multi-hop network method. This components guarantees an automated system with respect to the features.

To acquire the position of the drone, the control module utilizes inertial sensors, wireless ranging sensors and/or the Global Positioning System (GPS). There are many research works utilizing these devices for positioning and navigation [22,23]. We use inertial sensors and GPS for positioning and navigation, since other methods require additional devices or infrastructure.

2.2.3. Communication Modules

In the new network paradigm, sensors and actuators can have different communication interfaces. Therefore, DaaG needs to support the various communication interfaces, such as Wi-Fi, Bluetooth, ZigBee, and so on. However, since beaconing for all communication interfaces results in the overhead of the battery, a communication scheduler uses the backoff-based beaconing algorithm for energy efficiency. Moreover, DaaG supports both single-hop and multi-hop communications. Drones can communicate with each other over other drones as long as there is a connected path. This component guarantees the connectivity of the features.

2.3. Main Functions of DaaG

In this section, the main functions of DaaG are described in detail. The main functions are composed of communication scheduling, smart data processing, path-planning and data transfer method selection. The communication scheduling function continuously runs during DaaG operation and communicates with the sensors and the actuators. The collected data from the sensors and the actuators are utilized by other functions. The smart data processing function runs after DaaG visits every sensor node. Then, the function generates a problematic sensor list (PSL), which contains problematic sensors. The path-planning function consists of two parts: path-planning in the sensor network and path-planning in the actuator network. The former part of the function plans the flight path of DaaG and runs until DaaG collects data from every sensor. The latter part of the function determines which actuator has to be visited immediately and runs after DaaG has finished collecting sensor data. The data transfer method selection function determines which data transfer method to

use. This function continuously runs during the DaaG operation. If this function decided to change the data transfer method, then all of the operating DaaGs would change the data transfer method. A more detailed description of the functions is explained in the following subsections.

Algorithm 1 Backoff-based beaconing algorithm.

```
1: ∀c_i ∈ C : c_{i,bf} = c_{i,bmax} = 1
2: while isDaaGWorking() do
3:     for each c_i in C do
4:         c_{i,bf} = c_{i,bf} − 1                          ▷ Update remaining backoff time of c_i
5:         if c_{i,bf} ≤ 0 then
6:             c_{i,flag} = runBeaconing(c_i)               ▷ Broadcast beacon message with c_i
7:             if c_{i,flag} > 0 then
8:                 c_{i,bf} = c_{i,bmax} = 1
9:                 D = runCommunication( c_i)               ▷ Collect data D from nearby nodes with c_i
10:            else
11:                c_{i,bf} = c_{i,bmax} = min (b_max, α × c_{i,bmax})   ▷ Update backoff time of c_i
12:            end if
13:        end if
14:    end for
15:    if DaaG moves more than reset boundary length then
16:        ∀c_i ∈ C : c_{i,bf} = 1
17:    end if
18:    sleep(t_period)                                      ▷ Wait for t_period
19: end while
```

2.3.1. Communication Scheduling

In sensor network, sensors and actuators use sleep mode to minimize their power consumption. This technique works in the traditional environment that many sensors are stationary, but it is not appropriate in the network environment where nodes are mobile. DaaG is mobile, so the sensors' communication interface uses sleep mode by default, and it should wake up in response to the beacon message of DaaG. Moreover, DaaG has many communication interfaces to support communicating with various sensors and actuators. There are certain advantages to reducing the energy consumption of every sensor, but DaaG consumes much energy to support beaconing for all communication interfaces.

We propose and adopt a backoff-based beaconing algorithm to reduce the energy consumption. The detailed algorithm is introduced in Algorithm 1, where C is the set of communication interfaces, $c_{i,bf}$ is the remaining backoff time and $c_{i,bmax}$ is the maximum backoff time of the i-th communication interface, b_{max} is the maximum of the possible backoff time, α is the increasing backoff factor and t_{period} is the time interval of the algorithm. The core idea of this algorithm is that each communication interface sends a beacon message with its inherent backoff time. Some communication interfaces increase their backoff time to reduce energy, when there are no nodes around using the same interface. Even if there are no nodes that use a certain communication interface at the current position, there is the possibility that the node appears when DaaG moves to a different position. Therefore, we adopt a maximum backoff time b_{max} in the algorithm to limit the backoff increase.

The difference between b_{max} and $c_{i,bmax}$ is that the former affects backoff time with respect to the movement of the drone, while the latter only considers the response of nearby nodes. However, when b_{max} is too high, DaaG can still pass a node in between two beacons. DaaG also sends beacons for every arbitrary distance it moves to prevent missing nodes.

2.3.2. Smart Data Processing

The smartness that differentiates DaaG from conventional mobile sinks is as follows. First, DaaG collects the data from the sensors as:

$$\langle id, pos, st, sv, timestamp \rangle ,$$

where *id* is the sensor's unique ID number, *pos* is the position of the sensor, *st* is the sensor type and *sv* is the value of the sensor. DaaG determines that there is a problem at some area or for some objects by using the collected sensor data sv_t and stored data sv_{t-1} with the following conditions:

- the difference between sv_t and sv_{t-1} is larger than threshold value sv_{th},
- sv_t is lower than predefined lower bound sv_{lb},
- sv_t is higher than predefined upper bound sv_{ub},

which can be described as follows,

$$|sv_t - sv_{t-1}| \geq sv_{th} \quad \text{or} \quad sv_t \leq sv_{lb} \quad \text{or} \quad sv_t \geq sv_{ub}. \tag{1}$$

Since we assumed the performance and the memory capacity of the sensor are low, we only consider sv_t and sv_{t-1}. With this approach, the sensor could reduce the processing time, memory usage, and also the energy consumption. However, if the performance of the sensor is considerably higher for a complicated operation, the sensor could accumulate the data for a fixed period and analyze the accumulated value to detect the problematic sensor.

Then, *id* of the sensor is added in a PSL. To manage the problem, DaaG calculates the correlation value between the problematic sensor and each actuator. The correlation value between the sensor and actuator is as follows,

$$corr_{s_i,a_j} = \frac{\lambda}{\text{dist}\left(s_i, a_j\right)} + (1 - \lambda\ ) \frac{av_t - av_{t-1}}{sv_t - sv_{t-1}}, \tag{2}$$

where $\text{dist}\left(s_i, a_j\right)$ is the distance between the i-th sensor s_i and the j-th actuator a_j, λ is the weight value, which can be determined according to the type of service, and *av* is the value from the actuator. Then, DaaG finds the actuator that has the highest correlation with the problematic sensor. This means that the chosen actuator is the best to monitor the problematic sensor out of all other actuators. This sensor-actuator matching can help to solve problems and issues by making the actuator focus on the sensors. In the paper, these correlation values are used to calculate the path-planning of DaaG.

2.3.3. Path Planning

Assume that DaaG knows the position of sensors and actuations, $pos_i = (x_i, y_i)$ for $\forall i \in V$, where undirected graph $G = (V, E)$, V is a set of nodes and E is a set of edges. First, DaaG calculates the edge weight for all links in the E as follows:

$$w_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}, \tag{3}$$

for $\forall i, j \in S$ where $S \subset V$ is a set of sensors.

To gather the data from the sensors, the position error of the sensor should not be larger than the communication range, in our case, about 10 m.

Then, DaaG runs a path-planning algorithm, which is described in Algorithm 2. In the path-planning algorithm, the beaconing algorithm is used to collect the sensor data. The purpose of DaaG is to retrieve the sensor data in the sensor network, which is similar to the traveling salesman problem with neighborhoods (TSPN) [24] and close enough traveling salesman problem (CETSP) [25]. Therefore, it does not need to visit all sensors in the sensor network because using wireless communication can help DaaG get the sensor data without approaching the sensor.

This point makes Algorithm 2 different from other path-planning algorithms, such as the nearest-based algorithm. However, since using this algorithm alone can result in missing a node in the sensor network, DaaG marks the flag value F_i of the sensed node as one, and sets the flag value as two for the visited node. Finally, DaaG moves toward the node that has the lowest weight among the nodes.

Algorithm 2 Path-planning algorithm in the sensor network.

for $i = 1$ to $|S|$ **do** ▷ Initialize all F_i to 0
  $F_i = 0$
end for
while any $F_i = 0$ exists **do** ▷ Run until DaaG collects data from every sensor
  if DaaG visits s_i **then**
    $F_i = 2$
    s_k = getSensorToVisit(s_i) ▷ Get unvisited sensor s_k, which has the highest w_{ik}
    (pos_{sx}, pos_{sy}) = getPosition(s_k)
    moveDaaGTo(pos_{sx}, pos_{sy}) ▷ Command flight controller to move to (pos_{sx}, pos_{sy})
  end if
  T = getSensorListFromCollectedData(D) ▷ Get list of sensed sensors from D
  for each s_i in T **do**
    if $F_i = 0$ **then**
      $F_i = 1$
    end if
  end for
end while

Once DaaG collects all sensor data in the sensor network, it uses Algorithm 3 to move DaaG to the actuator network and to give command messages to the appropriate actuator in the case of the data ferry network method.

Algorithm 3 Path-planning algorithm in the actuator network.

PSL = runSmartDataProcessing(D) ▷ Get PSL with smart data processing
VL = List(empty) ▷ Initialize visited list VL
ACL = List(empty) ▷ Initialize actuator correlation value list ACL
for j = 1 to $|A|$ **do** ▷ Calculate ac_j for all actuators
  $ac_j = \sum_{i=1}^{|PSL|} corr_{s_i,a_j}$
  ACL.insert(ac_j)
end for
if data ferry **then**
  while $VL \neq A$ **do** ▷ Run until DaaG visits every actuator
    a_j = getActuatorToCommunicate($A \backslash VL, ACL$) ▷ Get a_j with highest ac_j in $A \backslash VL$
    (pos_{ax}, pos_{ay}) = getPos(a_j)
    VL.insert(a_j) ▷ Put a_j into VL
    moveDaaGTo(pos_{ax}, pos_{ay})
  end while
else if multi-hop **then**
  a_j = getActuatorToCommunicate(A, ACL)
  (pos_{ax}, pos_{ay}) = getPosition(a_j)
  s_i = getNearestSensorFrom(a_j)
  (pos_{sx}, pos_{sy}) = getPosition(s_i)
  for k = 1 to n_d **do**
    moveDaaGTo$\left(\frac{k \cdot pos_{ax} + (n_d+1-k) \cdot pos_{sx}}{n_d+1}, \frac{k \cdot pos_{ay} + (n_d+1-k) \cdot pos_{sy}}{n_d+1}\right)$
  end for
end if

In Algorithm 3, D is the set of collected data, which is collected from Algorithm 1, ac_j represents the correlation value of actuator a_j, ACL contains ac_j values for every actuator a_j, $A \subset V$ is a set of actuators, $VL \subset A$ represents the set of actuators that are visited by DaaG and n_d is the number of drones deployed. First, DaaG runs the smartness algorithm, so that it extracts a list of problematic sensors. The list has a tuple of $\left\langle s_j, corr_{s_j,a_1}, corr_{s_j,a_2}, \cdots, corr_{s_j,a_{|A|}} \right\rangle$, where s_j is the identification of the sensor and a_i is the actuator. Finally, DaaG visits all actuators in the list in the order of the correlation value, so that command messages will be transmitted and actuators will focus on the problematic sensor. However, when DaaG selects the multi-hop network method, drones do not directly visit actuators; they only form a bridge between the sensor network and the actuator network. The sensor that is the nearest to the highest correlated actuator is selected in order to transmit the sensor data to the highest correlated actuator at first. Then, this sensor and this actuator will be connected through drones, and these nodes will act as a sink node for their network.

2.3.4. Data Transfer Method Selection

To select the appropriate data transfer method, DaaG has to recognize environment variables, such as network state, service type, etc. After that, DaaG calculates a data transfer method score, which directly determines the data transfer method, as follows:

$$S_{dt} = \sum_{i=1}^{n} \alpha_i \times \theta_i, \tag{4}$$

where θ_i is the environment variable, such as loudness, temperature, light intensity and humidity. α_i is its weight value. For example, in the case of a wildfire detection system, the weight of temperature and light intensity would be higher than that of loudness.

Furthermore, light intensity and temperature are usually low in normal situations. In this case, the system does not need to send data continuously. Therefore, using the data ferry method to deliver the data reduces the energy consumption with no setbacks. However, if a wildfire occurs, the light intensity and temperature of the surveillance area will be increasing, and an immediate and continuous data report is required to understand the situation with accuracy. In this case, continuous data transmission with a multi-hop network is required.

The effect of environment variables could be adjusted by modifying environment weight value α_i. This value could be determined by evaluating the sensors. Following the S_{dt}, DaaG chooses the data transfer method between data ferry and multi-hop network when $S_{dt} < Th_{S_{dt}}$. $Th_{S_{dt}}$ is the threshold value of the data transfer method score, and we calculate $Th_{S_{dt}}$ as follows:

$$Th_{S_{dt}} = k \times \frac{n_d}{(n_a + n_s) \times dist(A, S)} \tag{5}$$

where n_d is the number of drones, n_a and n_s are the number of actuators and the sensors, $dist(A, S)$ is the distance between the geographical center points of the actuating area and sensing area and k is the coefficient to scale the $Th_{s_{dt}}$. The number of drones n_d is proportional to the throughput of the data ferry method. However, the other values, n_a, n_s and $dist(A, S)$, make the travel time of the drone longer. Therefore, these values are inversely proportional to the throughput of the data ferry method. We set the value of k as 1600 based on our simulation result.

Furthermore, this equation can be used to optimize the number of drones to deploy. After the deployment of the sensors and the actuators, the manager of the system could collect data from the sensors and get S_{dt} in the normal state. With the accumulated information of S_{dt}, the manager could set the threshold value $Th_{S_{dt}}$ and calculate the minimum of drones to be deployed. With this approach, we could optimize the configuration.

In the multi-hop network method, data can be continuously transferred. Therefore, it is favored when the service wants guaranteed real-time information. The communication link, however,

suffers from interference from other drones. Moreover, many drones are needed to cover separate networks that are far from each other, which can cause high capital and operational expenses. This network structure can also cause a battery imbalance problem, which is a major issue in sensor networks.

In the data ferry method, the data are uploaded to the drone with maximum throughput; the drone flies to the destination, and it comes back for additional data uploading. As it is free from interferences, it encounters a much lower bit error rate. However, the total throughput of the data ferry case is related to the flying distance and speed of drones. Therefore, network efficiency may be decreased when the speed of a drone is slow or the data size is small.

3. Simulations

Before we implement our system on the drone, we have verified that our algorithms are suitable for connecting separate networks and can reduce the main service delay.

We have conducted a series of simulations that are designed to examine the network performance, energy efficiency,and path-planning capability of our system. The result of each simulation is given in the following subsections.

3.1. Network Performance

Two of the main features of our proposed system are the capability to interconnect isolated networks and to combine independent services. The performance of such features is heavily dependent on the network performance, which our proposed system tries to enhance. With the following simulations, we examine the performance of our proposed system's network. To simulate the methods as realistically as possible, we employ D-MUNS [26], a multiple UAV network simulator, which is based on ns3 [27] and RotorS [28].

3.1.1. Simulation Setup

We have deployed sensors and actuators as in Figure 3, and more detailed simulation configurations are as follows;

- The number of sensors n_s in the sensing area is 30.
- The number of actuators n_a in the actuating area is 10.
- The number of drone n_d is one.
- The distance between the sensing area and actuating area is 160 m.
- Drones going to the actuator pass the sensing area.
- The problematic sensor is the southern-most node in the sensing area.
- All sensors, actuators and drones are equipped with a Wi-Fi interface for communication.
- The maximum throughput of sensors is limited to 100 Kbps

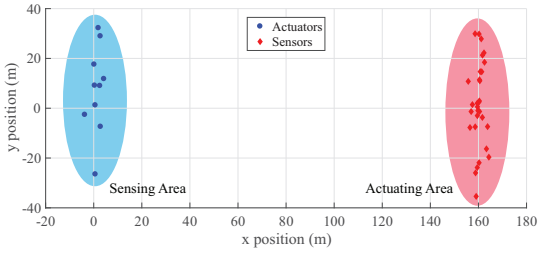


Figure 3. Node placement of the network performance simulation.

In our simulations, we select a priority service throughput as a metric to analyze the performance of our algorithms. The priority service throughput is the sum of weighted actuators' throughput. The weight value is the correlation coefficient of actuators regarding the problematic sensor. Therefore, the priority service throughput Th_{ps} is as follows,

$$Th_{ps} = \sum_{i=1}^{n_a} \omega_i \times Th_i. \tag{6}$$

The reason for adopting the new metric is that the main actuator (the highest-correlated actuator) is most important in the service, but other actuators can also monitor the problematic sensor imperfectly. In our simulations, ω_i is selected in accordance with the reciprocal of the distance between the problematic sensor and the actuators. The set of ω_i in our simulations is as follows: $\{0.23, 0.30, 0.39, 0.51, 0.67, 0.87, 1.13, 1.47, 1.92, 2.50\}$.

3.1.2. Simulation Results in the Data Ferry Method

We implemented two path-routing algorithms in the simulator, which are a nearest neighbor algorithm [29] and a rendezvous algorithm [30], for the comparison study, since these algorithms have been widely used for UAV path-finding. All three algorithms firstly visit the sensor network and then visit the actuator network. The nearest neighbor algorithm visits the unvisited nearest sensor node in a greedy manner and then visits the actuator network in the same way. In case of the rendezvous algorithm, eight sensors and four actuators are selected as rendezvous points. Then, the drone visits only 12 rendezvous nodes since other sensor nodes or actuator nodes are connected to rendezvous nodes. Our event-centric path-planning algorithm visits the sensor node following our algorithm, which is a little bit like the rendezvous algorithm, but it firstly visits the highest correlated actuator.

The data transmission is not continuous in the WSAN. And this makes the network throughput comparison difficult. Therefore, to ease comparison of the performance, we adopt normalized cumulative priority service throughput (NCPST). With the NCPST, the weighted size of transmitted data could be obtained. As we can see from the result of the simulation in Figure 4a, our event-centric path-planning algorithm in DaaG shows the best performance compared to the other two algorithms in the simulation. With respect to the time it takes for the drone to send the sensing data to one of the actuators successfully, our algorithm is 6.5% earlier than the rendezvous algorithm and 23.2% earlier than the shortest-path algorithm.

The rendezvous algorithm generates circular path which connects rendezvous nodes based on the geographic position of the rendezvous nodes. After the path has created, the drone follows the generated path to collect the data and returns to the starting point. However, our event-centric path-planning algorithm generates path based on the correlation between the actuators and the sensors. This is what differentiates our algorithm from the rendezvous algorithm. Therefore, the rendezvous algorithm's arrival time at the actuator network is delayed. In terms of the priority service throughput, that of DaaG showed the fastest growth rate. Because the first actuator that the other two algorithms visited is not a highly-correlated actuator, it has a low ω_i value. The rendezvous algorithm achieved the maximum throughput at first since the total service delay is lower than the other algorithms, but this is not more important than the main service delay. Hereby, we can verify that our path-planning algorithm can increase the main throughput in the data ferry method.

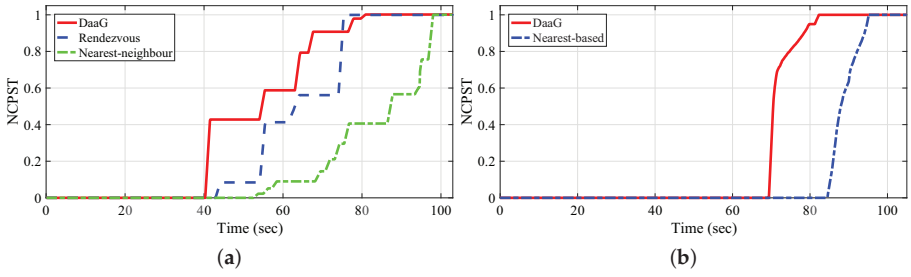


Figure 4. Normalized cumulative priority service throughput. (**a**) Data ferry method; (**b**) multi-hop network method.

3.1.3. Simulation Results in the DaaG Multi-Hop Network Method

In this simulation, we select a nearest-based multi-hop network as a comparison target, which organizes a multi-hop topology to connect a sensor and an actuator that has the shortest distance to the sensor. Similar to the previous simulation, drones visit the sensor network and gather the sensing data. The differences are that the number of drones is increased to four, and they create a bridge for the communication.

Since the distance between the sensor network and actuator network is too long to connect these two networks with a single drone, we use more drones to connect the networks with the multi-hop method.

Figure 4b shows the result of the simulation. Our video [31] shows the whole simulation procedure and result. In the result, our drones create a communication bridge from the highest correlated actuator to the sensor that is closest to the actuator. However, the nearest-based path-planning scheme connects a sensor and the actuator that is closest to the sensor, but not the highest correlated actuator. As shown in Figure 4b, DaaG's communication has started 17.7% earlier than the nearest-based path-planning scheme. Furthermore, we can notice that DaaG's priority service throughput increased more quickly than the other. Moreover, the DaaG system quickly finishes the data transmission compared to the nearest-based path-planning scheme since all actuators complete their job of monitoring the problematic sensor. This is because the highest correlated actuator is the first monitor in the actuator network, but it is the third monitor in the nearest-based path-planning scheme. Following the analysis, we can claim that DaaG can provide new services faster than other path-planning algorithms in terms of both the data ferry and multi-hop network method. Moreover, DaaG can select the message transfer schemes according to the network status and service type.

3.1.4. Network Throughput Comparison between the Data Ferry and Multi-Hop Method

Since DaaG selects the data transfer method for a given environment, we measured the performance of each data transfer method. The result is shown in Figure 5. When the connection is built, the throughput of the multi-hop network method is higher than the data ferry method.

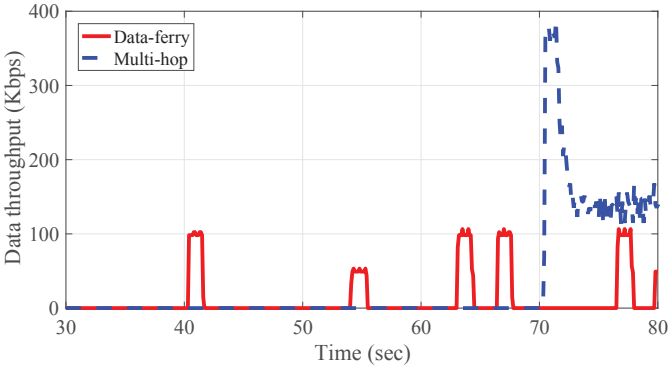


Figure 5. Throughput comparison between the data transfer methods.

The multi-hop network method could transmit data at the maximum throughput continuously. However, the data ferry method only transmits data at the maximum throughput when it communicates with nearby sensors or actuators. However, to construct a multi-hop network, drones have to fly to the points and hold their position. Therefore, the network construction time of the multi-hop method is higher than the time of the data ferry method. Furthermore, sensors and actuators have to continuously transmit/receive the data.

3.2. Energy Consumption

With size-constrained devices, energy efficiency is one of the most important issues. Most sensors/actuators operate on batteries, so the lifetime of the devices is limited if the energy efficiency is not handled well. In this subsection, we have simulated the energy consumption efficiency improvement of DaaG in the environment.

3.2.1. Simulation Setup in the Energy Consumption Simulation

The simulation has been conducted with the OPNET simulator. Throughout the entire simulation, 36, 64 and 100 sensors were uniformly deployed in an 180 m × 180 m, 240 m × 240 m and 300 m × 300 m area, respectively. Furthermore, the manet_station model of OPNET is used as the sensor model. Each node transmits a 200-byte packet with a 45-m communication range to the sink node at one of the corners. The packet generation follows a Poisson distribution with a rate of 0.1 packets per second. The optimized link state routing (OLSR) protocol is used in the multi-hop scenario, and DaaG collects the data physically.

To measure the energy consumption of each node, we used the first order radio model [32]. In this model, the energy consumption for transmitting 1 bit over a distance d is $E_{Tx} = E_{elec} + \epsilon_{amp} \times d^2$, where E_{elec} is the energy spent by the transmitter or receiver circuit and ϵ_{amp} stands for the energy dissipated in the transmit amplifier. The energy used to receive 1 bit of data is $E_{Rx} = E_{elec}$. In the series of simulations, E_{elec} was set to 50 nJ/bit, and ϵ_{amp} was set to 100 pJ/bit/m^2 [33,34]. Each simulation lasted 900 s.

3.2.2. Simulation Result

Figure 6 compares the average energy consumption of the two cases: multi-hop network and DaaG deployment. With any size of network, DaaG performs much better than the multi-hop network. Sensors with DaaG deployment only need to deliver data to the DaaG in a single-hop manner, so that the energy is kept low and is much less affected by the size of the network. It is also true that the

drone will consume a large amount of energy, but the energy constraint is not concerned with the total energy consumption, but the longevity of the sensors.

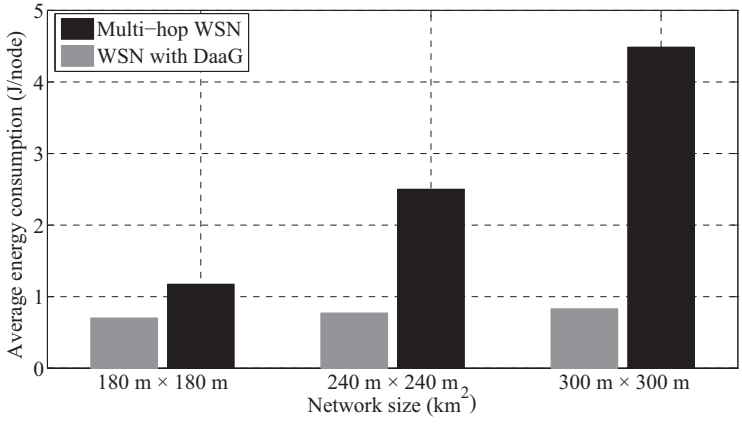
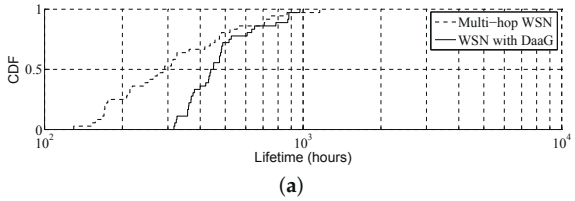
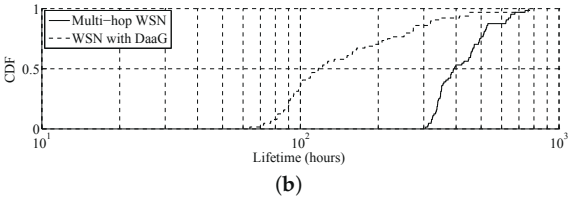


Figure 6. Average energy consumption with various network sizes.

Figure 7 presents the cumulative distribution function (CDF) of the lifetime for various network sizes. The lifetime of each sensor can be estimated with an arbitrary value of 1296 joules, a typical AAA battery size. As we are interested in the ratio of the two cases, not the exact time, the exact size of the battery is particularly important. As can be seen in the figure, DaaG deployment always shows a much longer lifetime than traditional multi-hop WSN. As the network size increases, the performance gap is widened. Moreover, the lifetime of the sensors varies largely in a multi-hop network, which is a crucial setback for the WSAN management. Figure 8 shows the energy consumption map of each scenario. We can observe that sensors located around the sink node consume more energy in the multi-hop network. This is because data packets are forwarded through the nodes around the sink. On the other hand, the energy consumption is balanced when DaaG is applied to the network. It can also be observed that the energy consumption over all regions of DaaG deployment is similar to the minimum energy consumption in the multi-hop network.

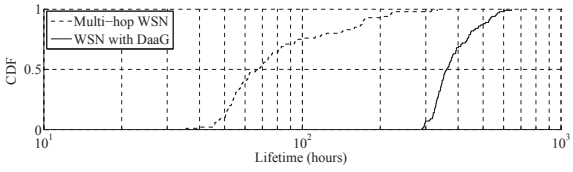


(**a**)



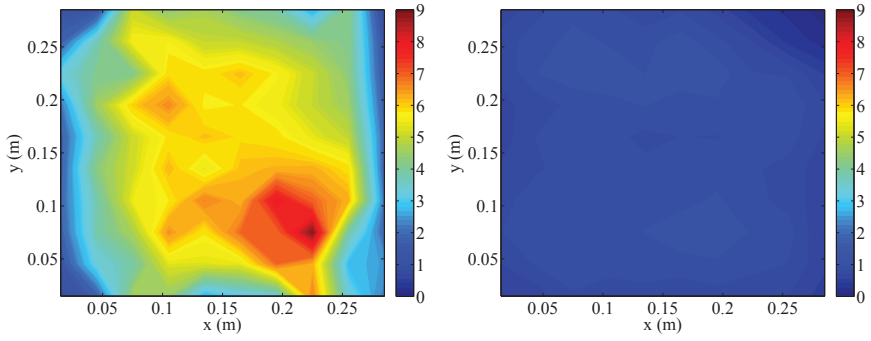
(**b**)

Figure 7. *Cont.*



(**c**)

Figure 7. Cumulative distribution function of the lifetime under various network size conditions. (**a**) 180 m × 180 m. (**b**) 240 m × 240 m. (**c**) 300 m × 300 m.



(**d**) (**e**)

Figure 8. The map of sensor energy consumption. (**a**) Multi-hop network. (**b**) DaaG deployment.

3.3. Main Focusing Delay

To validate our system, we have analyzed the delay of our system measured throughout the simulations. In the analysis, two metrics related to delay are considered: a main focusing delay and a total service delay. The main focusing delay means the interval in which a main actuator, which can best monitor problematic sensors, starts to monitor problematic sensors, after it completes collecting all sensor data and determining the problematic sensors. This delay is a core metric to determine whether the system can serve the main content rapidly. The total service delay is the elapsed time to complete visiting all actuators, which include both the main actuator and the non-main actuators, and send the operation message to each actuator, when the start time begins after all sensor data have been collected. Since the non-main actuators may monitor the problematic sensors or provide some help to achieve the requested service, the total service delay is one of the important metrics to determine whether the system can serve the requested content perfectly.

3.3.1. Simulation Setup

The following simulation has a setup that is similar to that of the simulation in Section 3.1. The details are as follows. The numbers of actuators in the actuating area are 5, 10, 20, 40, 60, 80, 100, 120, 160 and 200, and the number of sensors in the sensing area is three-times that of the actuators. The distance between the sensing area and the actuating area is 160 m, and a drone covers the entire region. The drone moves passed the sensing area to the actuators. With every simulation, one or more problematic sensors are randomly selected. It takes 10 s for the drone to complete the command operation. With each setup with the same number of sensors and actuators, 100 simulations were conducted. The following is the result.

3.3.2. Simulation Result

Figure 9 shows the result of delay measurement with our simulation. The result is desirable in that the main service delay is shorter than the nearest-based method. This is mainly due to the fact that the nearest-based path-planning method is likely to visit some nodes that are not main actuators. On the other hand, DaaG visits the main actuators with higher priority with the smartness function. The time saved with the smartness function enables DaaG to outperform the nearest-based method by far, under certain conditions. Figure 9 shows that the total service delay of DaaG is longer than the nearest-based method, as DaaG considers the task priority more important than the overall time delay. Yet, DaaG spends only 13.9% longer than the nearest-based path-planning method even with 200 actuators; thus, the scalability is good.

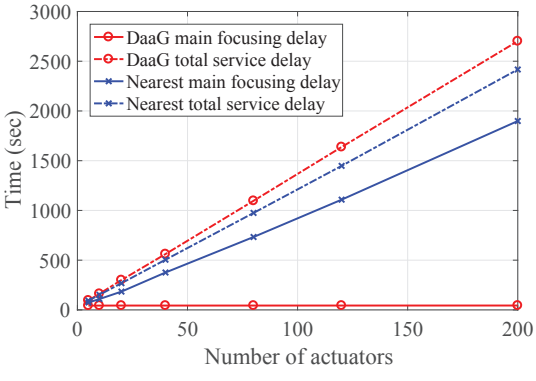


Figure 9. Delay measurement from the simulation.

4. Implementing and Evaluating DaaG

In this section, we empirically present the feasibility of DaaG based on a specific scenario where the sensing area is physically separated from the actuation area. We firstly describe a network configuration as a plausible scenario, then explain how to implement DaaG and relevant network nodes for sensors and actuators, and after that, we explain the empirical results and their analysis.

4.1. Exemplary Scenario

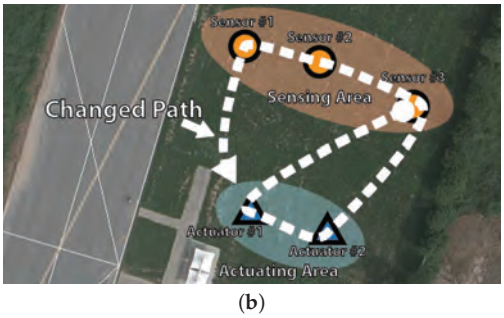
The purpose of our experiment scenario is to connect separate sensor and actuator networks.

To conduct the experiment, we have deployed sensors and actuators as in Figure 10a. The whole scenario of our experiment is as follows:

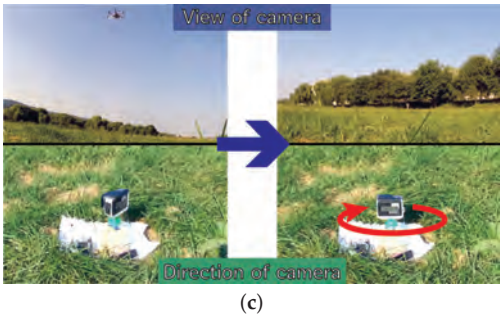
1. There are two separated areas, a sensing area and an actuating area, any one of which is not connected to the other.
2. Sensors in the sensing area measure the light intensity value.
3. Actuators in the actuating area are filming the sensors, but the number of actuators is smaller than the sensors.
4. Drone #1 (DaaG) flies over the sensing area, associates itself with sensors in the sensing area and retrieves the sensed data from the sensors.
5. Drone #1 visits the actuating area and calculates the angle of the actuators.
6. Drone #1 sends the angle data to the actuators to film the sensor that detects an event.

The following implementation, deployment and evaluation will be explained based on this configuration and scenario, but the simulation study will use different network configurations and scenarios.

(**a**)



(**b**)



(**c**)

Figure 10. The pictures of the demonstration scenario including path change information. (**a**) Experiment setting and expected DaaG path before the experiment begins. (**b**) Path change due to the retrieved data. (**c**) Actuator changed by DaaG .

4.2. Implementation

In this subsection, the implementation details of the system are elaborated. Based on our prior work [35,36], we extended the functionality of our novel drone system to realize the proposed DaaG. As in Figure 2, the DaaG implementation is composed of two parts: software and hardware. To reflect the scenario, we implement the two separate networks: the sensor network and the actuator network.

4.2.1. DaaG Hardware

The DaaG hardware is composed of a drone frame, an embedded computer and auxiliary sensors, as in Figure 2. The pictures of the implementation are given in Figure 11a,b. We use DJI F550 Flame Wheel as a hexarotor frame. The 3DR Pixhawk flight controller [37] is installed for basic flight control, such as attitude and position control. For advanced computing operations, we chose Odroid XU4. This computing unit manages tasks, plans the flight path, sends commands to the Pixhawk and communicates with nodes or other DaaGs. The Pixhawk and Odroid are connected via a USB-UART module and communicate with each other by using MAVLink [38].

4.2.2. DaaG Software

There is an Apache server running on DaaG, to handle requests from the nodes. Alongside the Apache server, MySQL DB and PHP scripts are used to enable easier manipulation of the server. Nodes in our implementation make HTTP requests to send and receive data. DaaG mimics the HTTP requests with a DTN-like communication style is needed. Normal communications are made using typical TCP communication.

In the implementation scenario, DaaG has to collect the light intensity value from the sensors, determine which area needs to be monitored and calculate the angle of the actuators to watch a certain area. Algorithm 4 shows the procedure of DaaG operation designed for the implemented scenario. In the algorithm, S is the set of sensors, A is the set of actuators, H is the desired heading for the actuators, $getIntensityDiff(s_i)$ is the function returning the intensity difference of node s_i and $calculateActuatorDirection(a_j, s_i)$ is the function that returns the heading direction of the given actuator a_j to the sensor s_i. The process of the algorithm is as follows. The drone visits all sensor nodes. While visiting all nodes, the drone receives light condition data and generates light condition change data. After the data collection, the drone calculates the command, which makes the actuator change direction. After the commands are generated, the drone visits the actuators and rotates their cameras toward the sensor with the highest light condition change data.

Algorithm 4 DaaG IoT control decision algorithm.

```
enableBeaconing()                                   ▷ Run Algorithm 1 in background
while isDaaGOperating() do
    Δ lux_max = 0
    s_max = 0                                       ▷ Initialize variables
    collectDataFromSensors()                        ▷ Run Algorithm 2 to collect data
    for s_i ∈ S do
        Δ lux_i = getIntensityDiff(s_i)
        if Δ lux_max < Δ lux_i then
            Δ lux_max = Δ lux_i
            s_max = s_i
        end if
    end for
    for a_j ∈ A do
        h_j = calculateAcutatorDirection(a_j, s_max)
    end for
    commandToActuators(H)                           ▷ Run Algorithm 3 to control actuators
end while
```

4.2.3. Prototypes for Sensors and Actuators

There are two node types, a sensor node and an actuator node, in our experiment. Figure 11c is a sensor-mounted prototype, and Figure 11d is an actuator-mounted prototype. We use Arduino Uno for the controller and the Xbee Wi-Fi module as a network interface. On the sensor node, the BH1750FVI light

sensor is attached to collect the light intensity. For the actuator node, a camera is mounted and films the environment. A servo motor is attached to the camera to adjust the viewing direction of the camera.

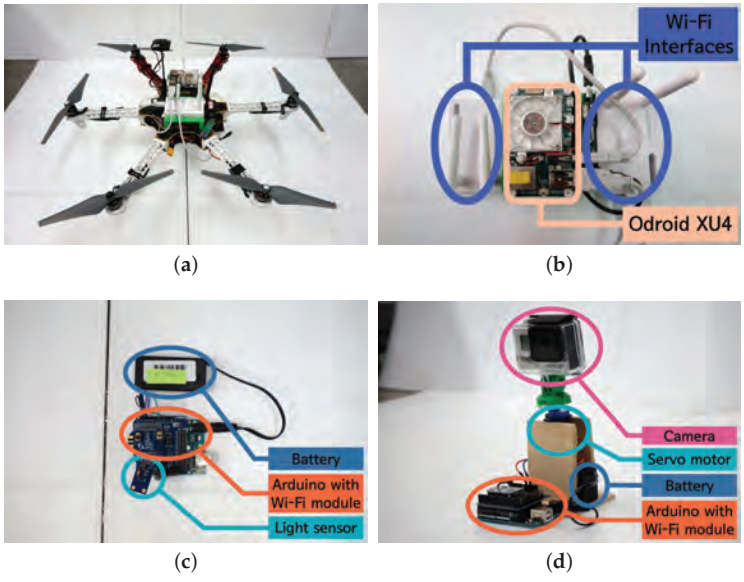


Figure 11. The implementation of DaaG, sensors and actuators. (**a**) The hexa-copter-based DaaG. (**b**) The computing module. (**c**) The IoT sensor module. (**d**) The IoT actuator module.

4.2.4. Drone Control System

The DaaG ground control station (GCS) is written in Python, which could run on almost any computing hardware. The GCS manages the information of every active DaaG and commands to the DaaGs in the situations of 'low battery', 'possibility of collision' and 'urgent change of point of interest'. To achieve the features mentioned above, the GCS receives information of DaaGs with a UDP socket and sends drone control commands with a TCP socket when needed. Since the mission duration is important for the performance of DaaG, we enabled the drone hand-off, which has been proposed in [39], to lengthen the duration.

4.3. Empirical Deployment

Following the experimental scenario, we have conducted the experiment in an outdoor environment. Figure 10a shows the basic layout of our experiment where the location of experiment is Seoul, Korea, where the latitude is 37.547329, and the longitude is 127.119588. The size of the experimental setting is 36.61 m by 40.14 m. The distance between sensors and actuators is about 31 m. In order to prevent sensors from communicating with actuators directly, we reduced the transmission power of the sensors' network interface to 1 dbm. The velocity of DaaG is set 3 m/s. The threshold value for distinguishing a problematic sensor node I_{th} is set to 10,000.

If the value of the light intensity sensor reaches I_{th}, then the DaaG system assumes that the sensor node is a problematic sensor node.

To prove that the DaaG system has smartness and works well in the separated network environment, we have chosen a nearest-based path-planning algorithm as a comparison target, which is a general algorithm for UAV that can reduce the travel time and battery consumption. With both algorithms, we conducted an empirical study to monitor sensors that have a value less than

the threshold value. Furthermore, we selected the data ferry method to deliver sensing and actuating data in our outdoor experiment.

4.4. Empirical Evaluation

With the implemented sensors, actuators and a drone with DaaG in the actual deployment, we conducted an experiment according to the experiment scenario, observed the results and examined them.

4.4.1. Observing Empirical Results

Figure 10 shows the demonstration of the actual experiment conducted. Our video [40] shows the whole experimentation procedure and result. The detailed procedure and result of the experiment are as follows:

- First, based on the starting position of Drone #1 and the location of the sensing and actuating area, the expected path of Drone #1 is in Figure 10a for both the DaaG and nearest-based planned path drone.
- Since the light intensity value of Sensor #1 is lower than the threshold in our scenario, Drone #1 runs Algorithm 3, then Sensor #1 is registered in PSL, and $corr_{s1,a1}$ will be the highest.
- After that, Drone #1 changes the path to visit Actuator #1 rather than Actuator #2, and Figure 10b shows the changed path after running Algorithm 3.
- When Drone #1 has arrived at Actuator #1, it sends an actuation message, which rotates the servo motor to focus on Sensor #1, to Actuator #1, as Figure 10c shows.

In the case of the nearest-based planned path drone, the path for traveling in the sensing area was the same as that of the DaaG and the planned path. However, the drone visits Actuator #2 first, following the planned path, which is different from the path DaaG chooses. The drone sends the order message to Actuator #2, then it goes to Actuator #1.

We can see from the experiment that the separated network infrastructure is connected immediately. Before using DaaG, the sensor network can only sense the light intensity, and the actuator network can only film some places. However, after using DaaG, these networks can act as a surveillance system for the brightness problem. This system can create a new value, which the existing network cannot provide.

4.4.2. Examining Observations

As a validation, we have analyzed the delay of our system with a series of experiments. In the analyses, both the main focusing delay and the total service delay are considered, so that the observations made for the field experiment match those for the simulations. In our scenario, the main actuator can be Actuator #1, because Sensor #1 is registered in PSL and Actuator #1 is close to Sensor #1 compared to Actuator #2.

Figure 12 shows the result of delay measurement based on our experiment. In the result, it can be seen that DaaG shows a better result compared to nearest-based path-planning with respect to the main focusing delay. The entire delay of our system for moving the drone and operating the actuator is 46.46% shorter than that of the nearest-based path-planning system. In our experiment, the nearest-based drone visited Actuator #2, which is not the main actuator. It can be seen from Figure 12 that the total service delay of DaaG is only 2.08% longer than that of the nearest-based path-planning system.

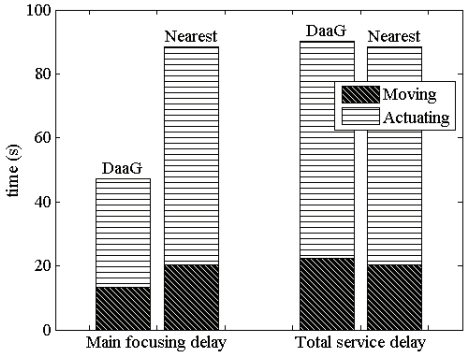


Figure 12. Delay measurement from the experiment.

With these results, we should notice that the essential purpose of this system is to serve a new service by connecting separate networks. Following this purpose, the main focusing delay is more important than the total service delay, because a short main focusing delay can provide the priority service rapidly. The results shows that our platform is much faster than the nearest-based path-planning system, so that the main service can be rapidly provided with our platform. Naturally, the total service delay is also important to increase the quality and accuracy of the service, as a lengthy delay can negatively affect the service quality. However, our system requires only 2.08% additional time, which is negligible, compared to the nearest-based path-planning system.

5. Conclusions

We have designed and implemented DaaG, a drone that provides connectivity between separate networks as a mobile gateway. Our simulation and experiment results show that DaaG can connect between the sensor network and actuator network. Furthermore, DaaG uses various algorithms to connect these separate networks efficiently.

There is a large variety of possible application scenarios for the system. The drone can fly to dangerous regions that people cannot reach. Therefore, a disaster situation is the most appropriate example for this scenario. Another example is an agricultural application of an IoT system. Examples of sensing devices in this scenario would be for light, temperature and moisture. Possible actuators would be sprinklers or spraying fertilizers and pesticides. With DaaG, we believe more applications in addition to these scenarios can be realized.

The smart and path-planning algorithm in DaaG can be improved, for example to be able to run appropriately in an environment in which the system does not have knowledge of the service type, nor the map of the actuators, which is our future work.

Acknowledgments: The first three authors equally contributed to this paper. This research was supported by the Unmanned Vehicles Advanced Core Technology Research and Development Program through the Unmanned Vehicle Advanced Research Center (UVARC) funded by the Ministry of Science, ICT and Future Planning, the Republic of Korea (NRF-2016M1B3A1A01937599).

Author Contributions: Hwangnam Kim managed the overall procedure of this research. Mungyu Bae, Seungho Yoo and Jongtack Jung designed and implemented DaaG. Mungyu Bae, Seungho Yoo, Kangho Kim and Joon Yeop Lee designed and performed the experiments. Seongjoon Park designed and performed the simulations. Mungyu Bae analyzed the data. Mungyu Bae, Seungho Yoo and Jongtack Jung wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

WSAN	Wireless sensor and actuator network
DaaG	Drone-as-a-gateway
IoT	Internet of Things
M2M	Machine-to-machine
DoF	Degree of freedom
UAV	Unmanned aerial vehicle
GPS	Global Positioning System
PSL	Problematic sensor list
ACL	Actuator correlation value list
TSPN	Traveling salesman problem with neighborhoods
CETSP	Close enough traveling salesman problem
NCPST	Normalized cumulative priority service throughput
OLSR	Optimized link state routing
CDF	Cumulative distribution function
GCS	Ground control station

References

1. Verdone, R.; Dardari, D.; Mazzini, G.; Conti, A. *Wireless Sensor and Actuator Networks: Technologies, Analysis and Design*; Academic Press: Cambridge, MA, USA, 2008.
2. Ericsson AB. Ericsson Mobility Report, November 2017. Available online: http://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-november-2017.pdf (accessed on 15 February 2018).
3. Deloitte. *Cyber-Physical Systems-Government 2020* ; Deloitte Development LLC: Oakland, CA, USA , 2014.
4. Lee, W.; Kim, J.; Kang, J. Automated construction of node software using attributes in a ubiquitous sensor network environment. *Sensors* **2010**, *10*, 8663–8682.
5. Wu, F.J.; Tseng, Y.C. Energy-conserving data gathering by mobile mules in a spatially separated wireless sensor network. *Wirel. Commun. Mob. Comput.* **2013**, *13*, 1369–1385.
6. Ngai, E.C.; Lyu, M.R.; Liu, J. A real-time communication framework for wireless sensor-actuator networks. In Proceedings of the 2006 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2006; IEEE: Piscataway, NJ, USA; p. 9.
7. Meyer, F.; Wymeersch, H.; Fröhle, M.; Hlawatsch, F. Distributed Estimation With Information-Seeking Control in Agent Networks. *IEEE J. Sel. Areas Commun.* **2015**, *33*, 2439–2456.
8. Ozdemir, S.; Xiao, Y. Secure data aggregation in wireless sensor networks: A comprehensive overview. *Comput. Netw.* **2009**, *53*, 2022–2037.
9. Lee, S.K.; Bae, M.; Kim, H. Future of IoT Networks: A Survey. *Appl. Sci.* **2017**, *7*, 1072.
10. Dardari, D.; Conti, A.; Buratti, C.; Verdone, R. Mathematical Evaluation of Environmental Monitoring Estimation Error through Energy-Efficient Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2007**, *6*, 790–802.
11. Ferdoush, S.; Li, X. Wireless Sensor Network System Design Using Raspberry Pi and Arduino for Environmental Monitoring Applications. *Procedia Comput. Sci.* **2014**, *34*, 103–110.
12. Zabini, F.; Conti, A. Inhomogeneous Poisson Sampling of Finite-Energy Signals With Uncertainties in $\mathbb{R}^d$. *IEEE Trans. Signal Process.* **2016**, *64*, 4679–4694.
13. Deaconu, I.; Voinescu, A. Mobile gateway for wireless sensor networks utilizing drones. In Proceedings of 2014 RoEduNet Conference 13th Edition: Networking in Education and Research Joint Event RENAM 8th Conference, Chisinau, Moldova, 11–13 September 2014; IEEE: Piscataway, NJ, USA; pp. 1–5.
14. Saleem, Y.; Salim, F.; Rehmani, M.H. Resource management in mobile sink based wireless sensor networks through cloud computing. In *Resource Management in Mobile Computing Environments*; Springer: Cham, Switzerland , 2014; pp. 439–459.
15. Tunca, C.; Isik, S.; Donmez, M.Y.; Ersoy, C. Distributed mobile sink routing for wireless sensor networks: A survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 877–897.

16. Trasviña-Moreno, C.A.; Blasco, R.; Marco, Á.; Casas, R.; Trasviña-Castro, A. Unmanned aerial vehicle based wireless sensor network for marine-coastal environment monitoring. *Sensors* **2017**, *17*, 460.
17. Cao, H.; Liu, Y.; Yue, X.; Zhu, W. Cloud-Assisted UAV Data Collection for Multiple Emerging Events in Distributed WSNs. *Sensors* **2017**, *17*, 1818.
18. Tzoreff, E.; Weiss, A.J. Path Design for Best Emitter Location Using Two Mobile Sensors. *IEEE Trans. Signal Process.* **2017**, *65*, 5249–5261.
19. Jung, J.; Kim, K.; Lee, W.; Kim, H. Poster: Asynchronous Acoustic Localization using Commercial Devices. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, Seoul, Korea, 1–4 November 2015; ACM: New York, NY, USA, 2015; pp. 393–394.
20. Win, M.Z.; Conti, A.; Mazuelas, S.; Shen, Y.; Gifford, W.M.; Dardari, D.; Chiani, M. Network localization and navigation via cooperation. *IEEE Commun. Mag.* **2011**, *49*, 56–62.
21. Bartoletti, S.; Dai, W.; Conti, A.; Win, M.Z. A Mathematical Model for Wideband Ranging. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 216–228.
22. Conti, A.; Dardari, D.; Guerra, M.; Mucchi, L.; Win, M.Z. Experimental Characterization of Diversity Navigation. *IEEE Syst. J.* **2014**, *8*, 115–124.
23. Liu, Z.; Dai, W.; Win, M.Z. Node placement for localization networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6.
24. Dumitrescu, A.; Mitchell, J.S.B. Approximation algorithms for TSP with neighborhoods in the plane. *CoRR* **2017**, arXiv:1703.01640.
25. Gulczynski D.J.; Heath J.W.; Price C.C. The Close Enough Traveling Salesman Problem: A Discussion of Several Heuristics. *Perspectives in Operations Research*; Springer: Boston, MA, USA, 2006; Volume 36, pp. 271–283, ISBN 978-0-387-39933-1.
26. La, W.G.; Park, S.; Kim, H. D-MUNS: Distributed multiple UAVs' network simulator. In Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 4–7 July 2017; pp. 15–17.
27. The ns-3 Network Simulator. Available online: https://www.nsnam.org/ (accessed on 15 February 2018).
28. Furrer, F.; Burri, M.; Achtelik, M.; Siegwart, R. Robot Operating System (ROS). In *Studies in Computational Intelligence*; Springer: Cham, Switzerland, 2016; Chapter 23, Volume 625, pp. 595–625, ISBN 978-3-319-26052-5.
29. Arya, S.; Mount, D.M.; Netanyahu, N.S.; Silverman, R.; Wu, A.Y. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *JACM* **1998**, *45*, 891–923.
30. Salarian, H.; Chin, K.W.; Naghdy, F. An energy-efficient mobile-sink path selection strategy for wireless sensor networks. *IEEE Trans. Veh. Technol.* **2014**, *63*, 2407–2419.
31. DaaG Simulation Result with Multi-Hop Network Method. Available online: https://youtu.be/cusJZlA4WMc (accessed on 15 February 2018) .
32. Heinzelman, W.R.; Chandrakasan, A.; Balakrishnan, H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 4–7 January 2000; IEEE: Piscataway, NJ, USA, 2000; p. 10.
33. Jafri, M.R.; Javaid, N.; Javaid, A.; Khan, Z.A. Maximizing the lifetime of multi-chain pegasis using sink mobility. *arXiv preprint* **2013**, arXiv:1303.4347.
34. Bagci, H.; Yazici, A. An energy aware fuzzy approach to unequal clustering in wireless sensor networks. *Appl. Soft Comput.* **2013**, *13*, 1741–1749.
35. Chung, A.Y.; Jung, J.; Kim, K.; Lee, H.K.; Lee, J.; Lee, S.K.; Yoo, S.; Kim, H. Poster: Swarming drones can connect you to the network. In Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, Florence, Italy, 19–22 May 2015; ACM: New York, NY, USA, 2015; p. 477.
36. Yoo, S.; Jung, J.; Chung, A.Y.; Kim, K.; Lee, J.; Park, S.; Lee, S.K.; Lee, H.K.; Kim, H. Empowering Drones' Teamwork with Airborne Network. In Proceedings of the 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), Taipei City, Taiwan, 27–29 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 678–685.
37. Meier, L.; Tanskanen, P.; Heng, L.; Lee, G.H.; Fraundorfer, F.; Pollefeys, M. PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight Using Onboard Computer Vision. *Auton. Robots* **2012**, *33*, 21–39.

38. Meier, L.; Camacho, J.; Godbolt, B.; Goppert, J.; Heng, L.; Lizarraga, M. Mavlink: Micro Air Vehicle Communication Protocol. Available online: http://qgroundcontrol.org/mavlink/start (accessed on 15 February 2018).
39. Lee, J.; Kim, K.; Yoo, S.; Chung, A.Y.; Lee, J.Y.; Park, S.J.; Kim, H. Constructing a reliable and fast recoverable network for drones. In Proceedings of the Communications (ICC), 2016 IEEE International Conference on, Kuala Lumpur, Malaysia, 22–27 May 2016; IEEE: Piscataway, NJ, USA , 2016; pp. 1–6.
40. Devising Mobile Sensing and Actuation Infrastructure with Drones. Available online: https://youtu.be/x_xYiXeFMrs (accessed on 15 February 2018).

sensors

MDPI

Article

Comparing RIEGL RiCOPTER UAV LiDAR Derived Canopy Height and DBH with Terrestrial LiDAR

Benjamin Brede [1,*], Alvaro Lau [1,2], Harm M. Bartholomeus [1] and Lammert Kooistra [1]

[1] Laboratory of Geo-Information Science and Remote Sensing, Wageningen University & Research, Droevendaalsesteeg, 36708 PB Wageningen, The Netherlands; alvaro.lausarmiento@wur.nl (A.L.); harm.bartholomeus@wur.nl (H.M.B.); lammert.kooistra@wur.nl (L.K.)

[2] Center for International Forestry Research (CIFOR), Situ Gede, Sindang Barang, Bogor 16680, Indonesia

* Correspondence: benjamin.brede@wur.nl (B.B.)

Received: 15 August 2017; Accepted: 13 October 2017; Published: 17 October 2017

Abstract: In recent years, LIght Detection And Ranging (LiDAR) and especially Terrestrial Laser Scanning (TLS) systems have shown the potential to revolutionise forest structural characterisation by providing unprecedented 3D data. However, manned Airborne Laser Scanning (ALS) requires costly campaigns and produces relatively low point density, while TLS is labour intense and time demanding. Unmanned Aerial Vehicle (UAV)-borne laser scanning can be the way in between. In this study, we present first results and experiences with the RIEGL RiCOPTER with VUX®-1UAV ALS system and compare it with the well tested RIEGL VZ-400 TLS system. We scanned the same forest plots with both systems over the course of two days. We derived Digital Terrain Models (DTMs), Digital Surface Models (DSMs) and finally Canopy Height Models (CHMs) from the resulting point clouds. ALS CHMs were on average 11.5 cm higher in five plots with different canopy conditions. This showed that TLS could not always detect the top of canopy. Moreover, we extracted trunk segments of 58 trees for ALS and TLS simultaneously, of which 39 could be used to model Diameter at Breast Height (DBH). ALS DBH showed a high agreement with TLS DBH with a correlation coefficient of 0.98 and root mean square error of 4.24 cm. We conclude that RiCOPTER has the potential to perform comparable to TLS for estimating forest canopy height and DBH under the studied forest conditions. Further research should be directed to testing UAV-borne LiDAR for explicit 3D modelling of whole trees to estimate tree volume and subsequently Above-Ground Biomass (AGB).

Keywords: UAV; LiDAR; ALS; TLS; forest inventory

1. Introduction

LIght Detection And Ranging (LiDAR) has become a valuable source of information to assess vegetation canopy structure. This is especially true for complex forest canopies that limit manual and destructive sampling. These capabilities are investigated to replace traditional forest plot inventories [1], but even more if they can deliver additional information that is not captured with traditional inventories [2]. One particular important variable in this context is Above-Ground Biomass (AGB) which makes up an essential part of the forest carbon pool. Terrestrial Laser Scanning (TLS) has the potential to accurately measure AGB on a plot scale [3,4], while Airborne Laser Scanning (ALS) from manned aircraft can serve as means to up-scale plot measurements to the landscape level. This is particularly interesting for calibration and validation activities of space-borne missions aiming at AGB assessment like ESA's BIOMASS [5] and NASA's GEDI (https://science.nasa.gov/missions/gedi) missions. Another important derivative of LiDAR point clouds is vertical forest canopy structure, which is linked to biodiversity [6,7].

ALS is typically acquired from manned aircraft, thereby covering large areas, but requiring substantial financial capital and available infrastructure. Acquisition density is typically in the order of 1 to 10 points/m^2, depending on flight altitude and scanner configuration. A straight-forward application for ALS point clouds is the generation of Digital Terrain Models (DTMs) and Digital Surface Models (DSMs), and derivation of canopy height by considering the difference between those two. More advanced products take into account the waveform of the returning pulses and reconstruct canopy attributes from that [8]. However, the relatively low density of ALS point clouds forces to approach actual canopy structure from a statistical point of view where each resolution cell contains a sample of the population of possible returns. In this respect, ALS products can be treated as 2.5D raster layers.

On the other hand, TLS produces point clouds with such a density—millions of points per scan position—that single canopy elements like stems and branches can be resolved. Geometrical models serve to reconstruct the 3D tree architecture, and allow estimation of wood volume and derivation of AGB [4,9,10] and other stand characteristics. A hard requirement for this approach is accurate co-registration of several point clouds acquired from different scan positions in the forest, which leads to time demanding field campaigns, mostly in the order of 3 to 6 days/ha [11]. Therefore, it is questionable if TLS in its current form will replace operational plot inventories, or rather supply higher quality information for selected samples [2].

Independent from the developments of LiDAR instruments, Unmanned Aerial Vehicles (UAVs) have found use as platforms for various types of sensors in forestry and many other fields [12,13]. Especially the introduction of affordable, ready-to-use systems on the consumer market has been boosting applications and widened the user community. Even consumer-grade RGB cameras in combination with dedicated software packages can serve for the production of high-resolution orthomosaics and surface models derived with Structure from Motion (SfM) techniques. More sophisticated prototype sensors also allow the production of hyperspectral images [14]. One of the most favourable aspects of UAVs as sensor platforms is their low demand in infrastructure, high mapping speed and price advantage compared to manned aircraft. The implementation of legal regulations for professional UAV users remains a hot topic however [12].

Recently, LiDAR sensors have been mounted on UAVs to combine the advantages of LiDAR and UAV technology. A variety of custom build systems with different degrees of off-the-shelf components were tested to derive forest inventory metrics. Jaakkola et al. [15] probably build the first UAV LiDAR system, the Finish Geodetic Institute (FGI) Sensei, integrating an Ibeo Lux and Sick LMS151 profile scanner. During test flights the Sensei produced point clouds with 100 to 1500 points/m^2 and could be successfully used to detect single trees. Another custom build system based on the Ibeo Lux scanner was presented by Wallace et al. [16]. During tests it produced point clouds with up to 50 points/m^2, but with a relatively low airborne time of 3 to 5 min owed to the capacity of the UAV. This same system was employed to conduct surveys of forest plots, and terrain and under-storey height, tree location, tree height, crown area and volume could be derived [17]. Chisholm et al. [18] constructed another light-weight LiDAR UAV system that did not require any means of positioning or orientation system, but rather used pattern-matching algorithms to produce a point cloud. However, due to assumptions in the processing the system and the low range of the laser scanner of 30 m had to be flown below canopy. They could successfully estimate Diameter at Breast Height (DBH) for their open grove study site. Wei et al. [19] employed the commercially available HawkScan1200, consisting of a VUX®-1LR scanner and Applanix AP20 Inertial Measurement Unit (IMU), and mapped a 60 km^2 area with a point density of 0.5 points/m^2 to perform vegetation filtering and DTM generation on the resulting point cloud.

Overall, these systems showcase that principal technological challenges such as component miniaturisation and suitable post-processing have been overcome in the recent years. Important forest inventory metrics like tree height, location and DBH could be derived. Nonetheless, custom-build systems have not yet achieved point density counts in same the order of magnitude as TLS. This would

open up opportunities that are at the forefront of LiDAR research in forestry, such as explicit structural modelling to precisely estimate AGB [4,9]. Moreover, even though custom build systems are low cost, at the same time they are typically not easily available for use by a wider audience.

The aim of this paper is to present the commercially available RIEGL RiCOPTER system and the work flow to process the acquired data. In a field experiment we concurrently collected RiCOPTER and TLS data in a forest site containing different canopy architectures. We compared the two point clouds in respect to their point distributions, different elevation models derived from both point clouds and estimates of DBH. With this comparison we want to test if the RiCOPTER performs comparable to TLS field acquisition.

2. RIEGL RiCOPTER with VUX®-1UAV

2.1. System Specifications

The RIEGL RiCOPTER with VUX®-1UAV (RIEGL Laser Measurement Systems GmbH, Horn, Austria) is an integrated UAV and sensor system. The RiCOPTER is a battery-driven octocopter with an empty weight (no batteries and equipment) of 9 kg that can carry a payload of up to 8 kg. For safety reasons it has two flight controller units. In case of system failure of the main controller, the backup takes over immediately. Together with the VUX®-1UAV scanner (3.75 kg), the system controller (0.9 kg), the IMU (0.7 kg) and optional cameras the total system weights just under 25 kg; hence, it is possible to operate it under light UAV regulations in many European countries [13]. The batteries allow flight times of up to 30 min at 30 km h^{-1} maximum cruise speed. This allows flying multiple overlapping flight lines to increase target coverage. However, during mapping of forest plots flight time and speed need to be reduced to guarantee flight safety and adequate point density.

The VUX®-1UAV is a survey-grade laser scanner that is mounted underneath the RiCOPTER. It uses a rotating mirror with a rotation axis in flight direction to direct the laser pulses and achieve an across-track Field Of View (FOV) of 330° perpendicular to the flight direction. This means that lateral flight line overlap is only restricted by the maximum operating range of the laser. Overall its attributes are comparable to the terrestrial VZ-400 despite its lower weight (Table 1). It should be noted that both operate at a wavelength of 1550 nm, which makes them eye-safe and sensitive to the same types of canopy elements. An Applanix AP20 IMU attached to the VUX®-1UAV and Global Navigation Satellite System (GNSS) antennas on top of the RiCOPTER record flight orientation and GNSS data. Apart from these devices and sensors essential for processing, two consumer-grade Sony Alpha-6000 system cameras can be mounted on the VUX®-1UAV. During later processing the point clouds can be overlaid with the RGB colour information from these cameras. The on-board instrument controller manages all sensors' data streams and includes a 220 GB SSD storage, which is sufficient for several missions.

Table 1. VZ-400 and VUX®-1UAV main characteristics.

Characteristic	VZ-400 [1]	VUX-1UAV [2]
Maximum Pulse Repition Rate (PRR) (kHz)	300	550
Maximum effective measurement rate (kHz)	120	500
Minimum | Maximum range (m)	1.5 | 350 [3]	3 | 920 [4]
Accuracy | Precision (mm)	5 | 3	10 | 5
Laser wavelength (nm)	1550	1550
Beam divergence (mrad)	0.3	0.5
Weight (kg) [5]	9.6	3.75

[1] high speed mode, incl. online waveform processing; [2] 550 kHz mode; [3] at target $\rho \geq 0.9$; [4] at target $\rho \geq 0.6$; [5] without battery and tilt mount.

Next to the RiCOPTER system a ground station is necessary for mission planning and in-flight coordination. Planar or spherical Ground Control Points (GCPs) should be set out in the field before flight to support co-registration during processing. These targets do not necessarily need to be

geolocated in case only internal point cloud registration is to be optimised. However, they should have an adequate size of >0.5 m—depending on flight altitude and scanning speed—to be properly covered. In case sufficient planar surfaces are available in the study area, these can also be used. However, this is typically not the case for forest plots.

2.2. Operations

Necessary legal requirements for professional operations are similar to other UAV operations and mainly involve RiCOPTER registration as an aircraft in the country of operations as well as the training and licensing of the pilot. Both processes can partly run in parallel and can take up to several months. Additional to regular licensing the pilot should also become familiar with the flight behaviour of the RiCOPTER, since it is considerably larger than typical mini-UAV. Also the proper operation of the two independent flight controllers needs to be trained. Moreover, operation in forest areas usually requires take off and landing in canopy openings with restricted viewing conditions and options to manoeuvre. Another general preparation includes the identification of a source of base station data that is necessary for processing the acquired data. Additionally, legal requirements for the transportation of the batteries need to be investigated.

Once these general prerequisites are fulfilled, practical mission planning can begin. This mainly involves getting access permissions to the study site especially from the landowner, arranging transport and notifying other airspace users. Furthermore, the weather forecast should be studied with respect to wind, visibility and humidity to identify the best suitable days for mission execution. As for other mini-UAV the RiCOPTER has a legal limit on wind speed up to which take off is allowed, which is $7\,\mathrm{m\,s^{-1}}$ for the Netherlands. However, wind limits are typically stricter in respect to data quality as crown movement hampers proper co-registration of point clouds from different flight lines, as is also the case for TLS [11].

Initial flight path planning should be performed in preparation of the field work. The target is a certain point density to be achieved by varying flying speed and altitude, and overlap of flight lines. Nonetheless, not anticipated on-site conditions like single emerging trees or lack of emergency landing locations can demand modification. Transport to the site should take into account the size and weight of the equipment. The RiCOPTER itself is delivered with a transport case of ~120 cm × 80 cm × 55 cm. The ground station has dimensions ~55 cm × 45 cm × 20 cm. At the study site, the area should be inspected to identify take-off and landing as well as emergency landing locations, obstacles close the intended flight path and positions for GCPs. After completion the equipment can be set up and the mission executed. After the mission, the raw data is downloaded from the instrument controller.

2.3. Data Processing

RIEGL provides a software suite together with the RiCOPTER system to convert the produced raw data into point clouds. Figure 1 gives an overview of the required steps. While most of the work can be done in RIEGL's software for airborne and mobile laser scanning, RiPROCESS, the trajectory preprocessing has to be accomplished with third party software, e.g., Applanix POSPac Mobile Mapping Suite. For this purpose additional GNSS base station data has to be acquired. During GNSS post-processing both data streams from the GNSS antennas and the IMU are taken into account to reconstruct the flight trajectory.

For each flown and logged scan line, the raw scan data has to be subjected to waveform analysis during which targets are detected within the stored flight line waveforms. Up to four targets can be detected per pulse. During this process, Multiple Time Around (MTA) range ambiguities have to be taken care of. MTA range ambiguity occurs when pulses are fired before their predecessor pulses can return. MTA 1 range, where no ambiguity can occur because a pulse always returns before the next is fired, is at around 100 m range for a Pulse Repition Rate (PRR) of 550 Hz. Thus MTA range ambiguity needs to be taken care of, but does not result in serious obstacles assuming flying heights below 120 m. The waveform processing detects targets in the scanners own coordinate system. Next, this data is

interpreted with help of the trajectory information and the scanner mounting orientation to produce the first point clouds, one per flight line. The within and across flight line registration is already of high quality as experienced by the authors during several missions and would serve when registration error of below 1 m is not an issue.

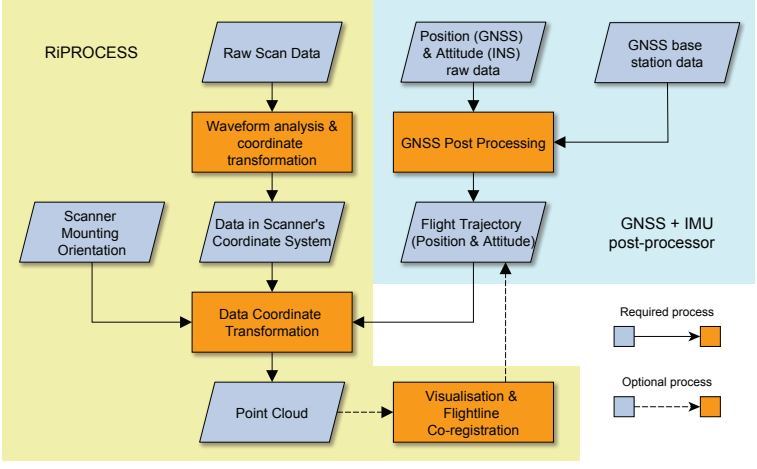


Figure 1. RiCOPTER processing flowchart (based on [20]).

However, when sub-centimetre accuracy is required, point clouds need to be fine registered. In principal this means to reduce errors in the flight trajectory as the position and orientation error of the RiCOPTER system (cm scale) is much bigger than the range error of the scanner (mm scale). This process can include two independent steps. One is the within flight line trajectory optimisation that is handled by the RiPRECISION package within RiPROCESS. Similar to Multi-Station Adjustment (MSA) for TLS [11] control planes are automatically searched for in the point clouds per flight line. So far this process demands no user interaction. It can be supported by GCPs that have been independently located to within millimetres, e.g., with Real Time Kinematic (RTK) GNSS. However, for forest situations this is impractical as GNSS reception is typically too low under closed canopies to achieve the required accuracy. The other possible optimisation is across flight line optimisation that can be performed with the Scan Data Adjustment (SDA) package within RiPROCESS. It assumes within flight line registration as optimal and only changes the overall position and orientation of flight lines to each other. This can be compared to linking point clouds from single scanning positions in TLS. Here, next to automatically detected planes, also manually digitised control objects, such as planes, spheres and points, can be included to improve the co-registration.

The point cloud processing can be finished off with removal of atmospheric noise, which is visible as returns close to the flight trajectory with typically low reflectivity, and target type classification. Finished point clouds can be exported from RiPROCESS in common file formats, e.g., ASCII and LAS, to continue analysis in dedicated software packages.

3. Field Experiment

The field experiment took place at the Speulderbos Fiducial Reference site in the Veluwe forest area (N52°15.15′ E5°42.00′), The Netherlands [21] (www.wur.eu/fbprv). The core site is established in a stand composed of European Beech (*Fagus sylvatica*) with occasional Pedunculate Oak (*Quercus robur*) and Sessile Oak (*Quercus petraea*), and a very open understorey with only few European Holly (*Ilex aquifolium*) (Figure 2). The stand was established in 1835 and had a tree density of

204 trees/ha. At the time of the experiment the overstorey was in the progress of spring bud break and leaf unfolding, so that only few trees carried a full leaf canopy. In an earlier inventory campaign the Beech stand has been equipped with a 40 m spaced wooden pole grid that has also been geo-located with RTK GPS and land surveying techniques to an accuracy of better than 0.5 m.

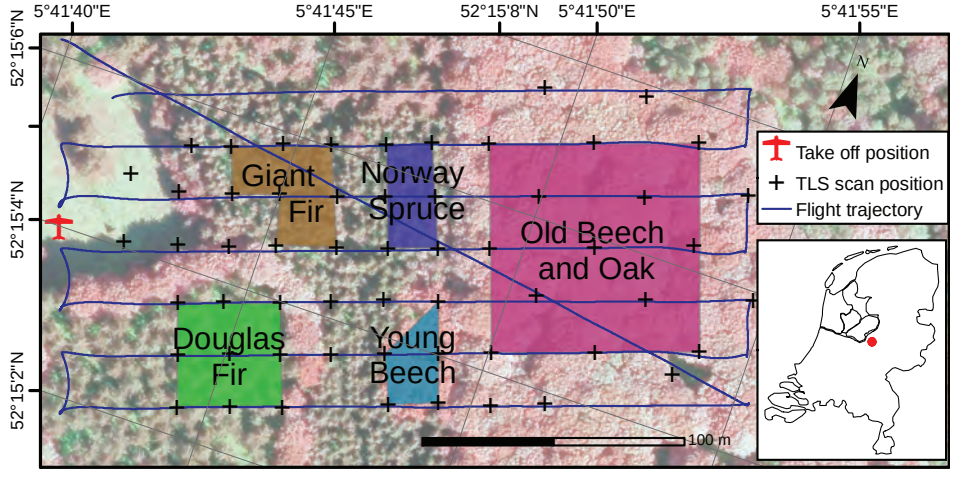


Figure 2. Map of the study site with TLS scan positions (crosses), take off position (red plane), flight lines (blue), target areas, study site location in the Netherlands (red dot in inset), airborne false colour composite as background image.

Additional to the Beech stand, sections of Norway Spruce (*Picea abies*), Giant Fir (*Abies grandis*), young beech and Douglas Fir (*Pseudotsuga menziesii*) have been scanned as well with the goal to capture different forest types in terms of species composition, tree density and canopy architecture. The Norway Spruce and Giant Fir stands were established in 1943 and 1967, respectively, and had no understorey species. However, the plots were relatively dense with 676 Trees/ha and 961 Trees/ha, respectively, and many low branches. The young beech stand was established in 1973 and had a density of 805 Trees/ha. There were no other species present in this stand, but most lower branches were carrying leaves. The Douglas Fir stand had a very open understorey where only few saplings of up to 2 m height could be found. The stand was established in 1959 and has been thinned since then as was obvious through the present stumps.

The total scanned area covered 100 m $\times$ 180 m, roughly 2 ha. In the study area, a forest road separates the old beech and oak from the other stands, and a bike path the Giant Fir and Norway Spruce stands. The UAV take-off area was located in an opening east of the stands that was wide enough to allow Visual Line of Sight (VLOS) operations.

TLS data acquisition was completed in the course of two days that were both marked by very low wind speeds (<3 m s^{-1}). During the first day the TLS scan position grid was set up. For that the wooden poles in the Beech stand were taken as starting positions. With the help of a theodolite the TLS positions were marked to form a grid of 40 m spacing in the Beech and 20 m spacing in the Douglas Fir stands. Additional positions in the grid centres have been added in the Beech stand. Cylindrical retro-reflective targets were set up for later coarse co-registration of scans [11]. The first 15 positions have been scanned during the first day, the remaining 43 during the second day. All scans were performed with 0.06° scan resolution. Due to the VZ-400's zenithal scan angle range of 30° to 130°, an upward and tilted scan had to be performed per scan location to cover the area directly over the scan position.

To support co-registration of RiCOPTER flight lines 4 large (120 cm × 60 cm) and 8 small (60 cm × 60 cm) ground control panels have been distributed under the trees and next to the take-off site. The panels consist each of 2 equally sized wooden panes connected via piano hinges. When set up the panes form a 90° angle between them, which makes them look like tents. Cars used for transport were deliberately parked on the forest road in the scanning area to provide additional control surfaces for the co-registration. The ground station was erected next to the take off site. The scan lines were designed to maximise lateral overlap and efficiently use air time (Figure 2). The RiCOPTER was flown at an altitude of 90 m a.g.l., with a cruise speed of 6 m s^{-1}. The VUX®-1UAV was operated with the full FOV of 330°, a PRR of 550 kHz and scan speed of 58 lines/s, which resulted in an average rectangular point spacing of ~8 cm and a point density of 140 points/m^2 for a single flight line at nadir. Mission time for active scan and non-active connection lines was 9 min to cover a distance of ~2300 m.

4. Methods

In case of TLS, the scans were first coarsely co-registered with automatically extracted tie-points based on the retro-reflective cylinders. These registrations had typical registration errors of <5 cm. Afterwards MSA was applied to refine the registration [11]. This approach automatically searches for tie-planes in the scans and iteratively adjusts orientation and position of each scan position to minimise the global fitting error of tie-planes. The resulting standard deviation of the errors over all tie-planes was 0.62 cm. All operations were executed with RIEGL's RiSCAN PRO® software package.

For processing of the RiCOPTER data the work-flow as described in Section 2.3 was applied. GNSS data was obtained from 06-GPS (Sliedrecht, The Netherlands) for a virtual base station in the centre of the study site and the period of the campaign to allow GNSS post-processing. RiPRECSION-UAV was applied to optimise within flight line registration. Automatic across-flight line registration with automatic search of tie-planes continuously failed to produce good results, probably due to missing planar surfaces in the study area. Therefore, the GCP panels were manually digitised as tie-planes and used for fine registration. Final standard deviation of the fitting errors of 0.97 cm.

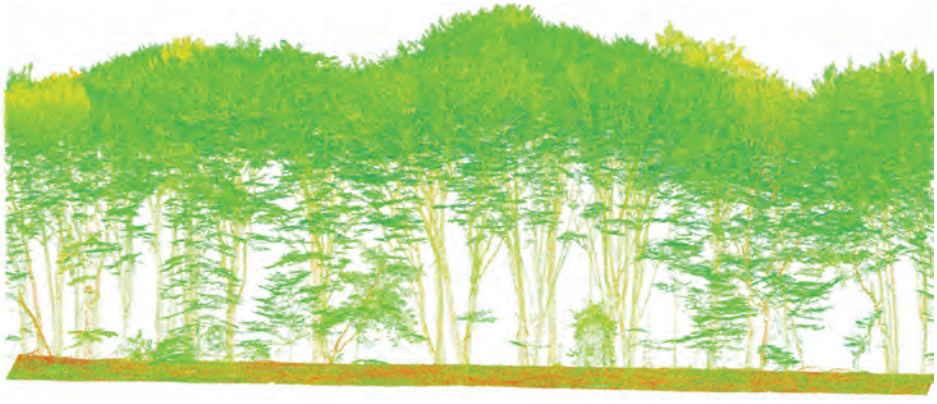
The resulting point clouds from TLS and ALS were co-registered via common tie-planes. These were the manually selected GCP panels. Then different raster products were produced at 0.5 m resolution with LAStools (https://rapidlasso.com/lastools/): scan density by counting all hits within a resolution cell, DTMs by selecting the lowest point in a resolution cell, DSMs with the highest, and Canopy Height Models (CHMs) by calculating the difference between DTMs and DSMs.

The lower stem parts of individual trees were manually extracted from the TLS and ALS point clouds from the 5 plots (Figure 2). For each tree all points at a height of 120 to 140 cm were selected to represent DBH. These subsets were manually inspected for the suitability to fit circles. In case of presence of branches at the DBH height, the corresponding points were further manually removed. Next, circles were fitted to the horizontal coordinates of these points separately for ALS and TLS. An iterative optimisation procedure was used to minimise the euclidean distance between points and circles according to Coope [22] as implemented in R's (http://www.r-project.org/) `circular` package. Next to the geometries, the points contained information about the return number and scan angle under which they were recorded. These were analysed to gain more insights which scan conditions can be beneficial to record stem points.

5. Results

The acquired TLS and ALS point clouds showed a number of differences. Figure 3 shows two sample transects through the Old Beech and Oak area (cf. Figure 2). The ALS point cloud clearly had a lower point density in the stem area, while the branch and leaf area appeared to be of comparable density. Nonetheless, single stems as well as larger branches could be made out. It should be noted that even though the images suggest the trees to have a full canopy, this was not the case during the time of acquisition. The canopy level rather has a distinctively different apparent reflectance than stem

and ground elements, because partial hits are much more likely in the crown area where branches do not completely fill the laser footprint.



(**a**) ALS

(**b**) TLS

Figure 3. Old Beech and Oak point clouds samples from same perspective, coloured according to apparent reflectance (blue = low, green = medium, red high).

Figure 4 gives an overview of point density over the whole of the study site. While ALS point density was highest in tree crowns, visible as mushroom forms in the Old Beech and Oak area, TLS point density peaked at stem locations, visible as black specks in the TLS map. Furthermore, higher density areas were created by slight horizontal course corrections of the UAV, which are visible as stripe patterns in the density map, especially in the forest opening in the Northwest. Also more points were observed along the centre line of the plot in WE direction due to the higher overlap of flight lines in that area, i.e., northern and southern flight lines contribute to the centre locations. This can be seen when comparing Beech areas close to WE centre line and Beech in upper right of Figure 4, around [160,60] and [210,110], respectively. In case of TLS fewer points were registered around scan positions, which stems from the restriction in zenithal scanning angle of the VZ-400 scanner. Overall, ALS point density was about 2 orders of magnitude lower than TLS for the given scan configurations. It was on average 5344 points/m^2, 3081 points/m^2, 3005 points/m^2, 2965 points/m^2

and 3004 points/m^2 for the Old Beech and Oak, Giant Fir, Norway Spruce, Young Beech, and Douglas Fir plots, respectively.

Figure 5 shows vertical return profiles of two focus areas representing different canopy architectures. While the Old Beech and Oak canopy had higher trees of up to 35 m with only few branches on the lower levels, the Norway Spruce canopy had trees up to 30 m and a considerable higher number of small, horizontally oriented branches below the crown level. The distribution of ALS and TLS points was similar for both canopies, but ALS hit relatively more often the upper canopy levels. This is clearly the effect of perspective of the ALS from above and TLS from below the canopy. Considering the distribution of return order in the upper canopy, the Old Beech and Oak canopy showed many more higher order returns than the Norway Spruce canopy. This could be explained by the foliage coverage of the Norway Spruce: while the water in the green needles allowed first order returns, it absorbed too much of the pulse energy to allow higher order return from within the clumped shoots. On the other hand, the not fully developed Old Beech and Oak canopy allowed partially intercepted beams and therefore multiple returns.

Despite the perspective of the ALS above the canopy, it hit the ground level in similar proportions as the TLS relative to all returns. This was clearly possible due to the multi-return capabilities of the VUX®-1UAV scanner. Returns up to 7th order could be recorded over the whole study area. This is in contrast to the TLS that is dominated by 1st order returns, which results from a higher proportion of hard targets like wood and ground elements. The ground returns were spread over some range of heights due to slightly sloped terrain and ground features like dead trees and hollows.

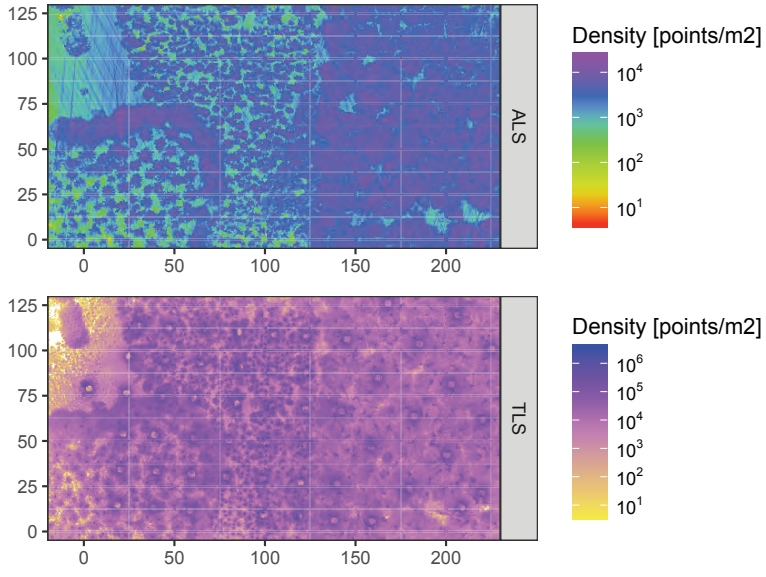
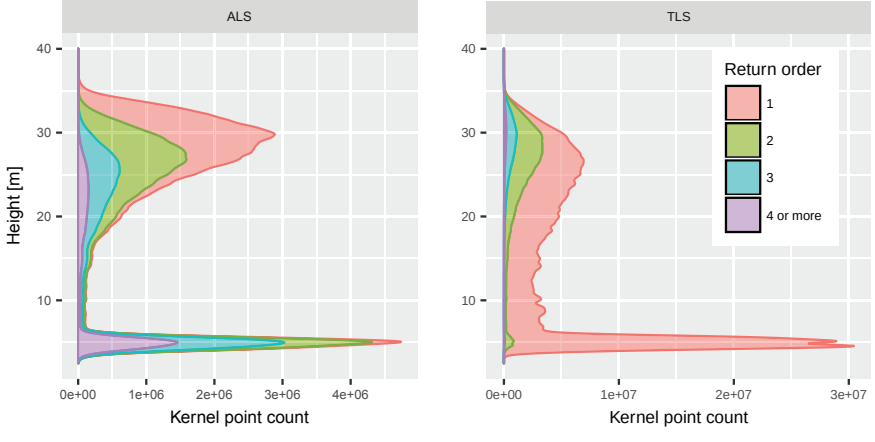


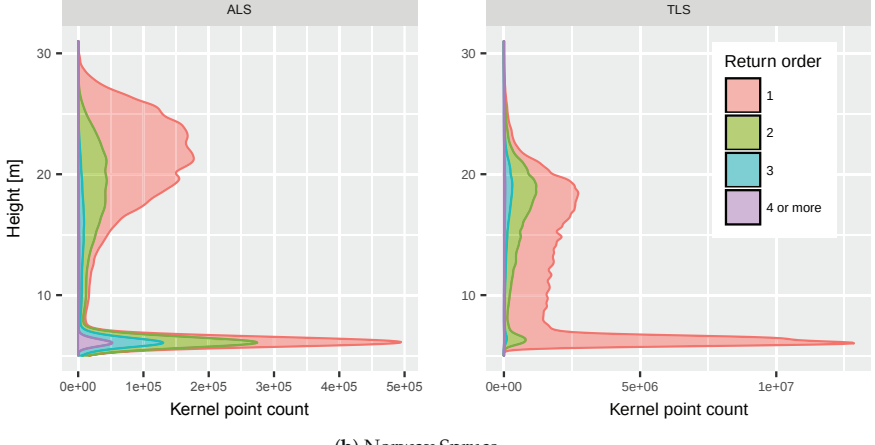
Figure 4. ALS and TLS point cloud density maps for whole study site at 0.5 m resolution in project coordinate system.

Figure 6 shows the CHM difference map. The underlying DTM, DSM and CHM maps can be found in the supplementary material. General agreement between both maps was very high with a dominance of small offsets of ±0.5 m. However, an overall pattern can be observed similar to relieve shading resulting in positive differences in eastern and negative in western directions The pattern is not regular over the whole study area. For instance, it is strong over the Douglas Fir plot in the

south-west and less pronounced over the Old Beech and Oak plot. This pattern stems from slight mis-registration of the point clouds of about 0.5 to 1 m at the crown level. One major source of error could have been crown movement by wind, a common problem in LiDAR field data acquisition [11]. This would also explain the pronounced pattern in the Douglas Fir plot with its high trees (up to 35 m) that are more susceptible to wind than the lower trees in the other plots. Finally, the TLS did not cover a small area in the Northwest that was occluded by a copse in the middle of the grass patch.



(**a**) Old Beech and Oak



(**b**) Norway Spruce

Figure 5. Vertical return density profiles smoothed with Gaussian kernel of all points in two areas of interest (see Figure 2), height reference is lowest point found in sub-area.

The mis-registration can also be found in the scatterplot of height differences in the plots in Figure 7: extreme cases can be found along the x and y axes. They represent cases when either the ALS or the TLS hit the crown and the other the ground. Outliers along the x and y axis represent mainly the western and eastern crowns sides, respectively. Nonetheless, all scatterplots confirm the high agreement of ALS and TLS. However, similar to the vertical profiles (Figure 5) also in the case of the CHMs ALS tended to detect higher points in the canopy, resulting in overall higher CHM.

For instance, the ALS CHM was 6.1 cm and 12.2 cm higher for the Giant Fir and Old Beech and Oak plots, respectively. The difference for all cells over all plots was 11.5 cm.

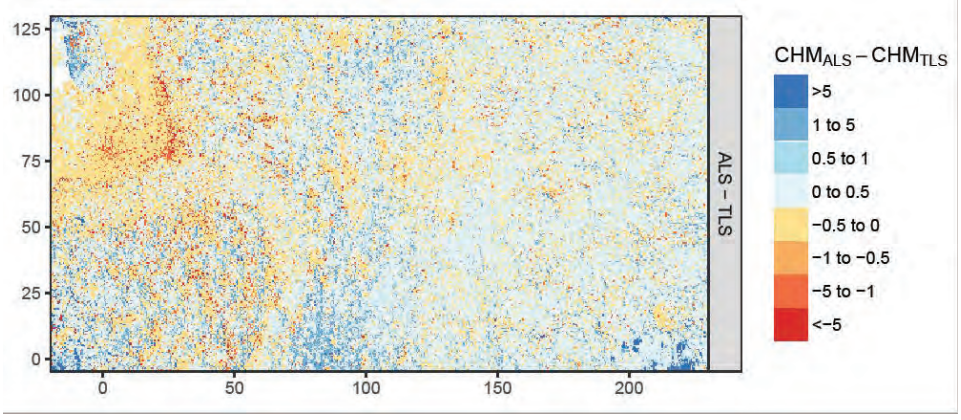


Figure 6. Differences in Canopy Height Models (CHMs). Colour height bins were chosen according to histogram and to facilitate evaluation of the sign of the difference.

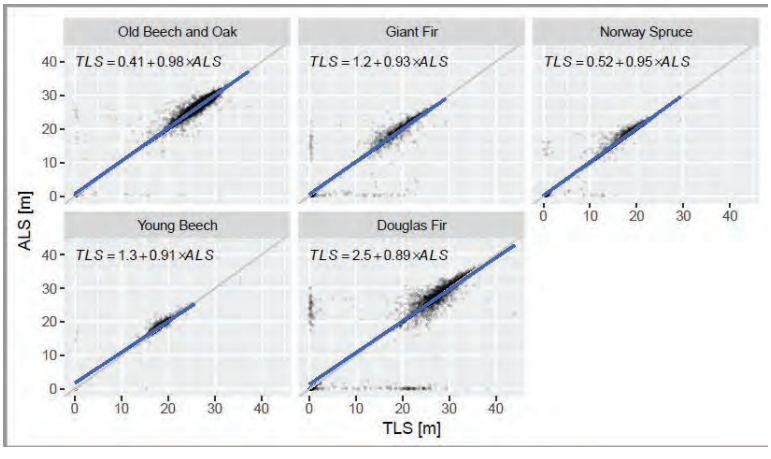


Figure 7. Differences in CHM per plot. Ordinary least squares regression lines in blue and formulas. Grey lines are 1:1 lines. Points are transparent to facilitate identification of high density clusters.

Out of the 58 extracted tree stems, 39 were found to be suitable for DBH estimation. Of the accepted 12 trees stems had low level branch points that had to be removed to make the circle fitting possible. The 19 unsuitable trunks would clearly fail to form a circle (17) or had bifurcated trunks that violated the assumption of a single, circular trunk (2). The rejected cases were mainly found in the dense Giant Fir and Norway Spruce plots, and in the Young Beech plot with small overall trunks and branches on the lower canopy levels. Each ALS and TLS point ring contained 40 and 5522 points on average, respectively. Figure 8 shows examples of 2 trunks and the resulting fitted circles. In both cases there was a large number of TLS points available, while only few ALS points covered the stem at the DBH height and in the case of Douglas Fir it was also only from one direction. However, it was still possible to approximate the stems with circles. In Figure 9 the performance of ALS fitted circles in comparison to TLS fits can be seen. Both agree well with a correlation coefficient of 0.98 and root mean

square error of 4.24 cm, while the range of estimated DBH was 19 to 93 cm. In comparison to TLS ALS estimates were 1.71 cm larger.

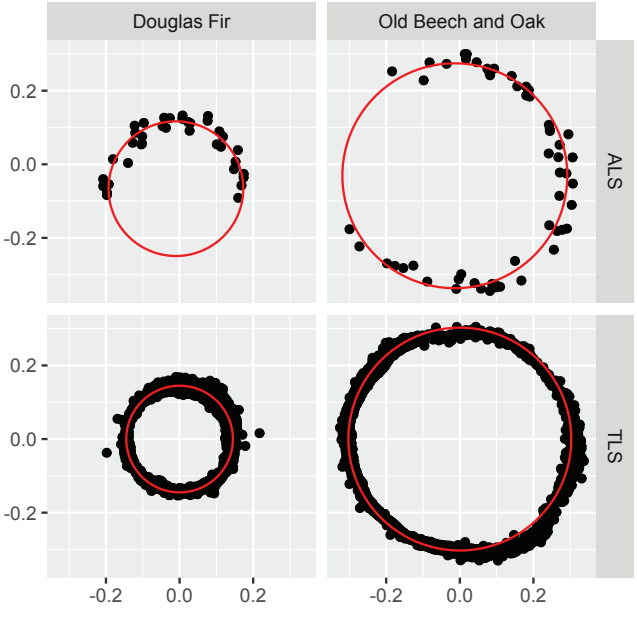


Figure 8. Samples of fitted circles to estimate DBH in the *x*-*y*-plane, *x* and *y* axis in cm.

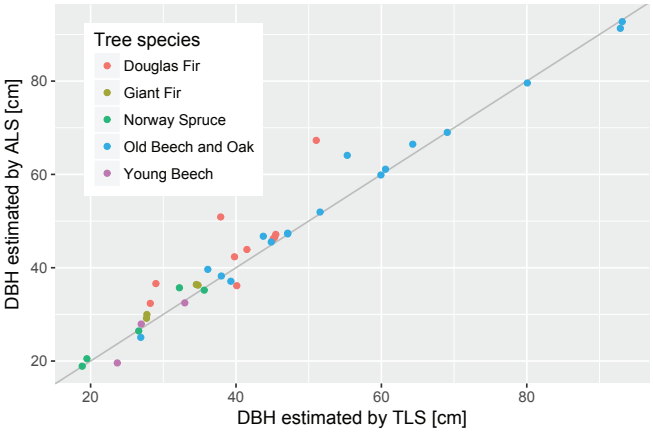


Figure 9. DBH of TLS compared to ALS. Grey line is 1:1.

Another interesting observation concerned the scan angles and return orders of the ALS points that were available for the DBH estimation. The scan or body across angle is the angle under which a laser pulse was shot from the scanner. It is 0° perpendicular to the rotor arms when mounted, i.e., a scan angle of 0° would describe nadir when the RiCOPTER flies perfectly horizontal. Large positive and negative scan angles result in low incidence angles. Figure 10 shows the distribution

of scan angles for the sampled points that were used for the DBH estimation. The distributions were strongly bimodal for the Old Beech and Oak, Norway Spruce and Douglas Fir plots with peaks between -20° to -40° and 20° to 40°. The distribution of the Young Beech points were dominated by two trees that were hit from flight lines with the same orientation, resulting in a peak around -30°. Even though the RiCOPTER does not always fly perfectly horizontal, because it has to compensate the wind direction, generally scan angles around 30° to 40° seem to be well suited for detection of trunks. Probably self-occlusion of the observed tree and occlusion by other trees is minimal at these angles. Furthermore, trunk points were constituted of 14.6% 1st, 25.8% 2nd, 29.4% 3rd, and 30.2% 4th or higher return order. This underlines that the multi-return capability of the VUX®-1UAV was beneficial to observe trunks.

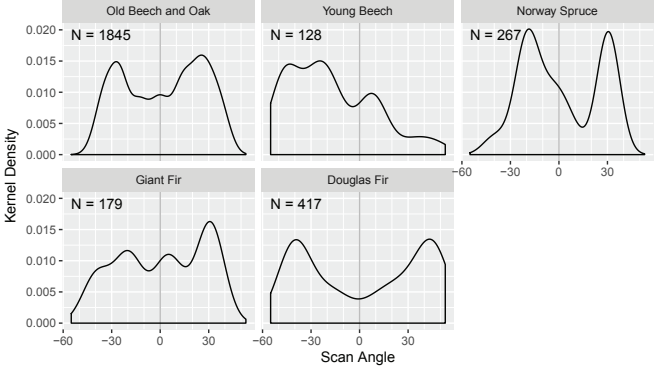


Figure 10. ALS scan angles under which the points considered for the DBH estimation were observed. N denotes the number of points for each plot.

6. Discussion

Development of LiDAR technology and algorithms in recent years has shown great potential to support forestry practices, in particular geometric characterisation of plots up to unbiased estimation of AGB [4]. In this context TLS yields a data source of unprecedented detail and accuracy. However, TLS acquisition can be labour intense and time consuming especially in challenging environments like tropical forests [11]. New UAV-borne LiDAR technology can possibly accelerate these field campaigns and provide a larger coverage.

In this context, the RIEGL RiCOPTER with VUX®-1UAV has proven useful to characterise the Speulderbos forest plot. CHMs were successfully derived and showed good agreement with TLS. Canopy height estimated by the ALS were generally higher. This could be expected by its viewing perspective from above the canopy and the known shortcoming of TLS to not always detect the top of canopy (e.g., [23]). However, the difference of on average 11.5 cm falls within the precision of traditional field measurements for tree height of 50 cm [24]. Concerning the estimation of individual tree height, Wallace el al. [17] found good agreement with field measurements of 0.35 m (mean absolute error) by using point clouds of up to 300 points/m^2 density.

Multi-return and side-looking capabilities proved to be important features of the VUX®-1UAV to scan trunks and estimate DBH for a number of trees under different canopy conditions (Figure 9). While other UAV LiDAR systems are also able to record multiple targets per pulse, not many systems are able to acquire data under larger scan angles (>30° off nadir). Nonetheless, a sufficient number of points could not be collected for all sampled trunks, mainly in the dense, narrow spaced needle-leaf plots. Repeated flights over the same plots with varying flight paths could result in better coverage. The average root mean square error of 4.24 cm between TLS and ALS is comparable to reported deviations

of TLS and traditional methods of 0.7 to 7.0 cm (RMSE) [1]. However, the DBH estimation error is still much larger than the precision of traditional methods of ~0.3 cm [24].

The scan angles that proved optimal to scan the trunk samples (Figure 10) have implications for the flight preparation. The targeted plots should be always well covered, possibly with flight lines that overshoot the plot area. For instance if the flight height is at 90 m and the optimal angle is assumed to be 30°, the flight trajectory should overshoot by ~52 m. However, it is difficult to say how general the optimal scan angles found in this study are. In any way, we found that multiple flight lines, made possible through the long air-borne time, were contributing to a better sampling from different directions. In this respect more lines at faster speed, should be preferred to fewer at lower speed assuming same airborne time. Maximising line crossings and multiple flights should be considered as well. The later will be primarily restricted by the number of battery packs available.

Initial point cloud production of VUX®-1UAV data solely relies on the on-board IMU and GNSS post processing. This is possible because of the IMU's accuracy, which on the other hand results in a large weight for the scanner. This drives the weight of the whole system, since heavier sensors also require larger payload capacity and thus larger UAVs. Together with the system requirement of long endurance this brought the total system weight up to just under the legal limit for mini-UAV (personal communication RIEGL, 2016). This design decision makes it unnecessary to employ algorithms that reconstruct the trajectory in the post-processing as shown in previous studies [16,18]. However, it makes the availability of GNSS base station data a hard requirement that might be difficult to fulfil in remote areas. Also the system weight can be a logistical challenge in such cases.

Even though this study did not aim to conduct full plot inventories, the data shows promising attributes to extend the analysis in that direction. One important step for this would be to detect single trees in all plots. Wallace et al. [25] produced detection rates of up to 98% with point clouds of 50 points/m^2 density. Therefore, detection should be achievable with the ~3000 points/m^2 RiCOPTER point clouds. Based on the detected trees, single tree height can be estimated. However, traditional forest inventory data would be necessary for validation.

Apart from characterising traditional forest metrics, UAV-borne LiDAR could also be utilised as a flexible, higher resolution alternative to manned airborne LiDAR, especially to study foliage. In that case several published algorithms could be employed [26–29] and tested if they are applicable on higher density point clouds. Moreover, reliable and mobile systems like the RiCOPTER are suitable for multi-temporal studies [15].

7. Conclusions

This study presented first results and experiences with the RIEGL RiCOPTER with VUX®-1UAV ALS with its components and processing work flow, and its performance in estimating CHM and DBH compared to TLS. As first steps we compared the RiCOPTER with the well tested RIEGL VZ-400 TLS by deriving CHM and estimating DBH. CHMs showed only small differences that could be explained by the perspective of the RiCOPTER above the canopy, resulting in different vertical detection profiles that facilitate the discovery of highest points in the canopy, which is not always possible with TLS. Additionally, the multi-return and side-looking capabilities of the VUX®-1UAV scanner proved beneficial to detect trunk elements. This feature will be valuable when more sophisticated 3D modelling is to be applied. However, not all sampled tree trunks were sufficiently covered with points, so that more flights or different flight patterns are necessary to achieve better coverage. Overall, the RiCOPTER could produce comparable results to the VZ-400. Further experiments should be directed to test explicit quantitative structural modelling to derive AGB from the RiCOPTER point clouds as well as co-registration strategies of multiple flights and with TLS systems.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/17/10/2371/s1, Figure S1: Digital Elevation Models at 0.5 m resolution in project coordinate system, blank cells did not contain points; Figure S2: Digital Surface Models at 0.5 m resolution in project coordinate system, blank cells did not

contain points; Figure S3: Canopy Height Models at 0.5 m resolution in project coordinate system, blank cells did not contain points.

Acknowledgments: This work was carried out as part of the IDEAS+ contract funded by ESA-ESRIN. A.L. is supported by CIFOR's Global Comparative Study on REDD+, with financial support from the International Climate Initiative (IKI) of the German Federal Ministry for the Environment and the donors to the CGIAR Fund. The access to the RiCOPTER has been made possible by Shared Research Facilities of Wageningen University & Research. The authors thank the Dutch Forestry Service (Staatsbosbeheer) for granting access to the site and 06-GPS for supplying the GNSS base station data. Further thanks go to Marcello Novani for help during the fieldwork and Jan den Ouden to identify the tree species. We thank two anonymous reviewers for their helpful comments to improve the quality of this manuscript.

Author Contributions: B.B. and H.M.B. conceived and designed the experiment; B.B., A.L. and H.M.B. conducted the field experiment; B.B. and A.L. analysed the data; H.M.B. and L.K. scientifically supported and reviewed the paper; L.K. arranged licensing of the UAV; B.B. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liang, X.; Kankare, V.; Hyyppä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; Yu, X.; Kaartinen, H.; Jaakkola, A.; Guan, F.; et al. Terrestrial laser scanning in forest inventories. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 63–77.
2. Newnham, G.J.; Armston, J.D.; Calders, K.; Disney, M.I.; Lovell, J.L.; Schaaf, C.B.; Strahler, A.H.; Danson, F.M. Terrestrial Laser Scanning for Plot-Scale Forest Measurement. *Curr. For. Rep.* **2015**, *1*, 239–251.
3. Gonzalez de Tanago, J.; Lau, A.; Bartholomeus, H.; Herold, M.; Avitabile, V.; Raumonen, P.; Martius, C.; Goodman, R.; Manuri, S.; Disney, M.; et al. Estimation of above-ground biomass of large tropical trees with Terrestrial LiDAR. *Methods Ecol. Evol.* **2017**, doi:10.1111/2041-210X.12904.
4. Calders, K.; Newnham, G.; Burt, A.; Murphy, S.; Raumonen, P.; Herold, M.; Culvenor, D.; Avitabile, V.; Disney, M.; Armston, J.; et al. Nondestructive estimates of above-ground biomass using terrestrial laser scanning. *Methods Ecol. Evol.* **2015**, *6*, 198–208.
5. Le Toan, T.; Quegan, S.; Davidson, M.W.J.; Balzter, H.; Paillou, P.; Papathanassiou, K.; Plummer, S.; Rocca, F.; Saatchi, S.; Shugart, H.; et al. The BIOMASS mission: Mapping global forest biomass to better understand the terrestrial carbon cycle. *Remote Sens. Environ.* **2011**, *115*, 2850–2860.
6. Eitel, J.U.; Höfle, B.; Vierling, L.A.; Abellán, A.; Asner, G.P.; Deems, J.S.; Glennie, C.L.; Joerg, P.C.; LeWinter, A.L.; Magney, T.S.; et al. Beyond 3-D: The new spectrum of LiDAR applications for earth and ecological sciences. *Remote Sens. Environ.* **2016**, *186*, 372–392.
7. Wallis, C.I.; Paulsch, D.; Zeilinger, J.; Silva, B.; Curatola Fernández, G.F.; Brandl, R.; Farwig, N.; Bendix, J. Contrasting performance of LiDAR and optical texture models in predicting avian diversity in a tropical mountain forest. *Remote Sens. Environ.* **2016**, *174*, 223–232.
8. Morsdorf, F.; Nichol, C.; Malthus, T.; Woodhouse, I.H. Assessing forest structural and physiological information content of multi-spectral LiDAR waveforms by radiative transfer modelling. *Remote Sens. Environ.* **2009**, *113*, 2152–2163.
9. Raumonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; Lewis, P. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. *Remote Sens.* **2013**, *5*, 491–520.
10. Hackenberg, J.; Morhart, C.; Sheppard, J.; Spiecker, H.; Disney, M. Highly accurate tree models derived from terrestrial laser scan data: A method description. *Forests* **2014**, *5*, 1069–1105.
11. Wilkes, P.; Lau, A.; Disney, M.I.; Calders, K.; Burt, A.; Gonzalez de Tanago, J.; Bartholomeus, H.; Brede, B.; Herold, M. Data Acquisition Considerations for Terrestrial Laser Scanning of Forest Plots. *Remote Sens. Environ.* **2017**, *196*, 140–153.
12. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97.
13. Torresan, C.; Berton, A.; Carotenuto, F.; Di Gennaro, S.F.; Gioli, B.; Matese, A.; Miglietta, F.; Vagnoli, C.; Zaldei, A.; Wallace, L. Forestry applications of UAVs in Europe: A review. *Int. J. Remote Sens.* **2016**, *38*, 2427–2447.

14. Suomalainen, J.; Anders, N.; Iqbal, S.; Roerink, G.; Franke, J.; Wenting, P.; Hünniger, D.; Bartholomeus, H.; Becker, R.; Kooistra, L. A Lightweight Hyperspectral Mapping System and Photogrammetric Processing Chain for Unmanned Aerial Vehicles. *Remote Sens.* **2014**, *6*, 11013–11030.
15. Jaakkola, A.; Hyyppä, J.; Kukko, A.; Yu, X.; Kaartinen, H.; Lehtomäki, M.; Lin, Y. A low-cost multi-sensoral mobile mapping system and its feasibility for tree measurements. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 514–522.
16. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR system with application to forest inventory. *Remote Sens.* **2012**, *4*, 1519–1543.
17. Wallace, L.; Musk, R.; Lucieer, A. An assessment of the repeatability of automatic forest inventory metrics derived from UAV-borne laser scanning data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7160–7169.
18. Chisholm, R.A.; Cui, J.; Lum, S.K.Y.; Chen, B.M. UAV LiDAR for below-canopy forest surveys. *J. Unmanned Veh. Syst.* **2013**, *01*, 61–68.
19. Wei, L.; Yang, B.; Jiang, J.; Cao, G.; Wu, M. Vegetation filtering algorithm for UAV-borne LiDAR point clouds: A case study in the middle-lower Yangtze River riparian zone. *Int. J. Remote Sens.* **2017**, *38*, 1–12.
20. RIEGL LMS RiProcess for RIEGL Scan Data. Available online: http://www.riegl.com/uploads/tx_pxpriegldownloads/11_Datasheet_RiProcess_2016-09-16_03.pdf (accessed on 13 September 2017).
21. Brede, B.; Bartholomeus, H.; Suomalainen, J.; Clevers, J.; Verbesselt, J.; Herold, M.; Culvenor, D.; Gascon, F. The Speulderbos Fiducial Reference Site for Continuous Monitoring of Forest Biophysical Variables. In Proceedings of the Living Planet Symposium 2016, Prague, Czech Republic, 9–13 May 2016; p. 5.
22. Coope, I.D. Circle fitting by linear and nonlinear least squares. *J. Optim. Theory Appl.* **1993**, *76*, 381–388.
23. Hilker, T.; van Leeuwen, M.; Coops, N.C.; Wulder, M.A.; Newnham, G.J.; Jupp, D.L.B.; Culvenor, D.S. Comparing canopy metrics derived from terrestrial and airborne laser scanning in a Douglas-fir dominated forest stand. *Trees* **2010**, *24*, 819–832.
24. Luoma, V.; Saarinen, N.; Wulder, M.A.; White, J.C.; Vastaranta, M.; Holopainen, M.; Hyyppä, J. Assessing precision in conventional field measurements of individual tree attributes. *Forests* **2017**, *8*, 38.
25. Wallace, L.; Lucieer, A.; Watson, C.S. Evaluating tree detection and segmentation routines on very high resolution UAV LiDAR data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7619–7628.
26. Tang, H.; Brolly, M.; Zhao, F.; Strahler, A.H.; Schaaf, C.L.; Ganguly, S.; Zhang, G.; Dubayah, R. Deriving and validating Leaf Area Index (LAI) at multiple spatial scales through LiDAR remote sensing: A case study in Sierra National Forest, CA. *Remote Sens. Environ.* **2014**, *143*, 131–141.
27. García, M.; Gajardo, J.; Riaño, D.; Zhao, K.; Martín, P.; Ustin, S. Canopy clumping appraisal using terrestrial and airborne laser scanning. *Remote Sens. Environ.* **2015**, *161*, 78–88.
28. Morsdorf, F.; Kötz, B.; Meier, E.; Itten, K.I.; Allgöwer, B. Estimation of LAI and fractional cover from small footprint airborne laser scanning data based on gap fraction. *Remote Sens. Environ.* **2006**, *104*, 50–61.
29. Detto, M.; Asner, G.P.; Muller-Landau, H.C.; Sonnentag, O. Spatial variability in tropical forest leaf area density from Multireturn LiDAR and modelling. *J. Geophys. Res. Biogeosci.* **2015**, *120*, 1–16.

sensors

MDPI

Article

Designing and Testing a UAV Mapping System for Agricultural Field Surveying

Martin Peter Christiansen [1,*], Morten Stigaard Laursen [1], Rasmus Nyholm Jørgensen [1], Søren Skovsen [1] and René Gislum [2]

[1] Department of Engineering, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark; msl@eng.au.dk (M.S.L.); rnj@eng.au.dk (R.N.J.); ssk@eng.au.dk (S.S.)

[2] Department of Agroecology, Aarhus University, Forsøgsvej 1, 4200 Slagelse, Denmark; rg@agro.au.dk

* Correspondence: mpc@eng.au.dk; Tel.: +45-40420617

Received: 30 September 2017; Accepted: 13 November 2017; Published: 23 November 2017

Abstract: A Light Detection and Ranging (LiDAR) sensor mounted on an Unmanned Aerial Vehicle (UAV) can map the overflown environment in point clouds. Mapped canopy heights allow for the estimation of crop biomass in agriculture. The work presented in this paper contributes to sensory UAV setup design for mapping and textual analysis of agricultural fields. LiDAR data are combined with data from Global Navigation Satellite System (GNSS) and Inertial Measurement Unit (IMU) sensors to conduct environment mapping for point clouds. The proposed method facilitates LiDAR recordings in an experimental winter wheat field. Crop height estimates ranging from 0.35–0.58 m are correlated to the applied nitrogen treatments of 0–300 $\mathrm{kg}\frac{\mathrm{N}}{\mathrm{ha}}$. The LiDAR point clouds are recorded, mapped, and analysed using the functionalities of the Robot Operating System (ROS) and the Point Cloud Library (PCL). Crop volume estimation is based on a voxel grid with a spatial resolution of $0.04 \times 0.04 \times 0.001$ m. Two different flight patterns are evaluated at an altitude of 6 m to determine the impacts of the mapped LiDAR measurements on crop volume estimations.

Keywords: aerial robotics; canopy estimation; crop monitoring; point cloud; winter wheat mapping

1. Introduction

Aerial mapping of agricultural and forestry land provides a means to estimate current production and environmental states, and monitor progress over time. Information on production and environmental states can be used in site-specific farming to tailor specific crop and soil treatments for each field [1,2]. However, the spatial resolution of satellite and aircraft sensory data is still low compared to that of an unmanned aerial vehicle (UAV), which fly at a much lower altitude. Low spatial resolution sensory data may underestimate productivity and environmental factors, and result in insufficient treatment coverage [3].

A UAV can inspect at a much closer range, and may provide spatial sensory information at a much higher resolution. UAVs have been used in agriculture to provide high-spatial resolution images to detect individual crops and weeds at the submillimeter scale [4]. In [5], image data from a DJI Phantom 2 UAV [6] were used to evaluate the results of seeding an experimental field by determining unseeded rows and bare soil. Plant height has also been estimated in experimental fields [7], using crop surface models derived from UAV-recorded red, green and blue (RGB) color images, and related to the extracted biomass. The relationship between barley plant height and extracted biomass was determined in [8], and developed into a prediction model for future use.

Other remote-sensing systems have been used in a broader context of UAV applications, such as Light Detection and Ranging (LiDAR) [9–13], for mapping forests and other outdoor environments. These systems are capable of transporting multi-spatial sensors (LiDAR, RGB,

thermal, and hyperspectral imaging) and recording the environment at altitudes between 40–120 m. In agriculture, LiDAR and hyperspectral sensory data have been combined to monitor vegetation [14]. The UAV with LiDAR and hyperspectral sensors was flying at an altitude of 70 m and a point density of 50 points/m^2. A lower flight altitude provides an increase in spatial resolution of the surroundings. An obstacle at lower altitudes is the draft from UAV propellers, which can affect the crop monitored.

The challenge is to select a UAV platform that matches the demands of the task. Smaller platforms can only carry a single spatial sensor at a time owing to load capabilities [15]. An approach could be to use single beam LiDAR units with a lower total weight to include spatial sensors such as RGB cameras. In [16,17], the authors use a single beam 2D LiDAR to measure corn height and soil surface distance. They are flying at an altitude of approximately 4 m with spacing of 0.75 between the crop rows. This spacing between the crop rows allows the LiDAR to observe the plants and the ground simultaneously when flying over a field.

LiDAR sensors have mainly been used in agriculture in ground-based systems for mapping soil [18] and crops [19,20]. LiDAR mapping data can then be used to evaluate the impact of agricultural production methods [21]. Orchards also tend to be an area of research where LiDAR have been used for mapping and monitoring [22–24].

Variable growth conditions within and between fields require different nutrient applications due to variations in soil type, soil phosphorus, and nutrient ranges of different crops. Nitrogen (N) is a major nutrient source, and must be applied every year to non-legume plants [25,26]; however, plant lodging and/or subsequent leaching of N have negative environmental impacts [27,28]. N requirements are based on soil and crop type, and understanding how to adjust N application according to crop need, thereby, minimizing pests and leaching, is a key topic for agricultural intensification.

Accurate N application according to crop needs depends on our ability to estimate crop biomass and crop N status (percentage N in crop biomass) to calculate crop N content (e.g., $\frac{gN}{biomass}$) and final yield [29]. The amount of biomass can be estimated using the volume of the crop canopy [7,30]. Current methods to estimate volume, and, thereby, the biomass, include UAV-based RGB imaging. Crop N concentration has been measured from multi spectral imaging recorded from a UAV [31].

Figure 1. unmanned aerial vehicle (UAV) and experimental field on the day of data recording. A Matrice 100 UAV platform from DJI [32] was used.

In this study, we instead use LiDAR data obtained by a UAV to map and estimate the height and volume of crops in an experimental field, as illustrated in Figure 1. Our multi sensor setup was specifically designed for mapping agriculture field area's by enabling simultaneous recording of LiDAR and RGB spatial data at low altitudes. The system design concerning Computer-Aided Design (CAD) models, schematics, source code and recorded data, can be found on the project homepage [33]. By making the design publicly available, the intention is that other researchers and developers can adapt and reuse our solution for similar projects.

We test different flight methods to determine the impact on spatial resolution and volume estimates. Compared to other studies that fly low altitudes, we map crop with a much higher density in terms of plant seeding. The higher crop density makes it hard to consistently observe the ground between the plants, so a novel approach have been utilised to determine crop height.

The purpose of the experiment is to create 3D LiDAR point clouds of the field to determine canopy volume and discriminate between different crop treatments using textural analysis. We evaluate the accuracy of mapped point clouds at estimating the structure of crop parcels, which is a significant factor that can be used as an alternative method for determining crop biomass. We expect that crop height, and, therefore, biomass accumulation can be observed using LiDAR point clouds.

2. Materials

2.1. Crop Area for Recording Data

A field experiment with winter wheat (*Triticum aestivum* L.) cv. Benchmark was established at Aarhus University, Flakkebjerg (WGS84 area: Lat 55.327297°, Lon 11.388465°, Lat 55.327297°, Lon 11.388344°, Lat 55.329608°, Lon 11.388480°, Lat 55.329608°, Lon 11.388359°). The field was seeded in autumn 2016 (gross parcel). The crops were sown in parallel rows facing perpendicular to the north as illustrated in Figure 2. The seed rate was 200 $\frac{\text{kg}}{\text{ha}}$, which equals approximately 400 plants m^2 and the seeds were sown in 0.12 m rows at a depth of 0.02–0.03 m. The planned single plot size (crop parcel) was 2.28 × 8 m, after 1 m was removed from each end.

Figure 2. Sketch of the gross and crop parcel structure and its alignment. Each crop parcel was seeded with 19 rows of winter wheat. The tire tracks separate the crop parcels and allow vehicles to access the winter wheat to provide treatment.

The experimental design, documented in Appendix A, involved four replicates and 21 N application strategies randomised within each block. Nitrogen was applied in the spring growing season as calcium ammonium nitrate, at different rates and times. Weed growth was regulated using Moddus. Nineteen buffer parcels, also containing winter wheat, were also added to the experimental field to protect against wind and divide parcels with different fertiliser conditions.

2.2. UAV and Sensor Setup

As a basis for the platform, a DJI Matrice 100 unmanned aerial vehicle (UAV) with firmware v. 1.3.1.10 and a TB48D battery pack was chosen. A sensor mount printed in 3D nylon was designed to fit as the payload, illustrated in Figure 3. The sensor mount was designed to utilise the precision dowel pin holes in the LiDAR and inertial measurement unit (IMU) sensor by including pins within the 3D print itself, dimensioned explicitly for a tight fit, as documented in Appendix B.

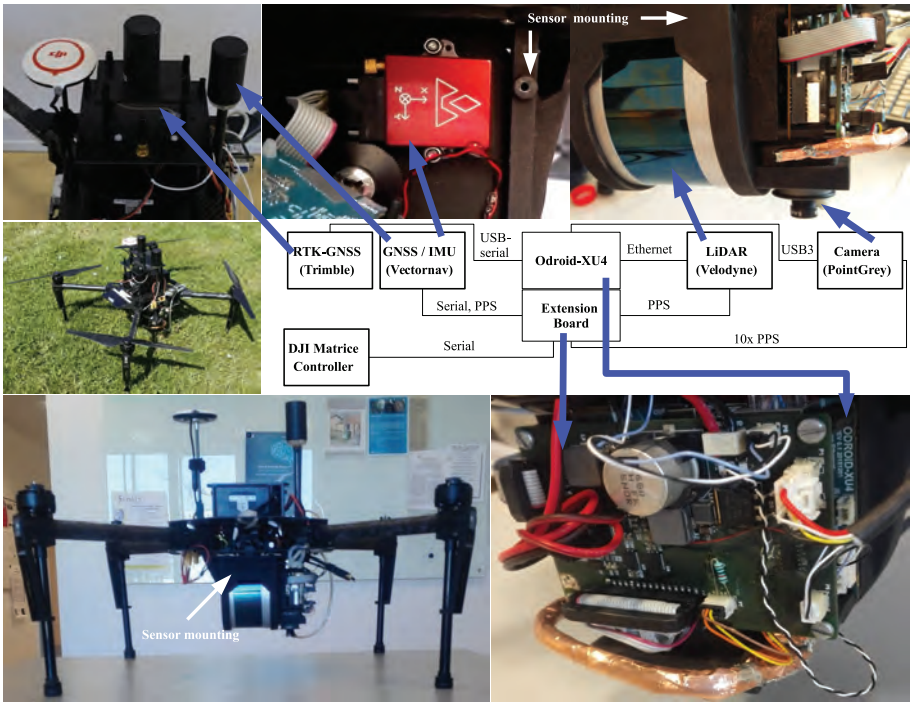


Figure 3. Sensor mounting added to the DJI Matrice 100 platform [32]. The block diagram shows how the sensors are connected to the Odroid XU4 (Hardkernel co., Ltd., GyeongGi, South Korea). The camera and LiDAR are facing downwards to observe the crops and soil.

For sensor datalogging, we used an Odroid XU4 with an in-house built extension board controlling input/output (IO) and power. The sensor system consisted of a Velodyne VLP-16 LiDAR (San Jose, CA, USA) connected via ethernet, a Point Grey Chameleon3 3.2 MP Color camera (Richmond, BC, Canada) with a Sony imx265 sensor (Tokyo, Japan) connected via USB3, a Vectornav VN-200 IMU (Dallas, TX, USA) with a MAXTENA M1227HCT-A2-SMA antenna (Rockville, MD, USA) connected via RS-232, a Trimble BD920 GNSS with real time kinematic (RTK) (Sunnyvale, CA, USA) module connected via a USB-serial (RS-232) and a RS-232 serial connection to the DJI UAV. Both camera and LiDAR faced downwards because the focus of this experiment was ground observations.

An updated schematic of the IO extension board can be found on the project homepage. The IO extension board handles logic level-shifting between sensors and the Odroid XU4. An ATmega32U4 (Microchip Technology, Chandler, AZ, USA) is also placed on the IO extension board and is programmed from the Odroid via serial peripheral interface-bus (SPI). The pulse-per-second (PPS) signal from the VN-200 was used for sampling synchronisation. The PPS signal was routed to the Velodyne VLP16 as an external time source. The Point Grey camera was triggered using a 10 Hz signal (10× PPS), phase-locked to the PPS using a hardware timer in the ATmega32U4. When a 10× PPS trigger pulse is sent to the Point Grey camera, an image capture exposure-response is transmitted back and logged by the VN-200 with the GPS time.

During post-processing, we matched nmea-0183 data from the Trimble RTK with nmea-0183 timestamps from the VN-200. This PPS based setup ensured a negligible time-tagging discrepancy in our system. In comparison to [16], all sensor timestamps are based on GPS time, instead of determining the retrieval time using the operating systems internal clock. The idea is to have sensory data from the UAV, which is in synchronization with the same time frame, to directly match the GPS, IMU and LiDAR.

2.3. Recording Software Setup

To record sensor data, we used the Robot Operating System (ROS) [34,35] running Ubuntu 14.04 (14.04.3, Canonical Ltd, London, UK) armhf and ROS release indigo. ROS nodes for sensors and the DJI UAV were implemented to run on a Odroid XU-4. In some instances, the ROS nodes were modified to suit our study: the VectorNav VN-200 driver was changed according to the ROS Enhancement Proposal (REP)-103 specification [36] and updated to log the trigger signal from the camera; the Velodyne ROS node was updated to handle both single and dual return measurements; the Point Grey ROS node was updated to capture images when the 10× PPS signal was received; and the built-in node rosbag in ROS was used to record data and timestamps for all active ROS nodes. Table 1 shows the data sampling rate and the relevant output data and software configuration setup for this experiment.

Table 1. ROS node configurations in relation to the sensors.

Sensor Output	Sampling Rate	Notes
DJI ROS sdk	50 Hz	(DJI OS time, attitude Quaternion), Baud = 230400
VectorNav IMU (1)	50 Hz	(Gyro, Acceleration, Quaternion, TimeGps), Baud = 115200
VectorNav IMU (2)	20 Hz	(INS, TimeUTC, TimeGps, TimeSyncIn), Baud = 115200
VectorNav IMU (3)	4 Hz	(GPS, TimeUTC, TimeGps, Fix, sats), Baud = 115200
Velodyne LiDAR	10 Hz	RPM = 600, strongest return
Point Grey Camera	10 Hz	Resolution = 2048 × 1536, 8 bits per pixel
Trimble GNSS (1)	10 Hz	GPGGA, Baud-rate = 115200, usb-serial
Trimble GNSS (2)	20 Hz	GPRMC, Baud-rate = 115200, usb-serial

For the chosen configuration the LiDAR has vertical resolution of 2°, a horizontal/azimuth resolution of approximiatly 0.2° at 10 Hz and a typical range accuracy of ±0.03 m [37]. The IMU has a static pitch/roll accuracy 0.5° root mean square (RMS), a dynamic pitch/roll accuracy of 0.1° RMS and a heading accuracy of 0.3° RMS according to the datasheet [38].

2.4. UAV Steering Using Litchi

To operate the Matrice 100 UAV [32], we used the UAV app Litchi [39]. Litchi is commercially released third party software for DJI UAVs, which can create pre-planned waypoint missions. The Litchi software enables pre-planning of UAV motion and speed when flying over individual crop parcels.

3. Methods

3.1. Data Recording

The experiment was performed on the 23 May 2017 in southwest Zealand, Denmark. The weather was sunny with wind speeds ranging from 1–2 $\frac{\mathrm{m}}{\mathrm{s}}$. To provide reference location data, the corner point of each crop-parcel was logged using a Topcon GRS-1 GNSS-RTK receiver (Tokyo, Japan). The logged corner points were based on manual judgment of the beginning and end of the winter wheat parcels. An approximate gross parcel boundary was determined based on the logged crop-parcel corner points, and its position was used to extract individual parcel areas from the LiDAR data.

3.2. Flight and Recording Procedure

UAV movement over the parcels was defined using two flight plans created in Litchi, as illustrated in Figure 4. Paths A and B were created to determine the impact of the LiDAR scanning resolution on mapping crop parcels and their border regions. The main difference between the plans is that the UAV moved along the borders of the crop parcels on Path A and along the crop rows on Path B. The increasing number of directional turns increases power consumption and reduces flight time on a battery. To account for lower flight time, Path B only covers the first third of the experimental field.

Figure 4. UAV flight Paths A and B over the experimental field. In Path A the UAV is set to move along the borders of the crop parcels. In Path B the UAV follows the crop rows alignment.

For both flight paths, the UAV flew at an altitude of 30 m to obtain an overview of the trial area and ensure stable IMU orientation tracking. Then, the UAV was lowered to an altitude of 6 m to observe the crop parcels. This altitude was chosen based on test flights to ensure compromise between high LiDAR point density in each scan and minimum downdraft on crops from the UAV's propellers.

Upon activating the flight plan, data recording was initiated using ROS. Data recording was stopped after the flight path was completed and the UAV had returned to the starting point. A break of 22 min between flight paths ensured optimal conditions for comparison, in which the battery was changed and the sensor was checked for anomalies.

3.3. Pose Estimation Post Processing

To use the LiDAR data for mapping crop parcels, the position and orientation (pose) of the UAV had to be estimated for each scan. We directly estimated the pose using GNSS, IMU, and UAV data.

Merging GNSS, IMU and DJI Data

Coordinate systems, in which the pose of a particular element is defined as a reference frame relative to its 3D position and orientation (quaternion). Each sensor on the UAV is represented by a reference frame relative to its position on the UAV (base frame), as documented in Appendix B. A base frame on the UAV was chosen relative to the surrounding environment (global frame), making it possible to map spatial data within a common frame. The base frame location of the UAV was chosen in the centre of the RTK-GNSS antenna, as it allows direct mapping between the base-frame and global-frame, based on the current GNSS position. Position data from the RTK-GNSS constitutes the latitude, longitude, and altitude (WGS84 format). We converted the GNSS data into universal transverse mercator (UTM) coordinates to represent the pose in Cartesian space.

To determine UAV orientation relative to its surroundings, we used data from the VectorNav and internal UAV IMU. Previous experience with the UAV platform has shown that the best combination is using VectorNav to determine the current roll and pitch angles, and Matrice 100 to determine the current yaw angle (UAV heading). Since all sensory data are stamped using the GNSS time, we can combine position and orientation directly.

The heading value provided by the UAV was estimated with regards to true north. As the UTM coordinate system was not aligned with true north, we applied a conversion, as defined in Equation (1):

$$\psi_{utm,LL}(lat_{gnss}, lon_{gnss}) = \arctan(\tan(lon_{gnss} - lon_{med}), \sin(lat_{gnss})), \quad (1)$$

where lat_{gnss}, lon_{gnss} are the latitude and longitude values measured in radians by the GNSS and and lon_{med} is the median longitudinal value of the UTM zone (in this case, $9^\circ = \frac{\pi}{20}$). Compensating with the calculated angle from Equation (1), we can determine the UAV heading in the UTM frame.

3.4. LiDAR Point-Cloud Mapping and Processing

After mapping the LiDAR scan into the same global reference frame (UTM32U), we extract and process information on individual parcels, as illustrated in Figure 5. To process the point-cloud data stored in ROS, we utilised the Point Cloud Library (PCL).

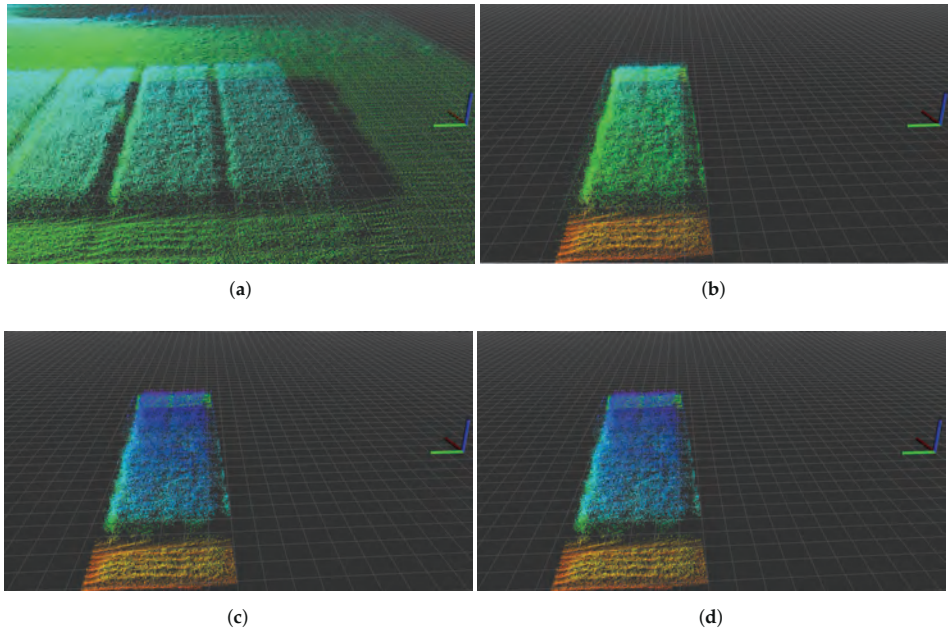


Figure 5. Processing steps for individual gross parcels. The ROS rviz 3D visualisation tool grid plane in the background is divided into 0.5 × 0.5 m grids. (**a**) raw mapped point-cloud; (**b**) point-in-polygon extraction; (**c**) statistical outlier removal; (**d**) voxelisation.

Using PCL, all LiDAR scans were stored in the same point cloud based on their estimated relative position. Relative UTM pose coordinates were used for PCL processing because single-precision floating-points are used in many PCL data containers.

3.4.1. Area Extraction Using Point-In-Polygon

The complete mapped point cloud is too large for direct processing; therefore, the estimated gross parcel areas is used to extract subparts.

The point-in-polygon approach is used to determine if each point in the mapped point cloud is inside the logged GNSS area. Point-in-polygon evaluation ensures that specific parts of the mapped point cloud are extracted for analysis and further processing, as illustrated in Figure 5b.

3.4.2. Statistical Outlier Removal

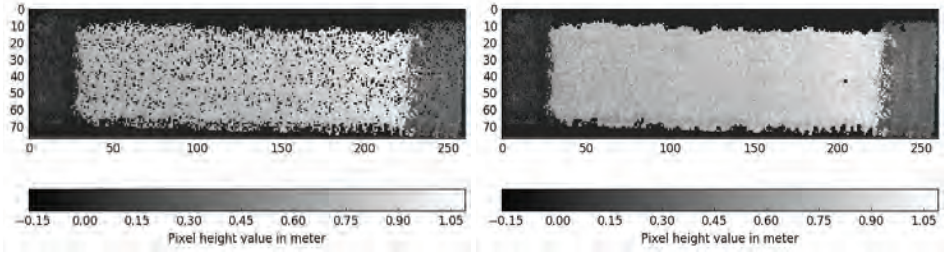
As illustrated in Figure 5a,b, points can be observed in the point cloud that do not seem to match the actual environment. These misaligned points are due to estimation of orientation errors or rapid rotations during UAV flight. To filter these outliers from the data, the statistical outlier removal functionality in PCL was used [40]. The statistical outlier removal functionality calculates the average distance of each point from its nearest k neighbours, and thresholds the value against the mean and standard deviation for the average distances set. Points with average values outside the threshold are removed from the point cloud.

3.4.3. Voxelisation

The density of the extracted point cloud differs throughout the mapped area, as it is hig y dependent on the LiDAR measurement cover from the UAV. Processing these non-uniformly distributed data-points was performed using a data-reduction based voxel grid. Using the voxel data structure, the geographical space was conceptualised and represented as a set of volumetric elements (voxels) [41]. Voxels are values in a regular grid in three-dimensional space, similar to pixels in two-dimensions. PCL was again used to perform the data reduction with the internal voxel grid functionality. The voxel resolution was chosen as $0.04 \times 0.04 \times 0.001$ m for the global x,y,z coordinate frame. Plant samples from the crop parcels also typically are cut in 0.5×0.5 m squares. Thus, the 0.04 m voxel grid size allows us to reproduce plant samples height using between 11×11 and 13×13 voxels stacks. The Velodyne LiDAR measurements have an accuracy of typical ± 0.03 m, as mentioned in Section 2.3, and a resolution of 0.002 m according to the manual [37]. To ensure we contain the original LiDAR measurement resolution for volume calculation and comparison, we have set the voxel grid resolution height to 0.001 m.

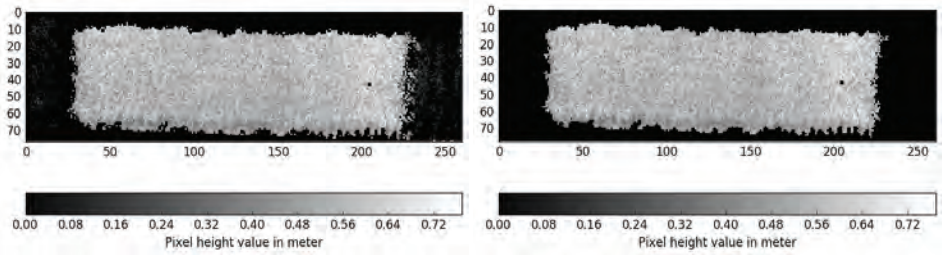
3.5. Voxel Grid Processing and Crop Parcel Parameter Estimation

After processing the extracted point cloud, we now have a voxel grid representing a subpart of the experimental field, as illustrated in Figure 5d. To estimate the crop height and volume, the algorithm uses the approach shown in Figure 6. In this estimation process, the voxel grid is converted into a pixel grid and estimated soil height subtracted, in order to calculate crop height and volume.



(**a**) (**b**)

Figure 6. *Cont.*



(**c**) (**d**)

Figure 6. Voxel grid crop parcel processing. (**a**) initial pixel grid after conversion from voxel grid; (**b**) pixel interpolation for the detected surface area; (**c**) ground-level removal from pixels; (**d**) crop parcel height estimation using region growing.

3.5.1. Conversion from Voxel to Pixel Grid

The voxel grid extracted from mapped LiDAR data was converted into a pixel grid in which each value represents the maximum active z-value in the voxel grid [42]. By numerically integrating all pixel values, the volume was estimated. The pixel grid uses the maximum value to ensure that the full length of each detectable plant is obtained. An example of output processing is shown in Figure 5d, which relates to the voxel grid in Figure 6a.

3.5.2. Interpolation of Missing Grid Pixels

The resulting 2D grid contains pixels with no value because the LiDAR provides sparse measurement data; therefore, a measurement is not present for each voxel stack. As we want to estimate crop height to calculate crop volume, we used iterative nearest neighbour interpolation [43] to determine whether to interpolate a pixel. This method requires that six of eight neighboring pixels contain a measurement for a pixel to be included. The pixel is then assigned the mean value of all valid measurements from the eight neighboring pixels. This process was repeated for the whole grid until no more pixels were included. A six-pixel criterion was used to ensure that the algorithm did not include areas without plant material.

3.5.3. Estimating Ground Level below the Crop Parcels

To estimate the volume that the crop covers inside the pixel grid, the ground level must be subtracted for all pixel values. The current LiDAR setup cannot consistently measure the ground level between crop plants. In many cases, plant density is too high and the LiDAR does not observe the parcels from an angle where the ground can be measured. In this study, we assumed that the ground level exhibits linear progression under the parcel compared to the surrounding area. The ground-level linear assumption is made because the soil was treated by agricultural machines prior to seeding. The ground plane was estimated using measurements from all regions of the parcel that are not part of the net parcel. Using least squares approximation, the processing algorithm provided a linear plane estimate for the surface below the crops. The linear plane is defined in Equation (2):

$$h_{ground}(x_p, y_p) = a_0 x_p + a_1 y_p + a_2, \tag{2}$$

where a_0, a_1, a_2 are constants and (x_p, y_p) is the pixel position. Figure 7 shows a 2D cross-section example of the ground-level approximation process of the linear plane.

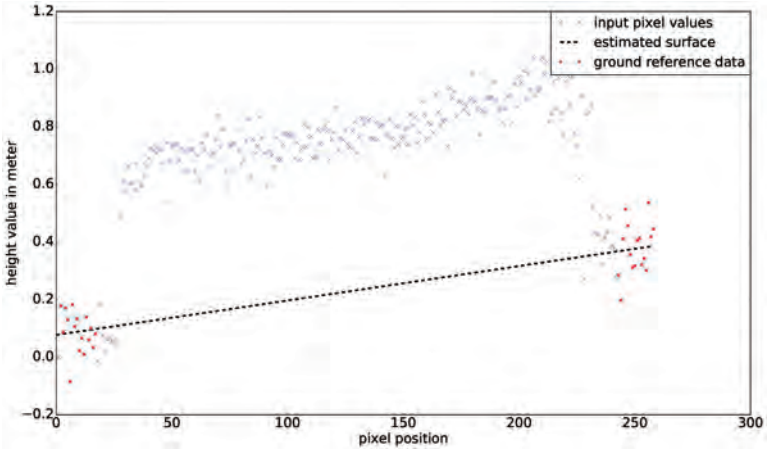


Figure 7. 2D cross-section of ground-level approximation using estimated soil points.

The estimated surface plane was subtracted from the pixel grid using Equation (2). Ground compensation resulted in a new pixel map, an example of which is illustrated in Figure 6c. Pixel values then represented the estimated crop height above ground level.

3.5.4. Crop Parcel Extraction

We used region growing [44] to determine all active pixels that belong to the crop area, beginning with the pixel with the maximum height value that fits the eight-connected neighborhood criteria. The gradient between the new test pixel and the average of pixels already in the region was used as the threshold value. We determined a threshold value of 0.1 m if the new pixel was included in the crop area. Figure 6d shows an example result of the growing region. The average crop height was then estimated using all pixels in the pixel grid extracted by region growing. All pixel values were then summed, enabling the estimation of volumes of individual crop parcels.

3.6. Correlating Crop Height to N-Application

From experience, we expect the amount of N to influence crop height. The assumption is that applied N will increase plant growth; however, it will saturate as the amount increases. To describe this relationship between crop height and N, we modeled it using a logistic curve [45,46], as defined in Equation (3):

$$h_{crop}(N_m) = \frac{c_1}{1 + e^{-c_0 N_m}} + h_{min}, \tag{3}$$

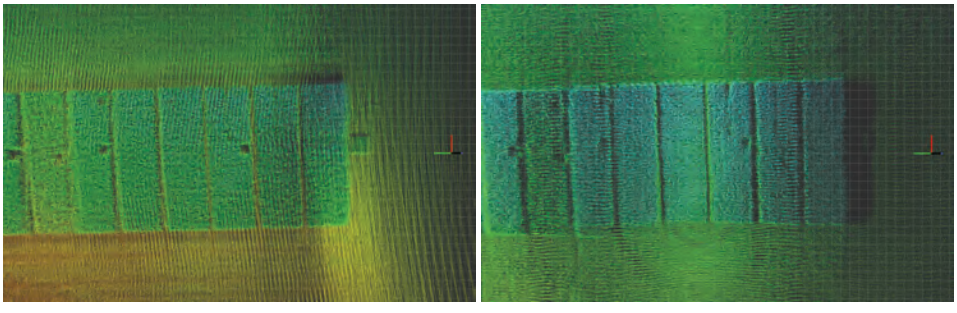
where $(c_1 - h_{min})$ is the curve's maximum value, h_{min} the average minimum height, N_m the applied nitrogen amount, and c_0 the gradient of the curve. $(c_1 - h_{min})$ denotes the curve's maximum value, as c_1 describes the estimated limit of increase in crop height due to nitrogen application. The model in Equation (3) was correlated to the estimated mean height for each crop parcel and the treatment plan in Appendix A. The individual nitrogen treatment for 84 crop parcels was added up to the data recording date. We used the average crop height from flight Path A as it covers all crop parcels in the experimental field.

4. Results

The results are divided into four subsections: experimental field mapping, mapping comparison, relation to treatment-plan, and crop parcel volume estimation.

4.1. Experimental Field Mapping

Figure 8 illustrates the mapping results of the different flight paths. Path B results in a much higher density point cloud for the crop parcel. Although we can distinguish more details using Path B data, the overall structure tends to be more heterogeneous. Moreover, the gaps between individual crop parcels are better defined in Path B, which is a direct result of higher LiDAR resolution.

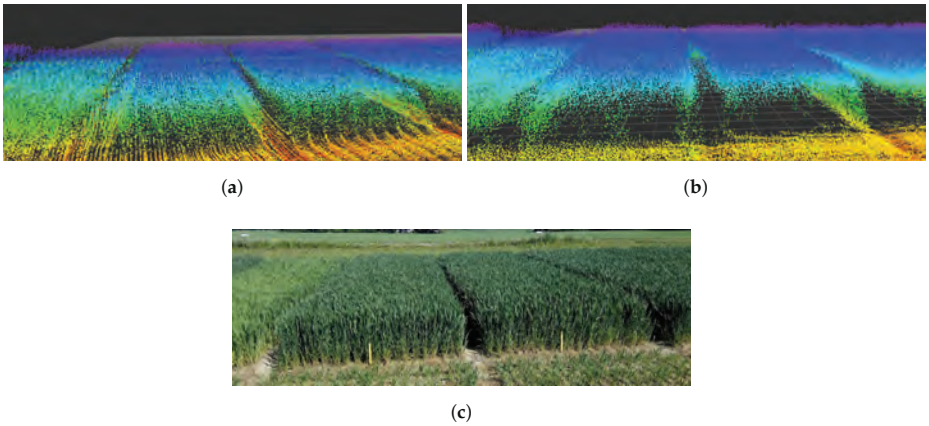


(**a**) (**b**)

Figure 8. Raw LiDAR mapping data of flight paths over the experimental field illustrated by the ROS rviz 3D visualisation tool. Both are shown as relative coordinates from the take-off location of the UAV. The colour of individual points represents their *z*-axis value in the global frame. (**a**) mapping result of flight Path A, indicating a more homogeneous point cloud distribution; (**b**) mapping result of flight Path B, indicating a more heterogeneous point cloud distribution.

4.2. Crop Parcel Comparison

The mapped point clouds constitute the output prior to PCL processing. Because the individual gross parcels were extracted, they could be recombined to produce the point cloud shown in Figure 9a,b. We compared the point cloud results (Figure 9) to a photograph taken on the same day (Figure 9c) with a similar observation position and angle. The mapped field parcels contained approximately 400–700 points per square meter.



(**a**) (**b**)

(**c**)

Figure 9. Mapped LiDAR data compared to actual conditions. (**a**) mapping result of flight Path A; (**b**) mapping result of flight Path B; (**c**) photograph of the same area in the experimental field.

4.3. Crop Height and Nitrogen Correlation

By comparing the total amount of N applied over the season to average crop height, we obtained the plot in Figure 10. Based on our estimates, the crop height would not increase during the current growth stage if more than 200 $\frac{\text{kgN}}{\text{ha}}$ was applied.

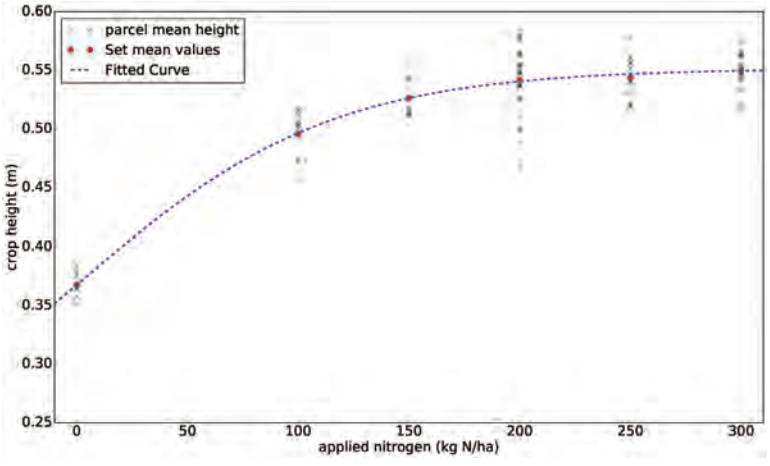


Figure 10. Relationship between crop height and nitrogen, and the model with estimated parameters.

4.4. Volume Estimates

The mapped point cloud data were processed and the calculated volumes of individual crop parcels are shown in Figure 11. The grey lines are included to indicate specific crop parcels that the UAV overflew when following Path B. Path A generally results in a lower volume estimate compared to Path B.

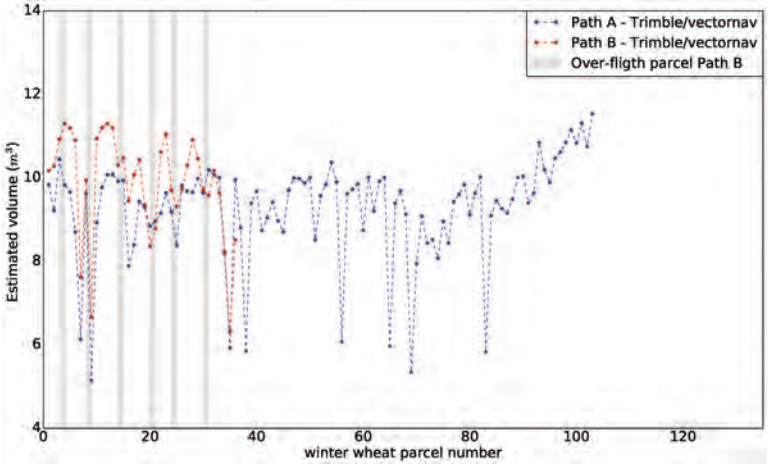


Figure 11. Volume estimates for flight paths A and B. The grey line marks the specific crop parcels that the UAV overflew when following Path B.

5. Discussion

Our results illustrate that the UAV sensor system can be used to collect spatial data from crop-fields, which can be post-processed to derive canopy volume estimates and textural analysis of individual crop parcels. Crop parcels were extracted based on known GNSS reference corner points, and crop volumes were then estimated. The resulting crop parcel volume and crop height estimates are in the range of 5.1–11.3 m^3 and 0.35–0.58 m, respectively.

The estimated volumes and their accompanying N application strategy represent the expected variation in the N status, biomass accumulation, and N content of a winter wheat crop during the spring growing season. These differences can be observed in terms of greenness, plant height, and crop biomass. On average, the crop parcel volume will not increase with N applications above 200 $\frac{\text{kgN}}{\text{ha}}$ in this experimental field. The impact of factor N_m in Equation (3) is hig y dependent on the level of N in the soil before seeding, and its maximum level will differ between fields. Even if we can estimate all or some of these unknown variables, we are still challenged by other random and controllable factors, such as climate, precipitation/irrigation, other nutrients, technological difficulties, and poor management in the field.

The variation in estimated volumes between Paths A and B arise from two factors. Path A observed all parcels in the same manner from both sides, creating a homogenous point cloud of all parcels, whereas Path B mapped more LiDAR data points per crop parcel, resulting in a denser point cloud with multiple overlapping voxel values. As we used maximum values to create the pixel maps, Path B provided a higher volume estimate for crop parcels farther from the flight path, as small orientation errors will have a larger impact farther away. In Path A, old cut plant-sample areas (Figure 8) were not mapped accurately because these areas were covered by surroundings crops. However, as these sample cut areas regrow with time, Path B was able to measure crop heights in these locations. The regrown spots also resulted in a higher volume estimate for Path B.

We can conclude that mapping using Path A can determine external crop volume and provide a homogeneous point cloud. This is because no information about plant density is available for inside the crop parcel. Mapping with Path B allows for a much denser point cloud of individual crop parcels because it includes multiple point values for the same plant area; thus, LiDAR permeation into the parcel could be calculated and used to estimate the density. We suggest that a combination of the two flight paths, that is, using flight Path A and flying over specific parcels of interest again, with a flight pattern similar to Path B, would be optimal, enabling fast flight times and denser point clouds for areas of interest. In the future, one could also fuse data from LiDAR and camera data directly, in order to provide additional information to the individual points in the point-cloud.

It is apparent that ground-based vehicles should be used if more dense point clouds are required. If the UAV were moved closer to the individual parcels, the wind impact from its rotors would disturb the plants, making a ground-based vehicle a more viable option. Because agricultural ground vehicles tend to drive on the same tracks in the field, they can also be used as a reference to distinguish between soil and plants.

The method presented in this study is dependent on known GNSS reference points for extracting crop parcels. The voxel and pixel grids of crop parcels can be used as a training data set for a neural network method to automatically detect and process individual crop parcels, similar to the process described in [47–49]. A future approach could involve weekly data recording to obtain a larger data set for crops at different stages of growth. Combining the trained neural network method with the ground vehicle solution could enable direct estimation of crop volume in the field.

Furthermore, determining the optimal point in the season for crop monitoring is beyond the capabilities of the current system. Continuous monotoring of each area using UAV would be hig y labour intensive and would need to be based on operator experience or other sources. However, satellite spatial imaging data could be combined with crop, soil, and weather data to provide an indicator of when and where LiDAR estimates of volume/biomass would be most advantageous.

6. Conclusions

We introduce a novel UAV design and mapping method for observing crops and estimating their current production and environmental states. The system design and utilised software components are made available, in order to allow adaption in similar projects. In this project, we mapped winter wheat with a row distance of 0.12 m at an altitude of 6 m, in 3D LiDAR point clouds. Textural analysis of the LiDAR data was performed to estimate the soil surface and total plant volume for individual crop parcels. The crop parcel heights vary from 0.35–0.58m, and correlate with their N treatment strategies. Individual crop parcels were mapped with a spatial resolution of $0.04 \times 0.04 \times 0.001$ m, based on the collected LiDAR data. As the UAV was flying at an altitude of six meters, the mapped field contained approximately 400–700 points per square meter.

Different flight methods were evaluated to determine the impact on spatial resolution and volume estimates. We concluded that flight Path B provides the highest LiDAR spatial resolution for mapping, but a lower coverage per battery because this approach increases energy consumption. A future approach could combine both flight paths for optimal mapping, where Path A is used to obtain an overview and Path B hig ights the details of specific areas of interest.

Supplementary Materials: Recorded ROS bags, videos, system info design and source code are available online at the project homepage [33].

Acknowledgments: We would like to acknowledge Kjeld Jensen at the University of Southern Denmark for providing technical support on the Trimble GNSS platform. Thanks are due to Uffe Pilegård Larsen at Research Centre Flakkebjerg for providing assistance in connection with the experiments. This research was financially supported by the Intelligente VIRKemidler til reduktion af reduktion af kvælstofudvaskningen (VIRKN) Project, funded by the Danish Ministry of Environment and Foods Grønt Udviklings- og Demonstrationsprogram (GUDP). The work presented here is partially supported by the FutureCropping Project [50] funded by Innovation Fund Denmark.

Author Contributions: Martin Peter Christiansen and Morten Stigaard Laursen have made substantial contributions in the development of the methods and algorithms presented in this paper. Morten has also designed the sensor mount printed in 3D nylon and the in-house built extension board. Søren Skovsen has made a substantial contribution in manuscript preparation. Rasmus Nyholm Jørgensen and René Gislum have made contributions to the definition of the research, data acquisition and manuscript preparation.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CAD	Computer-Aided Design
DJI	Dà-Jiāng Innovations Science and Technology Co.
GNSS	Global Navigation Satellite System
LiDAR	Light Detection and Ranging
IMU	Inertial Measurement Unit
IO	Input/Output
PPS	Pulse-Per-Second
PCL	Point Cloud Library
RGB	Red, Green and Blue
REP	ROS Enhancement Proposal
RMS	Root Mean Square
ROS	Robot Operating System
RTK	Real Time Kinematic
SPI	Serial Peripheral Interface-bus
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator
WGS	World Geodetic System

Appendix A

Table A1 contains the nitrogen treatment plan for the 84 crop parcels included in this experiment. However, the treatment plan is unknown for the buffer parcels (remaining 19 crop parcels). JumpStart2.0 is a microbial soil inoculant used to enhance crop growth.

Table A1. Treatment plan for winter wheat test parcels in the experimental field.

Treatment	Treatment Plan	Test Parcels
1	0 kg N/ha	7,33,54,67
2	50 kg N/ha (16th March) + 50 kg N/ha (20th April)	18,42,60,82
3	50 kg N/ha (16th March) + 100 kg N/ha (20th April)	20,35,50,83
4	50 kg N/ha (16th March) + 150 kg N/ha (20th April)	1,38,48,84
5	50 kg N/ha (16th March) + 200 kg N/ha (20th April)	13,30,57,66
6	50 kg N/ha (16th March) + 250 kg N/ha (20th April)	9,31,52,77
7	50 kg N/ha (16th March) + 50 kg N/ha (20th April) + 100 kg N/ha (5th May)	8,40,47,70
8	50 kg N/ha (16th March) + 50 kg N/ha (20th April) + 150 kg N/ha (5th May)	17,25,53,64
9	50 kg N/ha (16th March) + 50 kg N/ha (20th April) + 200 kg N/ha (5th May)	2,26,56,71
10	50 kg N/ha (16th March) + 100 kg N/ha (20th April) + 50 kg N/ha (5th May)	3,29,62,73
11	50 kg N/ha (16th March) + 100 kg N/ha (20th April) + 100 kg N/ha (5th May)	10,22,59,76
12	50 kg N/ha (16th March) + 100 kg N/ha (20th April) + 150 kg N/ha (5th May)	12,24,45,75
13	50 kg N/ha (16th March) + 50 kg N/ha (20th April) + 100 kg N/ha (05th May) + 50 kg N/ha (7th June)	6,37,61,78
14	100 kg N/ha (16th March)	14,39,49,72
15	200 kg N/ha (16th March)	11,27,51,80
16	0 kg N/ha + JumpStart2.0	5,36,63,81
17	50 kg N/ha (16th March) + 50 kg N/ha (20th April) + JumpStart2.0	19,32,43,74
18	50 kg N/ha (16th March) + 100 kg N/ha (20th April) + JumpStart2.0	4,41,44,79
19	50 kg N/ha (16th March) + 150 kg N/ha (20th April) + JumpStart2.0	21,28,55,65
20	50 kg N/ha (16th March) + 150 kg N/ha (20th April)	16,34,46,69
21	50 kg N/ha (16th March) + 150 kg N/ha (20th April) + 100 kg N/ha (5th May)	15,23,58,68

Appendix B

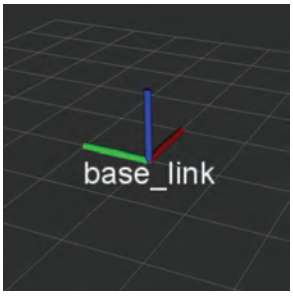
Transforms between different sensor frames are illustrated in Figure A1. The frame notation follow ROS REP-103 [36] and REP-105 [51] specification for coordinate frames for robots.

The coordinate frame called "base_link" is rigidly fixed to the robot base and defines the robots pose in relation to its surroundings. The "_link" notation after sensor frame names indicates that all sensors are rigidly fixed to "base_link". The "base_link" and "gnss_link" transforms are the same, as shown in Table A2 because we used the Trimble GNSS as the pose reference.

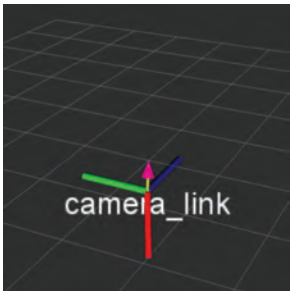
Table A2. Sensor frame transforms defined in ROS for the UAV.

Transform	x	y	z	ψ	θ	ϕ
base_link->camera_link	-0.0545	0	-0.424	0	$\frac{\pi}{2}$	0
base_link->gnss_link	0	0	0	0	0	0
base_link->imu_link	-0.039	-0.008	-0.294	π	$\frac{\pi}{2}$	0
base_link->laser_link	0	0.013	-0.304	$-\frac{\pi}{2}$	0.593233	$-\frac{\pi}{2}$

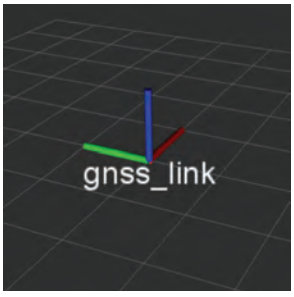
We use the CAD model of the sensor mount, found on the project homepage, to extract the orientation between the sensors and "base_link". The sensor mount was designed to utilise the precision dowel pin holes in the LiDAR and IMU as illustrated in Figure A2. As no dowel pin locations were available for the camera, we chose to use countersunk screws, as these self-center upon tightening. This combined again with a tight fit, which allows for repeatable mounting locations on the sensor frames. This sensor mount design ensures a significant reduction in the play of each sensor. Furthermore, as the nylon print can be expected to expand and contract uniformly, any changes in size caused by moisture and temperature should result in primarily a translation of the sensors relative to each other.



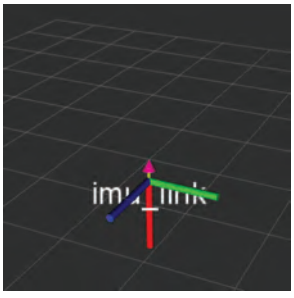
(**a**) The base_link frame.



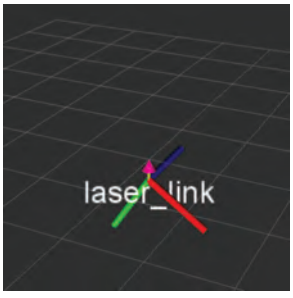
(**b**) The camera_link frame.



(**c**) The gnss_link frame.



(**d**) The imu_link frame.



(**e**) The laser_link frame.

Figure A1. Sensor frames defined in ROS and illustrated in the build in rviz 3D visualisation tool with regards to base_link.

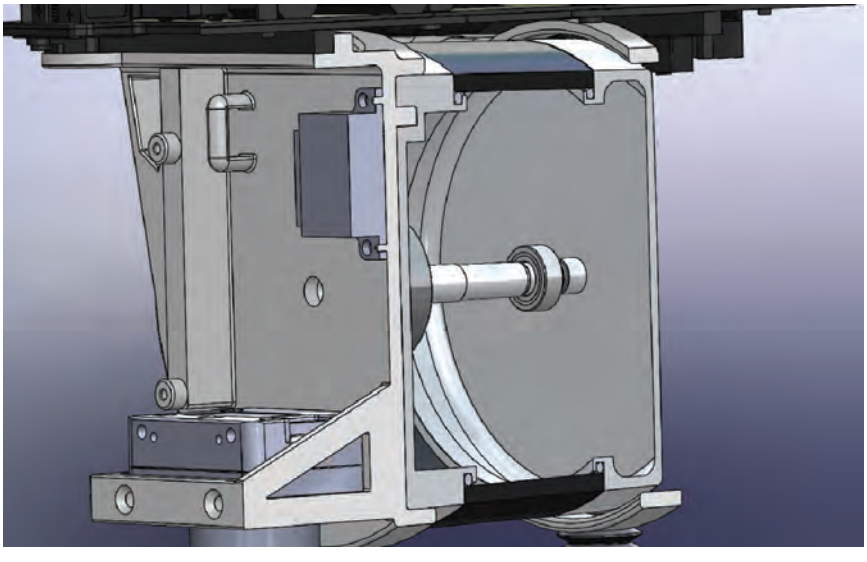


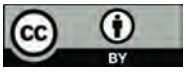
Figure A2. Image of the 3D CAD model of the sensor mount, designed to utilise the precision dowel pin holes in the LiDAR and IMU.

References

1. Senay, G.; Ward, A.; Lyon, J.; Fausey, N.; Nokes, S. Manipulation of high spatial resolution aircraft remote sensing data for use in site-specific farming. *Trans. ASAE* **1998**, *41*, 489.
2. Rudd, J.D.; Roberson, G.T.; Classen, J.J. Application of satellite, unmanned aircraft system, and ground-based sensor data for precision agriculture: A review. In *2017 ASABE Annual International Meeting*; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2017; p. 1.
3. Matese, A.; Toscano, P.; Di Gennaro, S.F.; Genesio, L.; Vaccari, F.P.; Primicerio, J.; Belli, C.; Zaldei, A.; Bianconi, R.; Gioli, B. Intercomparison of UAV, aircraft and satellite remote sensing platforms for precision viticulture. *Remote Sens.* **2015**, *7*, 2971–2990.
4. Madsen, S.L.; Dyrmann, M.; Laursen, M.S.; Jørgensen, R.N. RoboWeedSupport-Semi-Automated Unmanned Aerial System for Cost Efficient High Resolution in Sub-Millimeter Scale Acquisition of Weed Images. *Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng.* **2017**, *11*, 835–839.
5. Jørgensen, R.; Brandt, M.; Schmidt, T.; Laursen, M.; Larsen, R.; Nørremark, M.; Midtiby, H.; Christiansen, P. Field trial design using semi-automated conventional machinery and aerial drone imaging for outlier identification. In *Precision Agriculture'15*; Wageningen Academic Publishers: Wageningen, The Netherlands, 2015; pp. 146–151.
6. DJI Phantom 2 Series. Available online: http://www.dji.com/products/phantom-2-series (accessed on 20 November 2017).
7. Bendig, J.; Bolten, A.; Bennertz, S.; Broscheit, J.; Eichfuss, S.; Bareth, G. Estimating biomass of barley using crop surface models (CSMs) derived from UAV-based RGB imaging. *Remote Sens.* **2014**, *6*, 10395–10412.
8. Bendig, J.; Yu, K.; Aasen, H.; Bolten, A.; Bennertz, S.; Broscheit, J.; Gnyp, M.L.; Bareth, G. Combining UAV-based plant height from crop surface models, visible, and near infrared vegetation indices for biomass monitoring in barley. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *39*, 79–87.
9. Lin, Y.; Hyyppa, J.; Jaakkola, A. Mini-UAV-borne LIDAR for fine-scale mapping. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 426–430.
10. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR System with Application to Forest Inventory. *Remote Sens.* **2012**, *4*, 1519–1543.
11. Tullda , H.M.; Bissmarck, F.; Larsson, H.; Grönwall, C.; Tolt, G. Accuracy evaluation of 3D lidar data from small UAV. In Proceedings of the SPIE Conference on Electro-Optical Remote Sensing, Photonic Technologies, and Applications IX, Toulouse, France, 16 October 2015; pp. 964903–964911.
12. Khan, S.; Aragão, L.; Iriarte, J. A UAV–lidar system to map Amazonian rainforest and its ancient landscape transformations. *Int. J. Remote Sens.* **2017**, *38*, 2313–2330.
13. Guo, Q.; Su, Y.; Hu, T.; Zhao, X.; Wu, F.; Li, Y.; Liu, J.; Chen, L.; Xu, G.; Lin, G.; et al. An integrated UAV-borne lidar system for 3D habitat mapping in three forest ecosystems across China. *Int. J. Remote Sens.* **2017**, *38*, 2954–2972.
14. Sankey, T.T.; McVay, J.; Swetnam, T.L.; McClaran, M.P.; Heilman, P.; Nichols, M. UAV hyperspectral and lidar data and their fusion for arid and semi-arid land vegetation monitoring. *Remote Sens. Ecol. Conserv.* **2017**, doi:10.1002/rse2.44.
15. Jozkow, G.; Totha, C.; Grejner-Brzezinska, D. UAS Topographic Mapping with Velodyne Lidar Sensor. *ISPRS Ann. Photogram. Remote Sens. Spatial Inf. Sci.* **2016**, *3*, 201–208.
16. Anthony, D.; Elbaum, S.; Lorenz, A.; Detweiler, C. On crop height estimation with UAVs. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago, IL, USA, 14–18 September 2014; pp. 4805–4812.
17. Anthony, D.; Detweiler, C. UAV Localization in Row Crops. *J. Field Robot.* **2017**, *34*, 1275–1296.
18. Jensen, T.; Munkholm, L.J.; Green, O.; Karstoft, H. A mobile surface scanner for soil studies. In Proceedings of the Second International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture and Forestry, Madrid, Spain, 21–23 May 2014; pp. 187–194.
19. Andújar, D.; Rueda-Ayala, V.; Moreno, H.; Rosell-Polo, J.R.; Valero, C.; Gerhards, R.; Fernández-Quintanilla, C.; Dorado, J.; Griepentrog, H.W.; et al. Discriminating crop, weeds and soil surface with a terrestrial LIDAR sensor. *Sensors* **2013**, *13*, 14662–14675.

20. Reiser, D.; Vázquez Arellano, M.; Garrido Izard, M.; Griepentrog, H.W.; Paraforos, D.S. Using Assembled 2D LiDAR Data for Single Plant Detection. In Proceedings of the 5th International Conference on Machine Control & Guidance, Vichy, France, 5–6 October 2016.
21. Jensen, T.; Karstoft, H.; Green, O.; Munkholm, L.J. Assessing the effect of the seedbed cultivator leveling tines on soil surface properties using laser range scanners. *Soil Tillage Res.* **2017**, *167*, 54–60.
22. Jaeger-Hansen, C.L.; Griepentrog, H.W.; Andersen, J.C. Navigation and tree mapping in orchards. In Proceedings of the International Conference of Agricultural Engineering, Valencia, Spain, 8–12 July 2012.
23. Underwood, J.P.; Jagbrant, G.; Nieto, J.I.; Sukkarieh, S. Lidar-Based Tree Recognition and Platform Localization in Orchards. *J. Field Robot.* **2015**, *32*, 1056–1074.
24. Andújar, D.; Rosell-Polo, J.R.; Sanz, R.; Rueda-Ayala, V.; Fernández-Quintanilla, C.; Ribeiro, A.; Dorado, J. A LiDAR-based system to assess poplar biomass. *Gesunde Pflanz.* **2016**, *68*, 155–162.
25. Maguire, R.; Alley, M.M.; Flowers, W. *Fertilizer Types and Calculating Application Rates*; Communications and Marketing, College of Agriculture and Life Sciences, Virginia Polytechnic Institute and State University: Blacksburg, VA, USA, 2009.
26. Santi, C.; Bogusz, D.; Franche, C. Biological nitrogen fixation in non-legume plants. *Ann. Bot.* **2013**, *111*, 743–767.
27. Marschner, H. *Marschner's Mineral Nutrition of Higher Plants*, 2nd ed.; Academic Press: Cambridge, MA, USA, 1995; Chapter 11–12.
28. Mulla, D.; Schepers, J. Key processes and properties for site-specific soil and crop management. In *The State of Site-Specific Management for Agriculture*; ACSESS: Madison, WI, USA, 1997; pp. 1–18.
29. Gislum, R.; Boelt, B. Validity of accessible critical nitrogen dilution curves in perennial ryegrass for seed production. *Field Crops Res.* **2009**, *111*, 152–156.
30. Eitel, J.U.; Magney, T.S.; Vierling, L.A.; Brown, T.T.; Huggins, D.R. LiDAR based biomass and crop nitrogen estimates for rapid, non-destructive assessment of wheat nitrogen status. *Field Crops Res.* **2014**, *159*, 21–32.
31. Schirrmann, M.; Giebel, A.; Gleiniger, F.; Pflanz, M.; Lentschke, J.; Dammer, K.H. Monitoring Agronomic Parameters of Winter Wheat Crops with Low-Cost UAV Imagery. *Remote Sens.* **2016**, *8*, 706.
32. DJI Matrice 100. Available online: https://www.dji.com/matrice100 (accessed on 20 November 2017).
33. UAV LiDAR Project Homepage. Available online: https://vision.eng.au.dk/future-cropping/uav_lidar/ (accessed on 20 November 2017).
34. Quigley, M.; Conley, K.; Gerkey, B.P.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009.
35. Carvalho, J.P.; Jucá, M.A.; Menezes, A.; Olivi, L.R.; Marcato, A.L.M.; dos Santos, A.B. Autonomous UAV Outdoor Flight Controlled by an Embedded System Using Odroid and ROS. In *CONTROLO 2016: Proceedings of the 12th Portuguese Conference on Automatic Control*; Garrido, P., Soares, F., Moreira, A.P., Eds.; Springer: Berlin, Germany, 2017; pp. 423–437.
36. ROS REP-103. Available online: http://www.ros.org/reps/rep-0103.html (accessed on 20 November 2017).
37. Velodyne LiDAR. VLP-16 In *VLP-16 Manual: User's Manual and Programming Guide*; Velodyne LiDAR, Inc.: San Jose, CA, USA, 2016.
38. VectorNav. *VN-200 GPS/INS*; VectorNav Technologies: Dallas, TX, USA; Available online: https://www.vectornav.com/products/vn-200 (accessed on 20 November 2017).
39. Litchi for DJI Mavic/Matrice/Phantom/Inspire/Spark. Available online: https://flylitchi.com/ (accessed on 25 September 2017).
40. Hsieh, C.T. An efficient development of 3D surface registration by Point Cloud Library (PCL). In Proceedings of the 2012 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Taipei, Taiwan, 4–7 November 2012; pp. 729–734.
41. Popescu, S.C.; Zhao, K. A voxel-based lidar method for estimating crown base height for deciduous and pine trees. *Remote Sens. Environ.* **2008**, *112*, 767–781.
42. Wu, B.; Yu, B.; Yue, W.; Shu, S.; Tan, W.; Hu, C.; Huang, Y.; Wu, J.; Liu, H. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* **2013**, *5*, 584–611.
43. Parker, J.A.; Kenyon, R.V.; Troxel, D.E. Comparison of interpolating methods for image resampling. *IEEE Trans. Med. Imaging* **1983**, *2*, 31–39.

44. Revol, C.; Jourlin, M. A new minimum variance region growing algorithm for image segmentation. *Pattern Recognit. Lett.* **1997**, *18*, 249–258.
45. Richards, F. A flexible growth function for empirical use. *J. Exp. Bot.* **1959**, *10*, 290–301.
46. Lei, Y.; Zhang, S. Features and partial derivatives of Bertalanffy-Richards growth model in forestry. *Nonlinear Anal. Model. Control* **2004**, *9*, 65–73.
47. Li, B.; Zhang, T.; Xia, T. Vehicle detection from 3d lidar using fully convolutional network. *arXiv* **2016**, arXiv:1608.07916.
48. Becker, C.; Häni, N.; Rosinskaya, E.; d'Angelo, E.; Strecha, C. Classification of Aerial Photogrammetric 3D Point Clouds. *arXiv* **2017**, arXiv:1705.08374.
49. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3D. net: A new Large-scale Point Cloud Classification Benchmark. *arXiv* **2017**, arXiv:1704.03847.
50. FutureCropping Reseach Project. Available online: https://futurecropping.dk/ (accessed on 20 November 2017).
51. ROS REP-105. Available online: http://www.ros.org/reps/rep-0105.html (accessed on 20 November 2017).



sensors

MDPI

Article

Assessment of the Possibility of Using Unmanned Aerial Vehicles (UAVs) for the Documentation of Hiking Trails in Alpine Areas

Paweł Ćwiąkała, Rafał Kocierz, Edyta Puniach *, Michał Nędzka, Karolina Mamczarz, Witold Niewiem and Paweł Wiącek

AGH University of Science and Technology, Faculty of Mining Surveying and Environmental Engineering, 30-059 Cracow, Poland; pawelcwi@agh.edu.pl (P.Ć.); kocierz@agh.edu.pl (R.K.); mnedzka@agh.edu.pl (M.N.); karolina.mamczarz@op.pl (K.M.); w.niewiem@gmail.com (W.N.); paw.wiacek@gmail.com (P.W.)
* Correspondence: epuniach@agh.edu.pl; Tel.: +48-697-926-891

Received: 18 September 2017; Accepted: 25 December 2017; Published: 29 December 2017

Abstract: The research described in this paper deals with the documentation of hiking trails in alpine areas. The study presents a novel research topic, applying up-to-date survey techniques and top quality equipment with practical applications in nature conservation. The research presents the initial part of the process—capturing imagery, photogrammetric processing, quality checking, and a discussion on possibilities of the further data analysis. The research described in this article was conducted in the Tatra National Park (TNP) in Poland, which is considered as one of the most-visited national parks in Europe. The exceptional popularity of this place is responsible for intensification of morphogenetic processes, resulting in the development of numerous forms of erosion. This article presents the outcomes of research, whose purpose was to verify the usability of UAVs to check the condition of hiking trails in alpine areas. An octocopter equipped with a non-metric camera was used for measurements. Unlike traditional methods of measuring landscape features, such a solution facilitates acquisition of quasi-continuous data that has uniform resolution throughout the study area and high spatial accuracy. It is also a relatively cheap technology, which is its main advantage over equally popular laser scanning. The paper presents the complete methodology of data acquisition in harsh conditions and demanding locations of hiking trails on steep Tatra slopes. The paper also describes stages that lead to the elaboration of basic photogrammetric products relying on structure from motion (SfM) technology and evaluates the accuracy of the materials obtained. Finally, it shows the applicability of the prepared products to the evaluation of the spatial reach and intensity of erosion along hiking trails, and to the study of plant succession or tree stand condition in the area located next to hiking trails.

Keywords: UAV; monitoring; mountain trails

1. Introduction

Alpine areas are shaped by interacting morphogenetic processes [1,2]. Their activity depends on local environmental conditions, such as geological structure, relief, topoclimate, and the spatial reach of altitudinal zones [1]. Avalanches [3,4], landslides [5], and debris flows [6] are typical processes in alpine areas. They are the main cause of transformation of mountainous area landscapes in the world [7,8]. The Tatras slopes are being modelled by debris flows, mud-debris flows, and torrential flows, also by soil material flushing, rill erosion, nivation, solifluction, creep, aeolian processes, and the effects caused by needle ice [1,9]. These processes transform the Tatra slopes with varying intensity and their effects are differentiated in time and space.

An additional factor that increases the activity of erosive processes in high mountains is hiking [10–12]. The anthropogenic degradation on hiking trails on slopes is currently the subject of research, both in Poland [13,14] and worldwide [15–17].

In order to determine the effects of anthropogenic degradation on hiking trails it is necessary to examine the tourist flow and its spatial and temporal distribution. To this end, different methods were used. They include counting the number of tourists by people trained for this purpose [12], registration of tourists using the video system [18] or the use of pyroelectric sensors to accurately determine the distribution of tourist movements [19].

The effects of tourism are easily visible in the relief of hiking trails and paths and appear as various landforms created by erosion, denudation, or accumulation processes. Traditional methods of relief measurements include geomorphological mapping at various scales, test points, and surveys in slope catenas. The dynamics of the processes is typically determined with simple measurement techniques, such as a line of pegs driven into the ground at different depths, lines of marked debris to measure cover movements, nets to catch material falling off the rock walls, and instruments to measure the size of lifting [20]. Currently, surveying methods are used in topography. They cover traditional measurement techniques, such as satellite measurements [21] or total stations. They supply accurate and discrete data about the relief. Unfortunately, they are time-consuming and, therefore, they are used for small areas. In turn, remote sensing methods, such as terrestrial (TLS) and airborne laser scanning (ALS) [22,23], aerial and terrestrial photogrammetry [24,25], and satellite imaging, enable surveying large areas and provide quasi-continuous data. Their drawback is their high price. Despite this fact, they are increasingly being used in geomorphology and facilitate the acquisition of rich collections of data, with varying precision and detail.

Recently, UAVs, which are used to obtain spatial data, have become quite popular. Currently, the most common sensors installed on UAVs are non-metric cameras. However, devices equipped with infrared cameras, multi-spectral cameras, or light detection and ranging (LIDAR) scanners are becoming more and more popular [26,27]. Depending on the type of the flying unit, the UAV enables performing measurements over the areas ranging from a few acres to several square kilometres [28–30]. It is worth noting that this new and very popular method of spatial data acquisition ensures the required measurement accuracy. Ground sampling distance (GSD) is of crucial importance here. It depends mainly on the flight level and the camera sensors and lenses used to take measurements. The second important aspect is to provide the final products with georeference data. Currently, mainly ground points measured with traditional survey techniques are used for this purpose. There are also other solutions, such as the use of point coordinates obtained from other sources, for example, from satellite images [28]. Georeference information can be also supplied directly by measuring exact coordinates of images' projection centres. Such solutions are typically used in situations when the UAV is fitted with a satellite receiver which determines the UAV position in real time, and with inertial navigation systems (INS) [31,32].

Another factor which affects the accuracy of results obtained from the UAV data is topography of the area, which is the subject of this research. Most publications indicate that products generated by UAV flights are verified with satellite measurement methods (in the kinematic or static mode), tacheometry, and TLS. It is worth noting that, in many cases, both the products obtained from UAV and TLS data are burdened with the influence of vegetation, which negatively affects the quality of both datasets (from the UAV and TLS) [33]. Therefore, TLS is not always a reliable method for referential data acquisition. Taking this into account, analyses that rely on independent classical measurements (total station, satellite measurements) should be considered as more reliable [34]. In the areas where the landscape is not very diversified (flat areas) [35,36] the accuracy of the final products approximate 2–3 GSD [29,37]. However, according to our best knowledge, no studies have been conducted so far to determine the accuracy of UAV-based photogrammetric products in areas of such large height differences. The relative height differences of the area being subject to the tests described in this article are up to 1000 m.

Despite the fact that photogrammetric measurements taken from low levels have lower accuracies than the data obtained from TLS, their great advantages include even distribution of survey points in the study area, less effort needed to acquire data, and a lower price of the equipment. Considering the products generated during UAV flights (digital elevation models (DEM), orthophotomosaics), UAVs may be widely applied to monitor landslides [34,38–41], soil and slope erosion [29,42,43], and to measure seashores in order to check the evolution of beaches and cliffs [44–46]. It is worth noting that one may come across publications that suggest using UAVs for periodic monitoring of landslides and TLS for measuring large deformations of objects that occur after extreme events [41]. UAV surveys also find applications in vegetation monitoring [47], precision agriculture enhancement [48], information acquisition concerning damage caused by natural disasters [49,50], geological surveys [51], and visual investigation of large-scale structures, such as dams [52].

There are a variety of photogrammetric software tools available for UAV-based data. The use of web services and software packages that "automatically generate 3D points from arbitrary image configurations" was examined by Neitzel and Klonowski [53]. Vallet et al. [54] compared georeferenced DTMs produced from Pix4D and NGATE (in SOCET SET). The objective of the study presented in [55] is to investigate the level of accuracy that can be achieved using two of these software tools: Agisoft PhotoScan Pro and an open-source alternative, IGN MicMac, in sub-optimal survey conditions. On the basis of these studies, it can be concluded that the applied software influences the obtained results.

The most important aim of the research described in this article was to document the selected linear hiking trails in the Western Tatras between the lower subalpine zones and alpine glades. The current state of the trails was established using the photogrammetric method (UAV), whose accuracy was verified with other surveying methods (measurement of control profiles, TLS). The data obtained, after comparing them with archival data provided by the TNP (or data obtained in successive measurement series) comprise valuable information which enables, for example, the examination of the trails' erosion degree, the progress of forest succession, and the impact of anthropogenic denudation on tree line changes.

The outcome of the work comprise digital surface models (DSM) and/or digital terrain models (DTM) in two periods: primary and secondary. The current model was created on the basis of data obtained with the use of a UAV. The data for the creation of the original model came from ALS and was supplied by the TNP. It is worth noting that in the longer perspective, the follow-up measurements in the same area will facilitate a more in-depth analysis of the impact that tourism has on the environment.

2. Materials and Methods

2.1. Study Site

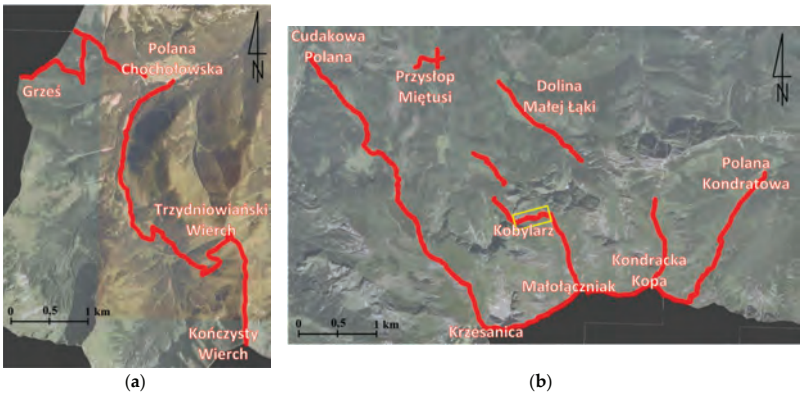
Research described in this article was conducted in the Tatra National Park, which is situated in the southern part of Poland (Figure 1). The Tatras comprise the only alpine area in Poland. Due to outstanding natural features of this area, the national park was founded here in 1955. It is one of the largest national parks in Poland with the total area of 21,197 ha [56]. In comparison to other national parks in Europe it is a relatively small park.

The TNP is one of the most visited parks in Poland. It is visited by 3.5 million tourists annually [56]. It is also one of the most visited parks in Europe. In comparison, alpine parks are visited on average by 700,000 tourists every year [57]. Although the density of hiking trails in TNP is 1.3 km/km^2 and is much higher than density of alpine and Pyrenean parks, a serious problem in the Tatras is the number of tourists, which exceeds the capacity of hiking trails [58]. The exceptional popularity of this place causes the destruction of vegetation and soil cover and contributes to the reinforcement of the morphogenetic processes, resulting in the development of numerous forms of erosion. The issue of excessive tourism in the Tatras and its impact on the landscape transformation has been widely discussed in the literature [12,14,57–59].

(**a**) (**b**)

Figure 1. Location of the study area: (**a**) location of the Tatra Mountains in Poland, (**b**) location of the Western Tatras and High Tatras (source: own study based on [60]).

The linear hiking trails in the Western Tatras were used as case studies to verify whether it is possible to use a UAV to evaluate the spatial reach and intensity of erosion in the alpine areas. It has also been verified how accurately can detect relief changes by this method. In total, about 27 kilometres of hiking trails were measured (Figure 2), which constitutes about 10% of all trails in the TNP. Some of these trails were overhauled recently (2014–2015). Their total length is 13.8 km. The length of the trails which have not been repaired is 13.2 km.



(**a**) (**b**)

Figure 2. Hiking trails in the Western Tatras which are the subject of this research: (**a**) western part of the research area, (**b**) eastern part of the research area (source: own study based on [61]).

2.2. Data Acquisition

For data acquisition, a model S1000+ UAV manufactured by the DJI Company (Shenzhen, China), (Figure 3) was used. It is an octocopter equipped with the DJI A2 flight controller with a GNSS receiver, a compass, and IMU, which enable the determination of the device's horizontal position with an accuracy of $\pm$1.5 m, and height with the accuracy of $\pm$0.5 m. According to information provided by the manufacturer, the device works properly when the wind does not exceed 8 m/s and the inclination is not more than 35° [62]. The UAV was fitted with a Alfa A7R camera (Sony, Tokyo, Japan) equipped with a Sony Zeiss Sonnar T* FE 35 mm F2,8 ZA lens, whose position was stabilised with a Zenmuse Z15-A7 gimbal (DJI, Shenzhen, China).

Measurements were carried out in August and September 2016. They included registration of vertical images of trails from onboard the UAV along with georeference information. All UAV flights were carried out by the holders of Unmanned Aerial Vehicle Operator (UAVO) qualifications certificate, while maintaining high security standards in the Visual Line of Sight (VLOS) conditions.

Figure 3. The device used for measurements, the DJI S1000 multicopter (source: authors' own study).

Field surveys with the UAV were preceded by drawing up UAV flight plans for each hiking trail which was subject of the research. It was assumed that the GSD would be:

- 15 mm in non-forested areas; and
- 20 mm in the areas covered with high trees.

Taking into account features of the equipment used for measurements and the characteristics of the study area, the following assumptions were made:

- the UAV real flight height (above ground level) over the non-forested areas was 100 m;
- the UAV real flight height (above ground level) over the forested areas was 150 m;
- the minimum side overlap between images in adjacent strips was 70%; and
- the minimum forward overlap between images in the strip was 85%.

The Tatras' steep slopes were a very demanding study area, so the DTM that was created on the basis of the ALS point cloud was used to plan the UAV flights. Therefore, flight plans considered the height differences along the trail, which exceeded 1000 m. For each hiking trail UAV flights were planned in the two parallel strips which were also parallel to the trail axis. In this way 37 flights were planned with lengths ranging from 0.7 km to 3.2 km. They were drawn up in the DJI Ground Station software. The example of the planned flight mission is shown in Figure 4. The real flight height above ground level and the maximum height differences between waypoints in each UAV flight are presented in Table 1.

The missions' projects also included the optimal take-off and landing sites. They had to be easily accessible from the trail, far from obstructions, such as high trees or rock walls, and it was also important to ensure that during the UAV flight mission the device could not be present below the start point height.

It should be emphasized that the planning of UAV flights over difficult terrain, which is a very steep slope of alpine mountains, requires great care and precision. This problem was highlighted in [63]. In that paper, the authors also pointed out the need for accurate flight planning in order to maintain the desired GSD. Resolving this problem for difficult alpine areas is described in detail above.

During the field works, before each planned UAV flight, it was necessary to set out, mark, and measure coordinates of control points and check points (Figure 5) in the area covered by the UAV flight. Previously, however, the project of the photogrammetric control point distribution (control points and check points) along the UAV flight routes was made. It was assumed that pairs of points

should be located (on both sides of the trail) in the interval of approximately 100 m. The correctness of the assumption was verified by analysing how the number and distribution of ground control points affect the accuracy of the DSM/DTM of the elongated areas, which is the subject of a separate publication [64]. Due to difficult terrain conditions, the project was also verified on the basis of orthophotomap and DTM provided by the TNP (archival aerial orthophotomap and archival DTM from ALS data), in order to evade wooded areas (lack of visibility from the air) and areas which are inaccessible due to the steepness of the slopes.

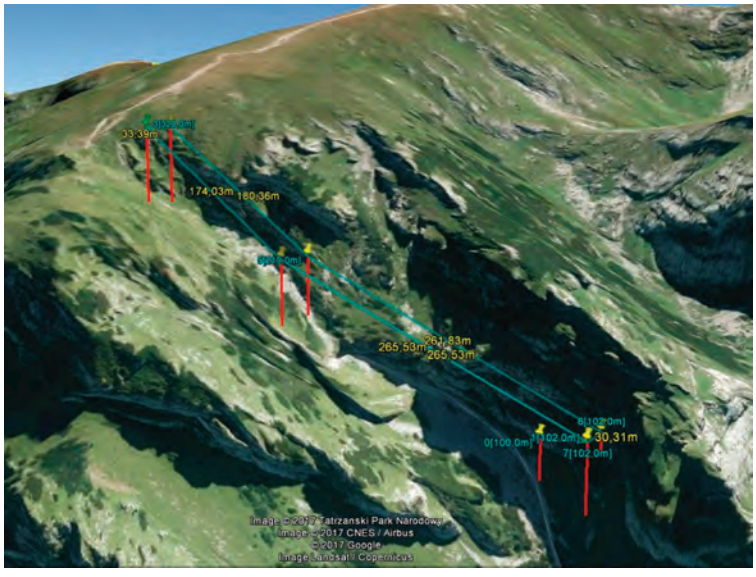


Figure 4. The UAV mission for a section of the trail from the Przysłop Miętusi glade to Małołączniak in the area of Kobylarzowy Żleb (source: authors' own study). The location of the mission area is indicated by a yellow rectangle in Figure 2b.

Table 1. Summary of the number of images, control and check points and flight parameters for each photogrammetric mission.

Mission	Number of Images	Real Flight Height above Ground Level (m)	Maximum Height Differences between Waypoints within One Mission (m)	Number of Points of the Photogrammetric Control	
				Control Points	Check Points
Czerwone Wierchy (13 missions)	2889	100–150	180–840	143	113
Kobylarz (three missions)	351	100	230–365	16	10
Grzes (five missions)	783	150	250–315	29	17
Trzydniowianski Wierch (nine missions)	1806	100–150	200–410	75	48
Jaworzynka-Siodłowa Perć (two missions)	254	150	230	9	7
Polana Olczyska (two missions)	488	150	90	11	8
Przysłop Miętusi (one mission)	172	100	90	9	5
Wielka Polana (two missions)	368	100	140	12	5

Figure 5. A control point on the image taken from the UAV (source: authors' own study).

Control points and check points were measured with the real-time network (RTN) GNSS method referenced to a network of reference stations NadowskiNET or the real-time kinematic (RTK) GNSS method referenced to one of ten base stations that were distributed evenly within the study area.

In the case of RTN GNSS measurements, the final coordinates were immediately calculated in the field and registered in the controller. The primary problem for the RTK GNSS method (using our own base station) was to determine the spatial coordinates of the base station. To this end, throughout the entire period of kinematic measurements the phase observations were registered on the base stations. These observations along with data from NTW1 (Nowy Targ) reference station, which belongs to the ASG-EUPOS system were used to determine coordinates of the base stations with the accuracy not lower than 1 cm. Based on the coordinates of the base stations, final coordinates of points were calculated. In total, 537 points were measured in the PL-2000/7 planar coordinate system (PL-ETRF2000) and Krondstat86 height system (PL-KRON86-NH) based on the levelling quasi geoid PL-geoid-2011. The accuracy of obtained coordinates is at the level of 2–3 cm.

2.3. Data Processing

Image processing was conducted in Agisoft PhotoScan Professional software (Agisoft LLC, St. Petersburg, Russia) which employs full SfM multi-view stereopsis and estimates internal optical parameters and spatial positioning of the respective cameras. Due to the large size of UAV data, it was also necessary to use appropriate hardware. The computer with two Intel Xeon E5 processors, 64 GB RAM, and an Nvidia Quadro K4200 graphics card was used for the photogrammetric processing.

The first step of image processing was to align the images. At this stage, the images were uploaded to the software and were given the initial orientation by adding approximate coordinates of images' projection centres. In total, 7111 images were used. Images were combined in pairs automatically with the limit of 4000 tie points per pair. Images were aligned when parameters that analyse the image are at the level of single pixels. After images had been aligned, control points and check points were indicated on the individual images. This was preceded by uploading coordinates of the terrestrial photogrammetric control to the software. Each marker was indicated on all photos where it was visible. Table 1 shows a list of the regions (photogrammetric missions combined in groups) along with the number of images, as well as control points and check points that were used for alignment.

With the block of photographs prepared in such a way for each mission, the initial alignment commenced. At this stage, some of points of the photogrammetric control worked as control points and the others as check points, which made it possible to evaluate the project accuracy. The block of photographs was aligned and, at the same time, the camera calibration parameters were determined. In this process the following values were determined: the principal distance (c), the location of the principal point (c_x and c_y) and distortion parameters (k_1, k_2, k_3, and p_1 and p_2). As a result of the

alignment the root mean square errors of the control points and check points were obtained. They are listed in Table 2. On this basis the accuracy of the generated products was initially assessed.

Table 2. The initial root mean square errors of control points and check points.

Mission		m_x (mm)	m_y (mm)	m_h (mm)	m_{xy} (mm)	m_{xyh} (mm)
Czerwone Wierchy (13 missions)	control points	16	13	17	20	26
	check points	19	17	29	25	38
Kobylarz (three missions)	control points	56	57	34	80	88
	check points	68	50	63	85	102
Grzes (five missions)	control points	8	9	14	11	18
	check points	15	15	22	21	30
Trzydniowianski Wierch (nine missions)	control points	16	18	23	24	33
	check points	15	18	31	23	39
Jaworzynka-Siodłowa Perć (two missions)	control points	14	22	30	26	40
	check points	18	21	37	28	47
Polana Olczyska (two missions)	control points	12	17	26	21	33
	check points	20	17	23	26	35
Przysłop Miętusi (one mission)	control points	17	5	10	18	20
	check points	15	6	20	16	26
Wielka Polana (two missions)	control points	18	15	10	23	26
	check points	18	20	96	27	100
Average	**control points**	**24**	**25**	**22**	**34**	**41**
	check points	**29**	**24**	**47**	**37**	**60**

Then, the final alignment (optimization) of blocks of photographs was performed. This process involved all points of the photogrammetric control (both control points and check points). The final mean square errors of these points coordinates are listed in Table 3. Mean values for these parameters calculated for all the missions are: m_x = 29 mm, m_y = 29 mm, and m_h = 31 mm, which correspond to the error of the horizontal point position m_{xy} = 41 mm and the error of spatial position m_{xyh} = 51 mm. The worst results of RMSE errors were obtained for the missions in the Kobylarz region. This was due to a small number of control points, which were difficult to set up due to field conditions (steep slopes and exposure).

Table 3. Final mean square errors of points coordinates of the photogrammetric control.

Mission	m_x (mm)	m_y (mm)	m_h (mm)	m_{xy} (mm)	m_{xyh} (mm)
Czerwone Wierchy (13 missions)	28	25	37	38	53
Kobylarz (three missions)	56	63	29	84	90
Grzes (five missions)	16	19	25	25	36
Trzydniowianski Wierch (nine missions)	18	21	30	27	41
Jaworzynka-Siodłowa Perć (two missions)	13	20	31	24	39
Polana Olczyska (two missions)	18	18	28	26	38
Przysłop Miętusi (one missions)	22	8	24	23	34
Wielka Polana (two missions)	35	24	38	42	56
Avarage	**29**	**29**	**31**	**41**	**51**

The next step was to create a dense point cloud with the method of dense matching. As a result, a total of 7,854,760,000 points were generated in all missions. Table 4 shows the size of point clouds for each region. In addition, a number of tie points, which were generated when stitching images, was given. After creating a point cloud for each mission, point clouds were connected in larger groups. The aim of this step was to generate the largest study areas possible in coherent blocks. Then DSMs were created. These were used as a basis to conduct orthorectification of images and to create orthophotomaps. In the end, the finished products (point clouds, orthophotomaps, DSMs) were exported. Table 4 shows the resolution of the final products. It is important to note that before the ultimate generation of an orthophotomap, it was necessary to deprive it of all artifacts, such as people who move.

Table 4. The basic parameters of final products generated on the basis of photographs.

Mission	Number of Tie Points	Number of Points in the Point Cloud	Pixel Size for the Orthophotomap	DEM Mesh Size
Czerwone Wierchy (13 missions)	2,012,363	360,257,300	15–20 mm	25–35 mm
Kobylarz (three missions)	356,169	428,923,000	15 mm	25–30 mm
Grzes (five missions)	751,761	799,493,000	20 mm	30–40 mm
Trzydniowianski Wierch (nine missions)	1,565,467	1,888,167,000	15–20 mm	30–40 mm
Jaworzynka-Siodłowa Perć (two missions)	187,664	253,844,000	20 mm	37 mm
Polana Olczyska (two missions)	350,874	312,586,000	20 mm	35 mm
Przysłop Miętusi (one missions)	117,043	151,814,000	15 mm	27 mm
Wielka Polana (two missions)	371,399	417,357,000	15 mm	29 mm

2.4. Control Measurements

To verify the accuracy of the remotely-sensed data, TLS measurements of selected sections of trails and measurements of control profiles in various places along the trails were conducted.

The TLS measurements covered a 400-metre section of the trail on Hala Kondratowa (Figure 6). A Leica ScanStation C10 laser scanner (Leica Geosystems, St. Gallen, Switzerland) and the set of Leica HDS 6" targets were used for measurements. The point clouds were registered with the traverse technique based on the three tripods method. Measurements were taken on 16 stations in the strip of land whose width was about 60 m. Each point cloud on the station was registered with the resolution of 10 mm/10 m. Spatial coordinates of 15 scanner stations were set with RTK GNSS measurements with the accuracy of ±1.5 cm. This enabled to significantly reduce propagation of errors between the scanner stations. The TLS data was processed with the Leica Cyclone software. At the beginning, the scanning traverse was aligned in the local coordinate system and the obtained data was merged into one point cloud. Then, the point cloud was transformed into the PL-2000/7 system and the final alignment was performed. The mean transformation error was 0.032 m. The next stage of work was clearing the point cloud of points representing, inter alia, people on the trail, surveying equipment, and artificial landscape features. Manual removal of the non-corrected data was carried out. In the end, data was filtered and the resulting point cloud was manually divided into two layers: ground and trees.

Figure 6. Location and spatial reach of the area where TLS was carried out (Hala Kondratowa) (source: authors' own study based on [61]).

In order to validate the accuracy of the DSM obtained from UAV-borne photogrammetry, cross-sections of trails were also measured within the study area. Measurements were taken by means of RTK GNSS or RTN GNSS methods. The measurement methodology was analogous to the one applied for the photogrammetric control. Control profiles were measured in such places as Małołączniak, Przełęcz Kondracka, along Dolina Kondratowa, along the trail at Ratuszowe Źródło (Czerwony Grzbiet), along the trail from Cudakowa Polana to Ciemniak.

3. Results and Discussion

3.1. Accuracy Analysis of Products Obtained from UAV-Borne Photogrammetry

As a result of image processing in Agisoft Photoscan Professional software (Agisoft LLC), the RMSE of the spatial position of control points and check points obtained during the initial alignment of the block of photographs are 41 mm and 60 mm, respectively (Table 2). Ultimately, for all the points of the photogrammetric control this parameter amounted to 51 mm (Table 3). These are the values which only indicate the possible accuracies of the products created. It is worth noting that the maximum errors m_{xyh} for control points and check points were over 100 mm and were obtained for Kobylarzowy Żleb. This is a particularly difficult area because of the significant height difference exceeding 300 m on the trail section of about 500 m. For other areas the analysed parameter did not exceed 50 mm.

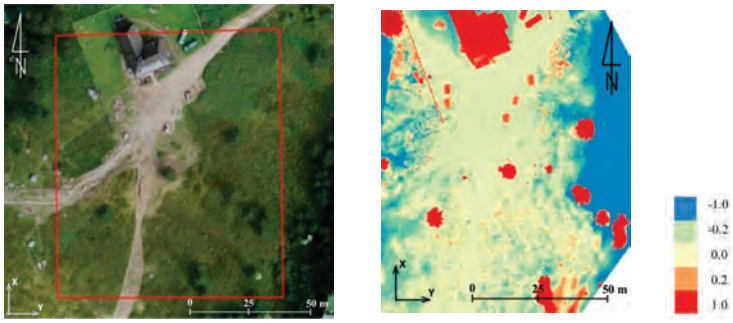
In many publications, the evaluation of accuracy of products obtained from UAV-borne photogrammeters is based only on RMSE analysis for check points. The authors of the paper [65] conducted a study of UAV usefulness for documenting landslides. Based on the results of the study, it was found that applying UAV and SfM enabled to create DSMs of the landslide surface with an accuracy of 4–5 cm in the horizontal and 3–4 cm in the vertical direction. The results obtained are similar to these presented in this paper, but are related to a small object with a slight height differences. However, in [63] the presented tests were conducted on an elongated object, but it was an area with a small inclination along the axis of the UAV flight. The flight plan was a single strip only. The accuracy of the orthophotomosaic was based on a model with only eight control points and 50 check points. The comparison between the coordinates of points extracted from the UAV orthophotomap and their counterparts obtained with the GNSS receiver (i.e., check points) showed a RMSE of 58 and 56 mm for coordinates X and Y, respectively. Our research, similarly as in the above mentioned paper, concerns elongated objects, but with a much more complicated relief. Our results enabled to determine the accuracy and usefulness of UAV data obtained for steep mountain trails for determining even minor changes in their relief. In order to perform a reliable assessment of the accuracy of the products obtained from UAV-borne photogrammetry, control measurements were additionally made (described under Section 2.4).

3.1.1. Evaluation of the Accuracy Based on TLS Data

Data obtained from TLS is of high accuracy and resolution, and the resolution varies substantially and depends on the distance from the station. These characteristics have determined that the TLS method is widely used to evaluate the accuracy of the products obtained from UAV-borne photogrammetry [33,35,55]. Unfortunately, the results of such comparisons do not always give reliable results to assess the accuracy of UAV-based photogrammetric products. In [55] mean differences between the photogrammetric DSMs and the TLS point cloud lower than 20 cm are achieved and standard deviations renged from 0.7 m to 1.1 m (depending on the software used). However, it should be noted that the offsets between the photogrammetric DSMs and the TLS point cloud were low within the control region. Deviations appear outside of the control region, growing with distance to the control region.

In this paper in the first step of accuracy assessment DTM obtained from TLS data were compared with of the DSM that was generated from UAV-borne photogrammetry. Data analysis was based on the comparison of raster layers. It was for two reasons. First, analyses of this type require much less

processing power, which is of great importance when handling data which has a very high spatial resolution. Secondly, the input data was characterized by similar spatial resolution (3 cm). Therefore, before the analyses could start, TLS point clouds were cleared off manually to remove objects of the land cover and then the open source project CloudCompare software was used to generate rasters whose resolution was 3 cm. Further calculations were done in QGIS (open source project) and results were presented in bitmap form (Figure 7).



(**a**) (**b**) (**c**)

Figure 7. The shelter on Hala Kondratowa: a comparison of DTM from TLS with DSM from UAV: (**a**) the study area, (**b**) results of comparison, and (**c**) colour scale (the units given in m) (source: authors' own study).

By comparing data from TLS, and data obtained from the UAV, it was possible to estimate the accuracy of the results obtained. The example presented in Figure 7, enables to draw the following conclusions:

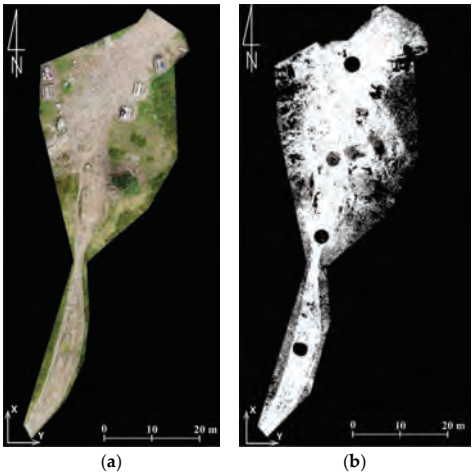
- data within the trail area not covered with vegetation feature high compliance (around 5 cm),
- significant differences may be noticed on objects of the land cover; they result from comparing DTM-TLS and DSM-UAV,
- large differences on the border of the study area are caused by interpolation of data from TLS and small scanning resolution in the distance above 30 m from the trail.

The problem of elimination of the influence of vegetation on the results of comparison of the TLS data and the UAV-based product was discussed, among others, in [66]. Studies presented in this paper were carried out on a landslide which extends over a horizontal distance of 850 m between elevations of 2105 m at the crown, and 1740 m at the toe, with an average slope of 25°. The quality of the photogrammetric DTM was assessed by subtracting the overlapping TLS DTM. In the vertical direction the RMS difference was 0.31 m although maximum deviations reached +3.44 to −4.08 m. The authors pointed out that the most significant errors are induced by some small trees and bushes, the effects of which could not be reliably removed from the photogrammetric DTM. The same conclusion was formulated in [33]. The article compared the data from the UAV equipped with RGB camera with TLS and total station measurements. The authors also presented a procedure of filtering point clouds, the results of which allow obtaining point clouds representing DTM without data representing vegetation. Application of filtration significantly reduced the errors.

In order to obtain figures that describe the accuracy of data obtained from the UAV only on hiking trails, which was the subject of research described in this article, the authors also conducted analyses that compare the point cloud from TLS with the point cloud and mesh generated from the UAV pictures. Two test fields were identified (Figure 8 and Figure 11). The first one covered the section of the trail by the shelter on Hala Kondratowa and the surrounding area. In the first step differences between data from the UAV and data from TLS were analysed in certain points. For this

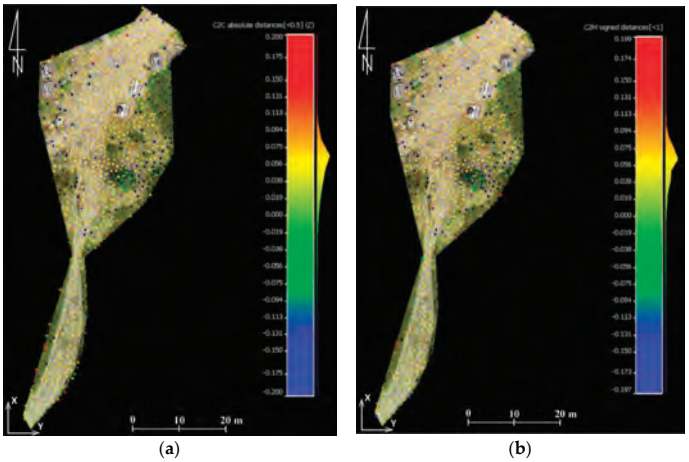
reason, a point cloud from TLS was thinned to the mesh with a cell size of 1 m and compared with a point cloud and mesh from the UAV (Figure 9). The average difference was about 0.05 m in both cases and standard deviation was around 0.05 m (Table 5). Similar calculations were done for the whole set of data obtained from TLS (Figure 10). In this case the average differences slightly exceeded 0.04 m (standard deviation was around 0.04 m) (Table 5).

The second area covers the part of the first area with a fragment of the trail in the immediate vicinity of the shelter on Hala Kondratowa. It is characterised by the lack of plants and a high density of points from TLS. For this area, differences between the point cloud from TLS and the point cloud generated from the UAV images were determined. For this area the average difference was 0.062 m and standard deviation came to 0.013 m (Table 5).



(**a**) (**b**)

Figure 8. Point clouds from the UAV (**a**) and TLS (**b**) which represent the first test field (source: authors' own study).



(**a**) (**b**)

Figure 9. Results of comparing the point cloud from TLS with the point cloud from the UAV (**a**) and mesh model from the UAV (**b**) in certain points for the first test field (source: authors' own study) (the units given are in m).

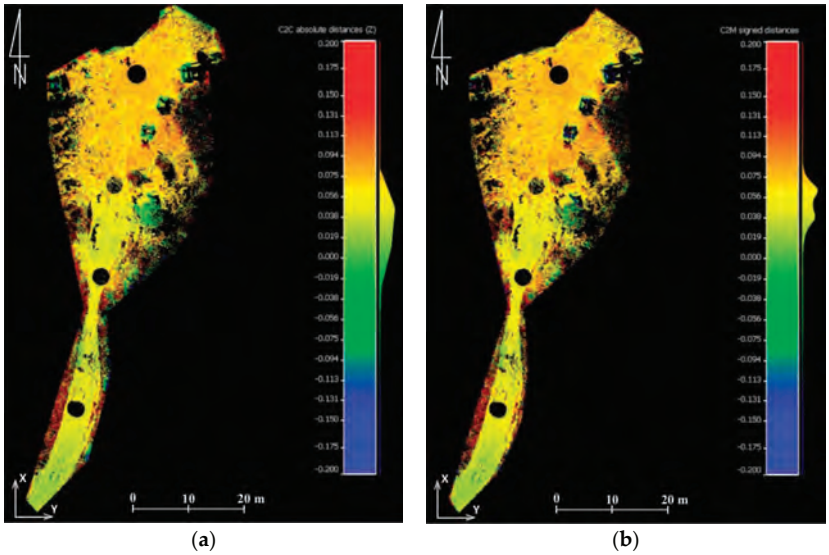


Figure 10. Results of comparing the point cloud from TLS with the point cloud from the UAV (**a**) and mesh model from the UAV (**b**) for the first test field (source: authors' own study) (the units given in m).

The low standard deviation obtained from the second test field (Table 5, Figure 11) indicates that both methods projected the relief in a similar way and the greatest differences occurred on the rock borders. The higher deviation values for the first test field result from different densities of the two sets of points and vegetation that appears in this area. The average distance value suggests that there is a difference between the methods. It may be caused by the accuracy with which georeference data is supplied or by not maintaining the scale parameter and it indicates accuracy level, at which these two methods can be compared.

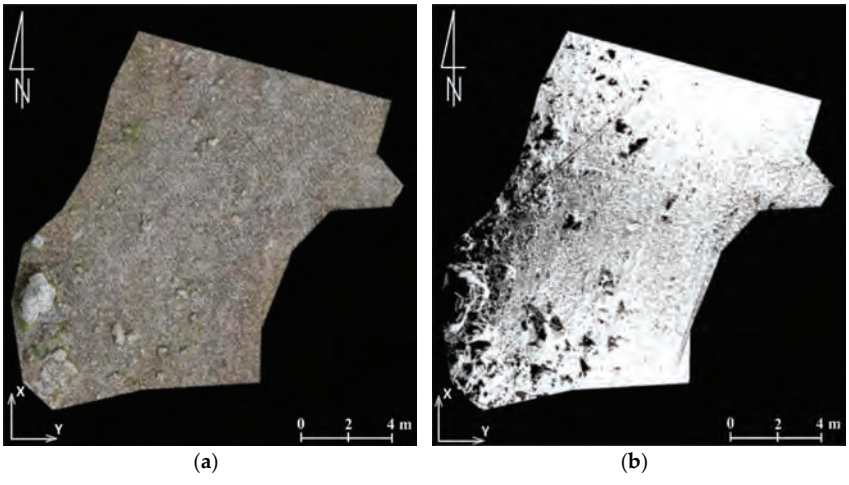
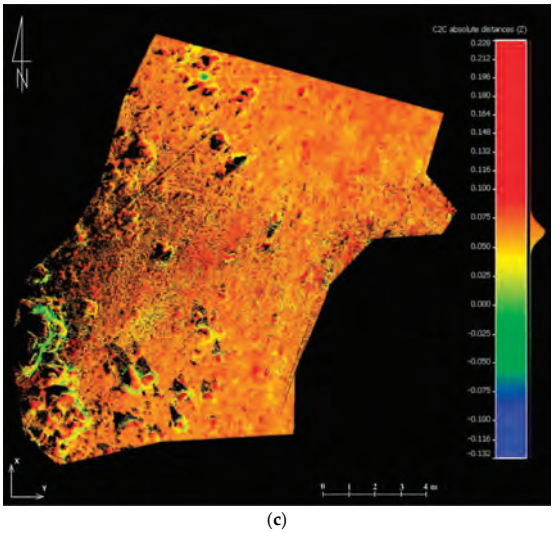


Figure 11. *Cont.*



(**c**)

Figure 11. Point clouds from the UAV (**a**) and TLS (**b**) which represent the second test field and results of their comparison (**c**) (source: authors' own study) (the units given in m).

Table 5. Results of comparing the point cloud from TLS with the point cloud and the mesh model from the UAV.

Test Field	Reference Object	Compared Object	Average Difference	Standard Deviation
first	UAV point cloud	thinned point cloud from TLS	0.050 m	0.058 m
	UAV mesh	thinned point cloud from TLS	0.048 m	0.054 m
	UAV point cloud	point cloud from TLS	0.042 m	0.044 m
	UAV mesh	point cloud from TLS	0.043 m	0.044 m
second	UAV point cloud	point cloud from TLS	0.062 m	0.013 m

Although TLS-based point clouds are as dense as the UAV-based point clouds, TLS data are subject to shadowing due to the oblique view point, what is shown in Figures 8–11. Such shadows are minimised in nadir UAV-acquired images and large-scale data acquisition can be obtained more effectively by UAV. The same conclusions were drawn in [66].

3.1.2. Evaluation of the Accuracy Based on the Control Profile Measurements

The accuracy of UAV-based products were also evaluated on the basis of control profiles. A similar solution was implemented in the studies presented in [67], but on a much smaller scale and without a detailed analysis of the results.

The points on the control profiles were measured by the RTK GNSS or RTN GNSS method. The method of measuring and calculating the coordinates of these points was the same as for the points of the photogrammetric control. For each control profile height differences between points in the profile recorded with different measurement methods were analysed. The authors used all types of information on relief that was available, that is:

- points measured with RTK/RTN GNSS methods: data of highest spatial accuracy (±2–3 cm), but also of low resolution. It is important to note the problem of relief generalisation that appears during discrete measurements;
- DTM acquired from TLS: data of high spatial accuracy (±5 cm) and of very high resolution (3 cm) along the trail;

- DSM acquired from the UAV: data of high spatial accuracy (±5 cm) and of very high resolution (3–5 cm); and
- DTM generated from the archival ALS points cloud: data of the lowest spatial accuracy (±30 cm) and of low resolution (35 cm).

Figure 12 presents an example of the cross-section. Although RTK/RTN GNSS measurements feature low density, it is easy to notice a very high compatibility of heights obtained with the model generated on the basis of the UAV images. This confirms the fact that both measurements were performed correctly. On the other hand, UAV DSM has lower elevations compared to other DTMs (TLS DTM and ALS DTM). In general, DSM should have higher elevation compared to DTM. Significant differences between UAV data and data from ALS (approx. 30 cm) result from the accuracy of this type of data. It must be emphasised that data from ALS comes from 2012. Therefore, it is necessary to consider the impact the small relief changes may have on the presented outcome.

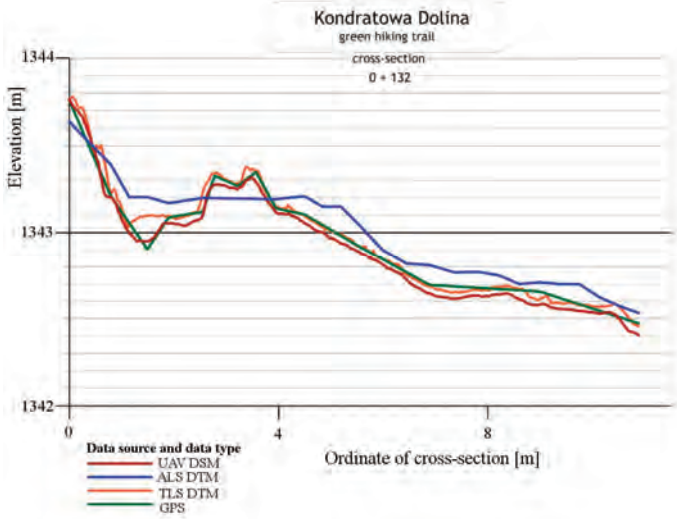


Figure 12. Example of the cross-section of the hiking trails determined with different survey methods (UAV, ALS, TLS, GPS) (source: authors' own study).

In the next part of the study, RTK/RTN GNSS method was recognised as the most accurate one and the authors of this paper identified parameters which enable the evaluation of the accuracy of the results generated with other methods, i.e., the mean height difference, maximum and minimum difference, and standard deviation of differences (Tables 6 and 7). The analysis was conducted using two scenarios. The first scenario considered all points measured with the RTK/RTN GNSS technique (points on the trail and points off the trail in the distance of up to 20 m) whereas, in the second scenario, only the points on the paved section of the trail were measured.

Table 6. Accuracy evaluation parameters of acquired materials for the first calculation scenario.

Parameter	GNSS–ALS	GNSS–UAV	GNSS–TLS
Average difference	0.174 m	0.043 m	0.074 m
Maximum difference	0.425 m	0.399 m	0.573 m
Minimum difference	−0.154 m	−0.100 m	−0.329 m
Standard deviation	0.115 m	0.132 m	0.139 m

Table 7. Accuracy evaluation parameters of acquired materials for the second calculation scenario.

Parameter	GNSS–ALS	GNSS–UAV	GNSS–TLS
Average difference	0.139 m	−0.027 m	0.016 m
Maximum difference	0.416 m	0.081 m	0.239 m
Minimum difference	−0.154 m	−0.100 m	−0.070 m
Standard deviation	0.099 m	0.041 m	0.060 m

For the first scenario, parameters for the accuracy evaluation were determined on the basis of 122 points on control profiles (Table 6, Figure 13a). In this case, average height differences for each analysed method are positive and positive deviations are much larger than the maximum negative height difference. Thus, one may come to the conclusion that points measured with the RTK/RTN GNSS method are located below the points determined using other methods. It is caused by the fact that products obtained from the UAV and TLS do not reflect the actual relief in the case of vegetated areas (they only show to a large extent vegetation over this area), whereas the average difference between data from ALS and GNSS measurements indicates systematic shift in height observations which does not exceed the average error of the ALS point cloud.

In order to evaluate the accuracy of detecting changes, which are caused by the erosion of trails, it was necessary to remove 30 points located outside the paved part of the trail (second calculation scenario) from the analysis. It considerably reduced the average height difference calculated for the UAV and TLS methods (Table 7, Figure 13). Parameters calculated for ALS did not change significantly in comparison to the first calculation scenario. It is important to note that in the case of the UAV, maximum and minimum height differences do not exceed 10 cm and standard deviation is 4.1 cm. The accuracy evaluation parameters for TLS data are less favourable, but they are also within the evaluated measurement accuracy.

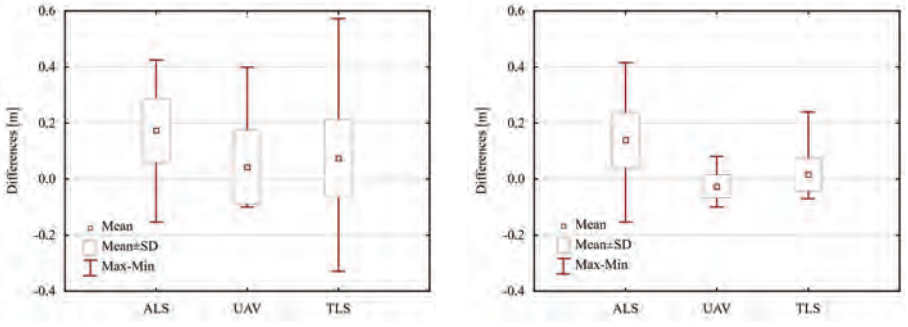


Figure 13. Box plot presenting parameters of accuracy evaluation for data obtained from ALS, UAV, and TLS against the RTK/RTN GNSS measurements: (**a**) for points located on trails and in their immediate vicinity; and (**b**) for points located on the trails (source: authors' own study).

Bearing in mind the above results, it may be noticed that the methodology adopted may be used to survey trail sections not covered with vegetation by means of the photogrammetric method based on the UAV with the accuracy of about 50 mm. This is a value of about three GSD and can be identified with the accuracy of UAV-based photogrammetric products in alpine areas, for linear objects, where the relative height differences exceed even 1000 m. It should be emphasized that the results obtained can be considered satisfactory, in particular in the light of the research results published so far [36,38,55]. Although obtaining accurate point clouds with UAVs seems to be easy to do nowadays, the results published so far refer to surface objects located in the alpine areas (landslides in particular)

or elongated objects with a slight inclination along their axis. This study has empirically proven that for extensive elongated objects located in alpine areas, obtaining highly-accurate products from UAV-borne photogrammetry is also possible, while maintaining the workflow described.

3.2. Possibilities of Using Products Obtained from UAV-Borne Photogrammetry for the Evaluation of Erosion along Hiking Trails

Castillo et al. [68] compared the accuracy, time, and cost of conventional and new methodologies for determining erosion of elongated objects (in this case gullies). Among others, 3D photo-reconstruction and TLS were compared. Despite its time-consuming character, 3D photo-reconstruction proved to be very competitive in terms of economy of work and quality of results. Unfortunately, this publication does not include UAV-borne photogrammetry. However, UAV imagery is successfully used to build DEM and to determine relief changes [29,34,65–69]. In [65] a UAV was used to collect a time series of high-resolution images over four years at seven epochs to assess landslide dynamics. The authors presented in [65] surface movement maps and the recorded displacement values exceed 7 m. Unfortunately, there is no ground validation for the movement vectors generated. Monitoring of a creek in an alpine environment was presented in [67]. The main result of the study was maps indicating the area-wide elevation differences based on grid files. Maximum elevation differences based on UAV DEMs were up to 2 m. The authors concluded that it is possible to evidently delineate erosion and deposition zones but, in general, results with higher accuracy were expected and the UAV-based approach is not suitable to record steep slopes. The results of our research presented above are in opposition to this conclusion.

Remotely-sensed data from a multi-rotor UAV were also used to estimate soil displacement from timber extraction trails in steep terrain (a vertical difference of over 120 m between the lowest and highest points in modelled terrain) [69]. To calculate the soil displacement volumes, a pre-harvest ALS-derived terrain model was to represent the terrain before the skid trails were constructed. In this study the authors focused only on the cut volume, in total, it was estimated that some 554 m^3 of earth had been displaced by the construction of skid trails over a cumulative length of 210 m. Although, some check points showed a significant error on the Z value of over 50 cm, the conclusion was that the UAV-based data shows considerable potential for improved environmental management and associated cost reduction. On the other hand the aim of the paper [28] was to monitor gully development by means of the UAV remote sensing. The authors concluded that the presented approach allows to map and measure erosion in very high detail. However, in this case soil erosion was only describe based on generated DTM with very high resolution, no quantitative assessment was provided.

On the other hand, an attempt to detect slight changes in the terrain relief on the basis of data obtained from UAV was described in [29]. In this paper surface changes at the level of single centimetres were detected, but it should be noted that UAV flight altitude during image acquisition was only 8–10 m above ground level.

Taking into account the lack of reliable data on this subject, it was decided to assess the applicability of UAV-based photogrammetry products to evaluation of erosion along hiking trails. To this end, point clouds obtained from the UAV were compared with archival data from ALS for the whole study area. On the basis of the results of these analyses, differential raster maps were generated which allow visual assessment of the degree and reasons for erosive phenomena. It has to be remembered, though, that the ALS archival data is characterised by far lower spatial resolution and, therefore, some spatial objects, such as route beds, boulders, are erosion craters, could have been generalized. Simultaneously, this data is features lower spatial accuracy which can lead to constant shifts in reference to actual data, which has been proved by the measurement of control profiles, as well as statistical analyses (Tables 6 and 7).

The analyses were performed using CloudCompare software (open source project). For the numeric calculations to be improved, the data from the UAV was subject to resampling where a

minimum distance between points was set at 3 cm. In this way, the number of points to be analysed was considerably reduced while the reliability of the analyses undertaken did not decrease. The ALS data was imported in two versions:

- the first one included all classes of points and therefore represented land forms along with the land cover (DSM ALS); and
- the second one included only points classified as ground points, which solely represent landforms (DTM ALS).

Subsequently, for the results of comparison to be as reliable as possible, TIN (triangulated irregular network) models were prepared for ALS data. It enabled the calculation of the distances between the clouds of points along normal vectors to the generated triangles and, therefore, the information was obtained about the direction of changes in the land form (decrease/increase). Moreover, the influence of the survey data density on the accuracy of determined distances was eliminated.

As a result of the analysis, for each point from UAV point clouds, a value of deviation from DSM ALS and/or DTM ALS was obtained which derives from the comparison of archival and current data. However, because browsing the point data results requires appropriate software, as well as a lot of RAM, the results obtained were presented as a raster dataset which enables convenient browsing of materials in the GIS environment. To this end, based on UAV point clouds and calculated differences, rasters with a resolution of 5 cm were generated.

Considering the characteristics of the compared data, the proper interpretation of the obtained maps of differences should consist of locating local disturbances of differences and comparing them with the image on the high-resolution orthophotomap. It enables establishing what causes these variations or to note that they are related to measurement errors (e.g., people on the trail during measurements).

Below, several hypothetical study areas are presented. They illustrate possible interpretations of the acquired materials. In all of the examples shown below the same colour scale has been used (Figure 14b). It must be emphasised that data from ALS was treated as referential data and, therefore, negative values (blue) prove the decrease within the period between measurement series; whereas positive values (red) prove the increase (accumulation) or the occurrence of objects in the UAV data that do not occur in the ALS data. In the area presented in Figure 14a, on the basis of differences between DSM UAV and DSM ALS identified for the roofing of the mountain shelter (Figure 14c) a relative accuracy of both sets of data can be calculated. Assuming that between the measurement periods the roof of the shelter was not subject to any displacements, this accuracy oscillates between −20 cm and +10 cm.

Another aspect which can be observed in the discussed example are the places marked with a distinct orange-red colour which, in the case of the DSM UAV–DSM ALS comparison, prove the increase in vegetation. However, this increase can depend on the period in which measurements were taken while, in case of the DSM UAV–DTM ALS comparison, it proves the occurrence of medium and high vegetation (Figure 14d).

When analysing the trail area, one may notice immediately the trail bed and small relative differences of colours, which indicate erosion (more blue ones) or accumulation (light yellow-orange ones). However, in the above-mentioned area such changes are insignificant.

Another comparison area was the yellow trail in the area of Jaworzynka-Siodłowa Perć (Figures 15–18) which shows how the collected materials can be interpreted. In the images presented below, the following changes occurring on the trail can be observed:

- violet line: places of increased erosion;
- yellow line: places of accumulation; and
- green line: errors in measurement data resulting from the presence of people on the trail during the measurement.

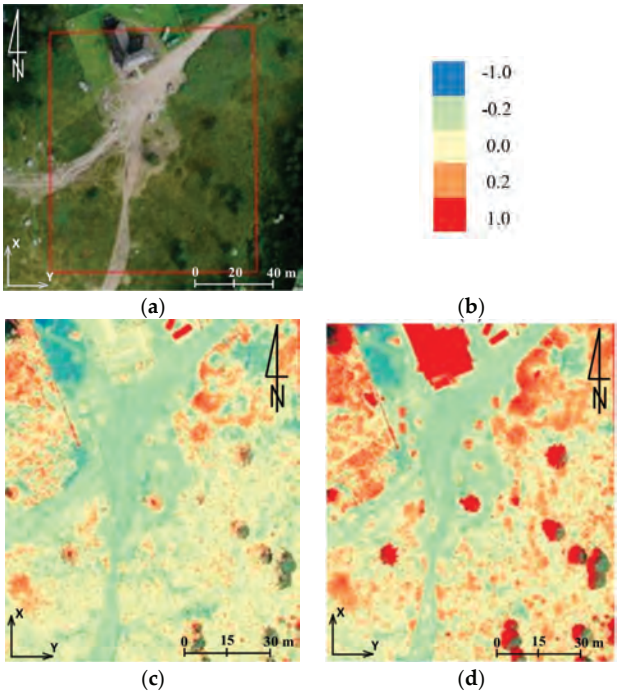


Figure 14. The mountain shelter in Hala Kondratowa: (**a**) the study area, (**b**) colour scale (the units given in metres), (**c**) the results of comparison of DSM UAV–DSM ALS, and (**d**) the results of comparison of DSM UAV–DTM ALS (source: authors' own study).

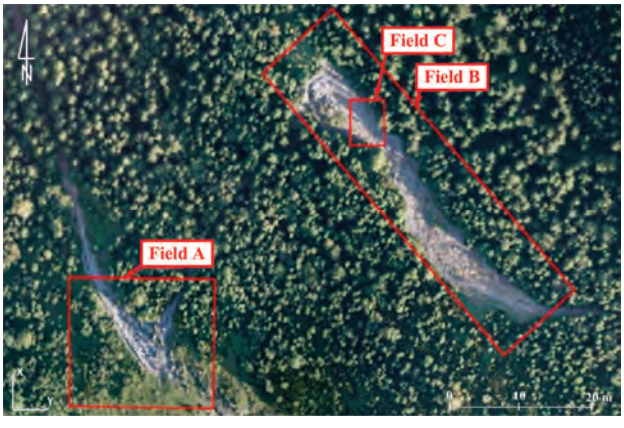
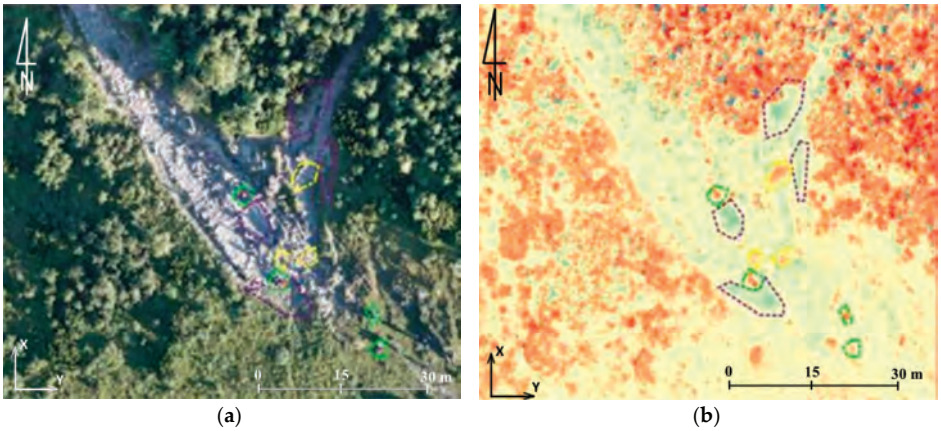
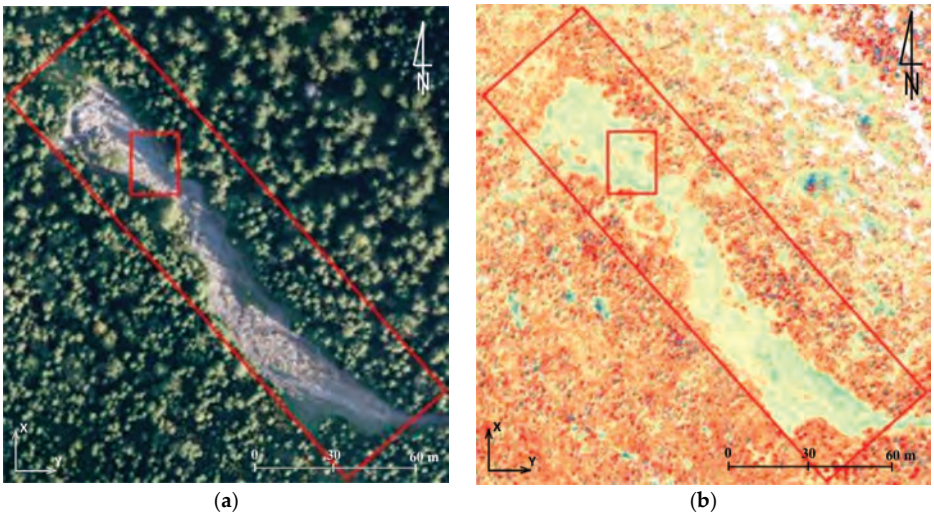


Figure 15. Test field: yellow trail in the area between Jaworzynka and Siodłowa Perć (source: authors' own study).



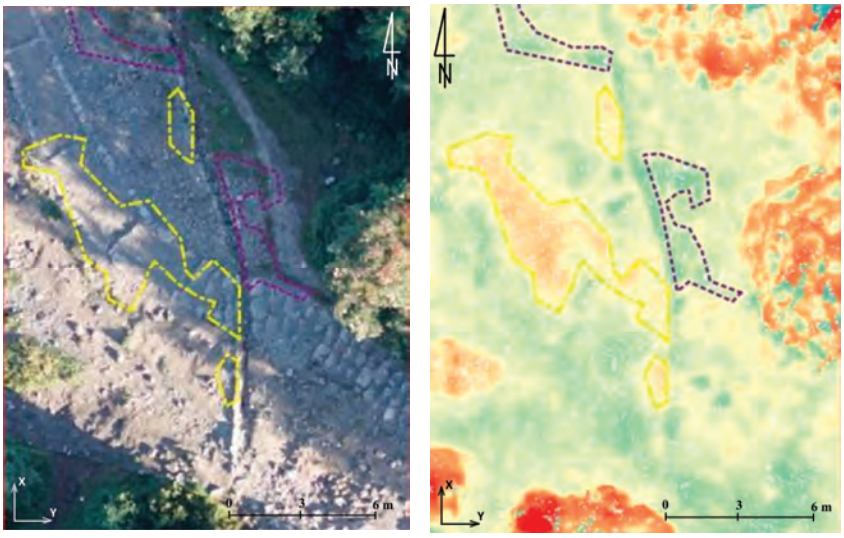
(**a**) (**b**)

Figure 16. Field A: (**a**) ortophotomap, and (**b**) the results of comparison of DSM UAV–DSM ALS (source: authors' own study).



(**a**) (**b**)

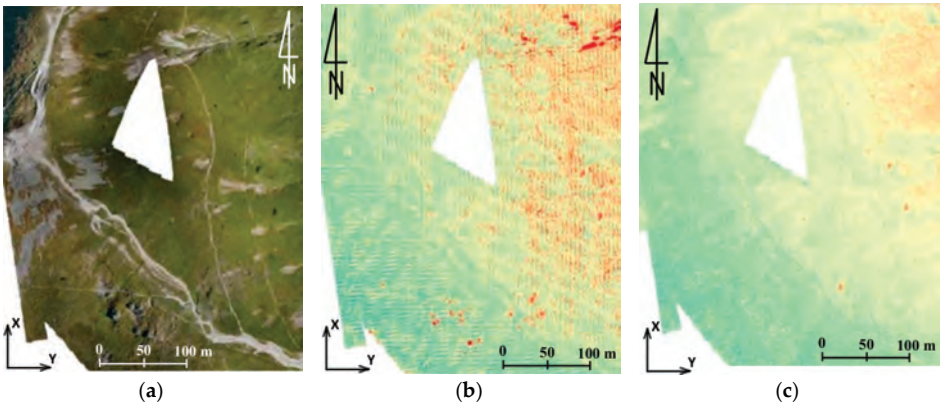
Figure 17. Field B: (**a**) ortophotomap, and (**b**) the results of comparison of DSM UAV–DSM ALS (source: authors' own study).



(**a**) (**b**)

Figure 18. Field C: (**a**) ortophotomap, and (**b**) the results of comparison of DSM UAV–DSM ALS (source: authors' own study).

Considering the sustainably lower resolution and pixel size in the archival data (35 cm), the analyses were conducted on the basis of clouds of points. The example presented below (Figure 19) illustrates the results obtained from the analysis based on the comparison of raster layers. On its basis, it can be ascertained that conducting such studies for the data of various resolution is not fully reliable and gives limited possibilities of interpretation (significant disturbances of results due to the resolution of data and the mountain slope inclination). Simultaneously, results based on the direct comparison of the clouds of points were presented for the same region. It has to be highlighted, though, that operation on the clouds of points requires increased processing power and can be impossible for vast areas.



(**a**) (**b**) (**c**)

Figure 19. Kopa Kondracka: (**a**) the comparison area on the orthophotomap, (**b**) the result of the comparisons based on raster layers, and (**c**) the results of comparisons based on clouds of points (source: authors' own study).

3.3. The Examples of Interpretation of Raster Images Obtained from a UAV

The photogrammetric methods are some of the most popular sources of data used in work related to forest administration. They started to be used in 1887 when the first photogrammetric photographs were taken with a camera lifted by a balloon. The photographs were later applied for settling a forest [70]. Around 1890 the first successful attempts of using terrestrial photogrammetry for mapping forest stands were made [71]. A significant breakthrough in the development of photogrammetry for the needs of the analysis of forest stands came along with the development of plane aviation. One of the first photogrammetric flights used by forestry took place in Bavaria in 1924 [72].

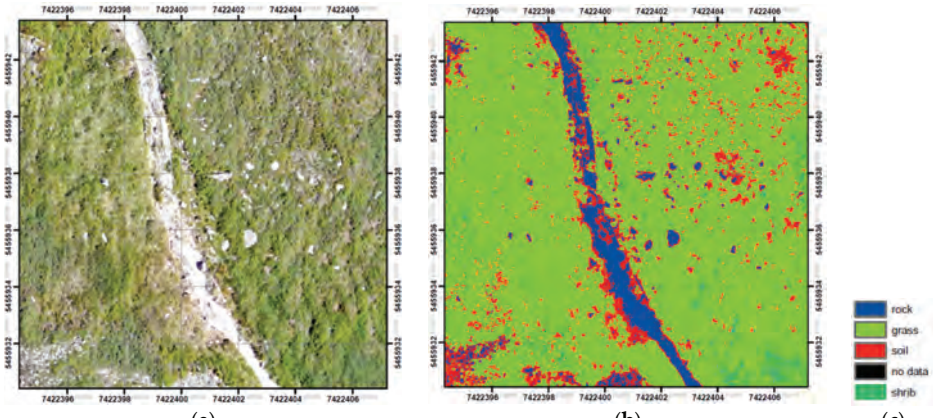
Currently, in the majority of cases, laser scanning measurements are performed concurrently with taking aerial photos. It is particularly crucial for densely wooded areas where the tree branches make it impossible to take photos from the level of the ground. Considering methods of obtaining more precise data, it is noted that ALS provides accurate information related to forests in vast areas. However, it is often insufficient to define the condition or features of individual trees located within the range of the laser beam. Therefore, TLS is a better method of estimating the condition of a forest stand, especially the density of afforestation [73].

It should also be mentioned that ALS is one of the most expensive methods of assessing the current state of land cover and land forms. Considering the data stored in the generally accessible public databases, it has to be remembered that the time in which the data was obtained might not correlate with the timing assumptions of the conducted project. Therefore, the potential of UAVs needs to be noted as these devices can supply information about the measured area in the chosen time, provided that weather conditions are favourable. An additional advantage is the scale of the photographs taken. One study [74] indicates that the scale of aerial photographs might not be sufficient for the needs of identifying individual tree tops. In the case of using UAV units this problem is minimised by the adequate selection of parameters of the mission which enables obtaining a high resolution of up to 1 cm.

The materials produced during the study can be used for analyses concerning the land cover, such as monitoring the anthropogenic changes of the environment or the extraction of tree tops [75]. Although it is common practice to use the infrared band during vegetation analysis, and it provides very satisfactory results [76,77], the objective of this article is to show the visible range data application in this case. The data presented in the form of a high-resolution orthophotomap allow for the supervised and unsupervised classification aimed at obtaining the land cover maps.

Methods of processing raster information are subject to modifications in relation to the clustering procedures and classifications compiled for satellite depiction [76]. Differences mostly in geometric, radiometric, and spectral resolution lead to a different approach to the processing of georasters. The supervised classification of the nearest neighbourhood was chosen considering the main aim of the classification, which is the separation of vegetated areas from bare ground and bedrock. However, the training sample needs to be selected every time when the areas are photographed in different times of a day or on different days. The changing morphology of the land forms is also significant [78].

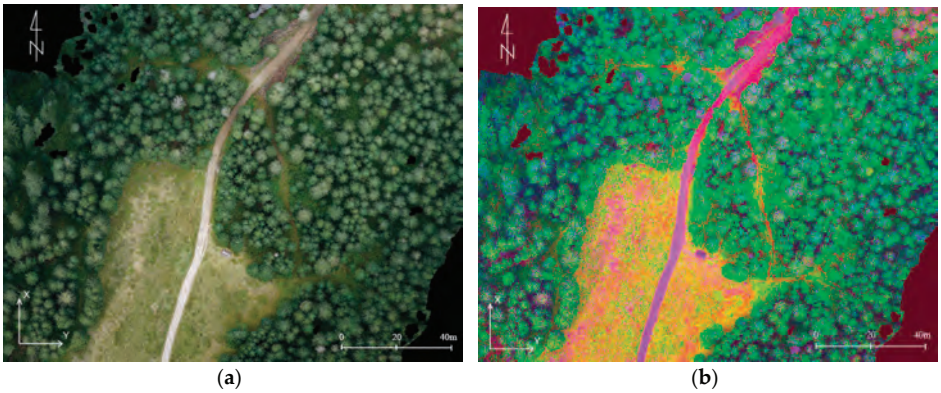
In the conducted analyses, training samples included from 2% to 5% of all pixels in the classes. The thematic map was created as a result of the division of the orthophotomap into five classes (rocks, grass and low flora, bushes, soils, lack of data). In the areas included in the research limestone and granite rocks are dominant, which are characterised by high spectral response within the range of visible light [79]. This simplifies the classification. By creating the land cover map (Figure 20) it is possible to conduct all of the quantitative analyses concerning relations between vegetation, bedrock, and soils. Defining the floral succession is possible due to the statistical comparison of the classified data. Another method which provides similar results is the calculation of greenness index (GI) [76]. In the studied case it was, however, less effective than the supervised classification.



(**a**) (**b**) (**c**)

Figure 20. (**a**) An extract from the orthophotomap including the area of Kopa Kondracka (the area undergoing the floral succession); (**b**) the land cover map created by means of supervised classification; (**c**) legend (source: authors' own study).

A method supporting the identification of objects is decorrelation stretching based on PCA (principal component analysis). The decorrelation stretch is a process to enhance the colour separation. The algorithm starts from each band centre and standardizes, then a covariance matrix is computed. To convert the data into the PCA space, it is necessary to multiply the RGB data by covariance matrix eigenvectors. To produce the decorrelation stretched image, the principal component image is modified by the linear transformation. Subsequently, then the inverse PCA transformation is conducted [80,81]. The application of the histogram decorrelation stretching facilitated the identification of trees that succumb to diseases or atrophy (Figure 21). The sole analysis of components allows the separation of tree tops. Decorrelation stretching enables efficient processing of large amounts of data and it does not take a great deal of time. The analysis of the DSM is a significant research tool. An analysis of slopes calculated on the basis of high-resolution DSM enables the detection of the hiking trail location which cannot be identified on a raw or pre-processed orthophotomap (Figure 22).



(**a**) (**b**)

Figure 21. (**a**) A fragment of the orthophotomap with the tourist trail between Dolina Chochołowska and Trzydniowiański Wierch; and (**b**) decorrelation stretching conducted on the basis of the analysis of the main components (source: authors' own study).

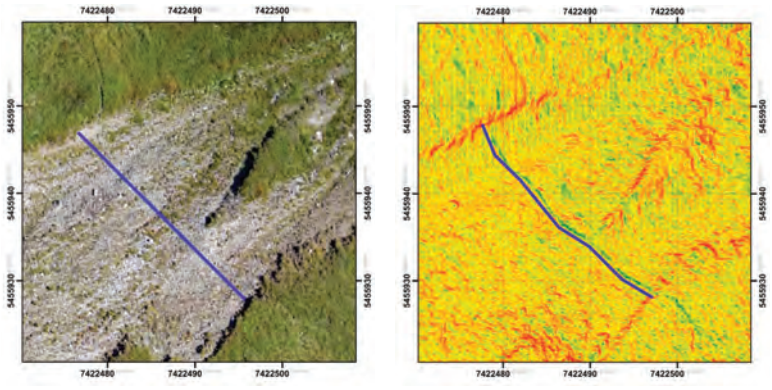


Figure 22. Rock debris: (**a**) orthoimage covering the invisible path; and (**b**) a map of slope allowing the detection of the trail (source: authors' own study).

4. Conclusions

This article describes, in detail, the process of spatial data acquisition by means of a UAV in difficult alpine areas. Moreover, it presents various possibilities of survey data analysis which are aimed at the assessment of erosion along hiking trails, as well as at studying land cover changes, particularly the floral succession in the immediate vicinity of hiking trails.

The area in which the research was conducted is demanding and imposes the necessity of precise planning of the UAV flights, as well as positioning control points with regard to a great number of factors. At this stage it is particularly important to focus on the land forms and forest cover.

Basic products resulting from UAV flights (point clouds, DTMs, orthophotomosaics) make a valuable source material for advanced spatial analyses. Their accuracy assessment parameters, calculated by comparing them with the results of measurements conducted with GNSS and TLS methods, are satisfactory. The methodology adopted enabled the surveying of trails in alpine areas not covered with vegetation by means of UAV imagery with an accuracy of about 50 mm. Although similar accuracies have been repeatedly confirmed in various published studies, they never referred to large, elongated objects with a very high inclination along their axis. It should be noted that the trails subjected to examination were from 1 m to 15 m wide and a vertical difference between the lowest and highest points in modelled terrain exceeded 1000 m.

Advanced analyses consisting in the comparison of data from various measurement periods were conducted in the next part of the study. It has to be emphasised that some of these analyses were made by comparing the obtained materials with historical data that features lower accuracy and spatial resolution. Despite this, the comparisons of point clouds facilitate the interpretation of changes occurring along the hiking trails. By analysing the generated differential maps, it is possible to define places with locally disturbed values of differences. However, such places should be additionally analysed on high-resolution orthophotomaps in order to eliminate factors which can distort the results of analyses, such as the increase in vegetation, or the presence of people on the trail during measurements. The application of various methods of digital analysis of images allows for automatic detection of phenomena and relations which are invisible for an observer, while being more efficient than manual data processing.

The presented studies do not exhaust the potential of analyses or their interpretations. Additionally, it is worth emphasizing that the most effective method would be to compare the data of similar accuracy and resolution parameters conducted in a similar period of the year. Therefore, it seems justifiable to conduct further measurements using the UAV in the following years. The data

obtained in such a way would certainly result in the increase in the reliability of defining even small (a few centimetres long) and short-term changes in the morphology of the terrain or forest succession. By means of this, it will be possible to define the effectiveness of protective actions taken by the TNP. This is because of the high density and high accuracy of the collected data. It also seems justified to examine the possibilities of limiting the number of necessary terrestrial control points (without losing the spatial accuracy) which, in such a differentiated and spatially difficult survey area, is going to allow for the optimisation of the data acquisition time.

Acknowledgments: The research and publication were completed as part of the project: "The assessment of the range and intensification of erosion along tourist trails in reference to the condition of these trails" financed by the forest fund of the State Forests National Forest Holding donated to the Tatra National Park (TNP) in 2016. Costs for open access publication are provided through the AGH University of Science and Technology via Fund No. 11.11.150.005 for statutory research.

Author Contributions: Experiment conception and design: Paweł Ćwiąkała. UAV flight planning: Paweł Ćwiąkała, Witold Niewiem, Michał Nędzka, and Paweł Wiącek. Field measurements and data processing: Paweł Ćwiąkała, Rafał Kocierz, Edyta Puniach, Michał Nędzka, Karolina Mamczarz, Witold Niewiem, and Paweł Wiącek. Manuscript drafting, analysis, and interpretation of results: Paweł Ćwiąkała, Rafał Kocierz, Edyta Puniach, Michał Nędzka, Karolina Mamczarz, Witold Niewiem, and Paweł Wiącek.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kotarba, A.; Kaszowski, L.; Krzemień, K. *High-Mountain Denudational System of the Polish Tatra Mountains*; Ossolineum: Wrocław, Poland, 1987.
2. Boltiziar, G.; Bugar, G.; Hresko, J.; Kohut, F. The dynamics of recent geomorphic processes in the Alpine Zone of the Tatra Mountains. *Geogr. Pol.* **2008**, *81*, 53–65.
3. Decaulne, A.; Saemundsson, T. Geomorphic evidence for present-day snow-avalanche and debris-flow impact in the Icelandic Westfjords. *Geomorphology* **2006**, *80*, 80–93. [CrossRef]
4. Rączkowska, Z.; Długosz, M.; Rojan, E. Geomorphological conditions of snow avalanches in the Tatra Mountains. *Zeitschrift für Geomorphol.* **2016**, *60*, 285–297. [CrossRef]
5. Crozier, M.J. Landslide geomorphology: An argument for recognition, with examples from New Zealand. *Geomorphology* **2009**, *120*, 3–15. [CrossRef]
6. Crosta, G.B.; Dal Negro, P.; Frattini, P. Soil slips and debris flows on terraced slopes. *Nat. Hazards Earth Syst. Sci.* **2003**, *3*, 31–42. [CrossRef]
7. Jacobson, R.B.; Miller, A.J.; Smith, J.A. The role of catastrophic geomorphic events in central Appalachian landscape evolution. *Geomorphology* **1989**, *2*, 257–284. [CrossRef]
8. Barnikel, F.; Becht, M. A historical analysis of hazardous events in the Alps—The case of Hindelang (Bavaria, Germany). *Nat. Hazards Earth Syst. Sci.* **2003**, *3*, 625–635. [CrossRef]
9. Rączkowska, Z. Recent geomorphic hazards in the Tatra Mountains. *Stud. Geomorphol. Carpatho-Balc.* **2006**, *11*, 45–60.
10. Cole, D.; Monz, C. Trampling disturbance of high-elevation vegetation, Wind River Mountains, Wyoming, USA. *Arct. Antarct. Alp. Res.* **2002**, *34*, 365–376. [CrossRef]
11. Bielawska, M.; Tsermegas, I. Anthropogenic degradation of the tourist trail in the Samaria Gorge (Western Creta). *Landf. Anal.* **2009**, *10*, 5–10.
12. Fidelus, J.; Płaczkowska, E. Contemporary transformation of slopes in the Western Tatras—As exemplified by fragments with tourist footpaths or else lacking human impact. *Pol. Geogr. Rev.* **2013**, *85*, 417–434. [CrossRef]
13. Buchwał, A.; Ćwiąkała, P.; Grudowicz, J. Utilization of geodetic survey to man-induced denudation assessment (Sokolica, Babia Góra, Flysch Carpathians). In *Interdisciplinary Topics in Mining and Geology: IX PhD Students' Scientific Conference*; Milczarek, W., Ed.; Oficyna Wydawnicza Politechniki Wrocławskiej: Wrocław, Poland, 2009; pp. 35–44.
14. Buchwał, A.; Fidelus, J.; Rogowski, M. Relief transformation along footpaths in The Riła, Piryn and Western Tatra Mountains. *Landf. Anal.* **2009**, *10*, 18–25.
15. Rechefort, R.M.; Swinney, D.D. Human Impact Survey in Mount Rainer National Park: Past, Present and Future. In *Proceedings of the Wilderness Science in a Time of Change Conference-Volume 5: Wilderness Ecosystems,*

Threats, and Management; Missoula, MT, USA, 23–27 May 1999, RMRS-P-15-VOL-5; Cole, D.N., McCool, S.F., Borrie, W.T., O'Loughlin, J., Eds.; U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station: Ogden, UT, USA, 2000; Volume 5, pp. 165–171.
16. Li, W.; Ge, X.; Liu, C. Hiking trails and tourism impact assessment in protected area: Jiuzhaigou Biosphere Reserve, China. *Environ. Monit. Assess.* **2005**, *108*, 279–293. [CrossRef] [PubMed]
17. Mihai, B.; Reynard, E.; Werren, G.; Savulescu, I.; Sandric, I.; Chitu, Z. Impacts of tourism on geomorphological processes in the Bucegi Mountains in Romania. *Geogr. Helv.* **2009**, *64*, 134–147. [CrossRef]
18. Arnberger, A.; Haider, W.; Brandenburg, C. Evaluating Visitor-Monitoring Techniques: A comparison of Counting and Video Observation Data. *Environ. Manag.* **2005**, *36*, 317–327. [CrossRef] [PubMed]
19. Wolf, I.D.; Hagenloh, G.; Croft, D.B. Visitor Monitoring along Roads and Hiking Trails: How to Determine Usage Levels in Tourist Sites. *Tour. Manag.* **2012**, *33*, 16–28. [CrossRef]
20. Rączkowska, Z. *Present-Day Periglacial Relief in High Mountains of Europe*; IGSO PAS: Warsaw, Poland, 2007.
21. Kaab, A. Monitoring high-mountain terrain deformation from repeated air- and spaceborne optical data: Examples using digital aerial imagery and ASTER data. *ISPRS J. Photogramm. Remote Sens.* **2002**, *57*, 39–52. [CrossRef]
22. Perroy, R.L.; Bookhagen, B.; Asner, G.P.; Chadwick, O.A. Comparison of gully erosion estimates using airborne and ground-based LiDAR on Santa Cruz Island, California. *Geomorphology* **2010**, *118*, 288–300. [CrossRef]
23. Barneveld, R.J.; Seeger, M.; Maalen-Johansen, I. Assessment of terrestrial laser scanning technology for obtaining high-resolution DEMs of soils. *Earth Surf. Process. Landf.* **2013**, *38*, 90–94. [CrossRef]
24. Stumpf, A.; Malet, J.-P.; Allemand, P.; Pierrot-Deseilligny, M.; Skupinski, G. Ground-based multi-view photogrammetry for the monitoring of landslide deformation and erosion. *Geomorphology* **2015**, *231*, 130–145. [CrossRef]
25. Perez, E.; Garcia, P. Monitoring soil erosion by raster images: From aerial photographs to drone taken pictures. *Eur. J. Geogr.* **2017**, *8*, 116–128.
26. Berni, J.A.J.; Zarco-Tejada, P.J.; Suarez, L.; Fereres, E. Thermal and Narrowband Multispectral Remote Sensing for Vegetation Monitoring From an Unmanned Aerial Vehicle. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 722–738. [CrossRef]
27. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR System with Application to Forest Inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [CrossRef]
28. D'Oleire-Oltmanns, S.; Marzolff, I.; Peter, K.D.; Ries, J.B. Unmanned Aerial Vehicle (UAV) for Monitoring Soil Erosion in Morocco. *Remote Sens.* **2012**, *4*, 3390–3416. [CrossRef]
29. Eltner, A.; Mulsow, C.; Maas, H.-G. Quantitative measurement of soil erosion from TLS and UAV data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-1/W2*, 119–124. [CrossRef]
30. Hackney, C.; Clayton, A. Unmanned Aerial Vehicles (UAVs) and their application in geomorphic mapping. In *Geomorphological Techniques*; Clarke, L.E., Nield, J.M., Eds.; British Society for Geomorphology: London, UK, 2015; pp. 1–12.
31. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [CrossRef]
32. Benassi, F.; Dall'Asta, E.; Diotri, F.; Forlani, G.; Morra di Cella, U.; Roncella, R.; Santise, M. Testing Accuracy and Repeatability of UAV Blocks Oriented with GNSS-Supported Aerial Triangulation. *Remote Sens.* **2017**, *9*, 172. [CrossRef]
33. Gruszczyński, W.; Matwij, W.; Ćwiąkała, P. Comparison of low-altitude UAV photogrammetry with terrestrial laser scanning as data-source methods for terrain covered in low vegetation. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 168–179. [CrossRef]
34. Tahar, K.N.; Ahmad, A.; Wan Mohd Akib, W.A.A. Unmanned aerial vehicle technology for low cost landslide mapping. In Proceedings of the 11th South East Asian Survey Congress and 13th International Surveyors' Congress, PWTC, Kuala Lumpur, Malaysia, 22–24 June 2011.
35. Mancini, F.; Dubbini, M.; Gattell, M.; Stecchi, F.; Fabbri, S.; Gabbianelli, G. Using Unmanned Aerial Vehicles (UAV) for High-Resolution Reconstruction of Topography: The Structure from Motion Approach on Coastal Environments. *Remote Sens.* **2013**, *5*, 6880–6898. [CrossRef]
36. Uysal, M.; Toprak, A.S.; Polat, N. DEM generation with UAV Photogrammetry and accuracy analysis in Sahitler hill. *Measurement* **2015**, *73*, 539–543. [CrossRef]

37. Barry, P.; Coakley, R. Accuracy of UAV photogrammetry compared with network RTK GPS. *Int. Arch. Photogramm. Remote Sens.* **2013**, *XL-1/W2*, 27–31.
38. Lucieer, A.; de Jong, S.M.; Turner, D. Mapping landslide displacements using Structure from Motion (SfM) and image correlation of multi-temporal UAV photography. *Prog. Phys. Geogr.* **2014**, *38*, 97–116. [CrossRef]
39. Fernández, T.; Pérez, J.L.; Cardenal, F.J.; López, A.; Gómez, J.M.; Colomo, C.; Delgado, J.; Sánchez, M. Use of a light UAV and photogrammetric techniques to study the evolution of a landslide in Jaen (Southern Spain). *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *XL-3/W3*, 241–248. [CrossRef]
40. Smith, M.W.; Vericat, D. From experimental plots to experimental landscapes: Topography, erosion and deposition in sub-humid badlands from Structure-from-Motion photogrammetry. *Earth Surf. Process. Landf.* **2015**, *40*, 1656–1671. [CrossRef]
41. Hsieh, Y.-C.; Chan, Y.-C.; Hu, J.-C. Digital Elevation Model Differencing and Error Estimation from Multiple Sources: A Case Study from the Meiyuan Shan Landslide in Taiwan. *Remote Sens.* **2016**, *8*, 199. [CrossRef]
42. Neugirg, F.; Kaiser, A.; Schmidt, J.; Becht, M.; Haas, M. Quantification, analysis and modelling of soil erosion on steep slopes using LiDAR and UAV photographs. *Proc. Int. Assoc. Hydrol. Sci.* **2015**, *367*, 51–58. [CrossRef]
43. Gillan, J.K.; Karl, J.W.; Elaksher, A.; Duniway, M.C. Fine-Resolution Repeat Topographic Surveying of Dryland Landscapes Using UAS-Based Structure-from-Motion Photogrammetry: Assessing Accuracy and Precision against Traditional Ground-Based Erosion Measurements. *Remote Sens.* **2017**, *9*, 437. [CrossRef]
44. Dewez, T.J.B.; Leroux, J.; Morelli, S. Cliff collapse hazard from repeated multicopter UAV acqusitions: Returns on experience. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B5*, 805–811. [CrossRef]
45. Lu, C.-H. Applying UAV and photogrammetry to monitor the morphological changes along the beach in Penghu islands. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B8*, 1153–1156. [CrossRef]
46. Papakonstantinou, A.; Topouzelis, K.; Pavlogeorgatos, G. Coastline Zones Identification and 3D Coastal Mapping Using UAV Spatial Data. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 75. [CrossRef]
47. Venturi, S.; di Francesco, S.; Materazzi, F.; Manciola, P. Unmanned aerial vehicles and Geographical Information System integrated analysis of vegetation in Trasimeno Lake, Italy. *Lakes Reserv. Res. Manag.* **2016**, *21*, 5–19. [CrossRef]
48. Honkavaara, E.; Saari, H.; Kaivosoja, J.; Pölönen, I.; Hakala, T.; Litkey, P.; Mäkynen, J.; Pesonen, L. Processing and Assessment of Spectrometric, Stereoscopic Imagery Collected Using a Lightweight UAV Spectral Camera for Precision Agriculture. *Remote Sens.* **2013**, *5*, 5006–5039. [CrossRef]
49. Cao, C.; Liu, D.; Singh, R.P.; Zheng, S.; Tian, R.; Tian, H. Integrated detection and analysis of earthquake disaster information using airborne data. *Geomat. Nat. Hazards Risk* **2016**, *7*, 1099–1128. [CrossRef]
50. Dominici, D.; Alicandro, M.; Massimi, V. UAV photogrammetry in the post-earthquake scenario: Case studies in L'Aquila. *Geomat. Nat. Hazards Risk* **2016**, *8*, 1–17. [CrossRef]
51. Piras, M.; Taddia, G.; Forno, M.G.; Gattiglio, M.; Aicardi, I.; Dabove, P.; Lo Russo, S.; Lingua, A. Detailed geological mapping in mountain areas using an unmanned aerial vehicle: Application to the Rodoretto Valley, NW Italian Alps. *Nat. Hazards Risk* **2017**, *8*, 137–149. [CrossRef]
52. Ridolfi, E.; Buffi, G.; Venturi, S.; Manciola, P. Accuracy Analysis of a Dam Model from Drone Surveys. *Sensors* **2017**, *17*, 1777. [CrossRef] [PubMed]
53. Neitzel, F.; Klonowski, J. Mobile 3D mapping with a low-cost UAV system. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *38*, 1–6. [CrossRef]
54. Vallet, J.; Panissod, F.; Strecha, C. Photogrammtric Performance of an Ultralightweight Swinglet UAV. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *XXXVIII-1/C22*, 253–258. [CrossRef]
55. Jaud, M.; Passot, S.; Le Bivic, R.; Delacourt, C.; Grandjean, P.; Le Dantec, N. Assessing the Accuracy of High Resolution Digital Surface Models Computed by PhotoScan® and MicMac® in Sub-Optimal Survey Conditions. *Remote Sens.* **2016**, *8*, 465. [CrossRef]
56. Tatra National Park, Poland. Available online: http://tpn.pl (accessed on 2 March 2017).
57. Pociask-Karteczka, J.; Baścik, M.; Czubernat, S. Ruch turystyczny w Tatrzańskim Parku Narodowym w latach 1993–2005. In *Studia nad Turystyką. Tradycje, Stan Obecny i Perspektywy Badawcze. Geograficzne, Społeczne i Ekonomiczne Aspekty Turystyki*; Kurek, W., Mika, M., Eds.; Instytut Geografii i Gospodarki Przestrzennej UJ: Kraków, Poland, 2007; pp. 271–280.
58. Fidelus, J. The Role of Pedestrian Tourist Traffic in the Landform Transformation in the Area of Tatra National Park. *Prace Geograficzne, Instytut Geografii i Gospodarki Przestrzennej Uniwersytetu Jagiellońskiego* **2008**, *120*, 19–29.

59. Gorczyca, E.; Krzemień, K. Wpływ turystyki pieszej na przekształcanie rzeźby Tatr. In *Geografia i Sacrum*; Domański, B., Skiba, S., Eds.; IGiGP: Kraków, Poland, 2005; pp. 77–85.
60. Google Maps. Available online: https://www.google.pl/maps (accessed on 5 March 2017).
61. Geoportal 2. Available online: http://mapy.geoportal.gov.pl/imap/?gpmap=gp0&actions=acShowWgButtonPanel_kraj_ORTO (accessed on 5 March 2017).
62. DJI. Available online: http://www.dji.com (accessed on 5 March 2017).
63. Mesas-Carrascosa, F.J.; Rumbao, I.C.; Berrocal, J.A.B.; Porras, A.G.-F. Positional Quality Assessment of Orthophotos Obtained from Sensors Onboard Multi-Rotor UAV Platforms. *Sensors* **2014**, *14*, 22394–22407. [CrossRef] [PubMed]
64. Wdowiak, P. Analysis of Distribution of Ground Control Points during a UAV Flights in Mountainous Areas. Master's Thesis, AGH University of Science and Technology, Kraków, Poland, 2016.
65. Turner, D.; Lucieer, A.; de Jong, S.M. Time Series Analysis of Landslide Dynamics Using an Unmanned Aerial Vehicle (UAV). *Remote Sens.* **2015**, *7*, 1736–1757. [CrossRef]
66. Niethammer, U.; James, M.R.; Rothmund, S.; Travelletti, J.; Joswig, M. UAV-based remote sensing of the Super-Sauze landslide: Evaluation and results. *Eng. Geol.* **2012**, *128*, 2–11. [CrossRef]
67. Seier, G.; Stangle, J.; Schöttl, S.; Sulzer, W.; Sass, O. UAV and TLS for monitoring a creek in an alpine environment, Styria, Austria. *Int. J. Remote Sens.* **2017**, *38*, 2903–2920. [CrossRef]
68. Castillo, C.; Péreza, R.; Jamesb, M.R.; Quintonb, J.N.; Taguasc, E.V.; Gómez, J.A. Comparing the Accuracy of Several Field Methods for Measuring Gully Erosion. *Soil Sci. Soc. Am. J.* **2011**, *76*, 1319–1332. [CrossRef]
69. Pierzchała, M.; Talbot, B.; Astrup, R. Estimating Soil Displacement from Timber Extraction Trails in Steep Terrain: Application of an Unmanned Aircraft for 3D Modelling. *Forests* **2014**, *5*, 1212–1223. [CrossRef]
70. Hildebrandt, G. 100 Jahre forstliche Luftbildaufnahme—Zwei Dokumenta aus den Anfängen der forstlichen Luftbildinterpretation. *Bildmess. Luftbildwes.* **1987**, *55*, 221–224.
71. Wodera, H. Die Holzmassenermittlung nach Luftbildern. *Allgemein. Forst- Holzwirtsch. Zitung* **1948**, *59*, 109–112.
72. Spellmann, H. Zustandserfassung in Kiefernbeständen mit Hilfe des Luftbildes. Ph.D. Thesis, Georg-August-Universität Göttingen, Göttingen, Germany, 1984.
73. Wencel, A.; Wężyk, P.; Zasada, M. Możliwości zastosowania naziemnego skaningu laserowego w leśnictwie. In *Techniki Geomatyczne w Inwentaryzacji Lasu—Potrzeby i Możliwości*; Zawiła-Niedźwiecki, T., Zasada, M., Eds.; Wydawnictwo SGGW: Warsaw, Poland, 2008; pp. 77–89.
74. Będkowski, K. *Fotogrametryczna Metoda Oceny Stanu i Zmian Wysokościowej Struktury Warstwy Koron w Drzewostanach*, 1st ed.; Wydawnictwo SGGW: Warsaw, Poland, 2005; pp. 1–208.
75. Wallace, L. Assesing the Stability of Canopy Maps Produced from UAV-LIDAR Data. In Proceedings of the 2013 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Melbourne, VIC, Australia, 21–26 July 2013; pp. 3879–3882. [CrossRef]
76. Hunt, R.E.; Daughtry, C.S.T.; Mirsky, S.B.; Hively, W.D. Remote Sensing With Simulated Unmanned Aircraft Imagery for Precision Agriculture Applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 4566–4571. [CrossRef]
77. Ghazal, M.; Al Khalil, Y.; Hajjdiab, H. UAV-based remote sensing for vegetation cover estimation using NDVI imagery and level sets method. In Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, Abu Dhabi, UAE, 7–10 December 2015; Available online: http://ieeexplore.ieee.org/document/7394354/ (accessed on 5 November 2017). [CrossRef]
78. Dunford, R.; Michel, K.; Gagnage, M.; Piegay, H.; Tremelo, M.-L. Potential and constraints of Unmanned Aerial Vehicle technology for the characterization of Mediterranean riparian forest. *Int. J. Remote Sens.* **2009**, *30*, 4915–4935. [CrossRef]
79. Geological Map of the Polish Tatras. Available online: http://mbc.malopolska.pl/dlibra/publication?id=92274&tab=3 (accessed on 7 July 2017).
80. Gillespie, A.R.; Kahle, A.B.; Walker, R.E. Color enhancement of highly correlated images. Decorrelation and HSI contrast stretches. *Remote Sens. Environ.* **1986**, *20*, 209–235. [CrossRef]
81. Campbell, N.A. The decorrelation stretch transformation. *Int. J. Remote Sens.* **1996**, *17*, 1939–1949. [CrossRef]

sensors

MDPI

Article

Seamline Determination Based on PKGC Segmentation for Remote Sensing Image Mosaicking

Qiang Dong [1,2] and Jinghong Liu [1,*]

[1] Chinese Academy of Science, Changchun Institute of Optics Fine Mechanics and Physics, #3888 Dongnanhu Road, Changchun 130033, China; dongqiang13@mails.ucas.ac.cn
[2] University of Chinese Academy of Science, #19 Yuquan Road, Beijing 100049, China
* Correspondence: liu1577@126.com; Tel.: +86-186-2693-5658

Received: 12 June 2017; Accepted: 24 July 2017; Published: 27 July 2017

Abstract: This paper presents a novel method of seamline determination for remote sensing image mosaicking. A two-level optimization strategy is applied to determine the seamline. Object-level optimization is executed firstly. Background regions (BRs) and obvious regions (ORs) are extracted based on the results of parametric kernel graph cuts (PKGC) segmentation. The global cost map which consists of color difference, a multi-scale morphological gradient (MSMG) constraint, and texture difference is weighted by BRs. Finally, the seamline is determined in the weighted cost from the start point to the end point. Dijkstra's shortest path algorithm is adopted for pixel-level optimization to determine the positions of seamline. Meanwhile, a new seamline optimization strategy is proposed for image mosaicking with multi-image overlapping regions. The experimental results show the better performance than the conventional method based on mean-shift segmentation. Seamlines based on the proposed method bypass the obvious objects and take less time in execution. This new method is efficient and superior for seamline determination in remote sensing image mosaicking.

Keywords: remote sensing; image mosaicking; seamline detection; image segmentation; graph cuts; multi-scale morphological gradient (MSMG)

1. Introduction

A large remote sensing image with a wide field of view and high resolution is often required for many applications, such as map-making, disaster management, and military reconnaissance [1,2]. However, the wide field of view and high resolution cannot be captured at the same time because of the limit of the sensor size of the aerial camera. Image mosaicking was used to solve this problem effectively. Image mosaicking is the instrument used to gain a remote sensing image that meets the requirements for both the field of view and resolution using a series of images with overlapping areas. Ideally, the transition in the overlapping region from one image to another should be invisible. Realistically, due to different illumination, exposure parameter settings, depth of field differences, shooting field changes, and other reasons, the overlapping area will inevitably have uneven brightness and geometric misalignment. The problem of brightness unevenness in the mosaicking image can be effectively solved after a series of color corrections, smoothing [3–5], and image fusion [6–8]. However, the apparent parallax caused by geometric misalignment cannot be solved by the above method. An effective way to solve this problem is to find an optimal seamline in the overlapping region, then take image content respectively on each side. The optimal seamline detection is to find the minimal difference between the two images in the overlapping area, e.g., the intensity difference, gradient difference, and color difference. The geometric misalignment can be eliminated by the above process, and two images can be mosaicked as a large field of view image without apparent parallax.

Since Milgram [9] proposed computer mosaicking technology, finding the optimal seamline to improve the quality of mosaicking image has become an important direction for many scholars to study. Many methods have been proposed to determine the location of the seamline. Yang et al. [10] smoothed the artificial edge effectively through the two-dimensional seam point search strategy. Afek and Brand [11] completed the geometric correction of the image by adding the feature-matching algorithm to the optimal seamline detection process. Fernandez et al. [12] proposed a bottleneck shortest path algorithm to realize the drawing of the aerial photography map by using the absolute value of pixel differences in the overlapping region. Fernandez and Marti [13] subsequently optimized the bottleneck shortest path with a greedy random adaptive search procedure (GRASP) to obtain a superior image. Kerschner [14] constructed an energy function based on color and texture similarity, and then used the twin snakes detection algorithm to detect the position of the optimal seamline. One snake is a profile that moves inside the image, changing its shape until its own energy function is minimal [15]. The twin snakes detection algorithm created start points for two contours on the opposite sides of the overlapping area, which passed through the overlapping area and constantly changed the shape until a new one was synthesized. Soille [16] proposed a mosaicking algorithm based on the morphology and marker control segmentation program, which rendered the seamline along the highlight structure to reduce the visibility of the joints in mosaicking image. Chon et al. [17] used the normalized cross correlation (NCC) to construct a new object function that could effectively evaluate mismatching between two input images. This model determined the horizontal expectations of the largest difference in overlapping region, and then detected the position of the best seamline using Dijkstra's algorithm. It could contain fewer high-energy pixels in a long seamline. Yu et al. [18] constructed a combined energy function with a combination of multiple image similarity measures, including pixel-based similarity (color, edge, texture), region-based similarity (saliency) and location constraint, and then determined the seamline by dynamic programming (DP) [19]. Li et al. [20] extracted the histogram of oriented gradient (HOG) feature to construct an energy function, then detected the seamline by graph cuts.

Many of the methods described above considered seamline detection as an energy optimization, characterizing the difference between input images in overlapping regions by constructing a special energy function (cost). Image information should be contained in the cost comprehensively, e.g., the color, the gradient, the texture feature and the edge strength, and then find the optimal solution through different optimization algorithms, such as dynamic programming, Dijkstra's algorithm [21], snake model [15], and graph cuts [22,23]. The core issue is how to avoid the seamline passing through the obvious objects in overlapping area. Owing to the differences of input images, there will be pixel misalignment and color differences near the seamline in the mosaicking image when the seamline passes through the obvious objects. This manifests as the obvious "seam" which can compromise the integrity of objects. Therefore, the seamline should pass through smooth texture region, such as roads, rivers, grass, i.e., background regions, bypassing the obvious objects, such as buildings or cars. It is beneficial to avoid the seamline from passing through the obvious objects to extract the pixel position of objects accurately. Namely, segmentation of the input image is necessary.

Pan et al. [24] proposed an urban image mosaicking method based on segmentation, which determined preferred regions by the mean-shift (MS) algorithm and calculated the color difference as the cost. Firstly, the input images were segmented by the mean-shift algorithm and then the span of every segmented region was computed. Then preferred regions were determined based on the given threshold of the span which was consistent with the size of the largest obvious object in the overlapping regions. Ideally, most of the obvious objects are smaller than the grass or street areas, which are segmented into smaller regions, so that the selected preferred regions do not contain obvious objects, such as buildings, cars, etc. Under the realistic condition, there are some difficulties in the implementation of the method. The result of segmentation strongly depends on the parameters of the mean-shift algorithm, such as bandwidth. In addition, the threshold of the span is related to the size of the objects of the overlapping region, which cannot be completed automatically and needs to be

given artificially for each image. Furthermore, the size of the obvious object is not always smaller than the street or the grass, so we cannot accurately extract the preferred regions without obvious objects. This method is too simple to construct the cost map, only considering the color difference, but not the other aspects, e.g., texture or edge. Saito et al. [25] proposed a seamline determination method based on semantic segmentation by training a convolution neural network (CNN). The model can effectively avoid the buildings and obtain a good mosaicking. However, the model needs large-scale image datasets for training, and it is very time consuming.

In this paper, a novel method for seamline determination is presented based on a parametric kernel graph cuts (PKGC) segmentation algorithm [26] for remote sensing image mosaicking. We determine the seamline via a two-level optimization strategy. Object-level optimization is executed firstly. The cost map is weighted by the background regions (BRs) determined by the results of the PKGC segmentation. The cost map contains the color difference, gradient constraint, and texture difference. Then the pixel-level optimization by Dijkstra's algorithm is carried out to determine the seamline in the weighted cost. This paper is organized as follows: Section 2 describes the novel method of this paper. Section 3 presents the experimental results and the discussion. Section 4 summarizes this paper.

2. Methods

Considering the integrity of the mosaicking image, the seamline should pass through flat areas of texture, such as rivers and meadows, bypassing the obvious objects, such as buildings. Therefore, we can set the background regions (BRs) and obvious regions (ORs) with an image segmentation method. The seamline prefers to pass through BRs and round ORs.

The corresponding relation of the overlap area of the input images will be determined after pretreatment and registration for input images [27]. Then the seamline can be detected in the overlapping region. Firstly, we determine BRs based on the segmentation by the PKGC algorithm. Then we construct the global cost considering the color difference, the multi-scale morphological gradient (MSMG) constraint, and texture difference. Finally, we determine the pixel position of the seamline by Dijkstra's algorithm based on the weighted cost map. Figure 1 shows the flowchart of this method.

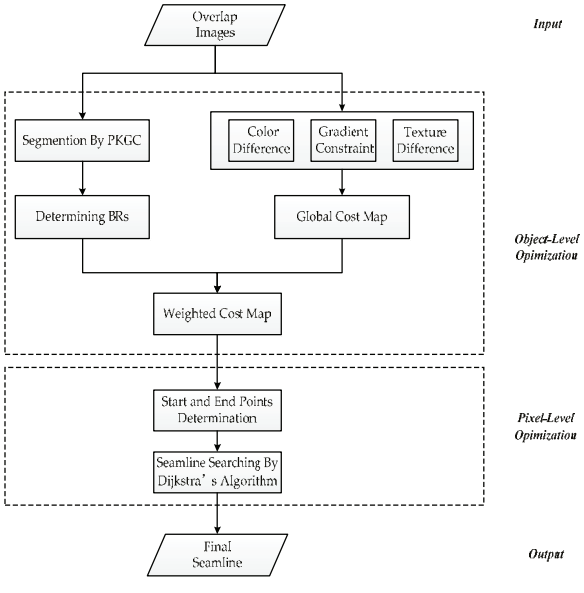


Figure 1. Flowchart of seamline determination.

2.1. BR Determination

Segmentation by PKGC: The PKGC algorithm borrows from the idea of kernel k-means, and a kernel function ϕ is introduced in the segmentation. The image data is implicitly mapped into the high-dimensional feature space. This makes it possible to highlight the slight difference between the image data, so that the original data, which cannot be divided, is linearly separable (or approximately linear), as Figure 2 shows. This is helpful to construct the piecewise constant model (PCM) containing only the dot product operation and the unsupervised segmentation function. By Mercer's theorem, any continuous, symmetric and positive semi-definite kernel function can be expressed as a dot product of higher dimensional space without knowing the mapping.

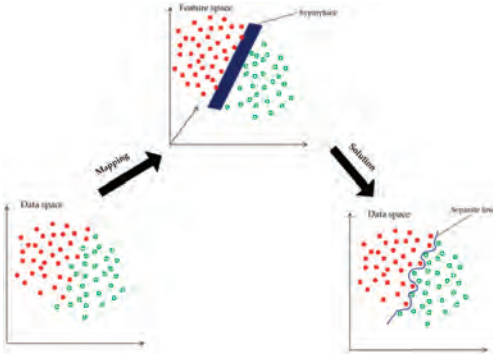


Figure 2. Illustration of nonlinear data separation by mapping to a higher dimensional feature space.

The kernel graph cuts model needs to set N_{seg} marks for the N_{seg} regions firstly, and then every pixel of the image is assigned a mask. Finally, determine which region that each pixel belongs to according to the mark. The segmentation by graph cuts method in the kernel-induced space is transformed into finding a mark allocation scheme to minimize the energy function. The energy function contains two items: the first is the kernel-induced distance term, which is used to estimate the deviation between the mapped data in each region of the PCM model, and the second is the smoothing term which can smooth adjacent pixels. The energy function is as follows:

$$E_K(\{\mu_l\},\lambda) = \sum_{l\in L}\sum_{p\in R_l} J_K(\phi(\mu_l)-\phi(I_p))^2 + \alpha \sum_{\{p,q\}\in N} r(\lambda(p),\lambda(q)) \quad (1)$$

where E_K is the non-Euclidean distance between the region's parameter and the observations. μ_l is the PCM parameter of region R_l, which can be acquired by the k-means clustering algorithm. λ is the indexing function assigning a label to the pixel. l is the label of the segmentation region. L is the number of segmentation regions. R_l means the region of label l. ϕ is the nonlinear mapping from image space I to the higher dimensional feature space J. The commonly-used function is the radial basis function (RBF), $K(y,z) = \exp(-||y-z||_2/\sigma^2)$. p and q represent two adjacent pixels. $r(\lambda(p),\lambda(q))$ is the smoothing function, $r(\lambda(p),\lambda(q)) = \min\left\{c, \left|\mu_{\lambda(p)} - \mu_{\lambda(q)}\right|_2\right\}$, where c is constant. α is a non-negative factor used to weigh the two terms. Then introducing the kernel function:

$$K(y,z) = \phi(y)^T \bullet \phi(z), \quad \forall (y,z) \in I^2 \quad (2)$$

where "•" is the dot product in the feature space. The non-Euclidean distance of the feature space can be expressed as follows:

$$\begin{aligned} J_K(I_p, \mu) &= \left|\left|\phi(I_p) - \phi(\mu)\right|\right|^2 \\ &= K(I_p, I_p) + K(\mu, \mu) - 2K(I_p, \mu), \quad \mu \in \{\mu_l\}_{1 \le l \le N_{reg}} \end{aligned} \tag{3}$$

Then, substitution of Equation (3) into Equation (1) results in the expression:

$$E_K(\{\mu_l\}, \lambda) = \sum_{l \in L} \sum_{p \in R_l} J_K(I_p, \mu_l)^2 + \alpha \sum_{\{p,q\} \in N} r(\lambda(p), \lambda(q)) \tag{4}$$

Clearly, the solution of Equation (4) depends only on the regional parameters $\{\mu_l\}_{l=1,2,\cdots,N_{reg}}$ and the indexing function λ. The iterative two-step optimization method is used to minimize the function. Firstly, fix the labeling results (image segmentation) and update the current statistics region parameter. Optimize E_K for the given kernel function. Then search for optimal labeling results (image segmentation) using the graph cuts iteration base on the region parameter obtained above.

Determining BRs: The image can be segmented into the foreground obvious objects regions and the background regions. Two input images are segmented using the PKGC algorithm independently. The BRs determine the intersection of the segmentation results of the left image and the right image. The remaining regions of overlapping area are regarded as ORs, i.e., the union of the segmentation results.

2.2. Constructing the Energy Function

We consider the following steps to construct a more accurate energy function. Firstly, calculate the global energy function $C(x, y)$ and then obtain the weighted cost $D(x, y)$ weighted by the BRs. Let the compound image I be the overlapping region of the input left image I_1 and the right image I_2. The global energy function $C(x, y)$ of pixel (x, y) contains several aspects as follows:

2.2.1. Color Difference

Color difference is the most common energy function in seamline detection for image mosaicking. We calculate the difference in the HSV (hue, saturation, value) color space instead of the common RGB space. The color difference $\delta_c(x, y)$ is defined as follows:

$$\delta_c(x, y) = \omega |V_1(x, y) - V_2(x, y)| + (1 - \omega) |(S_1(x, y) - S_2(x, y))| \tag{5}$$

where $V_1(x, y)$ and $S_1(x, y)$ is the intensity values of pixel (x, y) in the V and S channels of the HSV space of the left image I_1. Weight coefficient $\omega \in [0, 1]$ is used to balance the effects of V and S channels, and equals 0.95 in this paper. Similarly, $V_2(x, y)$ and $S_2(x, y)$ express analogous meaning.

2.2.2. MSMG Constraint

In the image morphological processing, the structural element is a common tool for image feature extraction and the structural element with different shapes can extract different image features. Furthermore, changing the size of the element can be extended to the multi-scale space [28]. The gradient can represent the sharpness of an image [29,30]. The comprehensive gradient feature will be extracted by the multi-scale morphological gradient operator [31,32]. In this paper, we propose a novel multi-angle linear structural element to extract the multi-scale morphological gradient (MSMG), extracting the multi-scale gradient of each angle and then combining them into the multi-scale morphological gradient, as Figure 3 shows. The details of this method are given as follows:

Firstly, construct the multi-scale element:

$$SE_j^\theta = \underbrace{SE_1^\theta \oplus SE_1^\theta \cdots \oplus SE_1^\theta}_{j},\ j \in \{1,2,\cdots,n\},\ \theta \in \{\theta_1,\theta_2,\cdots,\theta_m\} \tag{6}$$

where SE_1^θ is the basic linear structural element with length l and angle θ, and n is the sum of scales.

Then the gradient feature G_j^θ with scale j and angle θ will be extracted by the above operator. Let image $I = f(x,y)$.

$$G_j^\theta(x,y) = f(x,y) \oplus SE_j^\theta - f(x,y) \ominus SE_j^\theta \tag{7}$$

where $\oplus$ and $\ominus$ is the morphological dilation and erosion respectively, which are defined as:

$$f \oplus SE = \max_{(u,v)}(f(x-u,y-v) + SE(u,v)) \tag{8}$$

$$f \ominus SE = \min_{(u,v)}(f(x+u,y+v) - SE(u,v)) \tag{9}$$

where (x,y) is the coordinate of the pixel in the image, and (u,v) is the coordinate of the structural element.

According to the above definition, the maximum and minimum gray value of the local image region can be obtained by dilation and erosion operators, respectively. The morphological gradient is defined as the difference of the dilation and erosion, which can extract the local information effectively. Meanwhile, we can obtain more comprehensive information by changing the scale and angle of the linear structural element. The large scale indicates the gradient information within long distances, while the gradient information with short distances is indicated by the small scale. Angle 0° indicates the horizontal gradient information, and angle 90° indicates the vertical gradient information.

Finally, gradients of all scales and all angles are integrated into the multi-scale morphological gradient *MSMG*.

$$MSMG(x,y) = \sqrt{\frac{2\sum_{\theta=1}^{m}\left(\sum_{j=1}^{n}\varepsilon_j \cdot G_j^\theta(x,y)\right)^2}{m}} \tag{10}$$

where m is the number of angle θ, and $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ in this paper, i.e., $m = 4$. n is the numbers of scales, and $n = 5$ in this paper. ε_j is the weight of gradient in scale j, $\varepsilon_j = 1/(2 \times j + 1)$.

The MSMG constraint $\delta_g(x,y)$ of pixel (x,y) is defined as:

$$\delta_g(x,y) = \max(MSMG_1(x,y), MSMG_2(x,y)) \tag{11}$$

where $MSMG_1(x,y)$ and $MSMG_2(x,y)$ are the multi-scale morphological gradients of the pixel (x,y) in the left I_1 image and the right image I_2.

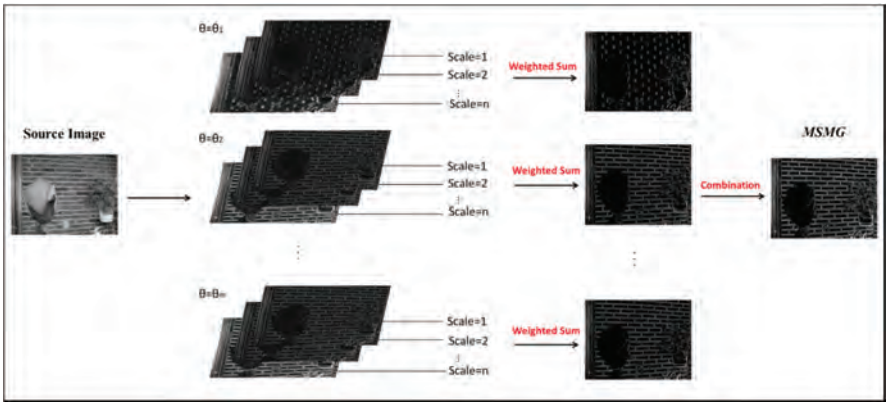


Figure 3. Demonstration of constructing the multi-scale morphological gradient.

2.2.3. Texture Difference

In this paper, we calculate the image entropy of the 3×3 neighborhood of the pixel to represent the local texture features of the image. We iterate through all pixels using the entropy filter of size 3 in the implementation. Image entropy is the measure of data randomness in an image's gray histogram, which is calculated by:

$$I^t = -\sum p \log_2 p \tag{12}$$

where p is the total number of histograms of image I. The detail texture information cannot be represented by image entropy due to only being based on the frequency of neighborhood data regardless of the intensity contrast. Considering the amalgamating of the variations of illumination and contrast in the image, it is considered adequate to regard image entropy as the coarse representation of texture features. The texture difference δ_t is defined as following.

$$\delta_t(x,y) = abs(I_1^t(x,y) - I_2^t(x,y)) \tag{13}$$

where $I_1^t(x,y)$is the image entropy of the 3×3 neighborhood of the pixel (x,y) in the image, similar to $I_2^t(x,y)$.

The global cost $C(x,y)$ is combined by the above three terms.

$$C(x,y) = (\delta_c(x,y) + \delta_g(x,y)) \times \delta_t(x,y) \tag{14}$$

Then we weight $C(x,y)$ by the BRs obtained in Section 2.1 to obtain the weighted cost $D(x,y)$.

$$D(x,y) = \begin{cases} vC(x,y), & f(x,y) \in BRs \\ C(x,y), & otherwise \end{cases} \tag{15}$$

where v is the weight coefficient of BRs, $0 < v < 1$. The seamline should preferentially pass through the BRs around the obvious objects, so we give a small weight value for the BRs.

2.3. Pixel-Level Optimization

The purpose of the pixel-level optimization is to optimize the location of the seamline in the local area. As shown in Figure 4, the overlap area of the image can be gained after determining the relation of the input images, and the intersection of the input images edges is determined as the start and end

point of the seamline. Then we detect the shortest path based on the weighted cost from the start point to the end point.

Dijkstra's algorithm is a global optimization technique to find the shortest path between the two nodes in the graph. Each pixel in the overlapping region is regard as one node which has eight neighbor nodes. As shown in Figure 5, the local cost is calculated based on the cost difference of neighbor nodes when detecting the shortest path using Dijkstra's algorithm. Let (x, y) be one node and (k_m, l_m) be a neighbor node of this node. The local cost of these two nodes is defined as:

$$d_{xy,k_m l_m} = |D(x,y) - D(k_m, l_m)|, \quad m = 1, 2, \cdots, 8 \tag{16}$$

where $D(x, y)$ and $D(k_m, l_m)$ are the weighted cost of pixel (x, y) and (k_m, l_m). Let $NBR(x, y)$ be all adjacent nodes of the node (x, y). $cost(x, y)$ and $cost(k_m, l_m)$ represent the global minimum cost from the start node to pixel (x, y) and (k_m, l_m), respectively.

$$cost(x,y) = \min(d_{xy,kl} + cost(k_m, l_m); (k_m, l_m) \in NBR(x,y)) \tag{17}$$

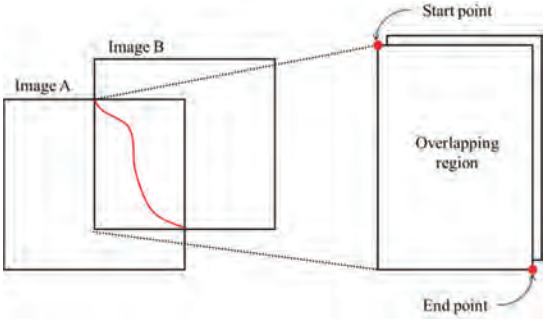


Figure 4. Determining overlapping region and start/end point.

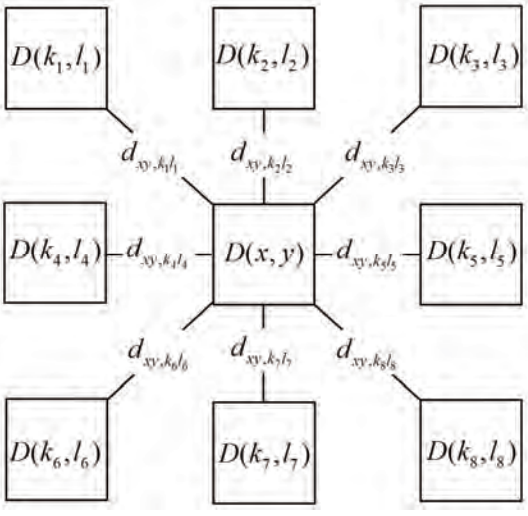


Figure 5. Node and neighbor nodes.

2.4. Multi-Image Seamline Detection

We introduce the method of seamline determination in a two-image overlapping region and can obtain a panoramic image with a wide field of view and high resolution by mosaicking a set of images using this method. As shown in Figure 6a, in the process of multi-image mosaicking, we hope that there is no multi-image overlap, i.e., the overlapping regions are all two-image overlap. In practical applications, the regions are always multi-image overlap. Figure 6b shows an illustrative example where the overlapping region is overlapped by three input images A, B, and C. The traditional method is to detect the seamline just between each of the two images, named frame-to-frame [33]. To mosaic multiple images more accurately, we propose a new optimization strategy to detect seamlines for multi-image overlap. Firstly we find the point $p(x, y)$ which is the weighted cost minimum in the setting rectangle. The center of this rectangle is the center of the multi-image overlapping region and its length and height are a quarter of the overlapping region. Then we detect the seamlines from the point $p(x, y)$ to the joint points of AB, AC, and BC. Finally, the panoramic image is mosaicked based on the optimal seamlines.

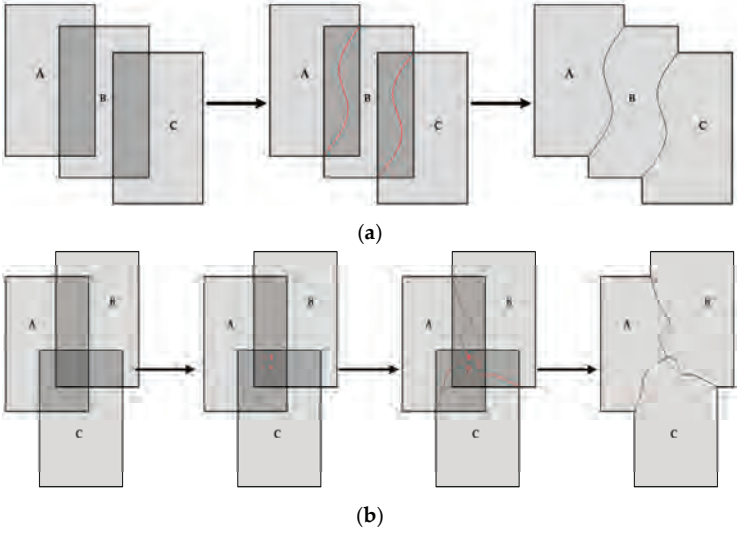


Figure 6. Seamline detection: (**a**) two-image overlap; (**b**) multi-image overlap.

3. Experiment and Discussion

The method we proposed was implemented by C++ (VC 10.0) and Matlab (2012b), combined. A desktop computer with an Intel Core i5 CPU at 3.2 GHz and 4 GB of memory was used. Experiments were conducted using three sets of orthoimages. In order to improve the efficiency, we built one pyramid level that was built by a reduction factor of three with a 3×3 average filter. Five sets of experiments were carried out to verify the effectiveness and superiority of our method.

3.1. Experiment of Image Segmentation

An image segmentation comparison was performed for Image Set 1 to verify the advantages of the BRs obtained by the PKGC segmentation we introduced in Section 2.1. Mean-shift was also carried out in the pyramid with parameter $(h_s, h_r, M) = (6, 5, 30)$, where (h_s, h_r) is the bandwidth and M is the smallest feature size. The weight v of BRs was 0.1.

The results of image segmentation are displayed in Figure 7. Figure 7a,b shows the left image and the right image, respectively, whose size is 3800×3200 pixels and the dotted rectangle indicates the

overlapping areas. The input images are segmented by mean-shift segmentation proposed in Pan's method. The span of regions that are larger than the given threshold S_T will be extracted as the BRs. S_T is the threshold of the span and usually equals the maximum size of significant targets. We test a set of different values ranging from 30 to 200 for determining the appropriate threshold S_T. When S_T is small, the BRs can be extracted, but the obvious objects with the larger size will be treated as BRs, as the red rectangles show in Figure 7c (S_T = 50). When S_T is larger, it can increase the extraction of obvious objects, but at the same time the background area cannot effectively be separated and will be regarded as ORs, as the blue rectangles show in Figure 7d (S_T = 180). When the background region is too small, or the object is too large, it cannot be effectively segmented no matter what value S_T is equal to, as the green rectangles show in Figure 7c,d. Figure 7e shows the segmentation result based on the PKGC algorithm. The objects and background can be segmented effectively by our method as the rectangles show. Therefore, the BRs obtained by our method are more accurate. The BRs getting by Pan's method depend on the value of the span threshold S_T. We must give a suitable S_T for every input image manually, which is apparently difficult. If the size of the object is larger than the size of the maximum background area, S_T, which can distinguish the large objects and the backgrounds, is non-existent, so the seamline passes through the obvious objects. Figure 7f shows the seamlines based on Pan's method (S_T = 80) and our method. The green line is Pan's method and the red line is ours. Figure 7g shows the enlarged details of the yellow rectangle regions in Figure 7f. We can see that the seamline by the proposed method successfully bypasses all of the obvious objects, but the seamline by Pan's method passes through the buildings.

(**a**) (**b**)

(**c**) (**d**)

Figure 7. *Cont.*

(**e**) (**f**)

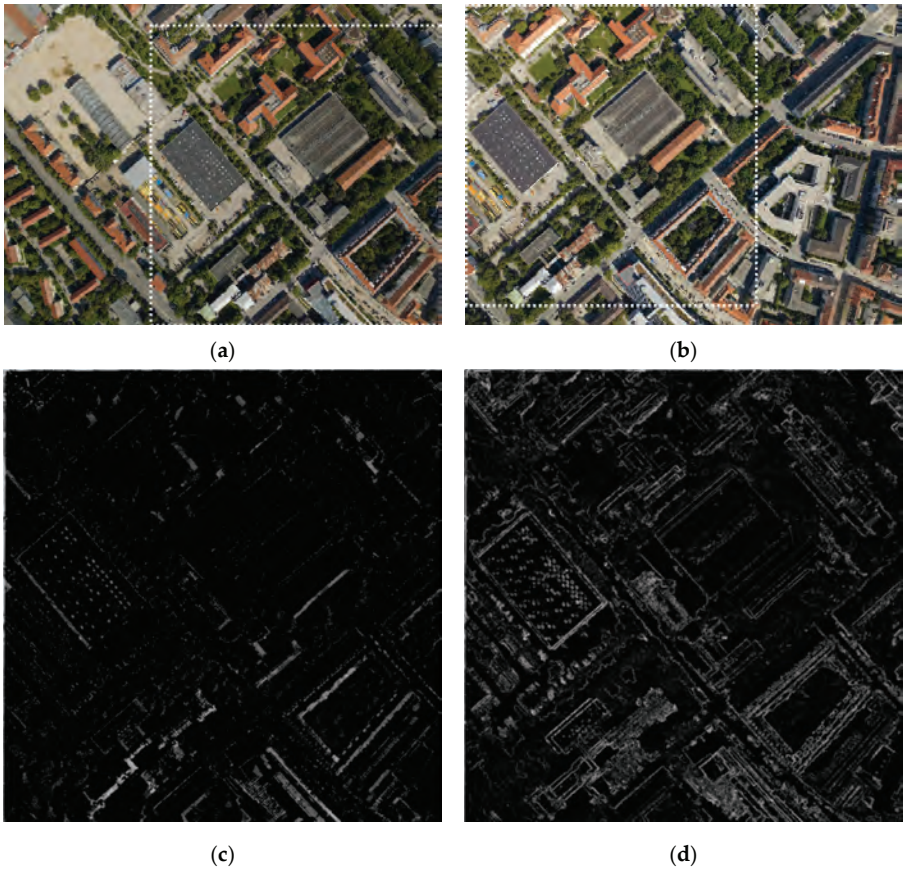
(**g**)

Figure 7. Experimental results of image set 1: (**a**) the left image; (**b**) the right image; (**c**) mean-shift segmentation result (S_T = 50); (**d**) mean-shift segmentation result (S_T = 80); (**e**) PKGC segmentation result; (**f**) seamlines by Pan's method (green line) and ours (red line); (**g**) details marked rectangles in (**f**).

3.2. Experiment of Energy Function

In order to verify the rationality of the energy function proposed in Section 2.2, we compared our cost and Pan's cost, then analyzed the results of the experiment for Image Set 2. The weight parameter of the color difference $\omega = 0.95$. In the process of calculating the MSMG, angle $\theta = \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ and the numbers of scales $n = 5$. The results of the energy function are displayed in Figure 8. Figure 8a,b shows the left image and the right image, respectively, whose size is 1900 × 1400 pixels, and the dotted rectangle indicates overlapping areas. The absolute value of the gray difference of the corresponding pixel is the same as the cost of the overlapping region directly, as Figure 8c shows. Although this method is simple, its stability is not strong, especially when the brightness difference of the overlapping region is small. Comprehensively considering the gray, texture, and edge information of the image, we proposed a new method to construct the energy function. Especially, the MSMG we proposed in Section 2.2.2 can extract the gradient information in scale space effectively, so the energy function is more accurate. As Figure 8d shows, the edge and the texture information are extracted comprehensively and clearly. Figure 8e shows the location of seamlines based on the cost of Pan's method and our

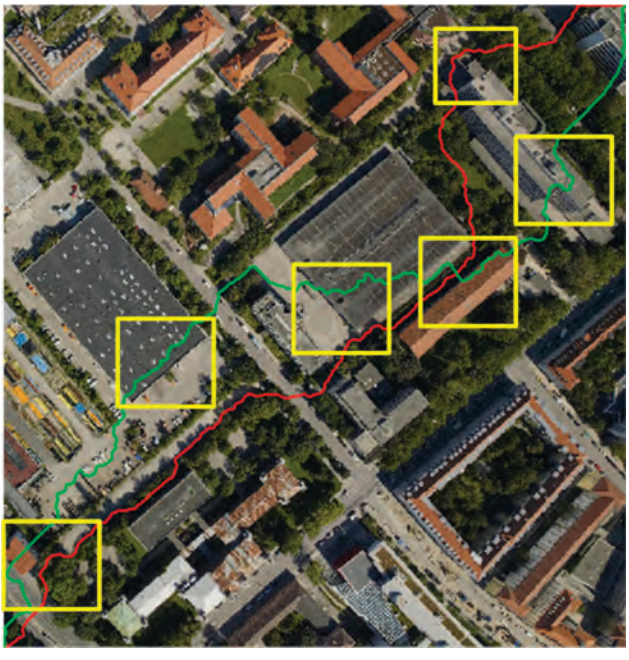
method. The green line is Pan's method and the red one is our method. The seamline based on our method can effectively avoid significant target edges and regions with complex textures. The cost we proposed can effectively reflect the salient information of the images. It can still extract the edge and the texture of the image, even though the gray values of input images are close. Contrarily, the cost based only on color difference is easily influenced by the brightness of the images. When the difference of brightness is too small, the obvious objects will be submerged in the BRs and the seamline will pass through the objects.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. *Cont.*



(**e**)

Figure 8. Experimental results of image set 2: (**a**) the left image; (**b**) the right image; (**c**) Pan's cost; (**d**) our cost; (**e**) seamlines based on Pan's cost (green line) and ours (red line).

3.3. Analysis of Time-Consumption

For Image Set 1 and 2, we determined the seamline using Chon's method [17], Li's method [20], Pan's method [24], and the proposed method, respectively, then recorded the time-consumption and the number of obvious objects passed through. Table 1 shows the performance comparison of different methods. To compute the NCC in Chon's method takes much time, and determining the seamline by iteration is intricate. Thus, the time-consumption is very large. Since the HOG feature of the 11×11 neighborhood is computed for every pixel, the calculation quantity is sizable. The main time-consumption of Pan's method is to determine preferred regions, which need to compute the span of every segmentation region. The more segmentation regions mean the more time-consumption. The proposed method takes less time than other methods. Meanwhile the seamline does not pass through any obvious object.

Table 1. Comparison of different methods.

Image Set	Items Method	Number of Pixels in Seamline	Number of Obvious Objects Passed Through	Processing Time (s)
1	Chon's	3694	2 buildings	835.63
	Li's	3546	2 buildings and 1 car	58.28
	Pan's	3975	5 buildings	37.20
	The proposed	4158	None	10.87
2	Chon's	1460	1 building	145.21
	Li's	1563	4 buildings	25.42
	Pan's	2254	6 buildings	13.76
	The proposed	1911	None	8.42

3.4. Analysis of Suppressing Noise

In order to test the performance of suppressing noise, we detected seamlines in the images perturbed with Gaussian noise and Gamma noise. Figure 9a shows the original image. Figure 9b,c are the images perturbed with Gaussian noise and Gamma noise. Figure 9d–f are the results of segmentation using the PKGC algorithm. The PKGC algorithm is very strong to suppress noise. As Figure 9e,f shows, obvious objects are segmented effectively. Thus, the seamlines based on the BRs bypass the obvious objects, as Figure 9h,i show. Correspondingly, the method we proposed in this paper is robust and stable to suppressing noise.

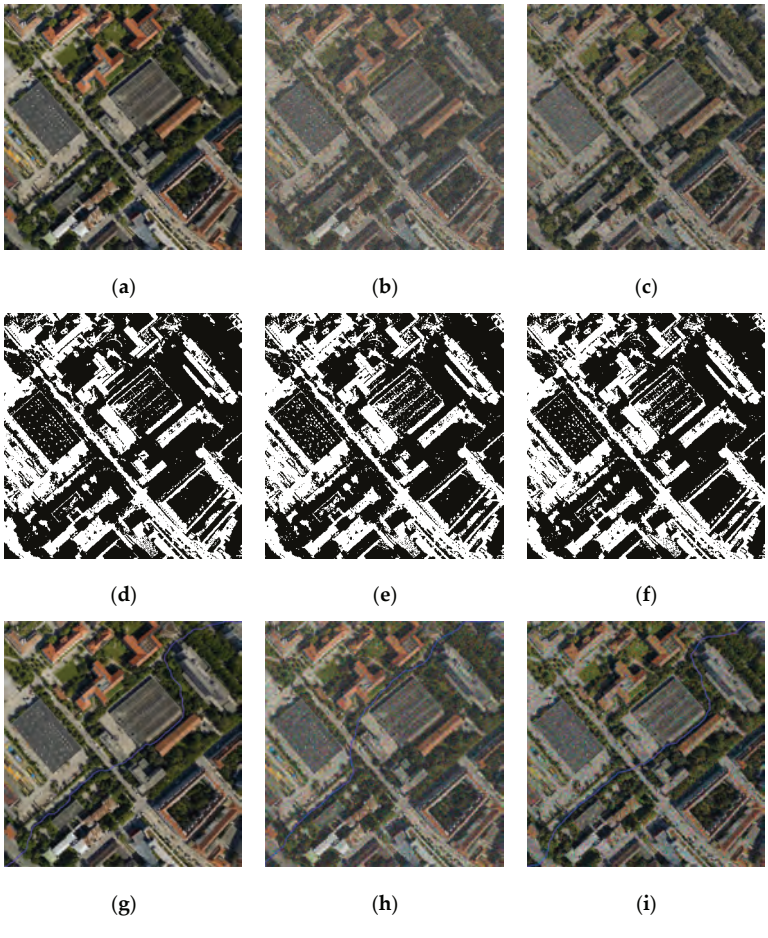


Figure 9. Experimental results of suppressing noise: (**a**) image without noise; (**b**) image with Gaussian noise; (**c**) image with Gamma noise; (**d**) segmentation result of (**a**); (**e**) segmentation result of (**b**); (**f**) segmentation result of (**c**); (**g**)seamline based on (**d**); (**h**) seamline based on (**e**), (**i**) seamline based on (**f**).

3.5. Experiment of Remote Sensing Image Mosaicking

The mosaicking dataset is captured by an aerial digital camera integrated in the airborne photoelectric platform. We use the unmanned aerial vehicle (UAV)-equipped airborne photoelectric platform to take images by whisk broom for a ground area. Figure 10 shows the process of obtaining the remote sensing image set. The experimental weather conditions are clear, the flight altitude

AGL is 2.1 km, the flight velocity is 30 m/s. The image is captured by an aerial visible light camera, the resolution is 3288 × 2192 pixels, the pixel size is 0.5 μm, the focal length f = 130.2 mm.

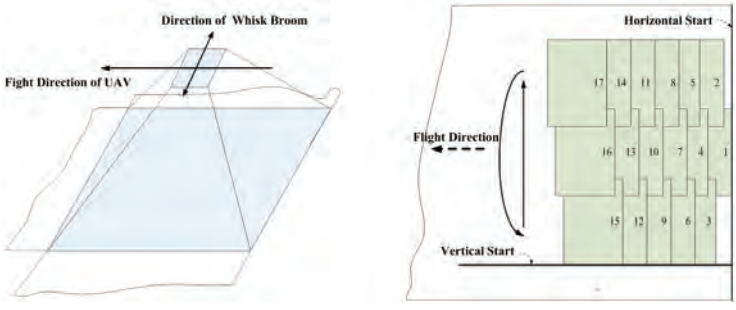


Figure 10. Obtaining remote sensing image set.

Figure 11a exhibits the remote sensing image dataset. Figure 11b gives the positional relations and overlapping regions between image sequences and the blue rectangle is the edge of each image. Figure 11c shows the seamlines detected in each overlapping region of the two images. Figure 11d are the optimal seamlines using the method we proposed in Section 2.4. Figure 11e shows the enlarged details of the yellow rectangle regions in Figure 11d. The detailed picture shows that the seamlines can effectively avoid the significant target, and pass through the background regions with a smooth texture, such as road, grass, and so on. The seamlines can provide a strong guarantee for the follow-up mosaicking. The comparison of seamlines detected in each two-image overlapping (TIO) region and multi-image overlapping (MIO) region is shown in Table 2. We can see that the optimal method for multi-image seamline detection proposed in Section 2.4 can determine a seamline with fewer pixels and takes less time, only 69.16% of TIO. Meanwhile, the optimal method can effectively resolve the problem that the seamline passes through the obvious objects near the edge of the overlapping region.

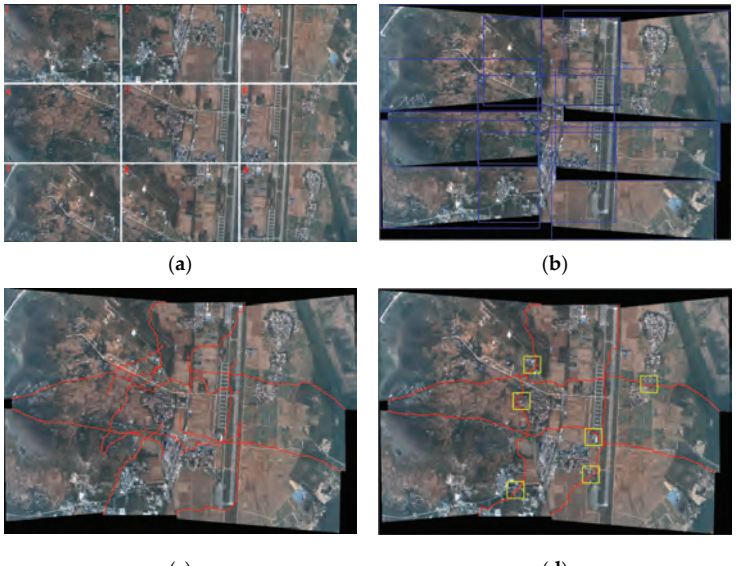


Figure 11. *Cont.*

(**e**)

Figure 11. Experimental results of image mosaicking: (**a**) the remote sensing image set; (**b**) positional relations and overlapping regions; (**c**) seamlines in two-image overlap; (**d**) optimal seamlines; (**e**) the details of yellow rectangles in (**d**).

Table 2. Comparison of seamlines.

Number of Image Overlapping Region	Type of Seamline	Number of Pixels in Seamline	Number of Obvious Objects Passed Through	Processing Time (s)
1-2	TIO	2916	1	2.19
	MIO	2277	0	1.68
1-4	TIO	3302	0	2.10
	MIO	2897	0	1.58
2-3	TIO	3255	0	1.92
	MIO	1863	0	1.42
2-5	TIO	3012	2	1.92
	MIO	1986	0	1.49
3-6	TIO	3706	1	2.52
	MIO	3163	1	2.01
4-5	TIO	3300	0	1.95
	MIO	1646	0	1.41
4-7	TIO	3693	2	1.58
	MIO	2124	1	1.06
5-6	TIO	2755	1	1.96
	MIO	1598	0	1.50
5-8	TIO	3621	1	1.48
	MIO	2320	0	0.95
6-9	TIO	4270	1	2.62
	MIO	2747	0	2.08
7-8	TIO	4114	0	2.56
	MIO	2793	0	2.03
8-9	TIO	3711	0	2.23
	MIO	3396	0	1.77

We mosaic this dataset by the pixel fusion method, two-image overlap seamlines, and the multi-image overlap seamlines method. Figure 12 shows the compare of mosaicking results by different methods, the first row of which is the result and details by pixel fusion method. The second row is the result based on two-image overlap seamlines and the third row shows the result and details based on the multi-image overlap seamlines. When using the pixel fusion method, the ghost effect will exist in the regions that the corresponding pixels are misaligned. The uneven brightness also exists when the brightness differences are too large. When using the two-image seamlines method, there will

be the geometrical dislocations in the misaligned regions and uneven brightness in the regions of the large brightness difference, showing the obvious "seam". The multi-image overlap seamlines method can solve the above problems effectively. There are no obvious seams, no geometric misalignments, no ghosting, and a high quality of reproduction in the panoramic mosaicking image.

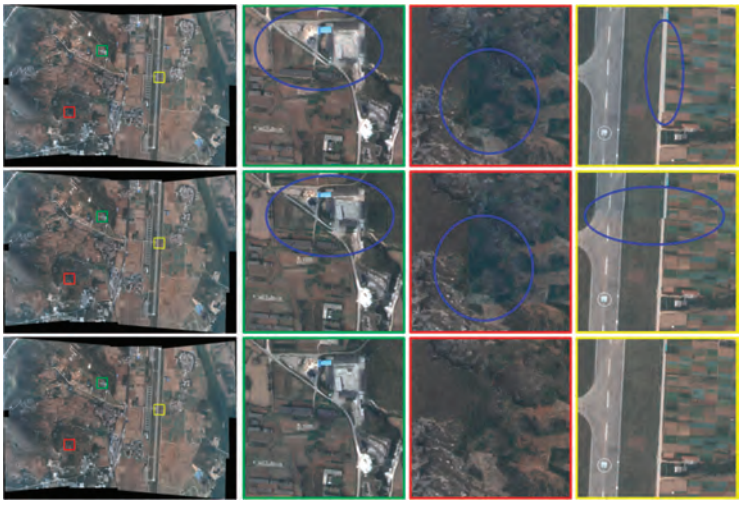


Figure 12. Comparison of different mosaicking methods: the first row is the pixel fusion method; the second row is the two-image overlap seamlines method and the third is the multi-image overlap seamlines method.

The proposed method is effective and excellent to determine seamlines for remote sensing image mosaicking. However, there is still room for improvement in this method. When the size of the objects is too small and the color is very close to the background, such as black cars in the shadow of buildings, they cannot be extracted. Moreover, although the time-consumption is greatly reduced compared with other methods, it is not sufficient for real-time processing. These issues will be solved in future research.

4. Conclusions

We proposed a novel method to determine the seamline based on PKGC segmentation and the combined cost for remote sensing image mosaicking. In this method, we segment the input images by the PKGC algorithm and determine the BRs and ORs based on the result of segmentation firstly. Then the global cost containing the color difference, MSMG constraint, and texture difference is weighted by BRs. Finally, the seamline is determined in the weighted cost from the start point to the end point. Dijkstra's shortest path algorithm is used to optimize the seamline at the pixel level. The BRs can automatically be obtained by PKGC segmentation in this method. Furthermore, the combined cost can indicate the image information accurately. The new method for multi-image seamline detection can effectively resolve the problems in multi-image mosaicking. Experimental results demonstrate the effectiveness and superiority of the proposed method. Seamlines go within roads or grass and successfully bypass the obvious objects. The performance of the proposed method is much better and faster than Pan's method. Moreover, this method is particularly suitable for images with objects of larger size. The proposed method is effective and shows potential for remote sensing image mosaicking. The mosaicking image based on the seamlines using the proposed method can satisfy the requirements of both field of view and resolution.

Author Contributions: Qiang Dong designed the algorithm, edited the source code, analyzed the experiment results and wrote the manuscript; Jinghong Liu made contribution to experiments design and revised paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xu, Y.; Ou, J.; He, H.; Zhang, X.; Mills, J. Mosaicking of Unmanned Aerial Vehicle Imagery in the Absence of Camera Poses. *Remote Sens.* **2016**, *8*, 204. [CrossRef]
2. Zhu, Z.; Hanson, A.R.; Riseman, E.M. Generalized Parallel-perspective Stereo Mosaics from Airborne Video. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 226–237.
3. Levin, A.; Zomet, A.; Peleg, S.; Weiss, Y. Seamless Image Stitching in the Gradient Domain. *Comput. Vis. ECCV 2004* **2004**, 377–389.
4. Xiong, Y.; Pulli, K. Mask-based Image Blending and Its Applications on Mobile Devices. In Proceedings of the SPIE 6th International Symposium on Multispectral Image Processing and Pattern Recognition, Yichang, China, 30 October–1 November 2009.
5. Sappa, A.D.; Carvajal, J.A.; Aguilera, C.A.; Oliveira, M.; Romero, D.; Vintimilla, B.X. Wavelet-Based Visible and Infrared Image Fusion: A Comparative Study. *Sensors* **2016**, *16*, 861. [CrossRef] [PubMed]
6. Huang, X.; Netravali, R.; Man, H.; Lawrence, V. Multi-sensor Fusion of Infrared and Electro-optic Signals for High Resolution Night Images. *Sensors* **2012**, *12*, 10326–10338. [CrossRef] [PubMed]
7. Szeliski, R.; Uyttendaele, M.; Steedly, D. Fast Poisson Blending Using Multi-splines. In Proceedings of the 2011 IEEE International Conference on Computation Photography (ICCP), Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, 8–10 April 2011; pp. 1–8.
8. Allène, C.; Pons, J.P.; Keriven, R. Seamless Image-based Texture Atlases Using Multi-band Blending. In Proceedings of the 2008 19th International Conference on Pattern Recognition (ICPR), Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
9. Milgram, D.L. Computer Methods for Creating Photomosaics. *IEEE Trans. Comput.* **1975**, *100*, 1113–1119. [CrossRef]
10. Shiren, Y.; Li, L.; Peng, G. Two-dimensional Seam-point Searching in Digital Image Mosaicking. *Photogramm. Eng. Remote Sens.* **1989**, *55*, 49–53.
11. Afek, Y.; Brand, A. Mosaicking of Orthorectified Aerial Images. *Photogramm. Eng. Remote Sens.* **1998**, *64*, 115–124.
12. Fernandez, E.; Garfinkel, R.; Arbiol, R. Mosaicking of Aerial Photographic Maps via Seams Defined by Bottleneck Shortest Paths. *Oper. Res.* **1998**, *46*, 293–304. [CrossRef]
13. Fernández, E.; Martí, R. GRASP for Seam Drawing in Mosaicking of Aerial Photographic Maps. *J. Heuristics* **1999**, *5*, 181–197. [CrossRef]
14. Kerschner, M. Seamline Detection in Colour Orthoimage Mosaicking by Use of Twin Snakes. *ISPRS J. Photogram. Remote Sens.* **2001**, *56*, 53–64. [CrossRef]
15. Kass, M.; Witkin, A.; Terzopoulos, D. Snakes: Active Contour Models. *Inter. J. Comput. Vis.* **1988**, *1*, 321–331. [CrossRef]
16. Soille, P. Morphological Image Compositing. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 673–683. [CrossRef] [PubMed]
17. Chon, J.; Kim, H.; Lin, C.S. Seam-line Determination for Image Mosaicking: A Technique Minimizing the Maximum Local Mismatch and the Global Cost. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 86–92. [CrossRef]
18. Yu, L.; Holden, E.J.; Dentith, M.C.; Zhang, H. Towards the Automatic Selection of Optimal Seam Line Locations When Merging Optical Remote-sensing Images. *Int. J. Remote Sens.* **2012**, *33*, 1000–1014. [CrossRef]
19. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
20. Li, L.; Yao, J.; Lu, X.; Tu, J.; Shan, J. Optimal Seamline Detection for Multiple Image Mosaicking via Graph Cuts. *ISPRS J. Photogramm. Remote Sens.* **2016**, *113*, 1–16. [CrossRef]
21. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
22. Boykov, Y.Y.; Jolly, M.P. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in ND Images. In Proceedings of the Eighth IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2001; pp. 105–112.

23. Boykov, Y.; Kolmogorov, V. An Experimental Comparison of Min-cut/max-flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Pattern Anal. Mach. Intel.* **2004**, *26*, 1124–1137. [CrossRef] [PubMed]
24. Pan, J.; Zhou, Q.; Wang, M. Seamline Determination Based on Segmentation for Urban Image Mosaicking. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1335–1339. [CrossRef]
25. Saito, S.; Arai, R.; Aoki, Y. Seamline Determination Based on Semantic Segmentation for Aerial Image Mosaicking. *IEEE Access* **2015**, *3*, 2847–2856. [CrossRef]
26. Salah, M.B.; Mitiche, A.; Ayed, I.B. Multiregion Image Segmentation by Parametric Kernel Graph Cuts. *IEEE Trans. Image Process.* **2011**, *20*, 545–557. [CrossRef] [PubMed]
27. Goshtasby, A.A. *Theory and Applications of Image Registration*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
28. Bai, X.; Zhou, F.; Xue, B. Fusion of Infrared and Visual Images through Region Extraction by Using Multi Scale Center-surround Top-hat Transform. *Opt. Express* **2011**, *19*, 8444–8457. [CrossRef] [PubMed]
29. Nayar, S.K.; Nakagawa, Y. Shape from Focus. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 824–831. [CrossRef]
30. Bai, X.; Zhang, Y.; Zhou, F.; Xue, B. Quadtree-based Multi-focus Image Fusion Using a Weighted Focus-measure. *Inf. Fusion* **2015**, *22*, 105–118. [CrossRef]
31. Vincent, L.; Soille, P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598. [CrossRef]
32. Wang, D.A. Multiscale Gradient Algorithm for Image Segmentation Using Watershelds. *Pattern Recognit.* **1997**, *30*, 2043–2052. [CrossRef]
33. Gracias, N.; Mahoor, M.; Negahdaripour, S.; Gleason, A. Fast Image Blending Using Watersheds and Graph Cuts. *Image Vis. Comput.* **2009**, *27*, 597–607. [CrossRef]

sensors

MDPI

Article

Using Unmanned Aerial Vehicles in Postfire Vegetation Survey Campaigns through Large and Heterogeneous Areas: Opportunities and Challenges

José Manuel Fernández-Guisuraga [1,*], Enoc Sanz-Ablanedo [2], Susana Suárez-Seoane [1] and Leonor Calvo [1]

[1] Biodiversity and Environmental Management Department, Faculty of Biological and Environmental Sciences, University of León, 24071 León, Spain; s.seoane@unileon.es (S.S.-S.); leonor.calvo@unileon.es (L.C.)
[2] Mining Technology, Topography and Structures Department, University of León, 22400 Ponferrada, Spain; enocsanz@unileon.es
* Correspondence: jofeg@unileon.es; Tel.: +34-696-865-431

Received: 9 January 2018; Accepted: 12 February 2018; Published: 14 February 2018

Abstract: This study evaluated the opportunities and challenges of using drones to obtain multispectral orthomosaics at ultra-high resolution that could be useful for monitoring large and heterogeneous burned areas. We conducted a survey using an octocopter equipped with a Parrot SEQUOIA multispectral camera in a 3000 ha framework located within the perimeter of a megafire in Spain. We assessed the quality of both the camera raw imagery and the multispectral orthomosaic obtained, as well as the required processing capability. Additionally, we compared the spatial information provided by the drone orthomosaic at ultra-high spatial resolution with another image provided by the WorldView-2 satellite at high spatial resolution. The drone raw imagery presented some anomalies, such as horizontal banding noise and non-homogeneous radiometry. Camera locations showed a lack of synchrony of the single frequency GPS receiver. The georeferencing process based on ground control points achieved an error lower than 30 cm in X-Y and lower than 55 cm in Z. The drone orthomosaic provided more information in terms of spatial variability in heterogeneous burned areas in comparison with the WorldView-2 satellite imagery. The drone orthomosaic could constitute a viable alternative for the evaluation of post-fire vegetation regeneration in large and heterogeneous burned areas.

Keywords: drone; megafire; multispectral imagery; Parrot SEQUOIA; UAV; WorldView-2

1. Introduction

Natural hazards, such as wildfires, constitute a serious global concern that is expected to increase in the future [1] mainly due to global warming predictions and changes in land use [2,3]. In particular, the increasing severity and recurrence of large forest fires in Mediterranean Basin ecosystems [4] can lead to severe long-term land degradation, including desertification [5,6]. Thus, post-fire monitoring of these systems through different tools should be a priority for management purposes [7].

Advances in geospatial technologies have led to an increase in the utilization of remote sensing techniques [3], which represent a major opportunity for conducting post-fire surveys in large and heterogeneous burned ecosystems [8]. High spatial resolution satellite imagery, such as that provided by Deimos-2, GeoEye-2, QuickBird or WorldView-2 on-board sensors, among others, have been used to assess post-fire regeneration in terms of fractional vegetation cover [8], species richness [9] or the basal area of tree species [10]. Nevertheless, satellite imagery shows certain weaknesses that could limit its applicability in the post-fire monitoring of highly heterogeneous and dynamic areas. First, the revisit periods of satellite platforms cannot be user-controlled for short-term time series monitoring

of plant communities with a fast regeneration rate, such as shrublands, after the occurrence of a disturbance [11]. Second, satellite imagery may be seriously affected by cloud cover and its projected shadows [12]. Third, even though the spatial resolution of multispectral satellite imagery has been improved, resolutions below one meter have not been achieved, which could become a problem when monitoring certain biophysical properties in spatially heterogeneous ecosystems [11]. For their part, sensors aboard manned platforms such as aircrafts, can also be used to conduct post-fire surveys on demand, but regular monitoring is constrained because of the high economic costs [13].

The use of lightweight unmanned aerial vehicles (UAVs) usually implies lower economic costs than other remote sensing techniques when surveying relatively small areas [11,13–16] and their low flight speed and flight altitude enables ultra-high spatial resolution (better than 20 cm) imagery [11] to be taken. In burned areas, vegetation recovery is not homogeneous due to the spatial variation in fire regime, pre-fire plant community composition and environmental characteristics [17]. Therefore, the use of ultra-high spatial resolution imagery would allow for conducting studies at the population level [18] to assess the effectiveness of post-fire management actions such as seedlings plantation strategies or seedling recruitment. Moreover, UAVs are flexible in terms of attaching different kinds of sensors (e.g., RGB, multispectral or LiDAR), also allowing the operator to schedule the exact flight time to gather data over target areas [19]. Nevertheless, UAV imagery may be difficult to manage because of its ultra-high spatial resolution [20] and the platform does not allow for a simultaneous image acquisition of large areas [19] and has a limited flight time [12].

Several research projects have used UAV on-board sensors for wildlife population monitoring [21–26], estimation of forest structural parameters [27–32], individual tree or species mapping [33–36], estimation of post-fire vegetation recovery from digital surface models [37], estimation of fire severity [18,38] and forest fire detection [39]. However, to our knowledge, the operational and processing challenges in the generation of multispectral mosaics derived from rotor-based UAVs imagery that allow very large burned areas to be monitored have not been assessed yet. In addition, it would be necessary to know the pros and cons of this tool on large burned surfaces in relation to fine-grained satellite imagery. In this context, the comparison between the spatial and spectral information provided by UAVs and satellites on heterogeneous surfaces would be essential to determine their suitability for representing fine scale ground variability. Some authors have compared the spatial information provided by ultra-high spatial resolution imagery captured by UAVs and high spatial resolution satellite imagery, for instance that provided by WorldView-2 satellite on agricultural systems such as vineyards [12] or crops [40], but not to our knowledge in heterogeneous burned areas.

Most common cameras employed in UAV surveys are digital RGB cameras [11,40] or digital cameras where one of the visible bands has been adapted for NIR imagery acquisition [16,32]. Also, multispectral cameras, such as Tetracam ADC Lite [41] or MicaSense RedEdge [42], have been chosen to perform aerial surveys. For its part, the Parrot SEQUOIA (Parrot SA, Paris, France) is a novel and affordable multispectral camera released on the market in early 2016, whose imagery quality has not been evaluated in scientific literature.

The main objective of this study is to evaluate the feasibility of using a rotor-based UAV with an on-board multispectral sensor (Parrot SEQUOIA) to obtain a multispectral orthomosaic at ultra-high spatial resolution, which could be useful for forestry management purposes in a heterogeneous and large burned area (3000 ha). Specifically, we intend to: (1) evaluate the quality of the raw imagery dataset captured with the Parrot SEQUOIA multispectral camera; (2) discuss the challenges encountered when dealing with the volume of data at ultra-high spatial resolution generated in the UAV survey carried out in a large area, and assess both the required processing capability and the quality of the obtained multispectral mosaic; and (3) compare the spatial information provided by the UAV ultra-high resolution multispectral mosaic with high spatial resolution satellite imagery (WorldView-2) in a heterogeneous burned landscape.

2. Materials and Methods

2.1. Study Area

The study area (Figure 1) is a 3000 ha framework located in the central section of a megafire of about 10,000 ha which occurred in a *Pinus pinaster* stand in Sierra del Teleno (León Province, northwest Spain) in August 2012. The survey framework was representative of the heterogeneity of the fire regime within the perimeter.

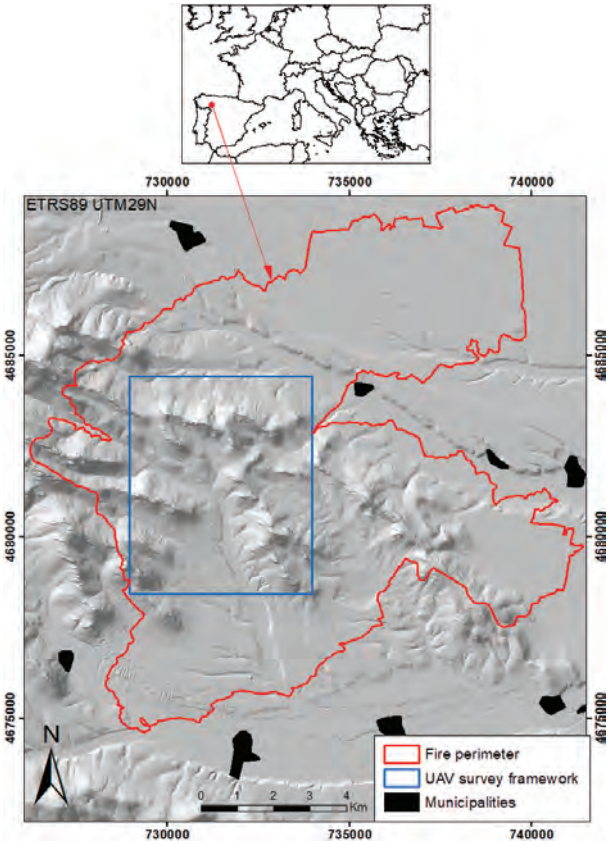


Figure 1. UAV survey framework within the megafire perimeter of Sierra del Teleno.

The study area is dominated by an Appalachian relief with prominent quartzite crests, wide valleys with moderate slopes on the upper two thirds of the study area, and sedimentary terraces on the lower third. The mean annual temperature in the area is 10 °C, with an average rainfall of 650 mm. The understory plant community after the occurrence of the megafire is composed by species such as *Halimium alyssoides*, *Pterospartum tridentatum* and *Erica australis* [43], with a great regeneration of *Pinus pinaster* seedlings.

2.2. UAV Platform and Multispectral Camera

A FV8 octocopter (ATyges, Málaga, Spain, Figure 2) was chosen to perform the aerial survey of a large burned surface of 3000 ha. This UAV is manufactured entirely from carbon fiber and titanium and it weighs 3.5 kg, with a maximum payload mass of 1.5 kg. The eight brushless motors (AXI-ATYGES

2814/22 260 W with a maximum efficiency of 85%) are powered by two lithium-ion polymer batteries (rated capacity and voltage of 8200 mAh and 14.8 V, respectively). The UAV has a cruising speed of 7 $m \cdot s^{-1}$ (10 $m \cdot s^{-1}$ max), with an ascent/descent rate of 5.4 $km \cdot h^{-1}$ (10.8 $km \cdot h^{-1}$ max). The maximum interference-free flight range is 3 km, with a flight duration of 10–25 min depending on the payload and weather conditions. The maximum flight height is 500 m above ground layer (AGL). The platform is remotely controlled by a 12-channel MZ-24 HoTT radio transmitter (Graupner, Kirchheim unter Teck, Germany) operating at 2.4 GHz. The UAV is equipped with a micro FPV camera with real-time video transmission at 5.8 GHz to a Flysight monitor. The core component of the UAV electronics is an ATmega 1284P flight controller (Microchip Technology Inc., Chandler, AZ, USA) with an integrated pressure sensor, gyroscopes and accelerometers. The navigation control board is based on an Atmel ARM9 microcontroller and it has a MicroSD card socket for waypoint data storage. The GPS module with integrated antenna is a LEA-6 (u-blox, Thalwil, Switzerland). This system allows for autonomous, semi-autonomous and manual takeoffs, landings and flight.

Figure 2. ATyges FV8 octocopter used for the aerial survey.

A Parrot SEQUOIA multispectral camera was installed underneath the UAV platform. The camera has four 1.2-megapixel monochrome sensors that collect global shutter imagery along four discrete spectral bands [44]: green (center wavelength -CW-: 550 nm; bandwidth -BW-: 40 nm), red (CW: 660 nm; BW: 40 nm), red edge (CW: 735 nm; BW: 10 nm) and near infrared -NIR- (CW: 790 nm; BW: 40 nm). The horizontal (HFOV), vertical (VFOV) and diagonal (DFOV) fields of view of the multispectral camera are 70.6°, 52.6° and 89.6°, respectively, with a focal length of 4 mm. With a flight altitude of 120 m, a ground sample distance (GSD) of 15 cm can be achieved. The camera was bundled with an irradiance sensor to record light conditions in the same spectral bands as the multispectral sensor. The weight of the multispectral camera plus the irradiance sensor is 107 g. 16-bit RAW files (based on 10-bit data) are stored in this camera during image shooting. ISO value and exposure time was set to automatic. Every image capture setting is saved in a text metadata file together with the irradiance sensor data. All this information is taken into account during the preprocessing stage to obtain absolute reflectance values for the final product.

2.3. UAV Survey Campaign

The aerial survey campaign was conducted for 100 h between June and July 2016. All flights (383) were performed within a 6-h window around the solar zenith to maintain relatively constant lighting

conditions. Though small variations in environmental conditions were rectified with the irradiance sensor, severe wind or cloud cover were avoided.

Mikrokopter Tools software was used to plan flights, which allowed the operator to generate an automatic flight route with waypoints depending on the camera's field of view (FOV), the chosen forward and side overlap between images and the required GSD [45]. A digital elevation model (DEM) was used to keep the same distance AGL in all flights tracks owing to the large difference in altitude (410 m) in the study framework. Flight tracks were uploaded in the UAV for each complete day. The flight height was fixed at 120 m AGL, providing an average ground resolution of 14.8 cm·pixel^{-1} given the specific camera characteristics. Each flight had an effective duration of 5–6 min (without including the takeoff and landing), with an average speed of 10 m s^{-1}. Battery change time and time needed to reach each takeoff site were not computed. However, both time lapses were included in the total flight time of 100 h. The camera trigger interval was set to a platform advance distance of 22.4 m in order to achieve an 80% forward image overlap at the fixed flight altitude. The waypoints route planned allowed an 80% side image overlap. The image overlap between adjacent flights was at least a flight line. The quality of the raw imagery dataset acquired during the UAV survey was evaluated to search for potentially undesired anomalies, such as: (1) horizontal banding noise (HBN) [46]; (2) non-homogeneous radiometry and issues related with hot-spot or opposition effect [47] or (3) blurring effects [48].

2.4. Image Data Processing

UAV imagery was processed into a multispectral mosaic with Pix4Dmapper Pro 3.0 [49] following the "Ag Multispectral" template. This software integrates computer vision techniques with photogrammetry algorithms [50] to obtain high accuracy in aerial imagery processing [51,52]. Pix4Dmapper Pro computes keypoints on the images and uses them to find matches between images. From these initial matches, the software runs several automatic aerial triangulation (AAT), bundle block adjustments (BBA) and camera self-calibration steps iteratively until optimal reconstruction is achieved. Then, a densified point cloud is generated to obtain a highly detailed digital surface model (DSM) that will be used to generate the reflectance maps. A pre-process or normalization was automatically applied to the imagery, where 16 bits TIF files (10 bit RAW data) were converted to standard 8 bit jpg files, taking into account the ISO, exposure time and irradiance sensor data.

A high-end computer with a 12-core Intel i7 processor and 64 GB of RAM was used to process the imagery. Most of the processing steps in Pix4Dmapper Pro need a large number of computational resources that grow exponentially as more images are simultaneously processed. Due to software and hardware limitations for very large projects (above 10,000 images), each of the nine projects was split into smaller subprojects. The subprojects could then be merged after completing the AAT-BBA stage for each one, being necessary only to further process the less demanding subsequent steps for the merged project. Flights, subprojects and projects processing workflows are detailed in Figure 3. Radiometric corrections were introduced based on camera setup parameters and sun irradiance measured by the irradiance sensor. Initial georeferencing was achieved by introducing camera locations in the AAT-BBA stage. At least ten ground control points (GCPs) evenly distributed per subproject were extracted from aerial orthophotos of the Spain National Plan of Aerial Orthophotography (PNOA) to improve global spatial accuracy. This dataset has a GSD of 25 cm with an accuracy better than 0.50 m in terms of $RMSE_{X,Y}$ [53]. The multispectral outputs (four reflectance maps with a GSD of 20 cm) of the Pix4D projects were mosaicked using ArcGIS 10.3.1 (Esri, Redlands, CA, USA) [54] without applying reflectance normalization to avoid the modification of the computed reflectance values in the radiometric correction process. Geospatial accuracy of the outputs was assessed in terms of root mean square error (RMSE) in X, Y and Z from the coordinates of 50 targets uniformly arranged through the UAV survey framework. The X, Y and Z coordinates of the Control Points (CPs) were measured with a high-accuracy GPS receiver (Spectra Precision MobileMapper 20 with accuracy better than 0.50 m in terms of $RMSE_{X,Y}$) in postprocessing mode.

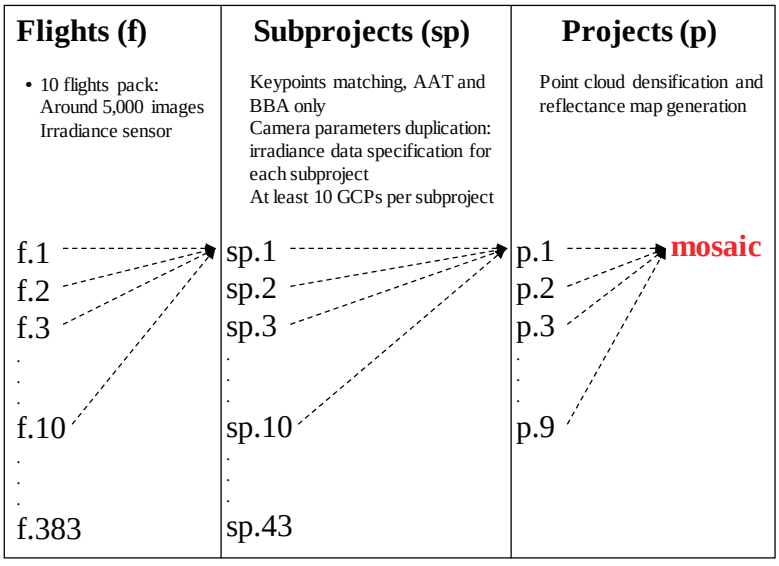


Figure 3. Processing workflow of the UAV imagery with Pix4Dmapper Pro.

2.5. WorldView-2 High Spatial Resolution Satellite Imagery and Image Comparison Statistical Analysis

A WorldView-2 image acquired on 23 June 2016 for the study framework was used to compare the spatial information provided by the UAV platform with high resolution satellite imagery in a heterogeneous burned landscape. The spatial resolution of the multispectral sensor on-board WorldView-2 satellite at nadir is 1.84 m, but the image was delivered by DigitalGlobe resampled to 2 m. This sensor has eight bands in the visible and NIR region of the spectrum [55]: coastal blue (400–450 nm), blue (450–510 nm), green (510–580 nm), yellow (585–625 nm), red (630–690 nm), red edge (705–745 nm), NIR1 (770–895 nm) and NIR2 (860–1040 nm). The raw image was orthorectified with a DEM (accuracy better than 20 cm in terms of $RMSE_Z$) and GCPs extracted from PNOA orthophotos. The image atmospheric correction was conducted by the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) algorithm [56]. The HyPARE algorithm implemented in ENVI 5.3 [57] was used to geometrically align the UAV multispectral orthomosaic and the WorldView-2 image achieving a UAV subpixel RMSE (<20 cm).

The image comparison was performed on the basis of the reflectance values and the Normalized Difference Vegetation Index (NDVI) of the UAV multispectral orthomosaic and the WorldView-2 image. UAV multispectral mosaic at original resolution (20 cm) was resampled to a GSD of 1 m (half of WorldView-2 spatial resolution) and 2 m (WorldView-2 spatial resolution) with a block average function for the input pixels within a set of non-overlapping windows with the required size (5×5 and 10×10 pixels). The function was computed with ArcGIS 10.3.1. Pearson bivariate correlations between the UAV multispectral mosaic (GSD of 20 cm, 1 m and 2 m) and WorldView-2 image (GSD of 2 m) were calculated on each comparable band to assess the spatial information provided by each sensor in our survey framework. To determine the reflectance variability between sensors, we computed the variance in the reflectance values in each band of the UAV images (native spatial resolution and resampled) and WorldView-2 image. For the more heterogeneous surface within the survey framework, which covers 1.5 ha, a basic statistic package was calculated on the UAV (at native resolution and 2 m) and WorldView-2 NDVI maps to compare the potentiality of these products in post-fire vegetation monitoring.

3. Results

3.1. Raw Imagery Dataset Quality

From 383 UAV flights, we acquired 45,875 images for each band, which made a total of 183,500 raw images that represented approximately 430 GB of information. The normalized UAV images had a balanced contrast. However, the red channel showed some saturation over highly reflective surfaces on this wavelength, such as forest tracks in our study area (Figure 4).

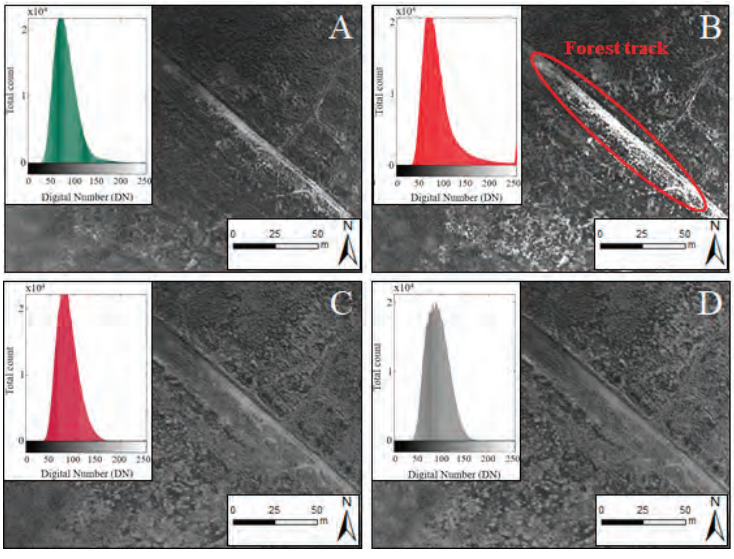


Figure 4. Example normalized UAV images from the dataset corresponding to green (**A**); red (**B**); red edge (**C**) and NIR (**D**) bands, as well as the image histograms.

A slightly horizontal banding noise (HBN) was observed within the four channels of the camera, especially in the green channel (Figure 5). The banding effect was more noticeable at the top and bottom of the image, where differences in the digital levels of alternate rows representing the same object were higher than 10%.

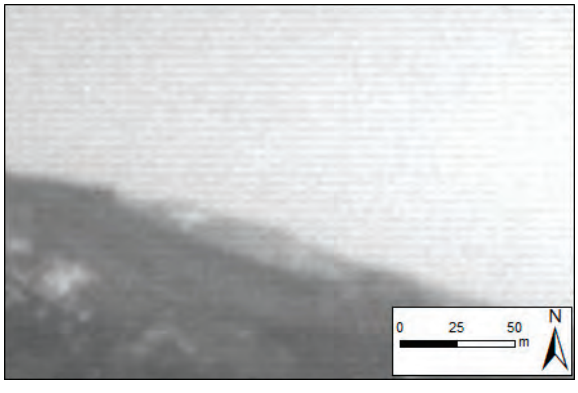


Figure 5. Horizontal banding noise in a raw green channel image.

Another undesired effect observed across the imagery was non-homogeneous radiometry across the image related with Bidirectional Reflectance Distribution Function (BRDF) [47]. In particular, a specific area of the imagery had systematically higher reflectance values than the remaining areas (Figure 6). This radiometric anomaly effect is commonly denominated hot-spot or opposition effect [58,59] and it appears as a consequence of the camera and sun position alignment [60]. For its part, the image dataset did not exhibit blurring effects that are usually associated with camera shaking [15].

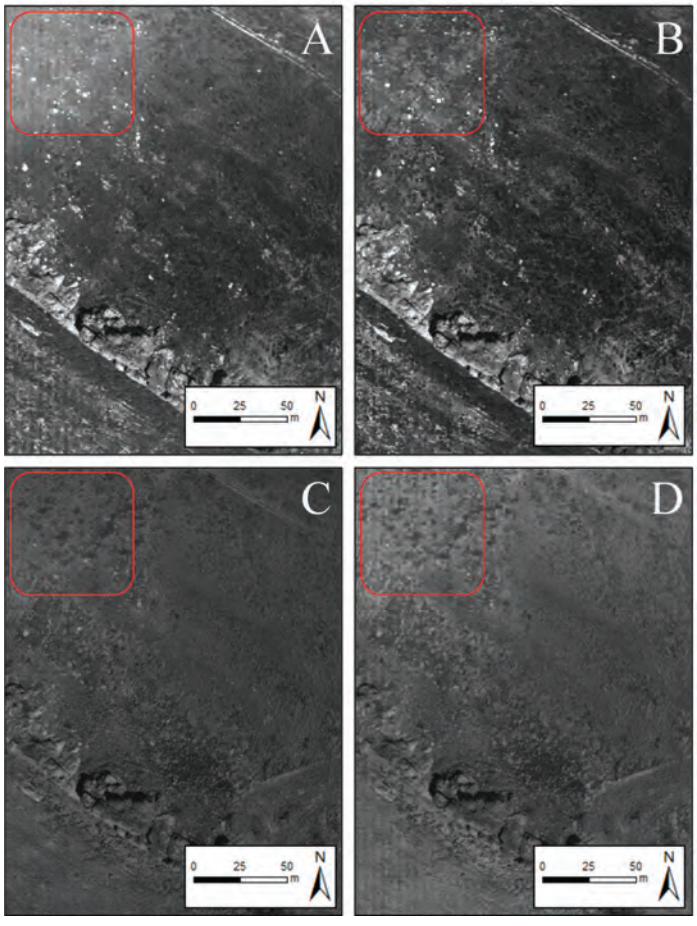


Figure 6. Hot-spot effect on green (**A**); red (**B**); red edge (**C**) and NIR (**D**) bands. Note that the upper left corner (framed in red) has higher reflectance values regardless of the terrain characteristics.

3.2. Multispectral Mosaic Processing and Product Quality

The processing of the multispectral orthomosaic was labor-intensive and time-consuming because of the large size of the surveyed area [19] and the ultra-high spatial resolution of the dataset [11]. Each subproject took 3–6 h to process the AAT, BBA and camera self-calibration. Point cloud densification and generation of the reflectance maps took up to 14 h for each project. The total amount of time required to process the whole dataset was about 320 h (20 days) with the available processing resources, including software failures.

For each of the 43 subprojects, the 3D reconstruction algorithm (AAT, BBA and self-calibration) obtained between 95% and 99% images aligned on the basis of more than 10,000 keypoints extracted from each image, with over 5500 keypoints matching with at least another two adjacent images. Green and NIR channels obtained the highest number of matches, whereas red channel systematically got the lowest number. The total number of 2D keypoint observations for BBA in each subproject was about 9 million, whereas the number of 3D matching points was 1.5 million, with a mean reprojection error of 0.2–0.3 pixels. The large forward and side overlap provided high accuracy in the keypoint matching step between adjacent images, as [45] pointed out. Changes between nominal and final parameters defining the geometrical model of the camera were as low as 0.01%. For its part, the point cloud densification at the merge step of the subprojects obtained between 6×10^6 and 7×10^6 3D densified points. For each of the nine projects, four reflectance maps (green, red, red edge and NIR) were obtained with a resampled GSD of 20 cm/pixel. Some areas of these maps were excluded (Figure 7) due to reflectance anomalies caused by USB-disconnections between the camera and the irradiance sensor.

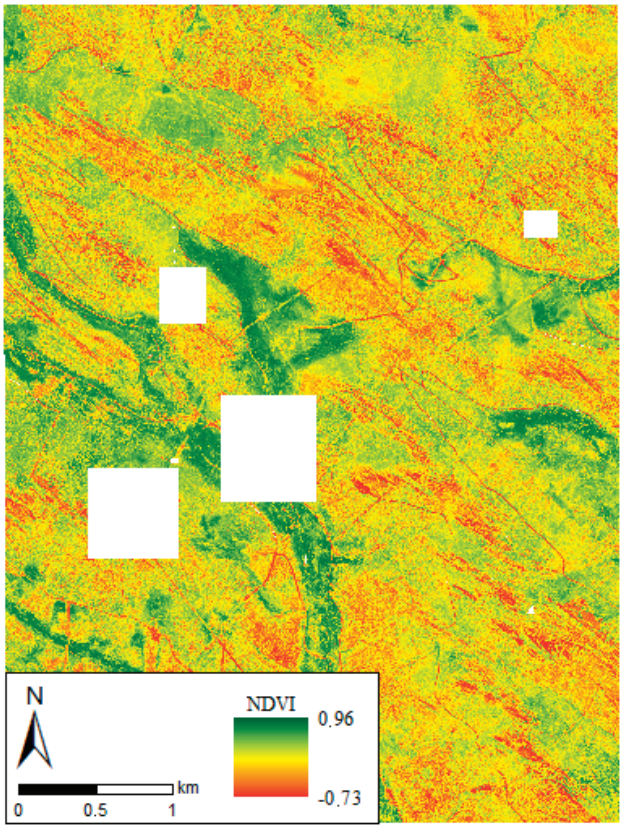


Figure 7. NDVI mosaic of the UAV survey framework. Blank areas are those masked due to the malfunction of the irradiance sensor.

Initial georeferencing was achieved by introducing the UAV's GPS positions taken at each camera shot in the bundle-block adjustment process within Pix4D workflow. The precision reported by Pix4D, calculated as the root mean square error (RMSE), was between 1.5–3 m in X-Y and between 2–4 m in Z.

The final georeferencing of the subprojects achieved by using ground control points (GCPs) extracted from PNOA orthophotos achieved an RMSE lower than 30 cm in X-Y and lower than 55 cm in Z. Horizontal and vertical accuracy was improved from initial georeferencing at least 80% and 73% respectively, after providing evenly distributed GCPs through the UAV survey framework.

3.3. Comparison of the Spatial Information Provided by UAV and WorldView-2 Imagery

Higher $r_{Pearson}$ values were obtained when the UAV mosaic resolution approached the resolution of the WorldView-2 image (2 m) (Table 1) for each band of the spectrum. The correlation between the two remote sensing platforms for each resolution was stronger for the visible region of the spectrum.

Table 1. Pearson correlation results between native and resampled UAV multispectral mosaics and WorldView-2 multispectral image.

	WV-2 Green	**WV-2 Red**	**WV-2 Red Edge**	**WV-2 NIR1**
UAV (20 cm) green	0.6832			
UAV (20 cm) red		0.7426		
UAV (20 cm) red edge			0.5873	
UAV (20 cm) NIR				0.6312
UAV (1 m) green	0.7385			
UAV (1 m) red		0.7649		
UAV (1 m) red edge			0.6134	
UAV (1 m) NIR				0.6476
UAV (2 m) green	0.7871			
UAV (2 m) red		0.7968		
UAV (2 m) red edge			0.6623	
UAV (2 m) NIR				0.6942

The largest variance in the reflectance values of each band was found for the UAV orthomosaic at 20 cm spatial resolution (Table 2). The variance of the UAV orthomosaic at 2 m of spatial resolution was similar to that of the WorldView-2 image.

Table 2. Variance in reflectance values computed for each band of the original and resampled UAV multispectral mosaics and the WorldView-2 multispectral image.

	Green	**Red**	**Red Edge**	**NIR**
UAV 20 cm	0.00083	0.00170	0.00134	0.00246
UAV 1 m	0.00079	0.00161	0.00122	0.00225
UAV 2 m	0.00071	0.00146	0.00101	0.00206
WV-2	0.00068	0.00132	0.00112	0.00211

The comparison between UAV and WorldView-2 NDVI maps derived from the imagery datasets at the original resolution of each sensor, corresponding to a heterogeneous surface of 1.5 ha within the survey framework, revealed greater variability in the UAV pixel values (Figure 8A,B). The horizontal structure of the vegetation observed in this area (Figure 9A) can be identified in the UAV mosaic (Figure 9B), but not in the WorldView-2 image (Figure 9C). The UAV NDVI map resampled to 2 m presented similar variability to the WorldView-2 image (Figure 8B,C).

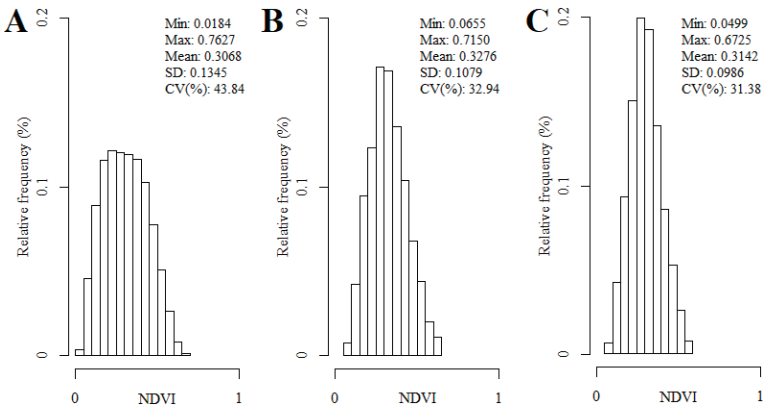


Figure 8. Histogram and statistics of the UAV (**A**) and WorldView-2 (**B**) NDVI map at native resolutions, and of the UAV NDVI map resampled to 2 m (**C**), corresponding to a 1.5 ha portion within pine plantation area.

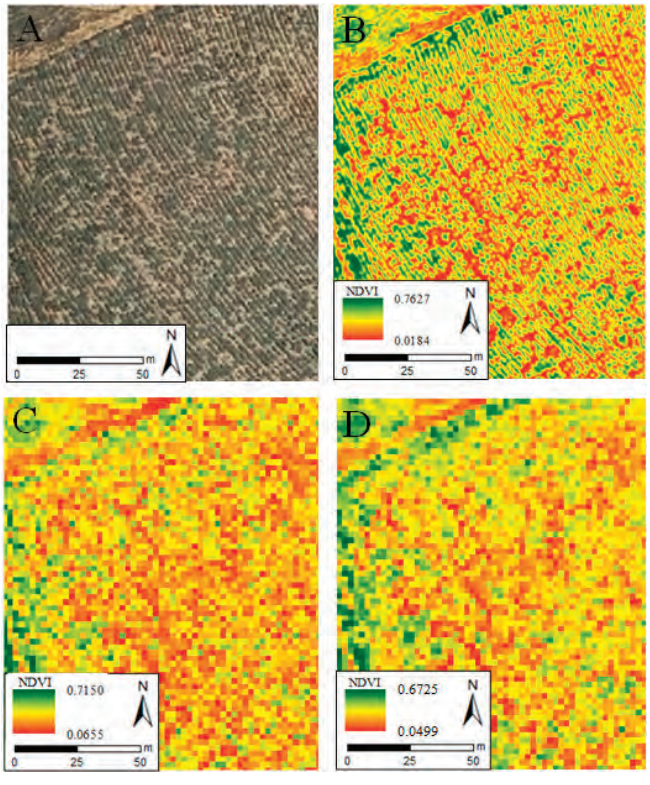


Figure 9. PNOA orthophoto (**A**), UAV (**B**) and WorldView-2 (**C**) NDVI maps at original resolutions, and UAV NDVI map resampled to 2 m (**D**), corresponding to a heterogeneous surface of 1.5 ha within the survey framework.

4. Discussion

This study evaluated the strengths and limitations of using a rotor-based UAV equipped with a novel multispectral camera (Parrot SEQUOIA) to conduct a field survey of a large (3000 ha) and heterogeneous burned surface. Our results indicate that the ultra-high spatial resolution UAV multispectral orthomosaic represents a valuable tool for post-fire management applications at fine spatial scales [18]. However, due to the ultra-high spatial resolution of the data and the large size of the surveyed area, data processing was highly time consuming.

Multispectral cameras onboard UAVs provide countless opportunities for remote sensing applications, but the technological limitations of these sensors [46] would require evaluation of the quality of the captured raw imagery data, particularly in novel sensors. In this study, we found that the raw imagery captured by the Parrot SEQUOIA multispectral camera presented some undesired radiometric anomalies. In the red channel we observed sensor saturation over highly reflective surfaces. This effect was not induced by radiometric down sampling from 10 to 8-bit performed by Pix4D during processing because it was present both in raw (10-bit) and in normalized (8-bit) images. The horizontal banding noise observed within the four channels of the camera is a common artifact of CMOS (complementary metal oxide semiconductor) rolling shutter sensors [46]. However, the Parrot SEQUOIA uses a global shutter system and this effect should not be significant in this multispectral sensor. To our knowledge, this camera has not been used in previous scientific studies and, therefore, this issue has not been reported so far. The issues related with Bidirectional Reflectance Distribution Function (BRDF) effect are magnified in sensors with a wide Field of View [61,62] such as the Parrot SEQUOIA. For its part, the hot-spot or opposition effect was more apparent at shorter wavelengths, as also highlighted by [47]. Some corrections to mask this effect have been proposed [59] that must be made individually for each image taking into account the time and position of the image acquisition, image orientation and solar positioning (azimuth and elevation), following some photogrammetric steps. Thus, the correction of this radiometric anomaly as well as the BRDF effect is very challenging and time consuming, becoming an unapproachable task when dealing with large imagery datasets [58]. The absence of a blurring effect in our dataset could be explained by the increased flight stability that the rotor-based UAVs offer over fixed-wing UAV platforms, also exhibiting fewer vibrations [13,29]. Moreover, the Parrot SEQUOIA camera was attached to the platform with a rubber damper to minimize vibrations, and the camera acquired imagery with the focal length set to infinity and fast shutter speed [15], preventing the occurrence of this effect. USB-disconnections between the camera and the irradiance sensor could be associated to a poor connection. However, the disconnections did not imply a major problem with the irradiance sensor, considering that it provided complete records for more than 90% of the survey framework with varying atmospheric conditions between adjacent flights, even performed on different days since the data acquisition from a rotor-based UAV platform could not be carried out in a single run over large areas due to restrictions in the flight range [12].

In relation to delivery times of on-demand imagery of commercial satellites and the usual times needed to implement post-fire management strategies within large burned areas [63], the length of the flight campaign (17 days) and the laboratory processing tasks (20 days) required a reasonable time. The computational demand of the project was very high due to the large amount of raw image data collected (183,500 raw images) and its ultra-high spatial resolution. The size of this dataset caused management difficulties in the laboratory in terms of data storage, backup and processing capability. This circumstance has already been reported by [20], data transfer between research teams being restricted by physical storage units or some processing options such as cloud computing. This computational demand may limit the execution of this type of projects to users who have access to high-end computers to process raw imagery. However, recent advances in computational capacity would allow a large-scale implementation of this type of workflow [64]. Other remote sensing products with reduced processing requirements such as on-demand satellite imagery offer a resolution from pan-sharpening techniques that is increasingly closer to what can be obtained with multispectral sensors on board UAVs. However, according to [65–67], the use of pan-sharpening techniques presents

several problems such as the appearance of spatial impurities or radiometric distortions in the merged product. This type of anomaly could represent a serious problem for providing the highest radiometric and spatial accuracy for fine scale applications. On the other hand, we consider that for this type of study, a UAV is more versatile than other types of remote sensing platforms, allowing flights to be carried out in the immediate post-fire situation given the provided control of the revisit time [18]. Another possible alternative to this highly demanding processing framework could be the performance of flights in small non-adjacent surfaces within the study area to reduce the campaign effort, but it would not be feasible to obtain a multispectral product that allows extrapolation of, for example, recovery models to other areas within the study area where the flights have not been carried out. The initial georeferencing precision ($RMSE_{X,Y}$ between 1.5–3 m and $RMSE_Z$ between 2–4 m) is not an optimum result considering that some authors, such as [51], have established as low accuracy an X-Y error higher than two times the GSD and a Z error higher than three times the GSD. Single frequency GPS receivers, such as the one used in the platform, which features a light antenna and chip power limitations, typically show important drifts throughout time. This is particularly important in our case since every subproject included flights carried out at different times or even on different days due to the large size of the surveyed area. Current research on the installation of dual frequency GPS onboard UAV platforms [68] would allow for direct georeferencing the generated geomatic products without the need of GCPs [15]. The geospatial accuracy of the final georeferencing achieved by using GCPs is a good result ($RMSE_{X,Y} < 30$ cm and $RMSE_Z < 55$ cm) considering the great extension of the UAV survey framework and taking into account that some studies reported a decrease in accuracy with large survey areas [64]. Other studies, such as that conducted by [11], obtained similar geospatial accuracy, but in our case, the error is closer to the lower limit that approximately matches the pixel size [69]. This accuracy was highly influenced by the even distribution of the GCPs through the UAV survey framework [70,71].

Within the comparison framework of the spatial information provided by UAV and WorldView-2 imagery, the higher correlations obtained between UAV orthomosaic resampled to match WorldView-2 image resolution, confirm that in the first successional stages of the vegetation on heterogeneous burned areas, the highest spatial resolution UAV mosaic (20 cm) does not provide redundant information [12] in relation to the satellite image. In this case, the ground variability scale associated with small vegetation patches, is larger than the coarser pixel sizes. Moreover, the stronger correlation between the UAV and WorldView-2 imagery found in the visible region of the spectrum was probably due to the similar relative spectral response in that region for the two sensors [44,55]. The NDVI map comparison between the UAV and WorldView-2 imagery conducted on a heterogeneous surface within the UAV survey framework, revealed again that coarser resolution satellite imagery cannot represent the spatial variability and patterns of areas characterized by very small vegetation patches [12]. The larger variance in reflectance values for each band of the highest spatial resolution UAV orthomosaic indicates that this product may be able to capture fine-scale ground patterns because of the greater spatial information provided by the dataset, improving the interpretation of landscape features. Some authors such as [18] stated that at this spatial scale, variations in sun azimuth and elevation will create variable shadow features throughout the day. This factor may introduce reflectance variability, and therefore, distort the calculation of spectral indices in ultra- high spatial resolution images. This effect in small targets is less significant in satellite imagery given its pixel size. However, within the NDVI map comparison framework, the sun azimuth and elevation of the UAV flight approximately matches the ones in WorldView-2 capture and the variability in reflectance values of both sensors was approximately the same as for the entire study area.

5. Conclusions

(1) The raw imagery acquired by the Parrot SEQUOIA multispectral camera presented some undesirable anomalies such as horizontal banding noise and non-homogeneous radiometry

across the image. Moreover, the irradiance sensor disconnections induced some radiometric anomalies across a small area of the multispectral mosaic that had to be discarded.

(2) The 16-bit imagery acquired on the UAV flights of the 3000 ha survey framework represents a large volume of data before processing it into a multispectral orthomosaic due to its ultra-high spatial resolution and the large size of the surveyed area. Nevertheless, this spatial resolution, which cannot be achieved with satellite platforms, could be crucial for developing spatial products to be used in post-fire management decision-making.

(3) Data processing was very labor-intensive, taking about 320 h to obtain the final multispectral orthomosaic. Due to the large imagery dataset generated on a UAV survey of a large area, the dataset processing must be subdivided regardless of the available processing capability. The obtained geospatial accuracy of the UAV multispectral orthomosaic was high ($RMSE_{X,Y}$ < 30 cm and $RMSE_Z$ < 55 cm) regarding the large extension of the surveyed area and the spatial resolution of the dataset.

(4) The spatial information provided by the ultra-high spatial resolution UAV multispectral orthomosaic was not redundant in these large and heterogeneous burned areas in comparison with high spatial resolution satellite imagery such as that provided by WorldView-2. The UAV orthomosaic could therefore improve the analysis and interpretation of fine-scale ground patterns.

Acknowledgments: This research was financially supported by the Spanish Ministry of Economy and Competitiveness, and the European Regional Development Fund (ERDF), within the framework of the GESFIRE project (grant number AGL2013-48189-C2-1-R); and by the Regional Government of Castilla y León within the framework of the FIRECYL project (grant number LE033U14). José Manuel Fernández-Guisuraga was supported by the European Social Fund and Youth Employment Initiative through the Spanish Ministry of Economy and Competitiveness (grant number PEJ2014-A-47268) and by a Predoctoral Fellowship of the Spanish Ministry of Education, Culture and Sport (FPU16/03070).

Author Contributions: S.S.-S. and L.C. conceived and designed the experiments; J.M.F.-G., E.S.-A., S.S.-S. and L.C. performed the experiments; J.M.F.-G. and E.S.-A. analyzed the data; E.S.-A. contributed materials/analysis tools; J.M.F.-G., E.S.-A., S.S.-S. and L.C. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

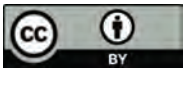
References

1. Seneviratne, S.I.; Nicholls, N.; Easterling, D.; Goodess, C.M.; Kanae, S.; Kossin, J.; Luo, Y.; Marengo, J.; McInnes, K.; Rahimi, M.; et al. Changes in climate extremes and their impacts on the natural physical environment. In *Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation*, 1st ed.; Field, C.B., Barros, V., Stocker, T.F., Qin, D., Dokken, D.J., Ebi, K.L., Mastrandrea, M.D., Mach, K.J., Plattner, G.K., Allen, S.K., et al., Eds.; Cambridge University Press: Cambridge, UK; New York, NY, USA, 2012; pp. 109–230. ISBN 978-11-0-702506-6.
2. Quintano, C.; Fernández-Manso, A.; Calvo, L.; Marcos, E.; Valbuena, L. Land surface temperature as potential indicator of burn severity in forest Mediterranean ecosystems. *Int. J. Appl. Earth Obs.* **2015**, *36*, 1–12. [CrossRef]
3. Poursanidis, D.; Chrysoulakis, N. Remote Sensing, natural hazards and the contribution of ESA Sentinels missions. *Remote Sens. Appl. Soc. Environ.* **2017**, *6*, 25–38. [CrossRef]
4. Álvarez, A.; Gracia, M.; Vayreda, J.; Retana, J. Patterns of fuel types and crown fire potential in *Pinus halepensis* forest in the Western Mediterranean Basin. *For. Ecol. Manag.* **2012**, *270*, 282–290. [CrossRef]
5. Vallejo, R.; Alloza, J.A. The restoration of burned lands: The case of eastern Spain. In *Large Forest Fires*, 1st ed.; Moreno, J.M., Ed.; Backhuys Publishers: Leiden, The Netherlands, 1998; pp. 91–108.
6. Tessler, N.; Wittenberg, L.; Greenbaum, N. Vegetation cover and species richness after recurrent forest fires in the Eastern Mediterranean ecosystem of Mount Carmel, Israel. *Sci. Total Environ.* **2016**, *572*, 1395–1402. [CrossRef] [PubMed]
7. Ruíz-Gallardo, J.R.; Castaño, S.; Calera, A. Application of remote sensing and GIS to locate priority intervention areas after wildland fires in Mediterranean systems: A case study from southeastern Spain. *Int. J. Wildland Fire* **2004**, *13*, 241–252. [CrossRef]
8. Chu, T.; Guo, X.; Takeda, K. Remote sensing approach to detect post-fire vegetation regrowth in Siberian boreal larch forest. *Ecol. Indic.* **2016**, *62*, 32–46. [CrossRef]

9. Viedma, O.; Torres, I.; Pérez, B.; Moreno, J.M. Modeling plant species richness using reflectance and texture data derived from QuickBird in a recently burned area of Central Spain. *Remote Sens. Environ.* **2012**, *119*, 208–221. [CrossRef]
10. Jung, M.; Tautenhahn, S.; Wirth, C.; Kattge, J. Estimating Basal Area of Spruce and Fir in Post-fire Residual Stands in Central Siberia Using Quickbird, Feature Selection, and Random Forests. *Procedia Comput. Sci.* **2013**, *18*, 2386–2395. [CrossRef]
11. Zhang, J.; Hu, J.; Lian, J.; Fan, Z.; Ouyang, X.; Ye, W. Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring. *Biol. Conserv.* **2016**, *198*, 60–69. [CrossRef]
12. Matese, A.; Toscano, P.; Di Gennaro, S.F.; Genesio, L.; Vaccari, F.P.; Primicerio, J.; Belli, C.; Zaldei, A.; Bianconi, R.; Gioli, B. Intercomparison of UAV, Aircraft and Satellite Remote Sensing Platforms for Precision Viticulture. *Remote Sens.* **2015**, *7*, 2971–2990. [CrossRef]
13. Anderson, K.; Gaston, K.J. Lightweight unmanned aerial vehicles will revolutionize spatial ecology. *Front. Ecol. Environ.* **2013**, *11*, 138–146. [CrossRef]
14. Tang, L.; Shao, G. Drone Remote Sensing for Forestry Research and Practices. *J. For. Res* **2015**, *26*, 791–797. [CrossRef]
15. Ribeiro-Gomes, K.; Hernandez-Lopez, D.; Ballesteros, R.; Moreno, M.A. Approximate georeferencing and automatic blurred image detection to reduce the costs of UAV use in environmental and agricultural applications. *Biosyst. Eng.* **2016**, *151*, 308–327. [CrossRef]
16. Zhou, J.; Pavek, M.J.; Shelton, S.C.; Holden, Z.J.; Sankaran, S. Aerial multispectral imaging for crop hail damage assessment in potato. *Comput. Electron. Agric.* **2016**, *127*, 406–412. [CrossRef]
17. Beaty, R.M.; Taylor, A.H. Spatial and Temporal Variation of Fire Regimes in a Mixed Conifer Forest Landscape, Southern Cascades, California, USA. *J. Biogeogr.* **2001**, *28*, 955–966. [CrossRef]
18. McKenna, P.; Erskine, P.D.; Lechner, A.M.; Phinn, S. Measuring fire severity using UAV imagery in semi-arid central Queensland, Australia. *Int. J. Remote Sens.* **2017**, *38*, 4244–4264. [CrossRef]
19. Torresan, C.; Berton, A.; Carotenuto, F.; Di Gennaro, S.F.; Gioli, B.; Matese, A.; Miglietta, F.; Vagnoli, C.; Zaldei, A.; Wallace, L. Forestry applications of UAVs in Europe: A review. *Int. J. Remote Sens.* **2016**, *38*, 1–21. [CrossRef]
20. Hardin, P.J.; Jensen, R.R. Small-Scale Unmanned Aerial Vehicles in Environmental Remote Sensing: Challenges and Opportunities. *GISci. Remote Sens.* **2011**, *48*, 99–111. [CrossRef]
21. Jones, G.P.; Pearlstine, L.G.; Percival, H.F. An assessment of small unmanned aerial vehicles for wildlife research. *Wildl. Soc. B* **2006**, *34*, 750–758. [CrossRef]
22. Koski, W.R.; Allen, T.; Ireland, D.; Buck, G.; Smith, P.R.; Macrander, A.M.; Halick, M.A.; Rushing, C.; Sliwa, D.J.; McDonald, T.L. Evaluation of an unmanned airborne system for monitoring marine mammals. *Aquat. Mamm.* **2009**, *35*, 347–357. [CrossRef]
23. Israel, M. A UAV-based roe deer fawn detection system. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Conference on Unmanned Aerial Vehicle in Geomatics, Zurich, Switzerland, 14–16 September 2011.
24. Chabot, D.; Bird, D.M. Evaluation of an off-the-shelf Unmanned Aircraft System for surveying flocks of geese. *Waterbirds* **2012**, *35*, 170–174. [CrossRef]
25. Sarda-Palomera, F.; Bota, G.; Viñolo, C.; Pallarés, O.; Sazatornil, V.; Brotons, L.; Gomáriz, S.; Sarda, F. Fine-scale bird monitoring from light unmanned aircraft systems. *IBIS* **2012**, *154*, 177–183. [CrossRef]
26. Vermeulen, C.; Lejeune, P.; Lisein, J.; Sawadogo, P.; Bouche, P. Unmanned aerial survey of elephants. *PLoS ONE* **2013**, *8*, e54700. [CrossRef] [PubMed]
27. Floris, A.; Clementel, F.; Colle, G.; Gubert, F.; Bertoldi, L.; De Lorenzi, G. Estimation of Wood Volume with Photogrammetric Data Sensing from UAV on Small Surfaces: A Case Study in Trentino. In Proceedings of the 16th ASITA National Conference, Vicenza, Italy, 6–9 November 2012.
28. Getzin, S.; Wiegand, K.; Schöning, I. Assessing biodiversity in forests using very high-resolution images and unmanned aerial vehicles. *Methods Ecol. Evol.* **2012**, *3*, 397–404. [CrossRef]
29. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV–LiDAR system with application to forest inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [CrossRef]
30. Dandois, J.P.; Ellis, E.C. High spatial resolution three-dimensional mapping of vegetation spectral dynamics using computer vision. *Remote Sens. Environ.* **2013**, *136*, 259–276. [CrossRef]

31. Lisein, J.; Pierrot-Deseilligny, M.; Bonnet, S.; Lejeune, P. A Photogrammetric Workflow for the Creation of a Forest Canopy Height Model from Small Unmanned Aerial System Imagery. *Forests* **2013**, *4*, 922–944. [CrossRef]
32. Puliti, S.; Orka, H.O.; Gobakken, T.; Naesset, E. Inventory of small forest areas using an Unmanned Aerial System. *Remote Sens.* **2015**, *7*, 9632–9654. [CrossRef]
33. Fritz, A.; Kattenborn, T.; Koch, B. UAV-based photogrammetric point clouds-tree stem mapping in open stands in comparison to terrestrial laser scanner point clouds. In Proceedings of the UAV-g2013, Rostock, Germany, 4–6 September 2013.
34. Gini, R.; Passoni, D.; Pinto, L.; Sona, G. Use of Unmanned Aerial Systems for Multispectral Survey and Tree Classification: A Test in a Park Area of Northern Italy. *Eur. J. Remote Sens.* **2014**, *47*, 251–269. [CrossRef]
35. Jaakkola, A. Low-cost Mobile Laser Scanning and its Feasibility for Environmental Mapping. Ph.D. Dissertation, Aalto University, Espoo, Finland, 2015.
36. Michez, A.; Piégay, H.; Lisein, J.; Claessens, H.; Lejeune, P. Classification of Riparian Forest Species and Health Condition Using Multi-Temporal and Hyperspatial Imagery from Unmanned Aerial System. *Environ. Monit. Assess.* **2016**, *188*, 1–19. [CrossRef] [PubMed]
37. Aicardi, I.; Garbarino, M.; Lingua, A.; Lingua, E.; Marzano, R.; Piras, M. Monitoring Post-Fire Forest Recovery Using Multitemporal Digital Surface Models Generated from Different Platforms. In Proceedings of the EARSeL Symposium, Bonn, Germany, 20–24 June 2016.
38. Fraser, R.H.; van der Sluijs, J.; Hall, R.J. Calibrating Satellite-Based Indices of Burn Severity from UAV-Derived Metrics of a Burned Boreal Forest in NWT, Canada. *Remote Sens.* **2017**, *9*, 279. [CrossRef]
39. Cruz, H.; Eckert, M.; Meneses, J.; Martínez, J.F. Efficient Forest Fire Detection Index for Application in Unmanned Aerial Systems (UASs). *Sensors* **2016**, *16*, 893. [CrossRef] [PubMed]
40. Pérez-Ortiz, M.; Peña, J.M.; Gutiérrez, P.A.; Torres-Sánchez, J.; Hervás-Martínez, C.; López-Granados, F. Selecting patterns and features for between- and within- crop-row weed mapping using UAV-imagery. *Expert Syst. Appl.* **2016**, *47*, 85–94. [CrossRef]
41. Misopolinos, L.; Zalidis, C.H.; Liakopoulos, V.; Stavridou, D.; Katsigiannis, P.; Alexandridis, T.K.; Zalidis, G. Development of a UAV system for VNIR-TIR acquisitions in precision agriculture. In Proceedings of the Third International Conference on Remote Sensing and Geoinformation of the Environment, Paphos, Cyprus, 16–19 March 2015.
42. Tian, J.; Wang, L.; Li, X.; Gong, H.; Shi, C.; Zhong, R.; Liu, X. Comparison of UAV and WorldView-2 imagery for mapping leaf area index of mangrove forest. *Int. J. Appl. Earth Obs.* **2017**, *61*, 22–31. [CrossRef]
43. Calvo, L.; Santalla, S.; Valbuena, L.; Marcos, E.; Tárrega, R.; Luis-Calabuig, E. Post-fire natural regeneration of a *Pinus pinaster* forest in NW Spain. *Plant Ecol.* **2008**, *197*, 81–90. [CrossRef]
44. Parrot. Available online: https://community.parrot.com/t5/Sequoia/bd-p/Sequoia (accessed on 3 June 2017).
45. Santesteban, L.G.; Di Gennaro, S.F.; Herrero-Langreo, A.; Miranda, C.; Royo, J.B.; Matese, A. High-resolution UAV-based thermal imaging to estimate the instantaneous and seasonal variability of plant water status within a vineyard. *Agric. Water Manag.* **2017**, *183*, 49–59. [CrossRef]
46. Kelcey, J.; Lucieer, A. Sensor Correction of a 6-Band Multispectral Imaging Sensor for UAV Remote Sensing. *Remote Sens.* **2012**, *4*, 1462–1493. [CrossRef]
47. Burkart, A.; Aasen, H.; Alonso, L.; Menz, G.; Bareth, G.; Rascher, U. Angular Dependency of Hyperspectral Measurements over Wheat Characterized by a Novel UAV Based Goniometer. *Remote Sens.* **2015**, *7*, 725–746. [CrossRef]
48. Koik, B.T.; Ibrahim, H. A literature survey on blur detection algorithms for digital imaging. In Proceedings of the 1st International Conference on Artificial Intelligence, Modelling and Simulation, Kota Kinabalu, Malaysia, 3–5 December 2013.
49. Pix4D. Available online: https://pix4d.com/product/pix4dmapper-photogrammetry-software/ (accessed on 8 December 2016).
50. McGlone, J.C. *Manual of Photogrammetry*, 6th ed.; American Society for Photogrammetry and Remote Sensing (ASPRS): Bethesda, MD, USA, 2013; ISBN 978-15-7-083071-6.
51. Ruzgienė, B.; Berteška, T.; Gečyte, S.; Jakubauskienė, E.; Aksamitauskas, V.C. The surface modelling based on UAV Photogrammetry and qualitative estimation. *Measurement* **2015**, *73*, 619–627. [CrossRef]
52. Zahawi, R.A.; Dandois, J.P.; Holl, K.D.; Nadwodny, D.; Reid, J.L.; Ellis, E.C. Using lightweight unmanned aerial vehicles to monitor tropical forest recovery. *Biol. Conserv.* **2015**, *186*, 287–295. [CrossRef]

53. PNOA. Available online: http://pnoa.ign.es/caracteristicas-tecnicas (accessed on 21 December 2017).
54. ESRI. Available online: http://desktop.arcgis.com/es/arcmap/10.3/main/get-started/whats-new-in-arcgis-1031.htm (accessed on 22 June 2016).
55. DigitalGlobe. Available online: http://global.digitalglobe.com (accessed on 7 January 2016).
56. Matthew, M.; Adler-Golden, S.; Berk, A.; Felde, G.; Anderson, G.; Gorodetzky, D.; Paswaters, S.; Shippert, M. Atmospheric correction of spectral imagery: Evaluation of the FLAASH algorithm with AVIRIS data. In Proceedings of the 32nd Applied Imagery Pattern Recognition Workshop, Washington, DC, USA, 15–17 October 2003.
57. ENVI. Available online: http://www.harrisgeospatial.com/SoftwareTechnology/ENVI.aspx (accessed on 18 September 2017).
58. Laliberte, A.S.; Goforth, M.A.; Steele, C.M.; Rango, A. Multispectral Remote Sensing from Unmanned Aircraft: Image Processing Workflows and Applications for Rangeland Environments. *Remote Sens.* **2011**, *3*, 2529–2551. [CrossRef]
59. Ortega-Terol, D.; Hernandez-Lopez, D.; Ballesteros, R.; Gonzalez-Aguilera, D. Automatic Hotspot and Sun Glint Detection in UAV Multispectral Images. *Sensors* **2017**, *17*, 2352. [CrossRef] [PubMed]
60. Tellidis, I.; Levin, E. Photogrammetric Image Acquisition with Small Unmanned Aerial Systems. In Proceedings of the ASPRS 2014 Annual Conference Proceedings, Louisville, KY, USA, 23–28 March 2014.
61. Stark, B.; Zhao, T.; Chen, Y. An analysis of the effect of the bidirectional reflectance distribution function on remote sensing imagery accuracy from Small Unmanned Aircraft Systems. In Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016.
62. Roy, D.P.; Li, J.; Zhang, H.K.; Yan, L.; Huang, H.; Li, Z. Examination of Sentinel-2A multi-spectral instrument (MSI) reflectance anisotropy and the suitability of a general method to normalize MSI reflectance to nadir BRDF adjusted reflectance. *Remote Sens. Environ.* **2017**, *199*, 25–38. [CrossRef]
63. Taboada, A.; Tárrega, R.; Marcos, E.; Valbuena, L.; Suárez-Seoane, S.; Calvo, L. Fire recurrence and emergency post-fire management influence seedling recruitment and growth by altering plant interactions in fire-prone ecosystems. *For. Ecol. Manag.* **2017**, *402*, 63–75. [CrossRef]
64. Koci, J.; Jarihani, B.; Leon, J.X.; Sidle, R.C.; Wilkinson, S.N.; Bartley, R. Assessment of UAV and Ground-Based Structure from Motion with Multi-View Stereo Photogrammetry in a Gullied Savanna Catchment. *ISPRS Int. Geo-Inf.* **2017**, *6*, 328. [CrossRef]
65. Laporterie-Dejean, F.; Boissezon, H.; Flouzat, G.; Lefevre-Fonollosa, M.J. Thematic and statistical evaluations of five panchromatic/multispectral fusion methods on simulated PLEIADES-HR images. *Inf. Fusion* **2005**, *6*, 193–212. [CrossRef]
66. Thomas, C.; Ranchin, T.; Wald, L.; Chanussot, J. Synthesis of multispectral images to high spatial resolution: A critical review of fusion methods based on remote sensing physics. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 1301–1312. [CrossRef]
67. Tu, T.M.; Su, S.C.; Shyu, H.C.; Huang, P.S. A new look at IHS-like image fusion methods. *Inf. Fusion* **2001**, *2*, 177–186. [CrossRef]
68. Karl-Lehmann, J.R.; Nieberding, F.; Prinz, T.; Knoth, C. Analysis of Unmanned Aerial System-Based CIR Images in Forestry—A New Perspective to Monitor Pest Infestation Levels. *Forests* **2015**, *6*, 594–612. [CrossRef]
69. Mesas-Carrascosa, F.J.; Torres-Sánchez, J.; Clavero-Rumbao, I.; García-Ferrer, A.; Peña, J.M.; Borra-Serrano, I.; López-Granados, F. Assessing Optimal Flight Parameters for Generating Accurate Multispectral Orthomosaicks by UAV to Support Site-Specific Crop Management. *Remote Sens.* **2015**, *7*, 12793–12814. [CrossRef]
70. Shahbazi, M.; Sohn, G.; Théau, J.; Menard, P. Development and Evaluation of a UAV-Photogrammetry System for Precise 3D Environmental Modeling. *Sensors* **2015**, *15*, 27493–27524. [CrossRef] [PubMed]
71. Harwin, S.; Lucieer, A. Assessing the Accuracy of Georeferenced Point Clouds Produced via Multi-View Stereopsis from Unmanned Aerial Vehicle (UAV) Imagery. *Remote Sens.* **2012**, *4*, 1573–1599. [CrossRef]



sensors

MDPI

Article

Spatial Scale Gap Filling Using an Unmanned Aerial System: A Statistical Downscaling Method for Applications in Precision Agriculture

Leila Hassan-Esfahani [1,*], Ardeshir M. Ebtehaj [2], Alfonso Torres-Rua [3] and Mac McKee [1]

[1] Civil & Environmental Engineering, Utah State University, Logan, UT 84322, USA; mac.mckee@usu.edu
[2] College of Science & Engineering, University of Minnesota, Minneapolis, MN 55455, USA; ebtehaj@umn.edu
[3] Utah Water Research Laboratory, Utah State University, Logan, UT 84322, USA; a.torres@aggiemail.usu.edu
* Correspondence: leila.esfahani@aggiemail.usu.edu; Tel.: +1-435-764-3151

Received: 13 June 2017; Accepted: 10 August 2017; Published: 14 September 2017

Abstract: Applications of satellite-borne observations in precision agriculture (PA) are often limited due to the coarse spatial resolution of satellite imagery. This paper uses high-resolution airborne observations to increase the spatial resolution of satellite data for related applications in PA. A new variational downscaling scheme is presented that uses coincident aerial imagery products from "AggieAir", an unmanned aerial system, to increase the spatial resolution of Landsat satellite data. This approach is primarily tested for downscaling individual band Landsat images that can be used to derive normalized difference vegetation index (NDVI) and surface soil moisture (SSM). Quantitative and qualitative results demonstrate promising capabilities of the downscaling approach enabling effective increase of the spatial resolution of Landsat imageries by orders of 2 to 4. Specifically, the downscaling scheme retrieved the missing high-resolution feature of the imageries and reduced the root mean squared error by 15, 11, and 10 percent in visual, near infrared, and thermal infrared bands, respectively. This metric is reduced by 9% in the derived NDVI and remains negligibly for the soil moisture products.

Keywords: Landsat; UAV; downscaling; NDVI; soil moisture; precision agriculture

1. Introduction

Precision agriculture (PA) is the art and science of using advanced computational and remote sensing technologies to produce overall gains in profitability and environmental stewardship. PA relies on airborne and satellite remote sensing to support adaptive management practices for different areas within agricultural fields based on varying soil types, landscape, water demand, and crop stresses [1,2]. Passive airborne remote sensing observations have been used to estimate several agricultural variables, such as soil moisture [3,4], chlorophyll [5], crop types [6], disease types [7], yield [8], and irrigation water demand [9]. In this study, we focus on downscaling the visible to near-infrared satellite imageries and the derived normalized difference vegetation index (NDVI), and surface soil moisture (SSM) estimates as agricultural variables of interest.

NDVI has been used extensively in several agricultural studies, and high-resolution soil moisture data are essential for understanding the dynamics and productivity of hydrologic [10,11] and agricultural [12] systems. In the past few decades, space-borne remote sensing of soil moisture has progressed significantly through the launch of several satellites, including the Soil Moisture and Ocean Salinity (SMOS) satellite by the European Space Agency (ESA), and the Soil Moisture Active Passive (SMAP) satellite by NASA. A major advantage of remote sensing in microwave bands is that soil moisture can be measured regardless of time of day or atmospheric conditions. However, while these soil moisture data provide unique opportunities to better understand mesoscale land

atmosphere interactions, they are incompatible to provide information at scaled resolve scales that are useful in the context of agricultural management or ecological modeling [13]. As a result, research devoted to increasing the spatial resolution of satellite soil moisture data [3,14,15] has led to the developments of a variety of spatial downscaling methods.

A range of techniques of varying complexities has been developed for downscaling. Merlin et al. [16] developed an empirical radiative transfer model to downscale the near-surface soil moisture data by a factor of 8 and reported a 3% reduction in the mean squared error. In a more recent study, [10] they improved the spatial resolution of a passive microwave-derived product from a resolution of approximately 40 km to approximately 4 km using a disaggregation method. This product used MODIS 1 km resolution data to provide more accurate soil moisture data for a range of agricultural and hydrologic applications. Similar research by Choi and Hur [17] improved the spatial representation of soil moisture products using disaggregation approaches and MODIS data. Chauhan et al. [18] developed an approach to evaluate soil moisture at a resolution of about one kilometer by applying optical/infrared data to passive microwave derived soil moisture data at a resolution of about 25 km. In that study, NDVI was used to improve soil moisture downscaling by applying the "Universal Triangle" approach.

Most of these studies have focused on developing microwave soil moisture downscaling methods that provide soil moisture products at a resolution of a few kilometers. However, this resolution is still too coarse for agricultural applications. Here, we propose a downscaling method that aims to increase the resolution of Landsat imagery by a factor of 2 to 4 to produce improved derivative products, such as soil moisture or NDVI.

The proposed downscaling model relies on super-resolution techniques that use sparse approximation methods, which have shown advantages in enhancing the resolution of precipitation fields [19]. Succinctly stated, sparse approximation methods allow us to better recover high-frequency features that have been lost as a result of the smoothing effects of a low-resolution sensor. The super resolution method needs a joint library of low- and high-resolution imagery that can be used to train the downscaling scheme. To this end, we use coincident observations of the low-resolution Landsat imageries (RGB, NIR, Thermal Infrared) and the derived agricultural (NDVI and SM) products at resolution 30 m and spectral data products derived from high-resolution airborne imageries at 0.15-m resolution by the AggieAir platform (AggieAir, 2016) to populate the low- and high-resolution libraries.

The paper is organized as follows. Section 2 introduces the study area and the remote sensing platforms used. The proposed downscaling scheme and calibration of the raw data for calculation of the derivative products are also explained in this section. Section 3 presents the results of the downscaling scheme for individual spectral bands, NDVI and SM products. Section 4 discusses the concluding remarks and points out to potential future directions.

2. Materials and Methods

2.1. Study Area

The study area includes two agricultural fields located in Scipio, Utah, USA, centered at 39°14′ N 112°6′ W with the area of 1.35 km^2 (Figure 1). Oats and alfalfa are the main crops, grown from May to September, which are irrigated by modern center pivot sprinkler systems. This study was based on data collected on 1 June, 9 June, and 17 June 2013, a time frame that covers a growing cycle for alfalfa. Figure 1 shows the natural color images of the field on 1 June.

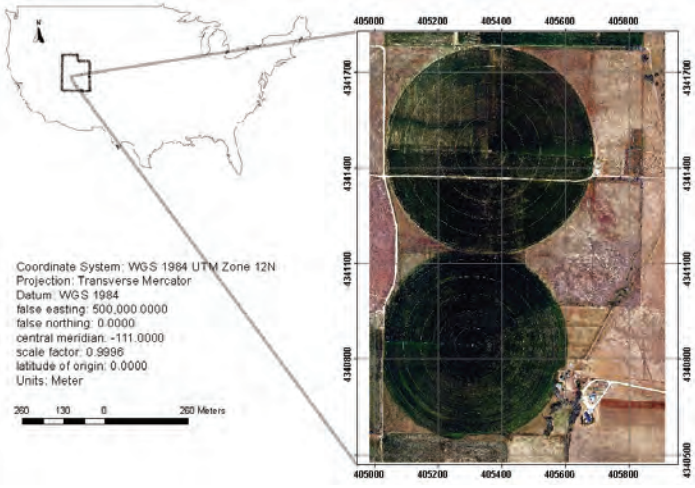


Figure 1. The location of the study area in Utah, United States.

2.2. AggieAir and Landsat

AggieAir is a collection of remote sensing platforms and payloads developed at the Utah Water Research Laboratory at Utah State University [20]. These completely autonomous unmanned aerial vehicles (UAVs) were programmed to navigate over the study area and capture images in the blue, green, red, and near infrared (NIR) spectra at 0.15 m resolution and in the thermal band at about 0.6 m resolution. The wavelength range peaks at around 420, 500, 600, and 800 nm, respectively, for the blue, green, red, and NIR bands. The thermal images were down-processed to 0.15 m resolution to achieve resolutions consistent with other AggieAir products [21]. In order to study and compare AggieAir and Landsat data, flight experiments were conducted to coincide with Landsat overpasses (on the same dates and at about 18:04 scene center time). In addition, the experiments were designed to cover the maximum variation of land cover as a result of crop growth stage. Therefore, the first growing cycle of alfalfa, from emergence to harvest, was selected for the experiments. This growing cycle was covered by four Landsat overpasses; however, one overpass was removed from the experimentation plan due to cloud cover (24 May). The UAV flew over the study area on 1 June, 9 June, and 17 June 2013.

2.3. Landsat and AggieAir Reflectance Homogenization

The visible camera used in the AggieAir flights is a Canon S-95 (with a 10-megapixel charge-coupled device (CCD) sensor with an International Standards Organization (ISO) range of 80–3200 [5]. NIR imagery was captured with the same camera modified to replace the manufacturer's optical filter with a Wratten 87 NIR filter that allows NIR wavelengths of 750 nm and larger. The relative spectral responses of the VIS–NIR cameras were obtained using the algorithm provided by Jiang et al. [22]. AggieAir also carries a small, low-power, microbolometer thermal camera from Infrared Cameras Inc. (ICI) [23] (Infrared Cameras Incorporated, 2012). The AggieAir and Landsat VIR spectral responses are shown in Figure 2.

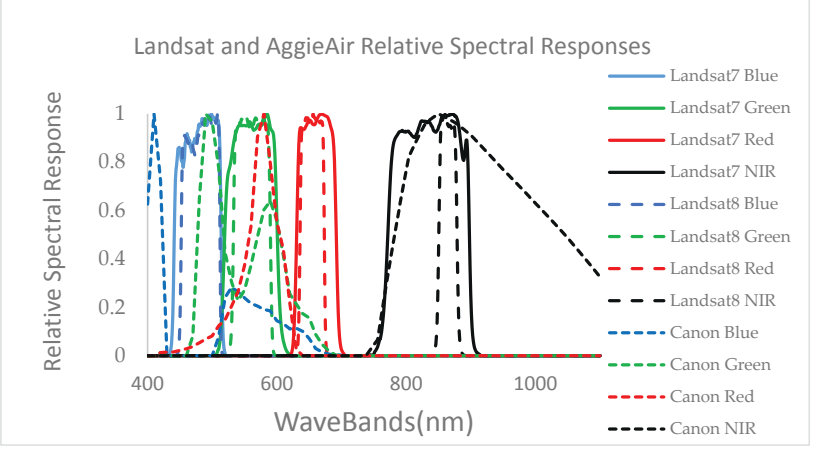


Figure 2. The spectral responses of AggieAir Canon cameras versus Landsat sensors (the spectral response is scaled for all individual bands from 0 to 1).

Figure 2 shows that the spectral responses of the AggieAir cameras are by no means identical to those of the Landsat sensors. Moreover, the significance of atmospheric contamination in the observations of the two sensors is drastically different. These differences often lead to some systematic differences between the observations by the two sensors. For potential biases, a de-biasing (bias fixing) step was conducted prior to downscaling. To that end, we assumed that the Landsat data represent the baseline product for bias correction and applied the bias correction to the high-resolution imagery acquired from the UAV platform.

Given the spectral characteristics of the Cannon cameras (Figure 2) used in the study (Figure 1), it is evident that 30-m aggregated AggieAir imagery was not congruent at the pixel level with the Landsat reflectance standard product (from Landsat 7 ETM+ and Landsat 8 OLI). Using USGS-developed Landsat Point Spread Function information for each of the Landsat 7 and 8 bands (USGS, 2015 personal communication), AggieAir reflectance bands were aggregated up to become compatible with the 30-m resolution Landsat imagery over the study area. We used histogram matching techniques for bias correction [24] on AggieAir imagery (Figure 3). This technique uses a probability matching function that adjusts the histogram of the deviated image from the AggieAir platform to match the histogram of the reference Landsat imageries.

Figure 3a shows the original Landsat NIR band surface reflectance for 1 June 2013 over the study area, and Figure 3b shows the aggregated NIR band reflectance image from AggieAir. Figure 3c shows the unbiased AggieAir NIR reflectance image at 30-m resolution with histogram matching applied. Figure 3d,e demonstrate the effectiveness of this approach with pixel-level scatter plots of the original and unbiased aggregated AggieAir NIR band versus the corresponding Landsat NIR band at a resolution of 30 m.

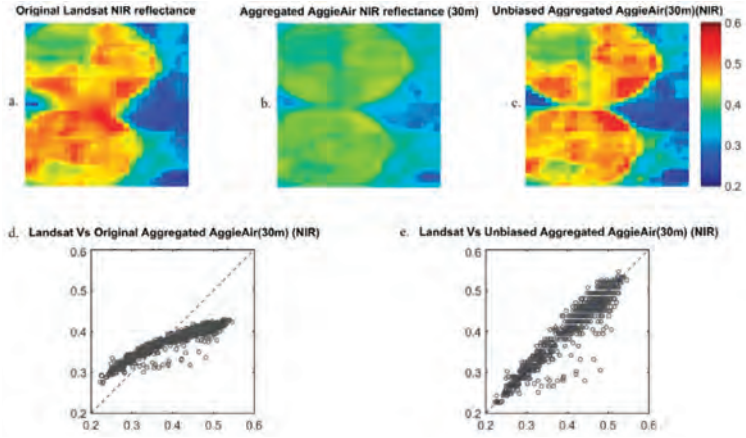


Figure 3. (**a**) Original Landsat NIR reflectance image; (**b**) aggregated AggieAir NIR image (resolution: 30 m); (**c**) Unbiased aggregated AggieAir NIR image (resolution: 30 m); (**d**) Scatterplot of Landsat versus the original aggregated AggieAir NIR image; (**e**) Scatter lot of Landsat versus the unbiased aggregated AggieAir NIR image.

2.4. Downscaling Individual Spectral Bands

To downscale the Landsat data using coincident high-resolution AggieAir observations, we conducted downscaling experiments on all Landsat red/green/blue and infrared (VIR) bands with scaling factors of 2 and 4. This corresponds to the 15 and 7.5 m spatial resolution grids, respectively.

As previously noted, several downscaling methods have been used for agricultural applications. Here, we focus on a super-resolution approach that has been successfully implemented for downscaling remote sensing data [19]. The method relies on a sparse approximation technique that allows us to recover high-resolution imagery from low-resolution counterparts by solving an inverse problem. In lieu of solving a large-scale inverse problem for the entire field of observation, the method focuses on small-scale constitutive elements of the observation field that consist of a few nearby pixels, often called a patch. In other words, the entire low-resolution observation field is broken down into patches, and then the method seeks to obtain their high-resolution counterparts through solving small-scale inverse problems that are computationally tractable. This method relies on a set of training examples that implicitly encodes the correspondence between the high- and low-resolution patches. These training examples are collected in two matrices called low- and high-resolution dictionaries. The method attempts to reconstruct the high-resolution patches with a few patches in the low-resolution dictionary by using a sparse approximation technique. It then assumes that the low- and high-resolution fields are geometrically similar and, thus, uses the same coefficients to combine the training patches in the high-resolution dictionary.

Specifically, let us assume that an m-by-n low-resolution (LR) field of the Landsat imagery is denoted by $X_l \in R^{m \times n}$ (e.g., 30-m grid), while its high-resolution (HR) counterpart from the AggieAir imagery is $X_h \in R^{sm \times sn}$ (i.e., 15 m or 7.5 m with downscaling factors, respectively, s = 2 and 4). Let us assume that the low-resolution images can be upscaled to the same size as the high-resolution images using an interpolation operator, denoted by $I_s(X_l) \in R^{sm \times sn}$. Consequently, we can obtain a residual field as $X_r = X_h - I_s(X_l)$, which contains the lost information between the high- and low-resolution fields that we wish to recover through the downscaling approach using a limited number of coincident low- and high-resolution images from Landsat and AggieAir. The optimal estimation of the residual field was carried out on a patch-wise basis, as previously described, to avoid a cumbersome optimization problem. The downscaling uses a large number of training examples of

LR and HR patches to reconstruct the residual field. Let us assume that those patches are vectorized and stored in column space of the low- $D_l = [x_{l1}, x_{l2}, \ldots, x_{lM}] \in R^{q \times M}$ and high-resolution $D_h = [x_{h1}, x_{h2}, \ldots, x_{hM}] \in R^{s^2 q \times M}$ dictionaries. For a given low-resolution image, the method breaks it down into N vectorized low-resolution patches $Y_l = \left\{ y_{li} \in R^{q \times l} \right\}_{i=1}^{N}$. Each patch is estimated with a linear combination of a few patches in the LR dictionary (i.e., column vectors of D_l). In this study, a Greedy method called orthogonal matching pursuit (OMP) by Mallat and Zhang [25] has been used. Initializing the estimation residual by the observed signal, at each iteration, first the support of the representation coefficients is updated by selecting an atom which has the maximum inner product with the estimation residual and then, given the support set, the values of the representation coefficients are being updated through an ordinary least squares. The iterations continue until a certain number of atoms is selected or the magnitude of the estimation residual falls below a certain threshold [19].

2.5. Developing Agricultural Variables

To demonstrate application of the proposed downscaling scheme for use in precision agriculture, NDVI and soil moisture (SM) products are derived from the downscaled original VIR imagery at resolutions of 15 and 7.5 m.

For SM estimations, original Landsat spectral imagery and the downscaled counterparts were fed into a previously developed soil moisture retrieval model [26]. The adapted method relies on Bayesian artificial neural network (Bay-ANN) algorithms that learn the soil moisture variability using a set of training examples of inputs (either the normal 30 m resolution Landsat imagery or downscaled imagery) and outputs (soil moisture). To provide the outputs, an intensive ground sampling program was conducted over the study area during the Landsat overpasses. The data collection procedure was designed to cover the maximum spatial distribution of soil moisture with respect to the irrigation system function, crop type, and soil texture characteristics [26,27].

3. Results and Discussion

3.1. Downscaled Spectral Bands

Imagery over the study area in the red, green, blue, NIR, and thermal spectral bands was used to study the quantitative performance of the proposed downscaling scheme. The scheme requires the development of a dictionary training set, i.e., a representative set of coincidental pairs of low- and high-resolution patches of images. The model restored the optimal high-resolution estimate using supervised learning from 8766 training patches obtained from three pairs of high- and low-resolution images from Landsat low- and the AggieAir high-resolution observations on 1 June, 9 June, and 17 June 2013. Hereafter, two different scenarios are defined; in the first scenario, all high and low resolution counterparts from three flight dates were used to populate the dictionary and, in the second scenario, one pair at a time is left out for validation purposes.

In the first scenario, all three pairs of high- and low-resolution images (all 8766 patches) were used to train the downscaling model. The downscaled products were compared with bias-corrected high-resolution AggieAir products as our benchmark. In the second scenario, the model was trained with two pairs (out of the three available high- and low-resolution pairs), and one pair was retained for testing.

To illustrate the effectiveness of the downscaling model in the first scenario, the 30 × 30 m low-resolution Landsat images (individual bands of red, green, blue, NIR, and thermal infrared) of the study area were downscaled to 15 and 7.5 m grid sizes. Figure 4 depicts the low-resolution NIR band imagery from Landsat (Figure 4a) acquired on 1 June 2013 at about 18:04 center time (UTC time), and its high-resolution counterpart from the AggieAir platform (Figure 4b). Figure 4c shows the NIR image downscaled from 30 m to the finer resolution of 15 m using the downscaling scheme. The results demonstrate how well the learning procedure performed and how some high frequency features can be recovered. The same procedure was repeated for a scale factor of 4, as shown in Figure 4e. The results show that suitable information in the training set leads to an improved algorithm success. As long

as the model can find very close or identical pairs of patches in the dictionary, it can successfully generate downscaled products. Figure 4d,f show the results from the same downscaling scheme for scale factors of 2 and 4 where the high resolution patches of interest were not part of the training dataset (second scenario).

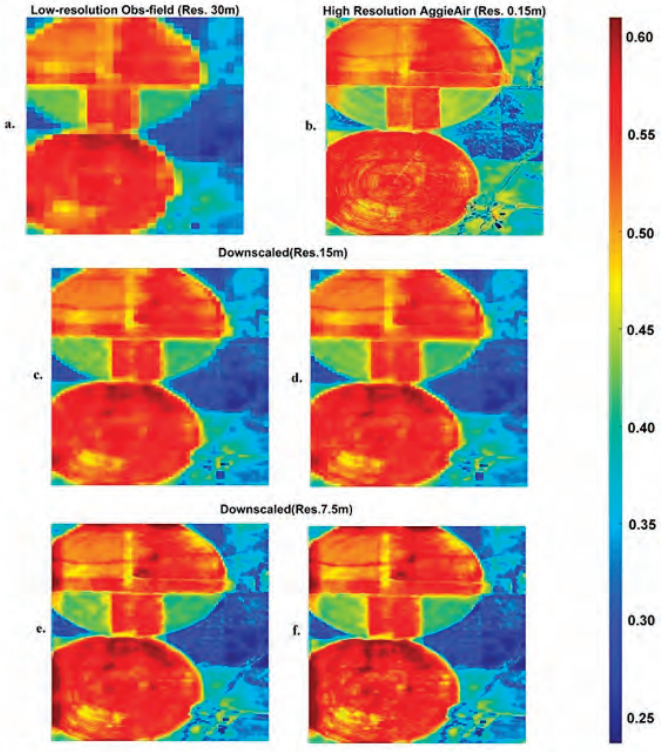


Figure 4. (**a**) Low-resolution observation (Landsat NIR); (**b**) high-resolution observation (AggieAir NIR); (**c**,**e**) downscaled NIR at 15 and 7.5 m resolution, respectively (first scenario); (**d**,**f**) downscaled NIR 15 and 7.5 m resolution, respectively (second scenario).

The differences between the downscaling results from the above two scenarios are illustrated in Figure 5 on a pixel-wise basis, comparing reflectance values of aggregated imagery from AggieAir with downscaled imagery resulting from our model. AggieAir images are aggregated to reach the same resolution as the downscaling results. This step has been carried out to achieve spatial consistency and make it feasible to perform a pixel-wise comparison. Figure 5 shows the one-by-one scatter plots of the downscaled fields from Landsat vs the upscaled field from AggieAir. Figure 5a,b show the scatter plots of the downscaled Landsat NIR field at scale factor of 2 (commensurate with resolutions of 15 m), versus the upscaled AggieAir image at the same resolution. The same plots are shown in Figure 5c,d for a scale factor of 4. The image of interest was included in the training dataset (first scenario) for the results shown in Figure 5a,c, while the information of the image of interest was not included in training examples (second scenario) for the results shown in Figure 5b,d. The downscaling quality metrics are reported in Table 1. These matrices are presented as percentage values that show the relative reduction in root mean square error as a quality metric. Note that the comparisons are performed after the biased correction approach was applied. The metrics show that the differences between the downscaled and benchmark images have decreased significantly in most of the cases.

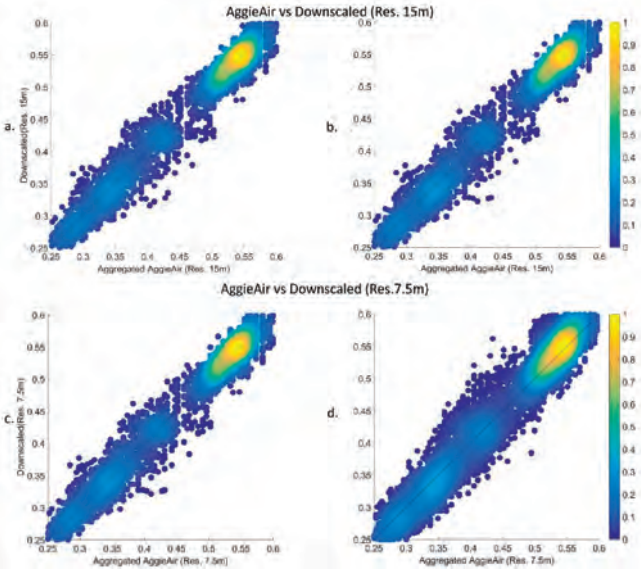


Figure 5. Pixel-wise comparison of AggieAir imagery versus downscaled NIR images as at resolution 15 and 7.5 m for the first (**a**,**c**) and second (**b**,**d**) scenarios, respectively.

Table 1. The root mean squared error (RMSE) quantifying the quality of the downscaling scheme for different scaling factors for all spectral bands.

Spectral Band	Downscaling Level	Including in the Training Set	RMSE		
			1 June 2013	9 June 2013	17 June 2013
			Improvement Ratio	Improvement Ratio	Improvement Ratio
Red	2	YES	14%	16%	15%
		NO	8%	11%	10%
	4	YES	25%	27%	24%
		NO	17%	20%	13%
Green	2	YES	14%	13%	15%
		NO	7%	8%	15%
	4	YES	25%	26%	23%
		NO	13%	15%	10%
Blue	2	YES	13%	15%	12%
		NO	5%	8%	5%
	4	YES	23%	29%	20%
		NO	9%	14%	9%
NIR	2	YES	13%	20%	16%
		NO	5%	10%	6%
	4	YES	14%	23%	5%
		NO	6%	14%	2%
Thermal	2	YES	9%	7%	6%
		NO	1%	1%	2%
	4	YES	0.12%	0.16%	0.16%
		NO	0.07%	0.09%	0.06%

As is evident from model statistics and a visual inspection of Figures 4d,f and 5b,d, the effectiveness of the model slightly degrades when the information of the downscaled image is not among the training examples. However, the decrease in accuracy is not significant, which confirms the generalization capability of the proposed method. This claim can be quantitatively supported by the information reported in Table 1, as the improvement of root mean squared values are mostly remained in the range of 10 ± 5 percent. Clearly, the degradation in downscaling quality increases as the downscaling level increases and the recovered high-resolution image becomes more blurred.

Figures 4 and 5 present the imagery of one representative date (1 June 2013) out of the three available dates, and Table 1 shows the quality matrices for all three dates in the experiment. Although the goodness-of-fit statistics do not noticeably depend on the scaling factor, the blurriness of the images in Figure 4e,f indicate degradation in the results for larger scaling factors. Table 1 shows that the downscaling models function very similarly for both scenarios in terms of qualitative and quantitative performance and the RMSE values show that the overall quality of the downscaling model has not significantly deteriorated for larger scaling factors.

Since the model learns from both scale levels, the resulting map conveys the information from both ends of Landsat and AggieAir. In Figure 4, the field exterior area is shown by lower values of NIR (less reflective) and more dark-reddish color (more reflective) on the upper side of the southern field that has come from the Landsat product. At lower scales (finer resolution), the influences of the AggieAir products become more visible. Visual inspections show that the color gradient of the downscaled fields tends to be closer to the AggieAir product, which demonstrates that the model is able to find and properly use relevant matches of high- and low-resolution patches in the dictionaries.

3.2. Remotely-Sensed Agricultural Variables Derived from Downscaled Bands

In order to test the downscaling scheme presented here for agricultural applications, the downscaled observations in VIR bands were used to infer high-resolution NDVI and soil moisture fields at resolutions of 15 and 7.5 m. NDVI was considered as a direct derivative of the spectral bands, having a diverse application in agricultural studies, which can be derived through a simple relationship between the red and NIR bands. Surface soil moisture (SSM) was considered as the indirect derivative of individual downscaled bands. As previously noted, here we use the model by Hassan-Esfahani et al. [27] to drive high resolution (0.15 m resolution) soil moisture products for the fields. NDVI and SSM were computed at high resolution (0.15 m) from AggieAir products, low resolution (30 m) from Landsat products, and downscaled levels (15 and 7.5 m resolution), and the resulting maps were compared versus aggregated AggieAir products as the baseline. Table 2 provides the goodness-of-fit statistics for these direct and indirect agricultural products versus aggregated AggieAir products. It is important to note that for the soil moisture product a model error is involved, which adds to the downscaling error. In other words, the low level of improvements in downscaled soil moisture products might be due to the insensitivity of soil moisture model to small subgrid variability of the downscaled spectral fields.

Table 2 shows that the downscaling model does not significantly degrade in derivation of high-resolution NDVI products compared to its performance for individual spectral bands with respect to the used quality measures. The goodness-of-fit matrices did not significantly degrade as the scaling level changed, nor when the image of interest was excluded from the training set. Here again the improvement in root mean squared values stayed in the range of 10 ± 5 percent which is absolutely comparable with the information in Table 1. The highest improvement is associated with the overpass in 9 June 2013. This improvement reflects higher association between Landsat 8 spectral response compatibility with AggieAir technology products. This compatibility drags less systematic error from data collection procedure and data homogenization step to further analysis.

Table 2. Root mean squared error (RMSE) for testing the performance of downscaling scheme in derivation of direct (NDVI) and indirect (SSM) agricultural products. High-resolution AggieAir products are considered as the reference.

Agricultural Product	Downscaling Level	Including in the Training Set	RMSE		
			1 June 2013	9 June 2013	17 June 2013
			Improvement Ratio	Improvement Ratio	Improvement Ratio
NDVI	2	YES	10%	15%	12%
		NO	5%	11%	12%
	4	YES	10%	11%	8%
		NO	7%	7%	6%
SSM	2	YES	1.81%	1.67%	1.54%
		NO	1.32%	1.42%	1.43%
	4	YES	1.49%	1.14%	1.78%
		NO	1.23%	1.12%	1.08%

Table 2 shows the degradation of results when the modeling is performed for indirect products (SSM). As presented here, the percent of improvement ratio noticeably decreases for almost one decimal point while comparing NDVI versus SSM models. One reason for this condition could be the fact that the downscaling error propagated and increased as a result of the nonlinear soil moisture Bayesian model. In addition, for direct NDVI products, we have compared observational data with same nature but from different sources while, for indirect products (SSM), we have compared modeling outputs, which may contain significant additive model errors.

The results suggest that the degree of improvements is dependent on the observed fields and perhaps it physical conditions such as vegetation coverage and soil moisture variability. The results from the second flight show the highest quality among the three sets of data. This holds true for the individual spectral bands and the direct and indirect products. For this flight date, there were fewer visible heterogeneities and the coarse-scale images from Landsat 8 did not require use of interpolation methods, as did those of Landsat 7, to fill missing data. Further development of the methodology and further availability of the AggieAir data provides the opportunity to extend the method for downscaling of the satellite data. However, the downscaling model performance is highly tied to numbers and the diversity of sample patches that are available in the dictionaries. In other words, more high-resolution counterparts that are being provided in the dictionaries create a higher quality trained reference dataset that can be applied for improved downscaling.

4. Conclusions

This study presents a statistical approach to compare remotely sensed agricultural variables at different spatial scales in qualitative and quantitative terms. The comparison is based on a new downscaling scheme that can add detailed spatial information to a low-resolution fields. The downscaling scheme was tested for individual spectral bands acquired from a variety of different sensors (Landsat 7, 8, and AggieAir). Tests were also conducted for direct and indirect derivatives of the imagery that have information value to agricultural management.

The downscaling algorithm uses a modern statistical learning method and can recover the lost higher frequency details in spectral bands and their direct and indirect derivatives. The method learns from a representative set of training data from low- and high-resolution patches by solving a constrained optimization problem.

The results presented here suggest future studies to explore the generality and performance of the methodology and its practical implications for use in applications such as precision agriculture.

Since the high-resolution imagery and derivative agricultural products are not available in many practical cases, these results are instructive as a quantitative evaluation of downscaling success rate for similar cases.

In future studies, the range of applicability and effectiveness of the downscaling methodology presented here should be precisely determined for larger datasets. The method should also be empirically tested for its performance on different direct and indirect products as well. This flexible methodology could also facilitate studies using Earth observation data from SSM missions, such as AMSR-E, SMOS.

Acknowledgments: This project was financially supported by the Utah Water Research Laboratory through an MLF Seed Grant. The authors would like to thank the Utah Water Research Laboratory, Utah State University, and the Provo, Office of the US Bureau of Reclamation for their support of this research. The authors acknowledge the extraordinary efforts of the AggieAir flight team. The authors appreciate the support of Ivan Robins, the farm owner, whose cooperation greatly improved of the data collection procedure.

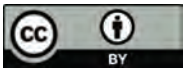
Author Contributions: Mac McKee and Alfonso Torres-Rua developed the research plan and supervised the work required for this paper. Ardeshir Ebtehaj has originally developed the downscaling scheme and supervised its modifications and application in the current study. Leila Hassan-Esfahani completed the literature review, data acquisition, method selection, modelling, manuscript preparation, and discussions. All authors shared equally in the editing of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ge, Y.; Thomasson, J.A.; Sui, R. Remote sensing of soil properties in precision agriculture: A review. *Front. Earth Sci.* **2011**, *5*, 229–238. [CrossRef]
2. Srbinovska, M.; Gavrovski, C.; Dimcev, V.; Krkoleva, A.; Borozan, V. Environmental parameters monitoring in precision agriculture using wireless sensor networks. *J. Clean. Prod.* **2015**, *88*, 297–307. [CrossRef]
3. Zaman, B.; McKee, M.; Neale, C.M.U. Fusion of remotely sensed data for soil moisture estimation using relevance vector and support vector machines. *Int. J. Remote Sens.* **2012**, *33*, 6516–6552. [CrossRef]
4. Hassan-Esfahani, L.; Torres-Rua, A.; Jensen, A.; MacKee, M. Topsoil moisture estimation for precision agriculture using unmmaned aerial vehicle multispectral imagery. In Proceedings of the IEEE International Geoscience and Remote Sensing Symp (IGARSS), Quebec, QC, Canada, 13–18 July 2014.
5. El-Arab, M.; Ticlavilca, A.M.; Torres-Rua, A.; Maslova, I.; McKee, M. Estimating chlorophyll with thermal and broadband multispectral high resolution imagery from an unmanned aerial system using relevance vector machines for precision agriculture. *Int. J. Appl. Earth Obs. Geoinf.* **2015**, *43*, 32–42. [CrossRef]
6. Mathur, A.; Foody, G.M. Crop classification by support vector machine with intelligently selected training data for an operational application. *Int. J. Remote Sens.* **2008**, *29*, 2227–2240. [CrossRef]
7. Franke, J.; Menz, G. Multi-temporal wheat disease detection by multi-spectral remote sensing. *Precis. Agric.* **2007**, *8*, 161–172. [CrossRef]
8. Shanahan, J.F.; Schepers, J.S.; Francis, D.D.; Varvel, G.E.; Wilhelm, W.; Tringe, J.M.; Schlemmer, M.R.; Major, D.J. Major Use of remote-sensing imagery to estimate corn grain yield. *Agron. J.* **2001**, *93*, 583–589. [CrossRef]
9. Ines, A.V.M.; Honda, K.; Gupta, A.D.; Droogers, P.; Clemente, R.S. Combining remote sensing-simulation modeling and genetic algorithm optimization to explore water management options in irrigated agriculture. *Agric. Water Manag.* **2006**, *83*, 221–232. [CrossRef]
10. Merlin, O.; Al Bitar, A.; Walker, J.P.; Kerr, Y. An improved algorithm for disaggregating microwave-derived soil moisture based on red, near-infrared and thermal infrared data. *Remote Sens. Environ.* **2010**, *113*, 2275–2284. [CrossRef]
11. Reichle, R.H.; Entekhabi, D.; McLaughlin, D.B. Downscaling of radiobrightness measurements for soil moisture estimation: A four-dimensional variational data assimilation approach. *Water Resour. Res.* **2001**, *37*, 2353–2364. [CrossRef]
12. Peng, J.; Loew, A.; Zhang, S.; Wang, J.; Niesel, J. Spatial downscaling of satellite soil moisture data using a vegetation temperature condition index. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 558–566. [CrossRef]

13. Ochsner, T.E.; Cosh, M.H.; Cuenca, R.H.; Dorigo, W.A.; Draper, C.S.; Hagimoto, Y.; Kerr, Y.H.; Njoku, E.G.; Small, E.E.; Zreda, M.; et al. State of the art in large-scale soil moisture monitoring. *Soil Sci. Soc. Am. J.* **2013**, *77*, 1888–1919. [CrossRef]
14. Hassan, Q.K.; Bourque, C.P.A.; Meng, F.R.; Cox, R.M. A wetness index using terrain-corrected surface temperature and normalized difference vegetation index derived from standard modis products: An evaluation of its use in a humid forest-dominated region of eastern Canada. *Sensors* **2007**, *7*, 2028–2048. [CrossRef]
15. Mallick, K.; Bhattacharya, B.K.; Patel, N.K. Estimating volumetric surface moisture content for cropped soils using a soil wetness index based on surface temperature and NDVI. *Agric. For. Meteorol.* **2009**, *149*, 1327–1342. [CrossRef]
16. Merlin, O.; Chehbouni, G.; Kerr, Y.; Goodrich, D. A downscaling method for distributing surface soil moisture within a microwave pixel: Application to the Monsoon'90 data. *Remote Sens. Environ.* **2006**, *101*, 379–389. [CrossRef]
17. Choi, M.; Hur, Y. A microwave-optical/infrared disaggregation for improving spatial representation of soil moisture using AMSR-E and MODIS products. *Remote Sens. Environ.* **2012**, *124*, 259–269. [CrossRef]
18. Chauhan, N.S.; Miller, S.; Ardanuy, P. Spaceborne soil moisture estimation at high resolution: A microwave-optical/IR synergistic approach. *Int. J. Remote Sens.* **2003**, *24*, 4599–4622. [CrossRef]
19. Ebtehaj, A.M.; Foufoula-Georgiou, E.; Lerman, G. Sparse regularization for precipitation downscaling. *J. Geophys. Res.* **2012**, *117*. [CrossRef]
20. AggieAir. 2015. Available online: http://aggieair.usu.edu/ (accessed on 6 September 2017).
21. Jensen, A.M. A Geospatial Real-Time Aerial Image Display for a Low-Cost Autonomous Multispectral Remote Sensing. Master's Thesis, Utah State University, Logan, UT, USA, 2009.
22. Jiang, J.; Liu, D.; Gu, J.; Susstrunk, S. What is the space of spectral sensitivity functions for digital color cameras? In Proceedings of the 2013 IEEE Workshop on Applications of Computer Vision (WACV), Tampa, FL, USA, 15–17 January 2013; pp. 168–179.
23. Infrared Cameras Incorporated. Available online: http://www.infraredcamerasinc.com (accessed on 6 September 2017).
24. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2008; p. 128.
25. Mallat, S.; Zhang, Z. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Signal Process.* **1993**, *41*, 3397–3415. [CrossRef]
26. Hassan-Esfahani, L.; Torres-Rua, A.; MacKee, M. Assessment of optimal irrigation water allocation for pressurized irrigation system using water balance approach, learning machines, and remotely sensed data. *Agric. Water Manag.* **2015**, *153*, 42–50. [CrossRef]
27. Hassan-Esfahani, L.; Torres-Rua, A.; Jensen, A.; MacKee, M. Assessment of surface soil moisture using high-resolution multi-spectral imagery and artificial neural networks. *Remote Sens.* **2015**, *7*, 2627–2646. [CrossRef]



sensors

MDPI

Article

Vision-Based Target Finding and Inspection of a Ground Target Using a Multirotor UAV System

Ajmal Hinas *, Jonathan M. Roberts and Felipe Gonzalez

Robotics and autonomous systems, Queensland University of Technology (QUT), Brisbane City QLD 4000, Australia; Jonathan.roberts@qut.edu.au (J.M.R.); felipe.gonzalez@qut.edu.au (F.G.)

* Correspondence: ajmal.hinas@hdr.qut.edu.au; Tel.: +61-0469752697

Received: 4 November 2017; Accepted: 15 December 2017; Published: 17 December 2017

Abstract: In this paper, a system that uses an algorithm for target detection and navigation and a multirotor Unmanned Aerial Vehicle (UAV) for finding a ground target and inspecting it closely is presented. The system can also be used for accurate and safe delivery of payloads or spot spraying applications in site-specific crop management. A downward-looking camera attached to a multirotor is used to find the target on the ground. The UAV descends to the target and hovers above the target for a few seconds to inspect the target. A high-level decision algorithm based on an OODA (observe, orient, decide, and act) loop was developed as a solution to address the problem. Navigation of the UAV was achieved by continuously sending local position messages to the autopilot via Mavros. The proposed system performed hovering above the target in three different stages: locate, descend, and hover. The system was tested in multiple trials, in simulations and outdoor tests, from heights of 10 m to 40 m. Results show that the system is highly reliable and robust to sensor errors, drift, and external disturbance.

Keywords: unmanned aerial vehicles; vision-based navigation; vision and action; OODA; remote sensing; inspection; target detection

1. Introduction

Development in Unmanned Aerial Vehicle (UAV) path planning [1–3], design [4–6], and remote sensing [7–10] has been an active field of research over the past 15 years. Several researchers have studied the possibility of using vision for navigation, target tracking, and landing site detection. A predictive navigation technique was developed for a wildlife detection scenario in [11]. In [12], a vision-based control system for a quadcopter for following a ground-moving target was demonstrated. Vision-based autonomous landing on a landing pad is presented in [13]. In most of the earlier work, the main focus has been on demonstrating the ability of the developed controllers and not on the reliability and robustness of the systems. Our motivation was to use a low-cost UAV system to remotely sense and find a target, and reliably descend and hover above the target with the purpose of closely inspecting it. The system can also be used for accurate and safe delivery of payloads or spot spraying applications in site-specific crop management.

Figure 1 illustrates the concept and a typical mission. Initially, the UAV has to take off (1). After takeoff, the UAV searches for a ground target (2). If a target is found, the UAV changes its original path (3) and hovers above the target for a few seconds for an action such as close inspection (4). After hovering, the UAV will fly to the next waypoint (5) and start to search for other targets. This paper focuses on target detection and navigation. A red-colored 0.2 m radius circular target was used in this study for illustrative purposes.

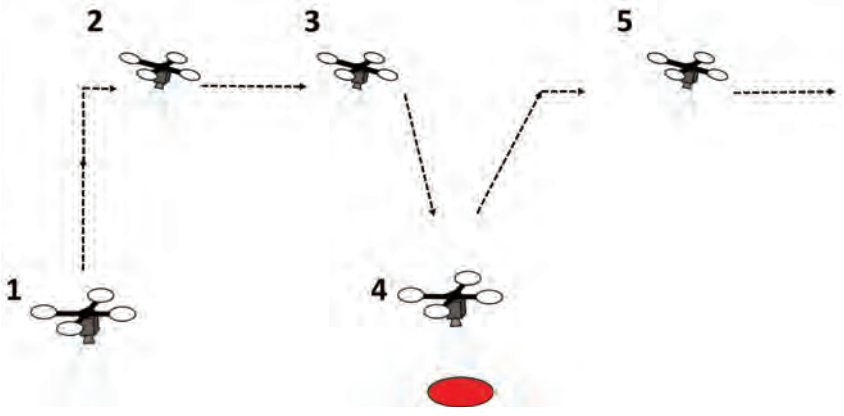


Figure 1. Typical mission performed by the multirotor UAV in a target finding and inspection mission.

The system uses a modular system architecture and a vision-based algorithm with the observe, orient, decide, and act (OODA) decision loop [14] for searching and hovering above a target. The OODA loop is commonly used by military pilots during the decision making process. The OODA loop has also been used in other contexts such as modeling the development of intelligent agents [15,16]. Our results from multiple outdoor flight tests show that the proposed algorithm is reliable and robust regardless of sensor errors and external disturbances.

2. Related Work

Related work in the field of vision-based navigation includes work by Mathe and Busoniu [17] who reviewed vision and control for UAVs in the context of inexpensive platforms for infrastructure inspection. They defined UAVs under $1500, diameter under 1 m, and weight under 4 kg as inexpensive platforms. The authors grouped the existing work, including the use of more costly UAVs, into three groups: power line inspection, building monitoring, and railway inspection. A general technique used in this application is tracking lines in infrastructure. Araar and Aouf [18] demonstrated a power line inspection task using a quadrotor. The authors developed a classical image-based visual servoing controller and a partial pose-based visual servoing controller for this purpose.

Mathe et al. [19] presented the task of inspecting railway semaphores using a proportional controller based on the distance to the object and flying the UAV around a semaphore in a 4–5 m radius arc. However, the study did not consider the part of navigating the UAV from a high altitude to the railway semaphore.

In other related work, a robust marker tracking algorithm was presented for precise UAV vision-based landing in [20]. The authors employed a pinhole camera model to calculate the relative position of the UAV from the landing target. The study was limited due to simulation and estimation errors and disturbances such as wind that were not considered.

Vision-based landing on a moving vehicle was demonstrated in [21]. The authors used the relative distance between the UAV and the vehicle to generate velocity commands to the UAV. Therefore, extending this method to multiple target-finding scenarios is difficult. In [22], a target detection and tracking algorithm was discussed with the aim of landing the UAV on a target. The method proposed was effective only for detection and tracking with low disturbances. Furthermore, the method was only tested in simulation.

An open-source computer vision-based guidance system for landing and hovering the UAV on a target was proposed in [23]. The authors used simple geometry to estimate the target's position and position control to guide the UAV towards the target. Their method was tested only with the UAV

cruising a few meters (3–5 m) above the ground. Even though there is literature on the topic related to this work, this paper contributes to the literature by extending the technique to high altitude flights.

3. Hardware System

The hardware system architecture is shown in Figure 2 and the complete hardware system is shown in Figure 3. A quadrotor UAV that uses the DJI 450 frame was developed for the purpose of this study. Four DJI 2212/920 KV motors and 8 × 4.7" propellers were used to produce thrust. The open source Pixhawk autopilot board was used as the main flight controller. The Pixhawk controller supports PX4 and APM firmware stacks. However, the PX4 firmware stack was used as it has more hardware support [24]. The Pixhawk autopilot board consist of gyroscopes, accelerometers, and a barometer as sensors. A ublox GPS + compass module was externally mounted for positioning. The GPS and compass modules were mounted externally and away from other electronics to reduce interference and increase accuracy. A Raspberry Pi 2 computer was attached to the UAV as an onboard computer to run the detection and navigation algorithm. Ubuntu 14.04 and ROS Indigo were installed on the onboard computer.

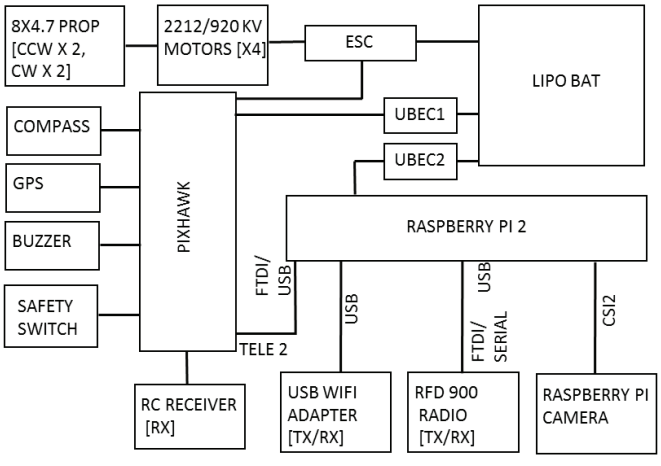


Figure 2. Hardware system architecture illustrating the hardware components and their interconnection.

Figure 3. Hardware system.

A downward-looking Raspberry Pi camera was attached to the UAV as an imaging sensor. This camera is capable of taking images with 640 × 480 resolution at 90 fps. A USB Wi-Fi adapter was connected to the Raspberry Pi computer for the purpose of debugging the code from the development PC. A terminal was configured for the onboard Ubuntu operating system to remotely access the UAV from the ground computer through a RFD 900 long-range radio. This terminal connection was used for monitoring flight log messages and initiating the autonomous flight. UBEC 1 and UBEC 2 were used to distribute the power to the Pixhawk and Raspberry Pi board, respectively.

4. Software System

The software system consists of four ROS [25] nodes (Figure 4): an image capture node that captures the images from the Raspberry Pi camera at a predefined frame rate; a target detection node that uses a modified version of the algorithm described in [26] for target detection; a main node that runs the main control loop of the system; and a Mavros node that provides communication between the ROS and PX4 autopilot.

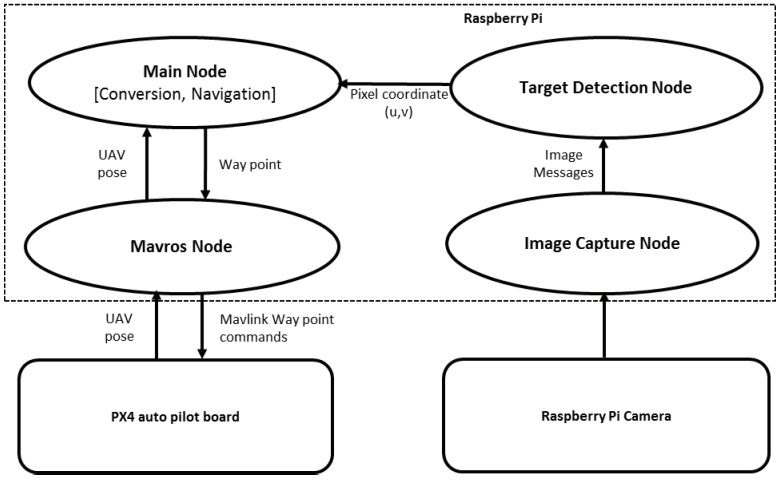


Figure 4. Software system architecture illustrating the ROS nodes and information flow.

The image capture node, target detection node, and the main node were developed in C++. Mavros is a prebuilt package available with ROS. OpenCV is used to capture images in the image capturing node. All the decision and guidance functions are performed by the main node. The following subsections describe in detail each function of the main node.

4.1. Conversion of 2D Image Coordinates to World Coordinates

The main node receives the center coordinates of the detected target from the target detection node as an input. This is then converted to the target location in the inertial frame by using the pinhole camera model (Figure 5). The coordinates of the target in world coordinates (inertial frame {L}) are given by Equation (1).

$$^{L}t = {}^{L}\xi_{C} \cdot {}^{C}t \tag{1}$$

The center of the target in the camera frame ^{C}t is given by Equation (2):

$$^{C}t = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = S\,{}^{C}RK^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{2}$$

where u and v are pixel coordinates of the center of the target in image frame $\{\mathrm{I}\}$. The projected pixel coordinates change with the rotation of the camera. Therefore, the rotation of the camera in its own frame ${}^{C}R$ is used for correcting the rotation. K is the camera matrix and is found by using the ROS camera calibration package as described in the tutorial [27]. S is an unknown constant. S can be solved by equating the last rows of the final matrix from the right side to the UAV's flying height z.

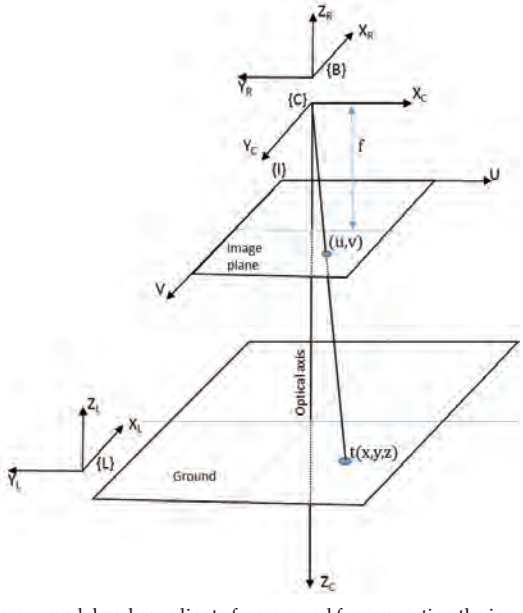


Figure 5. Pinhole camera model and coordinate frames used for converting the image target coordinates (u,v) into world coordinates (x, y, z).

The camera is directly attached to the UAV without any gimbal mechanism. Therefore, the camera rotation matrix ${}^{C}R$ is written as:

$$ {}^{C}R = R(-\theta)R(-\phi)R(-\psi) \tag{3} $$

where ϕ, θ, and ψ are roll, pitch, and yaw angles of the UAV in the body frame $\{\mathrm{B}\}$. The camera pose in inertial frame ${}^{L}\xi_{C}$ is given by Equation (4):

$$ {}^{L}\xi_{C} = {}^{L}\xi_{B} \,.\, {}^{B}\xi_{C} \tag{4} $$

where ${}^{L}\xi_{B}$ is the pose of the body frame $\{\mathrm{B}\}$ in inertial frame $\{\mathrm{L}\}$. It is given by the position of the UAV. The camera pose in body frame ${}^{B}\xi_{C}$ is dependent on the camera mounting arrangement. In our case, it can be written as a homogeneous matrix (Equation (5)):

$$ {}^{B}\xi_{C} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

4.2. Navigation Algorithm

Navigation of the UAV is achieved by continuously sending the local position messages from the main node via the Mavros [28]. The proposed algorithm performs the task of hovering above the target for close inspection through three different stages: locate, descend, and hover.

The OODA loop makes different decisions based on this stage of the UAV. In the locate stage, the estimated target position *TP* (line 30) is used to move the UAV laterally to the target's *x*,*y* position without descending (lines 32, 33 and 34). In the descend stage, the lateral deviation between the target's position *TP* and the UAV's position *CP* is calculated (line 37). If the deviation is within the tolerance *t*, the height of the multirotor is reduced by a predefined parameter d_h (line 39). Otherwise, the locate stage will be repeated (lines 45 and 46). This process continues until the multirotor reaches the hovering height h_h (line 38). This process happens in quick succession.

After reaching the hovering height, in the hover stage, if the distance between the center of the target and the center of the image in pixels is more than the tolerance value *g* (line 49), a proportional controller generates displacement vectors based on the pixel distance between the centers. These displacement vectors are used to adjust the position of the multirotor above the target (lines 50–54). In this stage, the camera field of view is very small. Therefore, if the target is out of the field of view, the last detected target pixel coordinate *CTL* is used for a timeout period t_o to laterally move the UAV above the target (lines 61–66). If the UAV is exactly above the target, the position is maintained for *s* seconds (line 56). Finally, the UAV flies to the next waypoint (lines 71 and 72).

Algorithm 1 Navigation Algorithm

```
1.  NWP= Number of waypoints
2.  NP=Next pose of UAV
3.  t=tolerance in meters
4.  WP=Waypoint
5.  HP=Hovering pose of UAV
6.  hh= hovering height
7.  s=hovering time
8.  dh= descending height
9.  CI=center of image frame in pixels
10. L=displacement constant
11. HR=horizontal resolution of camera
12. CTL=Centroid of target from last target detection in pixels
13. TL=Timestamp of last target-detected image
14. to=timeout from last target detection
15. u-horizontal image coordinate
16. v-vertical image coordinate
17. g-tolerance in pixels
18. i=0
19. TAKEOFF
20. IP=Initial position of UAV
21. stage=1
22. while (i<NWP)
23.   NP=WPi
24.   for each image Imj
25.     CP=read current pose of UAV
26.     if(target found)
27.       TLD=timestamp of(Imj)
28.       CT = Compute centroid of target(Imj)
29.       CTL=CT
30.       TP=Compute target pose(CT, CP)
31.       if(stage=1)
```

```
32.            HP=TP                                              ┐
33.            HP.z=CP.z                                          │
34.            NP=HP                                              ├ LOCATE
35.            stage=2                                            ┘
36.          else if(stage=2)                                     ┐
37.            if(distanceXY(TP.position, CP.position)<t)         │
38.              if(hh +IP.z< CP.z-dh)                            │
39.                HP.z=CP.z-dh                                   │
40.              else                                             │
41.                HP.z= hh+IP.z                                  ├ DESCEND
42.                stage=3                                        │
43.              end if                                           │
44.              NP=HP                                            │
45.            else                                               │
46.              stage=1                                          │
47.            end if                                             ┘
48.          else if(stage=3)                                     ┐
49.              if(|CI.u- CT.u|>g OR |CI.v- CT.v|>g)             │
50.                dx=(CI.u-CT.u)*L/HR                            │
51.                dy=(CI.v-CT.v)*L/HR                            │
52.                HP.x=CP.x+dx                                   │
53.                HP.y=CP.y+dy                                   │
54.                NP=HP                                          │
55.              else                                             │
56.                wait s                                         │
57.                break                                          ├ HOVER
58.              end if                                           │
59.          end if                                               │
60.        else                                                   │
61.            TLI=timestamp of(Imi)                            │
62.            If(stage=3 & TLI-TLD< to)                        │
63.              dx=(CI.u-CTL.u)*L/HR                           │
64.              dy=(CI.v-CTL.v)*L/HR                           │
65.              HP.x=CP.x+dx                                   │
66.              HP.y=CP.y+dy                                   │
67.              NP=HP                                          │
68.            end if                                           ┘
69.        end if
70.      end for
71.      i=i+1
72.      NP=WPi
73.    end while
```

4.3. Target Detection Algorithm

A robust target detection is needed for the navigation algorithm to perform properly. Different target detection techniques were tested. The Hough circle method described in [26] was used to detect the target. However, when increasing the altitude, the detection became less reliable. The color detection method described in [29] *was* also used and gives reliable detection at high altitudes; however,

false positives were observed at low heights. A detection algorithm (Algorithm 2) was developed by combining both techniques.

Algorithm 2 Target Detection Algorithm

```
1.  for each image Im_i
2.      u=v=−1
3.      Im_i=preprocess(Im_i)
4.      binaryImg= threshold (Im_i, colorRange)
5.      binaryImg=postProcessing(binaryImg)
6.      [u,v]=findHoughCircleCenter(binaryImg)
7.      if(u<1 OR v<1)
8.          if(moment00(binaryImg)>0)
9.              [u,v]=findCentroidOfBlob(binaryImg)
10.         end if
11.     end if
12. end for
```

In Algorithm 2, a binary image is created for red-colored blobs using a threshold operation (line 4). The center of the target is found using the OpenCV HoughCircle function (line 6). If no Hough circle is found, but blobs exist (lines 7 and 8), then the image moments are used to find the centroids of the blob (u, v) in pixels (line 9).

$$u = \frac{m_{10}}{m_{00}} \tag{6}$$

$$v = \frac{m_{01}}{m_{00}} \tag{7}$$

where m_{00} is 0th moment and m_{10} and m_{01} are the first-order moments.

A ground-based test was conducted where the target was moved by walking slowly away from the camera and the detection of the new algorithm was observed. The distance between the camera and the target was measured. From this test, the range of the target detection algorithm was determined as 88 m under bright sunlight.

5. Experiments

Simulated and real flight experiments were conducted to validate the developed system. There were two goals for the experiments. The first goal was to descend and hover the UAV reasonably close to the target with the purpose of inspecting the target. To achieve this goal, a criterion was defined such that if the UAV could hover above the target within an area of 1 m from the center of the target in the inertial xy plane and within 2.5 m above the ground, then the mission would be considered as successful. The second goal was to test the reliability of the system. To achieve this goal, tests were repeated multiple times with different parameters. The Experimental Scenario subsection describes the details.

5.1. Experimental Scenario

Figure 6 shows the experimental scenario used for both simulation and field tests. The UAV had to take off and move from one waypoint to another waypoint while scanning for a red circle target on the ground. A 0.2 m radius red-colored circular target was placed on the ground between the waypoints.

The UAV was flown at different heights ranging from 10 to 40 m. Each height was repeated five times. When repeating the heights, the coordinates of the waypoints were changed to ensure the ability of the system with different flight scenarios (Figure 6). The target was also placed randomly along the

path, to the right or left sides of the flight path, but so the target was visible to the UAV when the UAV flew along the flight path. In Figure 6, letters indicate the waypoints and arrows indicate the flight paths used. Paths AB and CD were flown according to the coordinates shown in the diagram. Paths similar to EF, GH, and IJ were chosen randomly.

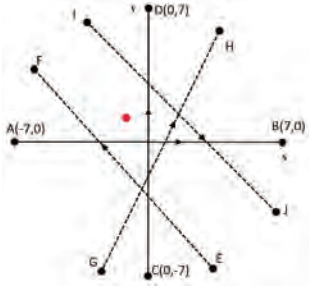


Figure 6. Typical flight paths used in each height of the experiment and an example target location (red circle). Waypoints are indicated by (●).

5.2. Simulation Experiments

Refining and testing the system is mandatory before doing any field flight test. Therefore, a Software in the loop (SITL) simulation environment was set up using the Gazebo robotics simulator [30] and the PX4 firmware for the Pixhawk autopilot hardware. A quadrotor model and simulated world were created to closely resemble the real experimental environment. Real images captured from the field test were also used in modeling the simulation environment. A downward-looking simulated camera model was attached to the simulated quadrotor. Simulated camera parameters were set to the real Raspberry Pi camera's parameters used in the real test. A very high frame rate increases the image processing load in the Raspberry Pi computer and may introduce additional delays. Therefore, the camera frame rate was set to 10 Hz in the real test as well as in the simulation. Horizontal flying velocities of the multirotor were kept similar to the field tests by setting the PX4 parameters.

Figure 7a shows the UAV approaching the target in a plane view and Figure 7b shows the UAV above the target at hovering height. Figure 7c shows the UAV flying at a height of 20 m and Figure 7d shows target detection using the simulated camera where the UAV is at a height of 20 m.

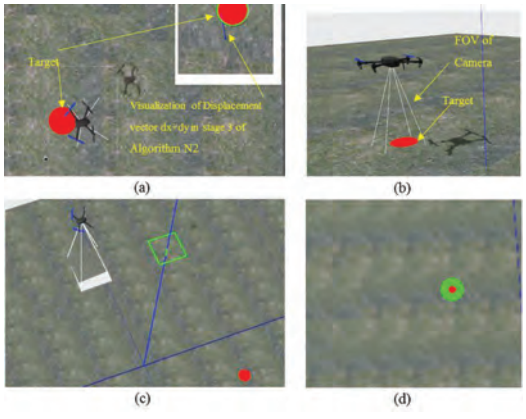


Figure 7. Simulation environment, (top) Unmanned Aerial Vehicle (UAV) is approaching the target. (**a**) Plane view; (**b**) 3D view; (**c**) UAV is flying at 20 m height; (**d**) target detection from 20 m height.

5.3. Flight Test Experiments

Flight tests were conducted at Queensland University of Technology's (QUT's) Samford Ecological Research Facility (SERF), Queensland, Australia on different days, including cloudy and overcast days, over a four-month period. The same experimental scenario described in Figure 6 was followed. Figure 8 shows the test site and the mission boundaries. Figure 9 shows the UAV hovering above the target during a field test.

Figure 8. The marked area shows the test site (Samford Ecological Research Facility, Queensland University of Technology (QUT), Queensland, Australia).

Figure 9. UAV is hovering above the target at hovering height (simulation of close inspection).

6. Results and Discussion

All of the simulated flight experiments were successful. In the real flight test, the overall success rate was 92%. Two flight tests failed: one at 15 m and one at 30 m altitude. Our analysis of the failed tests showed that the timeout period of 1 s had expired in the hover stage of the navigation algorithm. Therefore, the UAV had lost track of the target. However, we believe that experimentally adjusting the timeout period to a more suitable value can reduce the problem. Table 1 summarizes the results of the simulation and field tests.

Table 1. Results of simulations and flight tests.

Height (m)	Success Rate (Simulation)	Success Rate (Flight Test)
10	100% (5/5)	100% (5/5)
15	100% (5/5)	80% (4/5)
20	100% (5/5)	100% (5/5)
30	100% (5/5)	80% (4/5)
40	100% (5/5)	100% (5/5)

The task performed by the system is simple if all the sensor measurements are ideal and there are no external disturbances. However, target position estimations in the real world have considerable errors and uncertainty. Figure 10 shows these errors clearly. It shows the recorded target's X and Y position estimates and the UAV position when the UAV took off and flew at a height of 10 m from waypoint A to waypoint B without descending to the target.

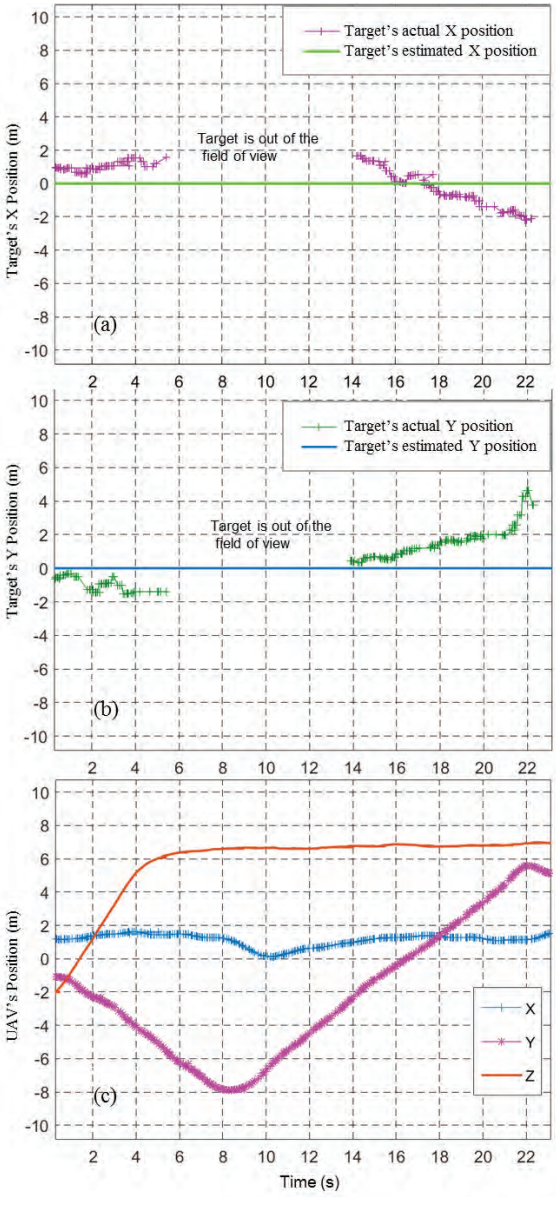


Figure 10. Image-based target position estimation and the actual position of the target from a 10 m height flying mission. (**a**) Target's actual and estimated X positions; (**b**) Target's actual and estimated Y positions; (**c**) UAV positions.

Nisi and Menichetti [31] listed and gave a good discussion of the error sources in their task of geo-referencing photovoltaic arrays using UAVs. Among the error sources discussed by Nisi and Menichetti are barometric altimeter resolution, UAV velocity, target height, lens distortion, GNSS accuracy, and imaging sensor resolution, which are also all applicable to our task. Moreover, experimental data collected showed considerable drift in the autopilot's inertial frame {L} (Figure 11). The UAV's position data shown in Figure 11 was recorded while the UAV was resting in a static position on the ground. The plot shows significant drift in the inertial frame of the UAV.

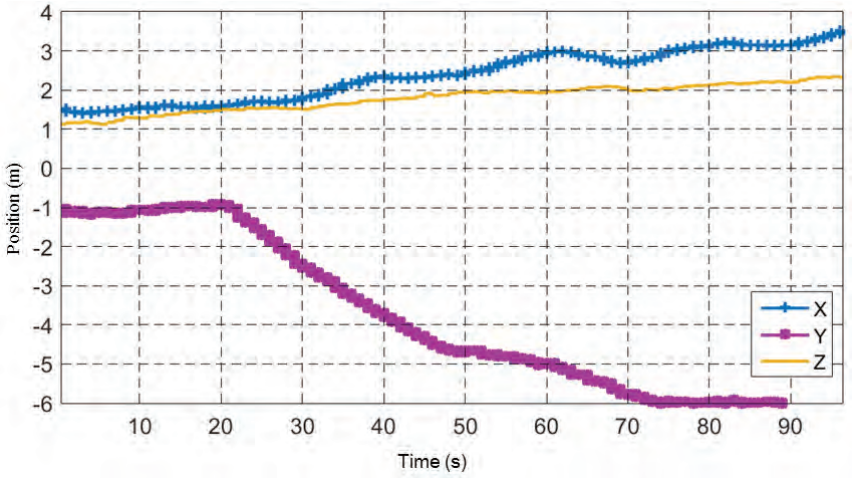


Figure 11. A plot of the autopilot's position estimation drift when the UAV is stationary on the ground.

Further to the error sources discussed in [31], the target position estimation method assumes that the camera is in perfect alignment with the UAV's body frame and that both the camera frame and the body frame have the same origin. There are practical limitations in aligning the camera and the Inertial Measurement Unit (IMU) of the UAV. Other sources such as errors in synchronization of data from different sensors (camera, IMU) and camera vibration also contribute to the overall error in target position estimation.

Figure 12a,b shows the 3D view and top view of the flight trajectories for a target finding and hovering mission at a 20 m height for both simulation and field tests. Here, the UAV was expected to move from waypoint C to waypoint D. The point E indicates the decision point where the system decided to move towards the target in the x,y before descending. The path in the field test, compared to the simulation, took a small spiral shape. This was mainly because of the image-based target position estimation errors and external disturbances such as wind. The top view of the field test shows a slight difference between the target's position and the hovering position. The drift in the inertial frame {L} was the cause of this effect. Figure 12c,d show X and Y coordinates of the UAV, estimated target position and real target position for the same 20 m target finding and hovering mission.

Figure 13 shows the results for a target finding and hovering mission at a 40 m height. Here, the UAV was expected to move from waypoint A to waypoint B. The point E indicates the decision point where the system decided to move towards the target. Compared to the 20 m flight test (Figure 12b, right), the 40 m flight test (Figure 13b, right) shows an increase of a spiral-like path when the UAV descended. An increase in the image-based target position error (Figure 13c,d, right) was the main reason for this.

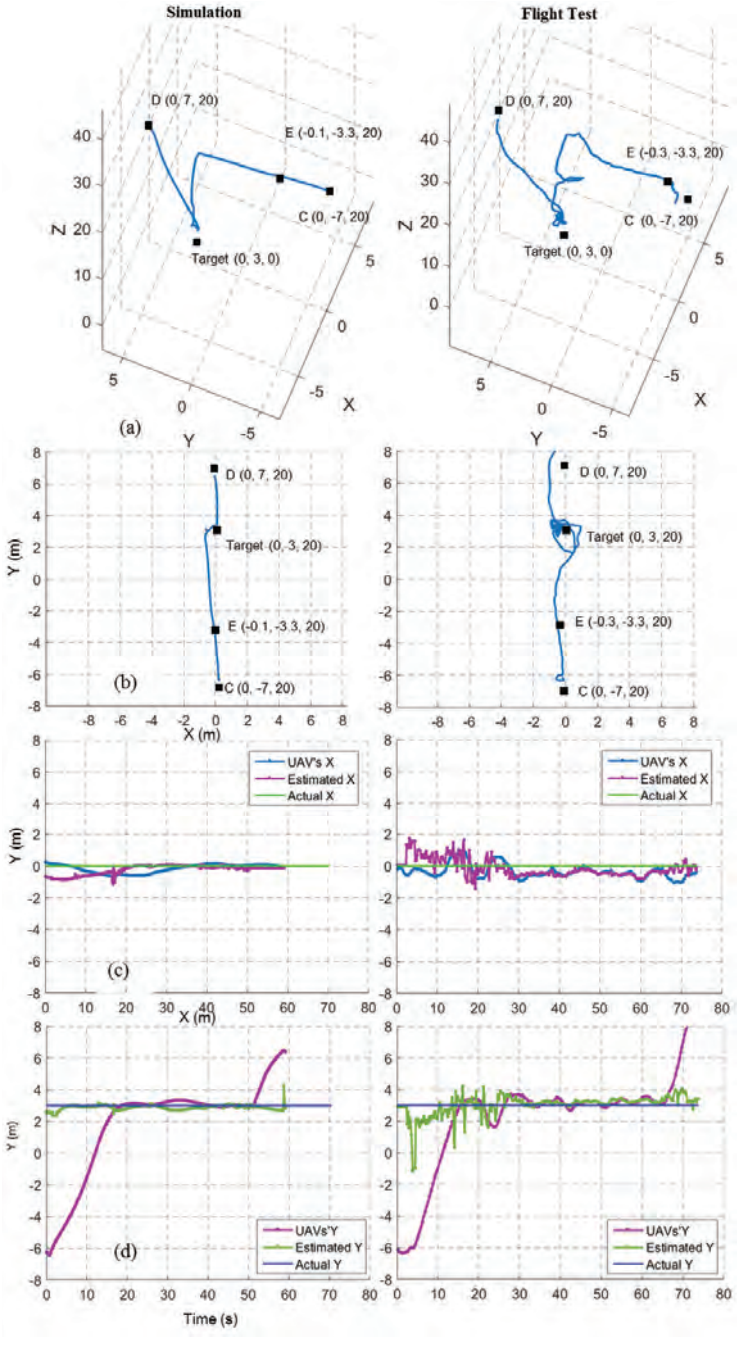


Figure 12. Flight trajectories from a 20 m simulated and a real target finding and hovering missions. Left column shows simulated results and the right column shows field test results. (**a**) 3D View; (**b**) top view; (**c**) X coordinates of UAV position, estimated target position and real target position; and (**d**) Y coordinates of UAV position, estimated target position and real target position.

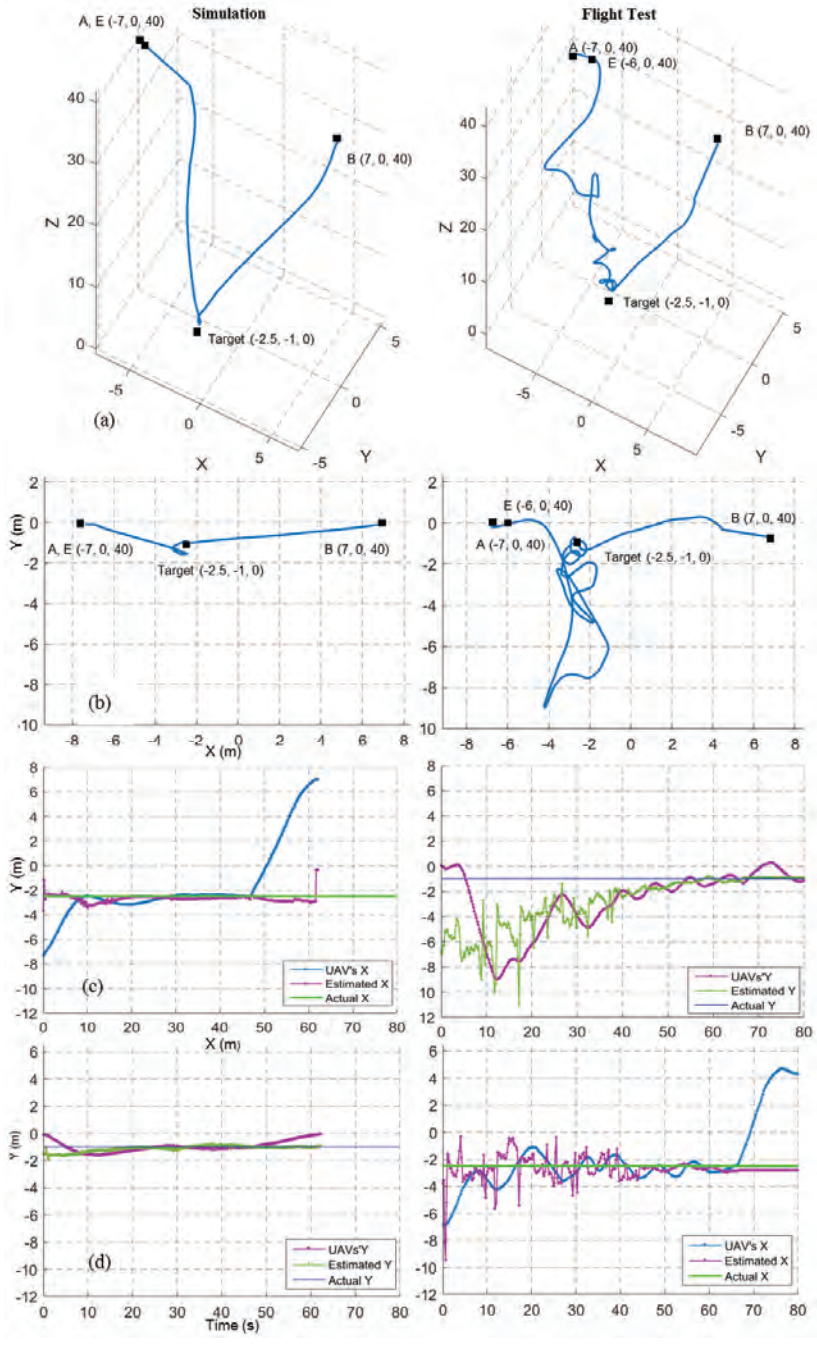


Figure 13. Flight trajectories from a 40 m simulated and a real target finding and hovering missions. Left column shows simulated results and the right column shows field test results. (**a**) 3D View; (**b**) Top View; (**c**) X coordinates of UAV position, estimated target position and real target position; (**d**) Y coordinates of UAV position, estimated target position and real target position.

In contrast, the 40 m simulation results show less error, similar to the 20 m simulation results. One possible reason might be that all the sensors in the simulation were ideal, except the IMU. Though the position estimation error increased with height in the flight test, the UAV could still compensate for the error and reach the target.

Infrastructure inspection with inexpensive UAVs is a growing field. Researchers have explored manually positioning the UAVs at a workable distance from the target. However, a fully autonomous system with the capability of finding and approaching the target is desirable. For instance, in railway infrastructure inspection tasks, the line tracking techniques discussed in earlier papers can be used to fly the UAV along the railway track at a 50–100 m altitude and the technique presented here can be used to approach an identified target and do a detailed inspection.

Moreover, purely image-based techniques without position estimation such as image-based visual servoing are useful in vision-based landing tasks. The technique may have limitations when multiple targets have similar features or in detecting distinguishable features when the targets are very far from the UAV. Generally, researchers have used relatively large targets with uniquely distinguishable features for vision-based landing tasks, where the challenges due to sensor errors and disturbances are less prevalent.

The navigation algorithm presented here tested with a reliable detection (>90%). However, target detection algorithms may be less reliable in real applications. Performance of the navigation algorithm might be affected when the target detection is unreliable. Further research is needed to make the algorithm robust for such cases.

7. Conclusions

In this paper, a system architecture and a reliable and robust algorithm for inspection of the ground target using an inexpensive multirotor UAV system is presented. Image-based target position estimation was used to guide the UAV towards the target. In an ideal sense, the task was very simple. However, uncertainties and errors introduced by the low-cost onboard sensors and disturbances in the outdoor environment present significant challenges to reliably perform the task. The authors applied a high-level decision making approach with the observe, orient, decide, and act (OODA) loop to address the problem. The results from multiple field tests show that the proposed algorithm is 100% reliable in simulation and 92% in real experiments and robust to sensor errors, drift, and external disturbances.

Future work will focus on extending the system for multiple targets. Multiple target detection and inspection needs real-time detection and tracking of each target. This is a more challenging task owing to the limited onboard computation power that is available with the inexpensive UAV system. Moreover, path planning and decision making are also important to optimally visit the targets with the limited battery capacity.

A video of the flight test can be found at the link https://youtu.be/6w_OFScBWIg.

Author Contributions: Ajmal Hinas developed the Hardware and the Software Systems and conducted experiments and wrote the manuscript. Jonathan M. Roberts provided important advice and guidance. Felipe Gonzalez contributed with the main concept, materials, and editing and revision.

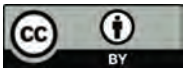
Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liang, X.; Wang, H.; Li, D.; Liu, C. Three-dimensional path planning for unmanned aerial vehicles based on fluid flow. In Proceedings of the 2014 IEEE Aerospace Conference, Big Sky, MT, USA, 1–8 March 2014.
2. Nolan, P.; Paley, D.A.; Kroeger, K. Multi-UAS path planning for non-uniform data collection in precision agriculture. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017.
3. Wirth, L.; Oettershagen, P.; Ambühl, J.; Siegwart, R. Meteorological path planning using dynamic programming for a solar-powered UAV. In Proceedings of the 2015 IEEE Aerospace Conference, Big Sky, MT, USA, 7–14 March 2015.

4. González, L.; Whitney, E.; Srinivas, K.; Périaux, J. Optimum multidisciplinary and multi-objective wing design in CFD using evolutionary techniques. In *Computational Fluid Dynamics 2004, Proceedings of the Third International Conference on Computational Fluid Dynamics, ICCFD3, Toronto, ON, Canada, 12–16 July 2004*; Groth, C., Zingg, D.W., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 681–686.
5. Lee, D.S.; Gonzalez, L.F.; Périaux, J.; Srinivas, K. Evolutionary optimisation methods with uncertainty for modern multidisciplinary design in aeronautical engineering. In *100 Volumes of 'Notes on Numerical Fluid Mechanics': 40 Years of Numerical Fluid Mechanics and Aerodynamics in Retrospect*; Hirschel, E.H., Krause, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 271–284.
6. Henderson, L.; Glaser, T.; Kuester, F. Towards bio-inspired structural design of a 3D printable, ballistically deployable, multi-rotor UAV. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017.
7. Alvarado, M.; Gonzalez, F.; Fletcher, A.; Doshi, A. Towards the development of a low cost airborne sensing system to monitor dust particles after blasting at open-pit mine sites. *Sensors* **2015**, *15*, 19667–19687. [CrossRef] [PubMed]
8. Pena, J.M.; Torres-Sanchez, J.; de Castro, A.I.; Kelly, M.; Lopez-Granados, F. Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle (UAV) images. *PLoS ONE* **2013**, *8*, e77151. [CrossRef] [PubMed]
9. Malaver, A.; Gonzalez, F.; Depari, A.; Corke, P.; Motta, N. Towards the development of a gas sensor system for monitoring pollutant gases in the low troposphere using small unmanned aerial vehicles. In Proceedings of Workshop on Robotics for Environmental Monitoring, Sydney, Australia, 11 July 2012.
10. Alsalam, B.H.Y.; Morton, K.; Campbell, D.; Gonzalez, F. Autonomous UAV with vision based on-board decision making for remote sensing and precision agriculture. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017.
11. Ward, S.; Hensler, J.; Alsalam, B.; Gonzalez, L.F. Autonomous UAVs wildlife detection using thermal imaging, predictive navigation and computer vision. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016.
12. Razinkova, A.; Cho, H.C. Tracking a moving ground object using quadcopter UAV in a presence of noise. In Proceedings of the 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Busan, Korea, 7–11 July 2015; pp. 1546–1551.
13. Yang, S.; Scherer, S.A.; Zell, A. An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle. *J. Intell. Robot. Syst.* **2013**, *69*, 499–515. [CrossRef]
14. Sholes, E. Evolution of a UAV autonomy classification taxonomy. In Proceedings of the 2007 IEEE Aerospace Conference, Piscataway, NJ, USA, 3–10 March 2007.
15. Heinze, C.; Goss, S.; Josefsson, T.; Bennett, K.; Waugh, S.; Lloyd, I.; Murray, G.; Oldfiel, J. Interchanging agents and humans in military simulation. *AI Magzine* **2002**, *23*, 37–47.
16. Sari, S.C.; Kuspriyanto; Prihatmanto, A.S. Decision system for robosoccer agent based on OODA Loop. In Proceedings of the 2012 International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, 11–12 September 2012.
17. Máthé, K.; Buşoniu, L. Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors* **2015**, *15*, 14887–14916. [CrossRef] [PubMed]
18. Araar, O.; Aouf, N. In Visual servoing of a Quadrotor UAV for autonomous power lines inspection. In Proceedings of the 2014 22nd Mediterranean Conference of Control and Automation (MED), Palermo, Italy, 16–19 June 2014; pp. 1418–1424.
19. Máthé, K.; Buşoniu, L.; Barabás, L.; Iuga, C.I.; Miclea, L.; Braband, J. Vision-based control of a quadrotor for an object inspection scenario. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 849–857.
20. Youeyun, J.; Hyochoong, B.; Dongjin, L. Robust marker tracking algorithm for precise UAV vision-based autonomous landing. In Proceedings of the 2015 15th International Conference on Control, Automation and Systems (ICCAS), Busan, Korea, 13–16 October 2015; pp. 443–446.
21. Hanseob, L.; Seokwoo, J.; Shim, D.H. Vision-based UAV landing on the moving vehicle. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Piscataway, NJ, USA, 7–10 June 2016; IEEE: Piscataway, NJ, USA, 2016.

22. Singh, G.; Anvar, A. Investigating feasibility of target detection by visual servoing using UAV for oceanic applications. In Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; pp. 1621–1626.
23. Choi, H.; Geeves, M.; Alsalam, B.; Gonzalez, F. Open source computer-vision based guidance system for UAVs on-board decision making. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016.
24. Why We Chose PX4 (vs. APM) as Luci's Default Firmware. Available online: https://medium.com/@Dronesmith/why-we-chose-px4-vs-apm-as-lucis-default-firmware-ea39f4514bef#.2a1pl81ao (accessed on 2 January 2017).
25. Robot Operating System. Available online: http://wiki.ros.org (accessed on 6 June 2017).
26. Sol. Detect Red Circles in an Image Using OpenCV. Available online: https://www.solarianprogrammer.com/2015/05/08/detect-red-circles-image-using-opencv/ (accessed on 3 February 2017).
27. Camera Calibration/Tutorials/MonocularCalibration. Available online: http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration (accessed on 6 June 2017).
28. Mavros. Available online: http://wiki.ros.org/mavros (accessed on 2 June 2017).
29. Stathopoulos, C. Visual Servoing Using a Webcam, Arduino and OpenCV. Available online: https://xstath.wordpress.com/2015/06/04/visual-servoing-using-a-webcam-arduino-and-opencv/ (accessed on 6 August 2016).
30. Gazebo. Available online: http://gazebosim.org/ (accessed on 7 August 2017).
31. Addabbo, P.; Angrisano, A.; Bernardi, M.L.; Gagliarde, G.; Mennella, A.; Nisi, M.; Ullo, S. A UAV infrared measurement approach for defect detection in photovoltaic plants. In Proceedings of the 2017 IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace), Padua, Italy, 21–23 June 2017; pp. 345–350.



sensors

MDPI

Article

Design of UAV-Embedded Microphone Array System for Sound Source Localization in Outdoor Environments †

Kotaro Hoshiba [1,*], Kai Washizaki [2], Mizuho Wakabayashi [2], Takahiro Ishiki [2,‡], Makoto Kumon [2], Yoshiaki Bando [3], Daniel Gabriel [1], Kazuhiro Nakadai [1,4] and Hiroshi G. Okuno [5]

1 Department of Systems and Control Engineering, School of Engineering, Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8552, Japan; gabriel@ra.sc.e.titech.ac.jp (D.G.); nakadai@ra.sc.e.titech.ac.jp (K.N.)

2 Graduate School of Science and Technology, Kumamoto University, 2-39-1 Kurokami, Chuo-ku, Kumamoto 860-8555, Japan; k.washizaki@ick.mech.kumamoto-u.ac.jp (K.W.); m.wakabayashi@ick.mech.kumamoto-u.ac.jp (M.W.); takahiro.ishiki@lixil.com (T.I.); kumon@gpo.kumamoto-u.ac.jp (M.K.)

3 Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan; yoshiaki@kuis.kyoto-u.ac.jp

4 Honda Research Institute Japan Co., Ltd., 8-1 Honcho, Wako, Saitama 351-0188, Japan

5 Graduate Program for Embodiment Informatics, Waseda University, 3-4-1 Okubo, Shinjuku-ku, Tokyo 169-8555, Japan; okuno@nue.org

* Correspondence: hoshiba@ra.sc.e.titech.ac.jp; Tel.: +81-3-5734-3883

† This is the extended version of Development of Microphone-Array-Embedded UAV for Search and Rescue Task, Kazuhiro Nakadai, Makoto Kumon, Hiroshi G. Okuno, Kotaro Hoshiba, Mizuho Wakabayashi, Kai Washizaki, Takahiro Ishiki, Daniel Gabriel, Yoshiaki Bando, Takayuki Morito, Ryosuke Kojima, Osamu Sugiyama, IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017.

‡ Current address: LIXIL Corporation, 2-1-1 Ojima, Koto-ku, Tokyo 136-8535, Japan.

Received: 7 October 2017; Accepted: 30 October 2017; Published: 3 November 2017

Abstract: In search and rescue activities, unmanned aerial vehicles (UAV) should exploit sound information to compensate for poor visual information. This paper describes the design and implementation of a UAV-embedded microphone array system for sound source localization in outdoor environments. Four critical development problems included water-resistance of the microphone array, efficiency in assembling, reliability of wireless communication, and sufficiency of visualization tools for operators. To solve these problems, we developed a spherical microphone array system (SMAS) consisting of a microphone array, a stable wireless network communication system, and intuitive visualization tools. The performance of SMAS was evaluated with simulated data and a demonstration in the field. Results confirmed that the SMAS provides highly accurate localization, water resistance, prompt assembly, stable wireless communication, and intuitive information for observers and operators.

Keywords: robot audition; sound source localization; multiple signal classification; outdoor-environment measurement; real-time measurement; unmanned aerial vehicle

1. Introduction

Research on remote sensing techniques involving unmanned aerial vehicles (UAV) is important to improve search and rescue in disaster-stricken areas because such technologies enable prompt action regardless of the terrain. Search and rescue tasks with UAV rely mainly on vision, which is

vulnerable to poor lighting conditions or occlusions. A UAV-embedded microphone array system is expected to be effective for the detection of people needing assistance in disaster-stricken areas. Since a UAV-embedded microphone array system receives rotor and wind noise as well as environmental sounds, the target sound is contaminated by ego-noise and other noise. Sound source processing should be able to localize, and discriminate a target sound from noise. Robot audition software [1–3] has been developed to cope with a mixture of sounds contaminated by noise. In particular, the open source robot audition software HARK (Honda Research Institute Japan Audition for Robots with Kyoto University) [4,5] provides noise-robust sound processing functions: sound source localization, source separation and recognition of separated sounds. Basic technologies of robot audition have been developed for use in indoor environments, and it is necessary to advance these technologies for use in outdoor environments, such as for search and rescue tasks using UAV. Five main challenges to developing such a system for UAV include:

1. sound source localization;
2. sound source separation and sound enhancement;
3. sound source classification;
4. real-time processing and intuitive visualization tools;
5. robustness of the device in outdoor environments.

The first challenge has been addressed in recent years in several studies including as a main research topic to find people in disaster situations, e.g., localization of an emergency signal from a safety whistle [6], and that of speech with a low signal-to-noise ratio (SNR) [7–9]. To locate the source of a sound, algorithms based on multiple signal classification (MUSIC) [10] are often used because they can effectively localize sound sources in highly noisy environments. In particular, MUSIC based on incremental generalized singular value decomposition with correlation matrix scaling (iGSVD-MUSIC-CMS) developed by Ohata et al. demonstrated good performance under dynamically changing noise [9]. iGSVD-MUSIC-CMS could localize sound sources in a low SNR environment, −15 dB. There is a severe trade-off between the speed and performance of the signal processing. Since only offline processing was reported in their evaluation, evaluation in real time is necessary for application to search and rescue tasks.

The second challenge, in order to identify a target sound source in extremely noisy environments, is also important, in two goals: to improve SNR of the target sound and to improve intelligibility of the separated signals. The first goal of the present study was to improve sound source classification (the third challenge). Recent studies on restoration of distorted signals have been reported including: sound source separation with a linear process [11] and integrated frameworks of sound source separation and classification using end-to-end training [12,13]. While the first goal targeted machine listening, the second goal targets human listening, that is, an operator tries to identify a target sound source manually, e.g., by inspecting sound spectrograms or by listening to separated sounds. In this case, intelligibility is a primary requirement. This second goal has not been reported as a function for UAV, although it is important to the UAV operator.

The third challenge, to effectively discriminate a target sound source, such as a human-induced sound, from other sound sources has been investigated [11–13], albeit these studies only reported offline processing and did not mention real-time processing.

Regarding the fourth challenge, latency in visualizing flight and sound source information should be as short as possible for efficient operation of UAV. Since UAV operators may be situated far from the UAV, visualization tools should be capable of displaying accurate information regarding UAV location and sound source. Finally, regarding the fifth challenge, to ensure system efficiency, all-weather acoustic sensors and reliability of wireless communication using a Wi-Fi signal that carries acoustic signals, which are necessary in outdoor environments, should be proven.

In this paper, we report the development of a UAV-embedded microphone array system that resolved four of the above five challenges. The third challenge, sound classification, which we regard to be at a higher level than the other four, will be investigated in a separate study. The remaining of the

paper is organized as follows: Section 2 describes the design method and details of the UAV-embedded microphone array system. Section 3 evaluates and discusses the performance of the system. Section 4 is the conclusion.

2. Methods

2.1. Design of Water-Resistant Microphone Array for Use Onboard UAV

To address the first and fifth challenges in Section 1, we designed and developed a microphone array.

Figure 1 shows our prototype hexagonal microphone array system (HMAS). Sixteen redMEMS (Micro Electro Mechanical Systems) microphones are set on a two-story hexagonal frame whose diagonal length is 1.8 m. The microphones and cables, being exposed, were vulnerable to water, and risk of disconnection. Additionally, the complexity of the frame demanded a lot of time for assembly of the HMAS. To solve these problems, we designed a spherical microphone array system (SMAS), which is water resistant and simple to assemble in a UAV (Figure 2). As shown in Figure 2a or Figure 2c, twelve MEMS microphones are embedded in a spherical body with a diameter of 0.1 m. Since a single strut and one cable connects the array to the UAV, its assembly is simple and the risk of disconnection is reduced. Unlike with the HMAS, where the microphones are equidistant around the UAV, the weight of the UAV is unbalanced with the SMAS. To solve this, we added two weights, each of the same size and mass, to counterbalance the SMAS. As shown in Figure 2d, the SMAS and weights are set at intervals of 120°, and the direction of the SMAS is 30° on the UAV coordinates. Figure 3 shows internal structure of the SMAS. To embed microphones into the body, gaskets were used. The microphone was attached to the gasket so that holes of the microphone and the gasket were coincident. When there is a gap between the microphone and the body, the microphone cannot receive acoustic signals precisely because of reverberations in the body. To fill gaps between microphones and the body, ring-shaped connectors were used. To ensure water resistance of the SMAS, holes of gaskets were covered with a water-resistant membrane and an antiweatherability tape. Since a water-resistant membrane and an antiweatherability tape are enough thin to pass acoustic signals through, they do not influence signals received by microphones.

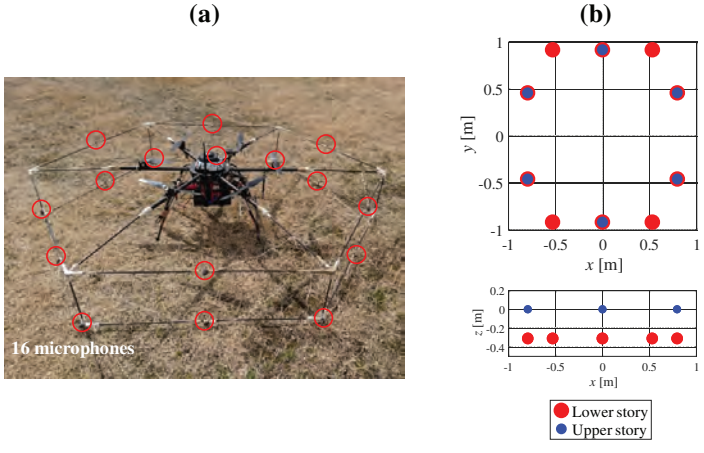


Figure 1. HMAS (hexagonal microphone array system). (**a**) the 16 microphones marked as red circles; (**b**) coordinates of the microphone positions in the HMAS.

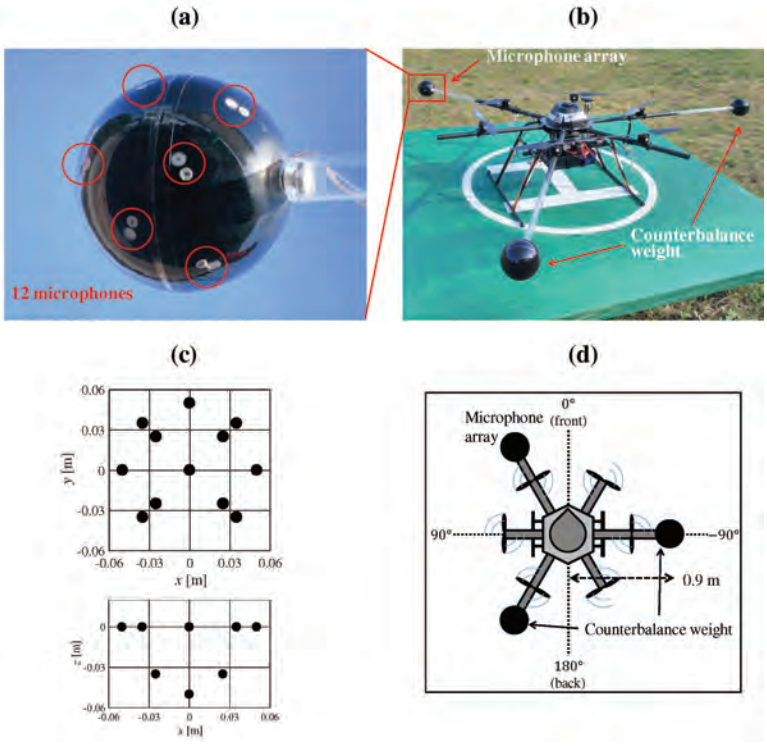


Figure 2. SMAS (spherical microphone array system). (**a**) the 12 microphones, and six of them marked as red circles; (**b**) UAV (unmanned aerial vehicles) with SMAS and two counterbalance weights; (**c**) coordinates of the microphone positions in the SMAS; (**d**) layout of the SMAS and two counterbalance weights in the UAV.

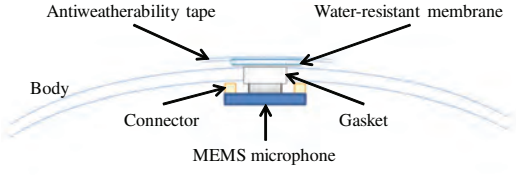


Figure 3. Internal structure of the SMAS.

2.2. Stabilization of Wireless Communication

To resolve the fifth challenge, to stabilize wireless communication, we incorporated a high-gain antenna that could receive a Wi-Fi signal from the UAV, which carries acoustic signals at a ground station. In addition, a communication protocol was also implemented to improve robustness.

For sound source localization, acoustic signals recorded by SMAS on the UAV are sent via wireless communication using a Wi-Fi signal to a ground station, and processed by a computer. A network system was constructed by assuming that the distance between the UAV and the ground station is short. However, in an outdoor environment, the network communication has the potential to be unstable as the distance increases. To ensure reliable wireless communication, two improvements were made.

First, we replaced an antenna at the ground station with the Yagi antenna (FX-ANT-A5, CONTEC (Osaka, Japan)) [14] to improve throughput of communication [15]. For acoustic signals recorded by 12 microphones, throughput of approximately 5 Mbps is necessary. Therefore, a high gain antenna was used for reliable wireless communication in outdoor environments. Figure 4 shows antennas (FX-ANT-A7, CONTEC) [16] on the UAV and the Yagi antenna at a ground station. On the UAV, two antennas was assembled to arms of UAV. At a ground station, the Yagi antenna was set on a tripod.

Figure 4. (**a**) antennas on the UAV marked as red circles; (**b**) the Yagi antenna at a ground station.

Second, we changed the communication protocol from TCP (Transmission Control Protocol) to UDP (User Datagram Protocol). In wireless communication tests at tens of meters of distance between the hovering UAV and the ground station, packet loss occurred approximately 400 times per minute. With TCP, each packet loss caused a retransmission request, greatly reducing acoustic signal throughput. Hence, the protocol was changed to UDP. Because UDP provides no guarantee of data integrity, no retransmission request is sent and throughput is maintained. When a packet loss occurs, the ground station receives defective acoustic signals. However, because the minimum frame size for sound source localization is much larger than one packet, the impact is negligible.

2.3. Development of Intuitive Visualization Tools for Operators

For the second and fourth challenges, we developed three visualization tools that display information regarding the sound source on the UAV coordinates and acoustic signals before and after their enhancement.

Essential to the system is a visualization tool to display sound source localization results. Several groups have developed such a tool for sound source localization [17–19]. Figure 5 shows MUSIC spectra produced by the MUSIC method [10]. It visualizes sound power arriving from each direction. The horizontal axis represents the frame number (time) and the vertical axis represents the azimuth angle θ (**a**) or the elevation angle ϕ (**b**). The sound power is represented as a color map. It is difficult to quickly determine the direction and time of a sound source with the MUSIC spectrum. In order to visualize sound source localization and the UAV location and orientation, we developed a tool to display such data on Google Earth^{TM} (Google (Los Angeles, CA, USA)) (Figure 6) [20]. Because users can change their viewpoint freely on Google Earth^{TM}, they can intuitively grasp the situation of the environment, the UAV, and the sound source. However, this tool is for observers only and not for the operator. Unlike an indoor environment, in an outdoor environment, the distance between the UAV and the operator may be large, necessitating a tool for its effective operation. Therefore, we developed visualization tools for operators.

Because an operator controls the UAV with reference to its coordinates, a user friendly method would be to display sound source directions on the UAV's coordinates. Therefore, the coordinate system shown in Figure 7 was defined. Forward direction of the UAV is defined as the positive direction of the y-axis, and the azimuth and elevation are projected to the circumferential and radial directions, respectively. Using this coordinate system, two visualization tools to display sound source directions were developed as shown in Figure 8. Figure 8a shows the MUSIC spectrum. The sound power in each direction is depicted by a color map. Figure 8b illustrates only the sound directions after threshold processing for the MUSIC spectrum. In addition, spectrograms of the recorded sound and

after sound enhancement by online robust principal component analysis (ORPCA) [21] are displayed as in Figure 8c. The left panel shows a spectrogram of the recorded acoustic signal, and the right panel shows it after enhancement. The horizontal and vertical axes represent time and frequency, respectively. By viewing these three sets of data in real time, even when located far from the UAV, the operator knows the relationship between the UAV and the sound source.

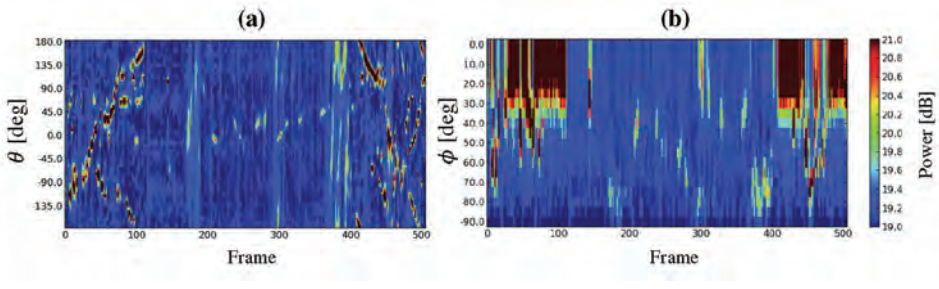


Figure 5. MUSIC (multiple signal classification) spectrum. (**a**) azimuth direction; (**b**) elevation direction.

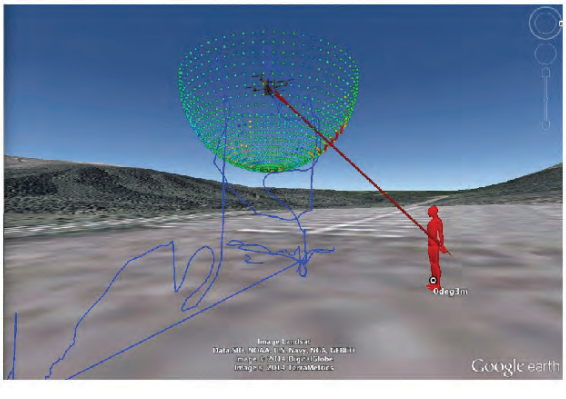


Figure 6. Previous visualization tool based on Google Earth™.

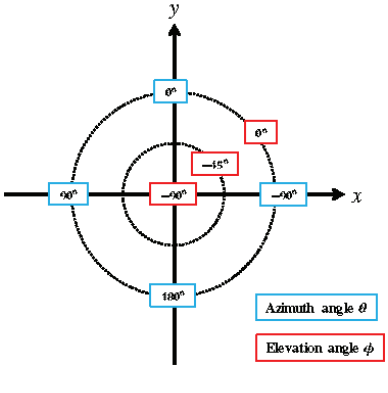


Figure 7. Visualization tool coordinate system.

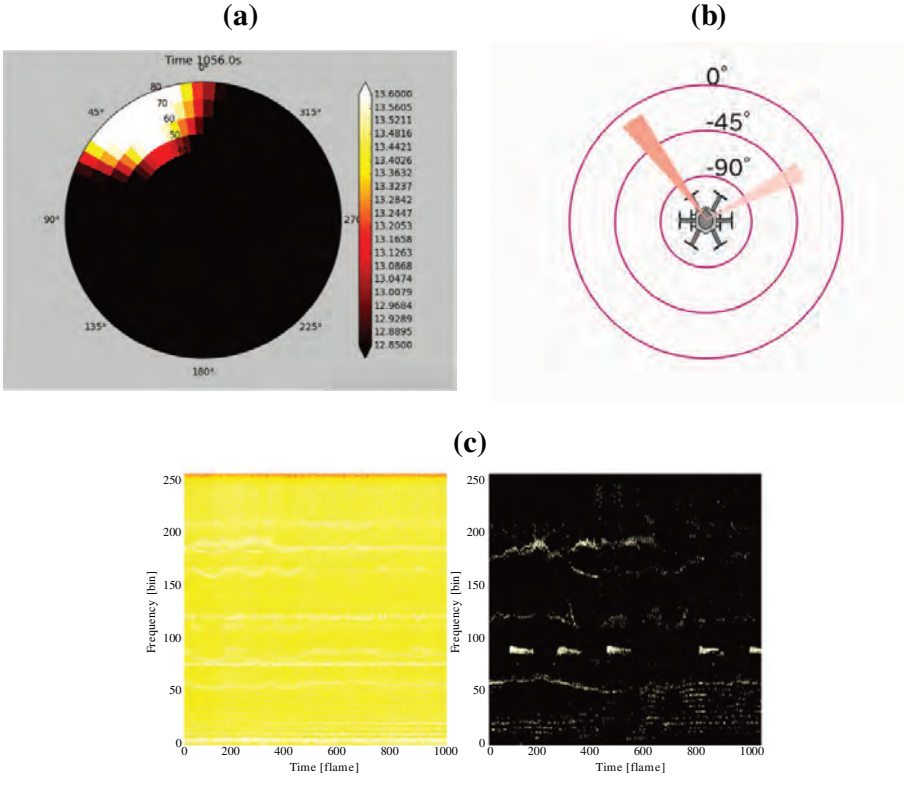


Figure 8. Visualization tools. (**a**) MUSIC spectrum; (**b**) sound direction; (**c**) spectrograms of captured sound (**left**) and after enhancement (**right**).

2.4. Sound Source Localization Method

We used two methods of sound source localization, namely SEVD-MUSIC (MUSIC based on Standard Eigen Value Decomposition), which is an original broadband MUSIC method [10], and iGSVD-MUSIC [9]. SEVD-MUSIC has low noise robustness and low computational cost, while iGSVD-MUSIC has high noise robustness and high computational cost. Either of these can be selected according to the circumstances. Algorithms of SEVD-MUSIC and iGSVD-MUSIC are described below.

2.4.1. SEVD-MUSIC

M channel input sound signals of the f-th frame are Fourier transformed to $\boldsymbol{Z}(\omega, f)$, from which a correlation matrix $\boldsymbol{R}(\omega, f)$ is defined as follows:

$$\boldsymbol{R}(\omega, f) = \frac{1}{T_R} \sum_{\tau=f}^{f+T_R-1} \boldsymbol{Z}(\omega, \tau)\boldsymbol{Z}^*(\omega, \tau). \quad (1)$$

ω is the frequency bin index, T_R is the number of frames used for the correlation matrix calculation, and $\boldsymbol{Z}^*$ is a complex conjugate transpose of $\boldsymbol{Z}$. The SEVD-MUSIC method calculates eigenvectors through an SEVD of the obtained $\boldsymbol{R}(\omega, f)$:

$$\boldsymbol{R}(\omega, f) = \boldsymbol{E}(\omega, f)\boldsymbol{\Lambda}(\omega, f)\boldsymbol{E}^*(\omega, f). \quad (2)$$

$\mathbf{\Lambda}(\omega, f)$ is a matrix with diagonal components that are eigenvalues in a descending order. $\boldsymbol{E}(\omega, f)$ is a matrix containing eigenvectors corresponding to $\mathbf{\Lambda}(\omega, f)$. Using $\boldsymbol{E}$, and a transfer function, $\boldsymbol{G}(\omega, \psi)$, corresponding to the sound source direction, $\psi = (\theta, \phi)$ in the UAV coordinates, the MUSIC spatial spectrum, $P(\omega, \psi, f)$, is calculated:

$$P(\omega, \psi, f) = \frac{|G^*(\omega, \psi)G(\omega, \psi)|}{\sum_{m=L+1}^{M} |G^*(\omega, \psi)e_m(\omega, \psi)|}. \tag{3}$$

L is the number of target sound sources, and e_m is the m-th eigenvector contained in $\boldsymbol{E}$. $P(\omega, \psi, f)$ is average over ω direction to estimate the direction of the sound source:

$$\bar{P}(\psi, f) = \frac{1}{\omega_H - \omega_L + 1} \sum_{\omega=\omega_L}^{\omega_H} P(\omega, \psi, f). \tag{4}$$

ω_H and ω_L are indices corresponding to the upper and lower limits of the used frequency bin, respectively. Threshold processing and peak detection is performed for $\bar{P}(\psi, f)$ and ψ of the obtained peak is detected as the sound source direction.

2.4.2. iGSVD-MUSIC

In iGSVD-MUSIC, for the f-th frame, the section of the length of T_N frames from the $f - f_s$-th frame is assumed to be a noise section, and the noise correlation matrix $\boldsymbol{K}(\omega, f)$ is calculated:

$$\boldsymbol{K}(\omega, f) = \frac{1}{T_N} \sum_{\tau=f-f_s-T_N}^{f+f_s} \boldsymbol{Z}(\omega, \tau)\boldsymbol{Z}^*(\omega, \tau). \tag{5}$$

The iGSVD-MUSIC method estimates noise in each frame and responds to dynamic change in noise. The noise component can be whitened by multiplying $\boldsymbol{K^{-1}}$ to $\boldsymbol{R}$ from the left. The iGSVD-MUSIC method calculates singular vectors through the GSVD of $\boldsymbol{K^{-1}}(\omega, f)\boldsymbol{R}(\omega, f)$:

$$\boldsymbol{K^{-1}}(\omega, f)\boldsymbol{R}(\omega, f) = \boldsymbol{Y_l}(\omega, f)\mathbf{\Sigma}(\omega, f)\boldsymbol{Y_r^*}(\omega, f). \tag{6}$$

$\mathbf{\Sigma}(\omega, f)$ is a matrix with diagonal components of singular values in a descending order. $\boldsymbol{Y_l}(\omega, f)$ and $\boldsymbol{Y_r}(\omega, f)$ are matrices containing singular vectors corresponding to $\mathbf{\Sigma}(\omega, f)$. Then, the MUSIC space spectrum is calculated:

$$P(\omega, \psi, f) = \frac{|G^*(\omega, \psi)G(\omega, \psi)|}{\sum_{m=L+1}^{M} |G^*(\omega, \psi)y_m(\omega, \psi)|}. \tag{7}$$

y_m is the m-th singular vector contained in Y_l. $P(\omega, \psi, f)$ is averaged over ω direction to estimate the direction of the sound source:

$$\bar{P}(\psi, f) = \frac{1}{\omega_H - \omega_L + 1} \sum_{\omega=\omega_L}^{\omega_H} P(\omega, \psi, f). \tag{8}$$

Threshold processing and peak detection is performed for $\bar{P}(\psi, f)$ and ψ of the obtained peak is detected as the sound source direction.

Both sound source localization methods based on MUSIC basically assume an acoustic far-field. However, by using the transfer function G according to the distance to sound sources, it is possible to localize sound sources at any distance. In addition, at the altitude at which a UAV flies normally (at least a few meters), an acoustic field is a far-field. Therefore, the accuracy of sound source localization depends on a SNR of an acoustic signal rather than a distance between a microphone array to a sound source.

2.5. Structure of Microphone Array System

By integrating the above components, the microphone array system was constructed. Figure 9 shows the SMAS configuration. The microphone array on the UAV was connected to a multi-channel sound signal recorder, RASP-ZX (System In Frontier (Tokyo, Japan)) [22] for synchronous recording of 12 ch sound signals. The sound signals were recorded at a sampling frequency of 16 kHz, and a quantization bit rate of 24 bits. Recorded acoustic signals, images from the wireless camera and data from a GNSS/IMU (Global Navigation Satellite System/Inertial Measurement Unit) sensor were transmitted through a wireless network to the ground station. Different frequencies were used for the wireless communications to prevent cross talk. In the SMAS, data from a GNSS/IMU sensor and images from the wireless camera were not used; therefore, only recorded acoustic signals were received by the Yagi antenna. The received data was integrated using ROS (Robot Operating System) to provide general versatility. The acoustic signals were processed by a PC using a sound source localization method. HARK was used for the algorithm implementation. The data after processing was shared by three PCs via a router. To reduce the processing load of one computer for real-time visualization, visualization tools were displayed using three laptops. PC1, PC2 and PC3 displayed the MUSIC spectrum (Figure 8a), sound direction (Figure 8b) and enhanced sound (Figure 8c), respectively. Since the SMAS is a separate system from the UAV, including its power supply, it can be applied to various UAVs.

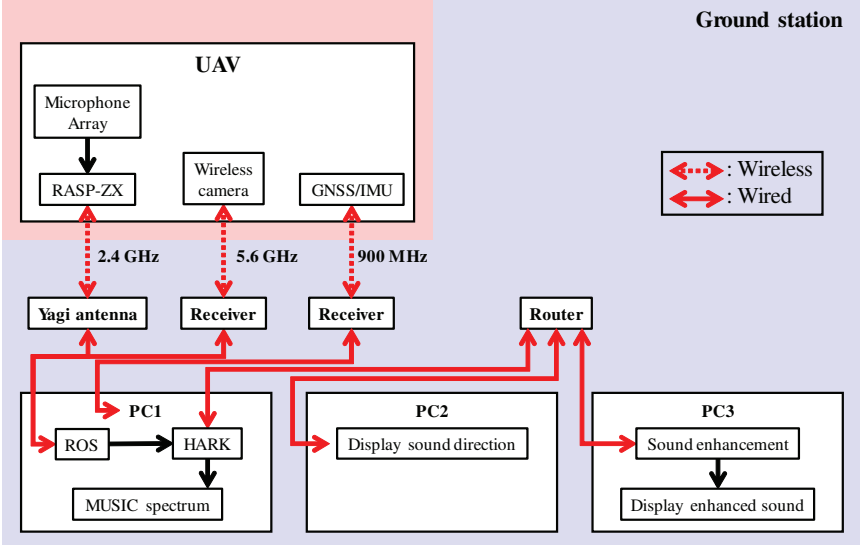


Figure 9. Configuration of SMAS.

3. Results and Discussion

The performance of the SMAS was evaluated using numerical sound simulation and by demonstration in an outdoor environment.

3.1. Evaluation Procedure

Sound localization performance was evaluated using acoustic signals created in a numerical simulation. Using transfer functions corresponding to two types (hexagonal and spherical) of microphone array and sound samples, acoustic signals arriving from every direction were created. Recorded noise of an actual flying UAV was added to the created signals. The direction was set as

every 5° in the azimuth range from −180° to 180° and the elevation range from −90° to 0°. As sound sources, a whistle and human voice were used. A Mini Surveyor MS-06LA (Autonomous Control Systems Laboratory (Chiba, Japan)) was used as the UAV. Spectrograms of the sound sources and the noise of the UAV recorded by each of the two microphone arrays are shown in Figure 10. Simulated signals were created in different SNR, −20, −10, 0, 10 and 20 dB. Simulated signals were processed by the SMAS and results were evaluated. Performance was also evaluated by demonstration in the field.

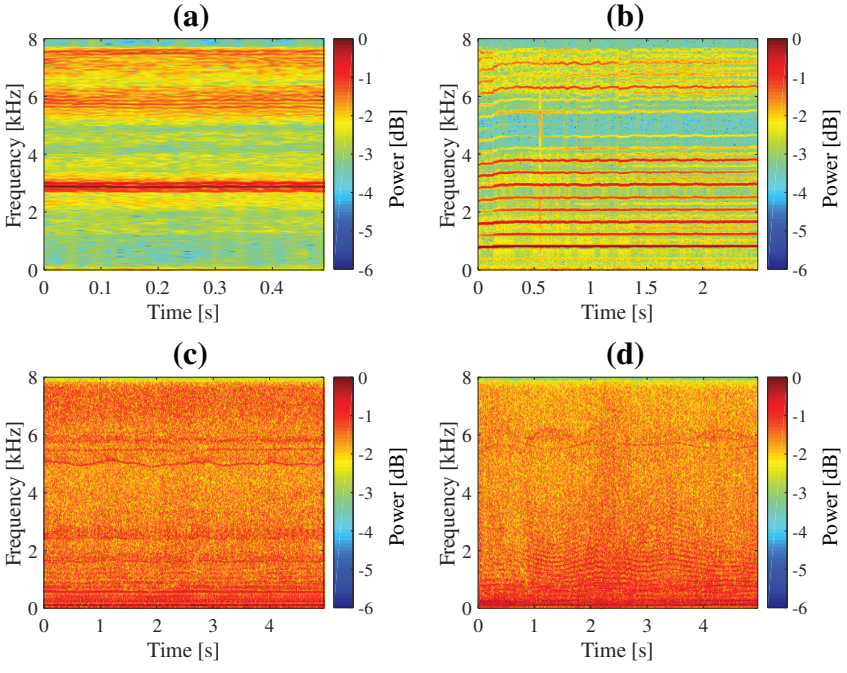


Figure 10. Spectrograms. (**a**) whistle; (**b**) voice; (**c**) noise of UAV recorded by hexagonal microphone array; (**d**) noise of UAV recorded by spherical microphone array.

3.2. Results of Simulation

The main differences between SEVD-MUSIC and iGSVD-MUSIC are noise robustness and computational cost. Since its computational cost is low, SEVD-MUSIC has a short delay; however, it has poor noise tolerance. Since iGSVD-MUSIC includes noise whitening, it has noise tolerance but a long delay. Thus, real-time property and noise tolerance are in a trade-off relationship. Figure 11 shows MUSIC spectra processed by SEVD-MUSIC (**a**) and by iGSVD-MUSIC (**b**) using spherical microphone array. The target sound is located around $\theta = 80°$. In both MUSIC spectra, the target sound source power can be seen. However, in Figure 11a, the noise power of the UAV can also be seen. Figure 12 shows the delay in the system when the frequency range, which is used in the MUSIC method, is changed. The horizontal axis represents the frequency range and the number of the frequency bin ($\omega_H - \omega_L + 1$), and the vertical axis represents delay. As shown in Figure 12, iGSVD-MUSIC has a time delay of 2 to 3 s longer than that of SEVD-MUSIC. In addition, as the frequency range increases, the time delay increases.

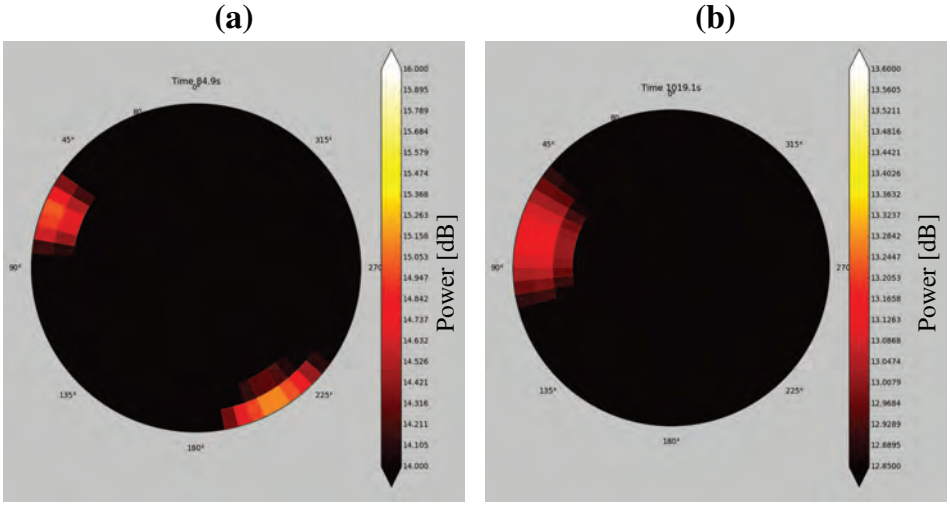


Figure 11. MUSIC spectra. (**a**) SEVD-MUSIC (MUSIC based on Standard Eigen Value Decomposition); (**b**) iGSVD-MUSIC (MUSIC based on incremental generalized singular value decomposition).

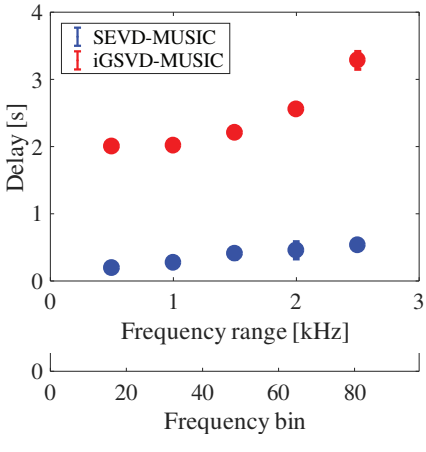


Figure 12. Time delay of the system.

Based on these results, localization performance was evaluated by its success rate. The success rate was calculated based on the UAV coordinates. When the angle of the maximum value of the MUSIC spectrum is matched with a set angle, it is defined that sound source localization succeeded. All simulated sounds were processed using sound source localization method, and the success rate was calculated. Figure 13 shows the success rate of localization for the hexagonal and a spherical microphone arrays, processed by SEVD-MUSIC and iGSVD-MUSIC. The frequency range used in the MUSIC method was from 500 to 3000 Hz. In the hexagonal array, the success rate of both MUSIC methods was almost 100% even with SNR less than 0 dB. In the spherical array, the success rate was lower than that of hexagonal. In particular, the success rate of SEVD-MUSIC was less than 30% when the SNR was −20 dB. This lower success rate was considered due to the smaller aperture diameter in the spherical array at 0.1 m compared to 1.8 m in the hexagonal. Therefore, the detection

area of the MUSIC spectrum was limited to increase the accuracy of localization with SEVD-MUSIC. As shown in Figure 11a, noise of the UAV appear in one direction constantly as directional noise, in this case at the azimuth angle of around −150°. To avoid the effect of such noise, the detection azimuth angle was limited to $-60° \leq \theta \leq 120°$. The success rate in a case when the detection angle was limited is plotted as the green line in Figure 13. By limiting the detection angle, the success rate of short-delay SEVD-MUSIC using the spherical microphone array with SNR −10 dB could be increased to approximately 97%. Since the SNR, when blowing a whistle or speaking to an actual flying UAV from a distance of around 10 m was approximately −10 to 0 dB, it was considered sufficient localization performance. This technique can be used with microphones located at one site on the UAV, unlike the HMAS in which microphones are dispersed around the UAV. Due to the location of our microphone array, parameters for sound source localization could be easily tuned to attain accurate localization with a small latency.

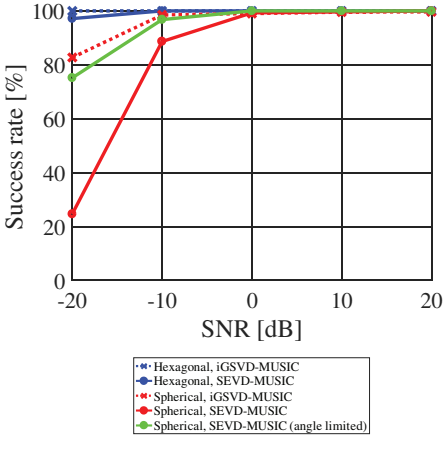


Figure 13. Success rate of localization.

3.3. Results of the Demonstration

Regarding efficiency in assembling the system, the HMAS took two hours to assemble, and especially time consuming was assembly of the frame and electric cables. In contrast, the SMAS took 40 min to assemble and 2 min to take off after switching on the UAV. Regarding water resistance, although the demonstration was performed in light rain, the SMAS worked without failure. To assess the reliability of wireless communication, throughputs were compared among four different antennas. Figure 14 shows the results of throughputs by antenna type: Diversity (FX-ANT-A1, CONTEC) [23], small Yagi (FX-ANT-A3, CONTEC) [24], Collinear (FX-ANT-A2, CONTEC) [25] and large Yagi (used in SMAS). Throughputs were measured by fixing the UAV on the ground at distances, 10, 30, 50, 75 and 100 m, with propeller rotation speeds, 0, 3000 and 5000 rpm. The required throughput (5 Mbps) is shown as a dotted line in Figure 14. It was found that throughput surpassed 5 Mbps even at 75 m by using the large Yagi-antenna. In the demonstration, the wireless network worked without disconnecting in the distance of tens of meters. To examine the intuitiveness of visualization tools, camera image, MUSIC spectrum, sound direction, and enhanced sound data were displayed as in Figure 15. These visualization tools provided directions of sound sources and other data in real time for the audience and operator, intuitively.

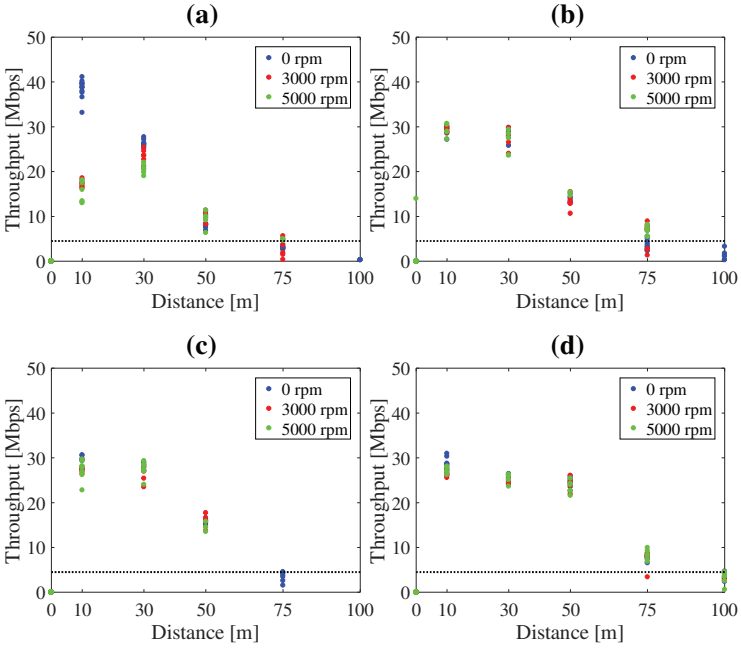


Figure 14. Throughputs by antennas type. (**a**) diversity; (**b**) Yagi (small); (**c**) collinear; (**d**) Yagi (large).

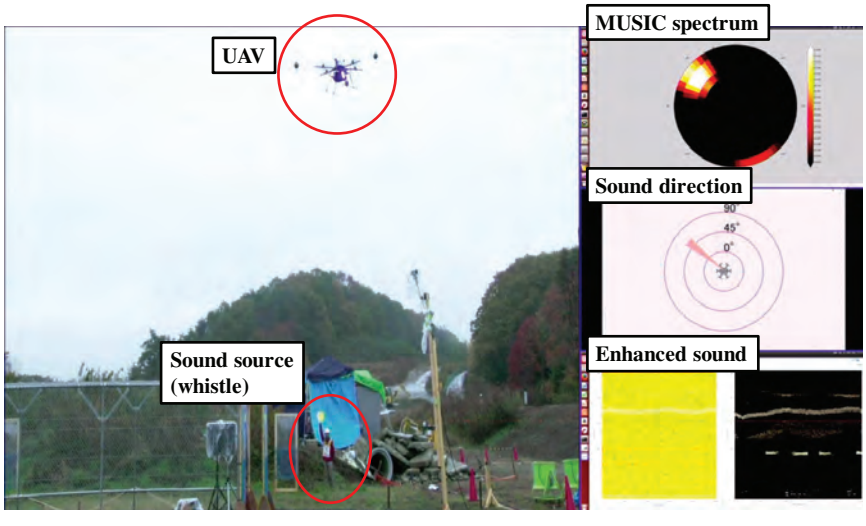


Figure 15. Data visualized in the demonstration.

3.4. Discussion

Before the demonstration, we conducted over 10 test flights, and all sound source localization trials were successfully completed. Thus, usability of the SMAS was verified. Tables 1 and 2 show a summary of pros and cons of each microphone array system and sound source localization method. For the microphone array system, the HMAS provides high accurate localization; however, it does not

have water resistance and efficiency in assembling. The SMAS provides lower accurate localization than the HMAS; however, we can increase the accuracy of localization depending on sound source localization method. For sound source localization method, SEVD-MUSIC has low noise tolerance and a small latency, while iGSVD-MUSIC has high noise tolerance and a large latency. Angle-limited SEVD-MUSIC can have high noise tolerance only when microphones located at one site on the UAV like the SMAS. Thus, because of their characteristics, we can select them according to the situation. In the demonstration, sound sources could be localized in real time with high accuracy using the SMAS and angle-limited SEVD-MUSIC because the SNR of the recorded acoustic signal was over −10 dB. However, in order to develop the system for the detection of people in a disaster-stricken area, a new sound source localization method with higher noise robustness and lower computational cost is needed. In addition, since there are several sound sources at an actual site, it is necessary to separate and identify human-related sound from recorded sounds. In future work, we will integrate the proposed sound source identification method using deep-learning [11–13] to the SMAS.

Table 1. Pros and cons of the HMAS and the SMAS.

	Accuracy of Localization	Water Resistance	Efficiency in Assembling
HMAS	◯	×	×
SMAS	△	◯	◯

Table 2. Pros and cons of SEVD-MUSIC, iGSVD-MUSIC and angle-limited SEVD-MUSIC.

	Noise Tolerance	Latency
SEVD-MUSIC	×	◯
iGSVD-MUSIC	◯	×
Angle-limited SEVD-MUSIC	△	◯

4. Conclusions

In this paper, we developed a UAV-embedded microphone array system for an outdoor environment. First, a novel microphone array was designed to ensure water resistance and efficiency of assembly. A 12 ch microphone array, including a spherical body of simple structure, was designed. By using coated microphones and a simple structure, water resistance and efficiency of assembly were ensured. Second, the antenna and communication protocol were changed to obtain reliable wireless communication. To improve throughput, the antenna at the ground station was changed to the Yagi antenna. To avoid reducing throughput, the communication protocol was changed from TCP to UDP. Third, intuitive visualization tools for a UAV operator were developed. By integrating the above improvements, the microphone array system was constructed. Tests showed that our microphone array system for an outdoor environment that is independent from the UAV provides highly accurate sound source localization performance in real time, and has effective intuitive operator visualization tools.

Acknowledgments: The authors would like to thank the members of System In Frontier Inc. for their support. This work was supported by JSPS (Japan Society for the Promotion of Science) KAKENHI Grant Nos. 16H02884, 16K00294, and 17K00365, and also by the ImPACT (Impulsing Paradigm Change through Disruptive Technologies Program) of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

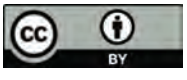
Author Contributions: K.H., K.W., M.W., T.I., M.K., Y.B., K.N. and H.G.O. developed proposed methods; All the authors conceived, designed and performed the experiments and the demonstration; K.H. analyzed the data and wrote the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakadai, K.; Lourens, T.; Okuno, H.G.; Kitano, H. Active audition for humanoid. In Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, TX, USA, 30 July–3 August 2000; pp. 832–839.
2. Okuno, H.G.; Nakadai, K. Robot audition: Its rise and perspectives. In Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015), Brisbane, Australia, 19–24 April 2015; pp. 5610–5614.
3. Okuno, H.G.; Nakadai, K. Special Issue on Robot Audition Technologies. *J. Robot. Mech.* **2017**, *29*, 15–267, doi:10.20965/jrm.2017.p0015.
4. Nakadai, K.; Takahashi, T.; Okuno, H.G.; Nakajima, H.; Hasegawa, Y.; Tsujino, H. Design and Implementation of Robot Audition System 'HARK'—Open Source Software for Listening to Three Simultaneous Speakers. *Adv. Robot.* **2010**, *24*, 739–761, doi:10.1163/016918610X493561.
5. HARK. Available online: http://www.hark.jp/ (accessed on 2 November 2017).
6. Basiri, M.; Schill, F.; Lima, P. U.; Floreano, D. Robust acoustic source localization of emergency signals from Micro Air Vehicles. In Proceedings of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 4737–4742.
7. Okutani, K.; Yoshida, T.; Nakamura, K.; Nakadai, K. Outdoor auditory scene analysis using a moving microphone array embedded in a quadrocopter. In Proceedings of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 3288–3293.
8. Furukawa, K.; Okutani, K.; Nagira, K.; Otsuka, T.; Itoyama, K.; Nakadai, K.; Okuno, H.G. Noise correlation matrix estimation for improving sound source localization by multirotor UAV. In Proceedings of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), Tokyo, Japan, 3–8 November 2013; pp. 3943–3948.
9. Ohata, T.; Nakamura, K.; Mizumoto, T.; Tezuka, T.; Nakadai, K. Improvement in outdoor sound source detection using a quadrotor-embedded microphone array. In Proceedings of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), Chicago, IL, USA, 14–18 September 2014; pp. 1902–1907.
10. Schmidt, R.O. Multiple emitter location and signal parameter estimation. *IEEE Trans. Antennas Propag.* **1986**, *34*, 276–280, doi:10.1109/TAP.1986.1143830.
11. Sugiyama, O.; Uemura, S.; Nagamine, A.; Kojima, R.; Nakamura, K.; Nakadai, K. Outdoor Acoustic Event Identification with DNN Using a Quadrotor-Embedded Microphone Array. *J. Robot. Mech.* **2017**, *29*, 188–197, doi:10.20965/jrm.2017.p0188.
12. Morito, T.; Sugiyama, O.; Kojima, R.; Nakadai, K. Reduction of Computational Cost Using Two-Stage Deep Neural Network for Training for Denoising and Sound Source Identification. In Proceedings of the IEA/AIE 2016 Trends in Applied Knowledge-Based Systems and Data Science Volume 9799 of the Series Lecture Notes in Computer Science, Morioka, Japan, 2–4 August 2016; pp. 562–573.
13. Morito, T.; Sugiyama, O.; Kojima, R.; Nakadai, K. Partially Shared Deep Neural Network in Sound Source Separation and Identification Using a UAV-Embedded Microphone Array. In Proceedings of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1299–1304.
14. FX-ANT-A5 Option Antenna for FLEXLAN. Available online: https://www.contec.com/products-services/computer-networking/flexlan-fx/fx-accessories/fx-ant-a5/ (accessed on 2 November 2017).
15. Ishiki, T.; Kumon, M. Continuous transfer of sensor data from multi-rotor helicopter. In Proceedings of the 33-th Annual Conference of the RSJ, Tokyo, Japan, 3–5 September 2015; RSJ2015AC1L3-03. (In Japanese)
16. FX-ANT-A7 Option Antenna for FLEXLAN. Available online: https://www.contec.com/products-services/computer-networking/flexlan-fx/fx-accessories/fx-ant-a7/ (accessed on 2 November 2017).
17. Sasaki, Y.; Masunaga, S.; Thompson, S.; Kagami, S.; Mizoguchi, H. Sound Localization and Separation for Mobile Robot Tele-Operation by Tri-Concentric Microphone Array. *J. Robot. Mech.* **2007**, *19*, 281–289, doi:10.20965/jrm.2007.p0281.
18. Kubota, Y.; Yoshida, M.; Komatani, K.; Ogata, T.; Okuno, H.G. Design and Implementation of 3D Auditory Scene Visualizer towards Auditory Awareness with Face Tracking. In Proceedings of the Tenth IEEE International Symposium on Multimedia (ISM), Berkeley, CA, USA, 15–17 December 2008; pp. 468–476.

19. Mizumoto, T.; Nakadai, K.; Yoshida, T.; Takeda, R.; Otsuka, T.; Takahashi, T.; Okuno, H.G. Design and Implementation of Selectable Sound Separation on the Texai Telepresence System using HARK. In Proceedings of the IEEE International Conference on Robots and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 2130–2137.
20. Hoshiba, K.; Sugiyama, O.; Nagamine, A.; Kojima, R.; Kumon, M.; Nakadai, K. Design and assessment of sound source localization system with a UAV-embedded microphone array. *J. Robot. Mech.* **2017**, *29*, 154–167, doi:10.20965/jrm.2017.p0154.
21. Feng, J.; Xu, H.; Yan, S. Online robust PCA via stochastic optimization. In Proceedings of the Neural Information Processing Systems Conference (NIPS), Stateline, NV, USA, 5–10 December 2013; pp. 404–412.
22. Acoustic Processing Unit (RASP-ZX). Available online: http://www.sifi.co.jp/system/modules/pico/index.php?content_id=36&ml_lang=en (accessed on 2 November 2017).
23. FX-ANT-A1 Option Antenna for FLEXLAN. Available online: https://www.contec.com/products-services/computer-networking/flexlan-fx/fx-accessories/fx-ant-a1/ (accessed on 2 November 2017).
24. FX-ANT-A3 Option Antenna for FLEXLAN. Available online: https://www.contec.com/products-services/computer-networking/flexlan-fx/fx-accessories/fx-ant-a3/ (accessed on 2 November 2017).
25. FX-ANT-A2 Option Antenna for FLEXLAN. Available online: https://www.contec.com/products-services/computer-networking/flexlan-fx/fx-accessories/fx-ant-a2/ (accessed on 2 November 2017).



sensors

MDPI

Article

Adaptation of Dubins Paths for UAV Ground Obstacle Avoidance When Using a Low Cost On-Board GNSS Sensor

Ramūnas Kikutis, Jonas Stankūnas, Darius Rudinskas and Tadas Masiulionis *

Department of Avionics, Vilnius Gediminas Technical University, 02187 Vilnius, Lithuania; ramunas.kikutis@vgtu.lt (R.K.); jonas.stankunas@vgtu.lt (J.S.); darius.rudinskas@vgtu.lt (D.R.)
* Correspondence: tadas.masiulionis@vgtu.lt; Tel.: +370-62-055157

Received: 12 July 2017; Accepted: 26 September 2017; Published: 28 September 2017

Abstract: Current research on Unmanned Aerial Vehicles (UAVs) shows a lot of interest in autonomous UAV navigation. This interest is mainly driven by the necessity to meet the rules and restrictions for small UAV flights that are issued by various international and national legal organizations. In order to lower these restrictions, new levels of automation and flight safety must be reached. In this paper, a new method for ground obstacle avoidance derived by using UAV navigation based on the Dubins paths algorithm is presented. The accuracy of the proposed method has been tested, and research results have been obtained by using Software-in-the-Loop (SITL) simulation and real UAV flights, with the measurements done with a low cost Global Navigation Satellite System (GNSS) sensor. All tests were carried out in a three-dimensional space, but the height accuracy was not assessed. The GNSS navigation data for the ground obstacle avoidance algorithm is evaluated statistically.

Keywords: UAV navigation; ground obstacle avoidance algorithm; Dubins paths; GNSS position measurements; position accuracy

1. Introduction

Nowadays, small unmanned aerial vehicles (UAVs) are becoming increasingly popular and are being more widely used for various practical applications. These vehicles can be used for various missions, such as small cargo delivery (e.g., delivering an automated external defibrillator) [1], electrical line monitoring or visual inspection of other ground infrastructure with a difficult access [2], fast response for post-earthquake consequences and initial help [3], surveillance, mapping and 3D visualization of ground objects [4], as a tool to get photos and other information for landscape and urban planning [5], etc. In spite of the wide application, a lot of restrictions at the international and national levels are issued by such organizations as the European Aviation Safety Agency (EASA), the Federal Aviation Administration (FAA) or local Civil Aviation Administrations (CAAs) [6]. Since these restrictions for UAV flights are due to the current lack of flight safety and reliability, research in various UAV related fields is of great importance. Researchers are searching for methods to ensure that automated UAV flights meet national airspace restrictions and limitations [7]. Otherwise, it will be impossible to implement and use small UAVs for all practical applications.

Research on the flight control systems of UAVs has a direct relationship with flight navigation accuracy and flight safety. This field is usually subdivided into three branches: guidance, navigation, and control. These areas can be analyzed at once, as in the paper on the control systems of two cubic blimps [8], or can be separated and analyzed independently. While research in these fields is important for UAV flight safety, currently, the greatest focus is on new navigation systems and sensing technologies, such as visual odometry, target relative navigation, terrain relative navigation,

simultaneous localization and mapping, simultaneous mapping and planning, and safe landing area detection. Guidance systems are also widely analyzed. Techniques from navigation systems and sensing technologies are applied for various flight procedures, whereas algorithms are used for mission planning, path planning and multi-UAV cooperation. Research in this field includes studies on potential fields, when a UAV mission is generally controlled by using the potential field to steer the UAV in the correct direction [9,10], and optimization methods, such as finding the most efficient flight trajectory, for example, in an urban environment [11]. Other studies focus on planning navigation under uncertainties, coordinated flight control, cooperative perception or coordinated flights with path planning by avoiding ground obstacles [12].

One of the goals for UAV guidance systems emphasized in recent research is path planning under uncertainties, which can be achieved in different ways. Some papers analyze the use of camera sensors to detect ground obstacles, while others propose combinations of various low cost sensors [13]. If a camera is used, the methods for obstacle detection, avoidance and UAV control can be implemented in areas even without a Global Navigation Satellite System (GNSS) signal [14]. Camera images can be used for ground obstacle detection and identification, tracking of other UAVs, and, if needed, as a navigation data sensor for other ground obstacle avoidance algorithms [15]. The image depth may be used to predict the distance to the obstacle [16] and, in some cases, can be shared between several unmanned vehicles [17]. Using these camera-based information methods, even auto-landings can be implemented [18]. However, in any case, the camera placement position and all angles should be calculated [19].

Obstacle detection is only one of the goals; others include the accurate avoidance of a ground obstacle or flights in a complex environment [20] and require the implementation of different methods or algorithms to attain them. If, for example, the ground obstacles are of various sizes and shapes or have a different height, different mapping techniques should be applied, as demonstrated in [21] and, where ellipsoidal geometry is used, in [22]. However, sometimes it is necessary to generate additional waypoints to safely overfly or to fly around a ground obstacle [23], or to change some additional UAV flight parameters (e.g., roll angle, airspeed) or the controller [24].

As noted earlier, if we want to use small UAVs for various practical applications, we need both to safely overfly or fly around the ground obstacles, and to meet all legal restrictions on UAV flights. One of the methods to attain this goal is to use Dubins paths. In the next section, a general idea of Dubins paths and their methodology is provided together with a review on current research on UAV navigation and obstacle avoidance.

2. Dubins Paths Methodology

In geometry, the term Dubins path typically refers to the shortest curve that connects two points. The calculations can be done in either a two-dimensional or a three-dimensional space with a constraint on the path's curvature and with prescribed initial and terminal tangents to the path. More precisely, for Dubins paths, each navigation leg corresponds to either the form CSC or CCC, where C stands for a concave or convex circle segment, and S—for a line segment. These flight path solutions are commonly called Dubins paths [25]. A general idea of Dubins paths and their implementation is proposed in [26,27]. Moreover, if we need to use in our research flight dynamics and kinematics, Dubins paths problems can be approached by using the Dubins aircraft model, which is presented and used in [28]. In some of the studies, Dubins paths are used in the Dubins traveling salesman problem (DTSP), which refers to a Dubins vehicle. This vehicle moves at a constant speed, cannot move in reverse, and has a minimum turning radius. Dubins vehicles are widely used for the motion planning of aircraft, ships or UAVs [25].

For UAVs, a simple flight mission, which has some straight flight segments, can be converted into Dubins paths by inserting filleted circular arcs near the turning points, as shown in Figure 1, which illustrates smoothed flight trajectory by inserting fillets near the turning points thus forming Dubins paths.

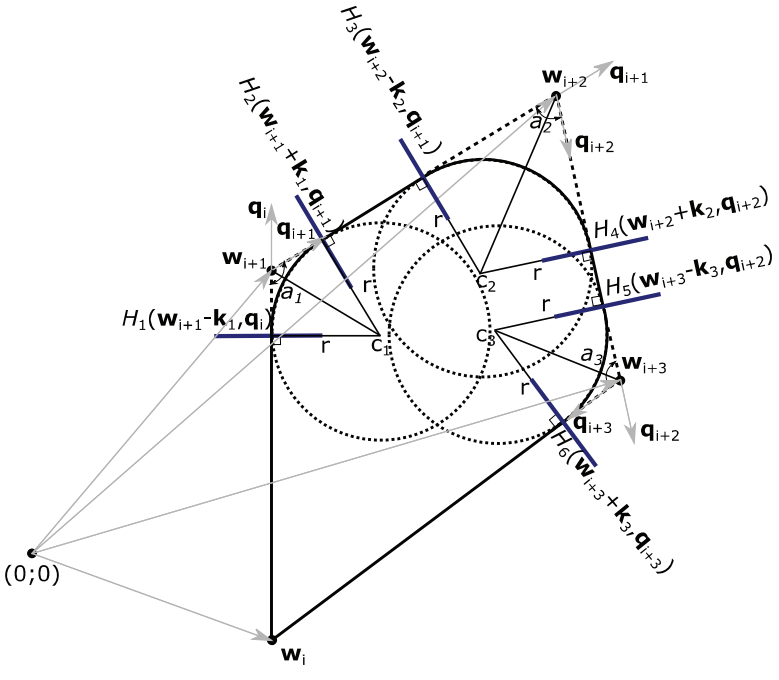


Figure 1. Conversion of a simple UAV mission into a Dubins trajectory.

To get the numerical data for the transition from the first flight segment which is connected by two navigation points $\mathbf{w}_i$ and $\mathbf{w}_{i+1}$ to the flight segment with points $\mathbf{w}_{i+1}$ and $\mathbf{w}_{i+2}$, unit vectors $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ (which show the directions of both flight segments) are needed. Unit vectors $\mathbf{q}_i$ and $\mathbf{q}_{i+1}$ can be calculated using Equations (1) and (2):

$$\mathbf{q}_i = \frac{\mathbf{w}_{i+1} - \mathbf{w}_i}{\|\mathbf{w}_{i+1} - \mathbf{w}_i\|} \tag{1}$$

$$\mathbf{q}_{i+1} = \frac{\mathbf{w}_{i+2} - \mathbf{w}_{i+1}}{\|\mathbf{w}_{i+2} - \mathbf{w}_{i+1}\|} \tag{2}$$

Since a UAV needs a command to switch from a straight to a circular flight path segment, the coordinate where the transition happens has to be defined. For this reason, angles a are needed, which are found by using Equation (3):

$$\angle a_x = \cos^{-1}(-\mathbf{q}_i^T \mathbf{q}_{i+1}) \tag{3}$$

Then, using angle a and the UAV turning radius, the distance from waypoint $\mathbf{w}_{i+1}$ for the transition between flight segments is found. With unit vectors $\mathbf{q}$ and distances $\mathbf{k}$ as in Equation (4), planes H_1 and H_2 can be found, representing the coordinates where a UAV does the transition from a straight flight segment into a turning—circular flight segment and then switches from a circular into a straight one by using Equations (5) and (6):

$$\mathbf{k}_1 = -\left(\frac{r}{\tan\frac{\angle a_1}{2}}\right)\mathbf{q}_{i+1} \tag{4}$$

$$H_1(\mathbf{w}_{i+1} - \mathbf{k}_1, \mathbf{q}_i) \tag{5}$$

$$H_2(\mathbf{w}_{i+1} + \mathbf{k}_1, \mathbf{q}_{i+1}) \tag{6}$$

If a UAV crosses the coordinates of plane H_1 or H_2, the transition is made. For complete calculations, the flight coordinates of the circular arcs must also be known. In this case, the UAV needs to know the center points of the Dubins circle which can be calculated with Equation (7):

$$\mathbf{c}_1 = \mathbf{w}_{i+1} - \left(\frac{r}{\sin \frac{\angle a_1}{2}} \right) \frac{\mathbf{q}_i - \mathbf{q}_{i+1}}{\|\mathbf{q}_i - \mathbf{q}_{i+1}\|} \tag{7}$$

The same technique can be used for all waypoints.

This or an equivalent methodology is used in various research papers. For example, a Dubins paths navigation methodology was proposed to achieve more precise and accurate UAV flights in [26,27,29,30]. In these papers, one of the tasks is to use the Dubins paths. For example, in [30], the authors propose to solve the Generalized Dubins Path Problem (GDPP), whereas in [27], a mathematical derivation of the Dubins paths is presented. In [27], the authors also use UAV flight dynamics and apply a vector field to keep the UAV on the generated Dubins paths flight mission. However, a great part of these papers is concerned with adapting Dubins paths to generate a path with the shortest distance or to search for the shortest time distance [10,31–34]. For example, as stated in [10], the UAV joins a predesigned admissible circular trajectory around a target, which is a fixed point space. Afterwards, this method is used to generate a time-optimal control synthesis for tracking a circle by a Dubins vehicle [10]. In other studies, Dubins paths and other autonomous UAV navigation techniques are also evaluated in windy conditions [35]. In [31], the authors consider the problem of determining a time-optimal path for a fixed-wing Miniature Air Vehicle (MAV) also taking into account the wind. As an alternative to wind, in [34], the authors try to find the minimum time of Dubins paths in the presence of a mathematically generated drift field.

In some publications on Dubins paths, optimization methods are applied or updated for ground obstacle avoidance, as proposed in [36]. In this paper, Dubins paths were applied for both static and moving ground obstacle avoidance by using a variation of the Rapidly-exploding Random Tree (RRT) planner. Very similar research was done by [37]. In this study, a combination of both static and moving obstacles was used in an enclosed territory, as well as a quadcopter and a developed search-and-avoid algorithm for obstacles. Search-and-avoid algorithms for Dubins paths can be developed by using several different techniques that fall broadly into three categories:

1. Methods where a suitable order of the obstacles is determined first and then suitable overflight directions are found;
2. Methods where suitable overflight directions are found first and then the order of the obstacles is optimized;
3. Methods that do not separate the order of operations (an overflight direction is found and an order of obstacles is chosen at the same time).

These algorithms when Dubins paths are used can also be applied to avoid moving aerial obstacles—other UAVs or aircraft [38].

Various ground obstacle avoidance methods are proposed in different studies. For example, in [39], a virtual force field is generated around the ground obstacle. This field has various strength levels, depending on how close to the obstacle the vehicle is. Based on the field strength, a vehicle makes either smaller or larger corrections to avoid a collision. In [40], the authors generate an obstacle avoidance algorithm by adding three additional flight legs around the ground object. Still, these flight legs cannot represent real Dubins paths.

Obstacle avoidance in an open territory is an even harder task. In [41], convex polygon flight paths, which should be used to avoid ground obstacles of various shapes, are generated. Alternatively, as in [42], a certain safety buffer zone around the ground obstacles is applied and Dubins paths are used to look for an optimal flight path.

A great part of the related papers tackle ground obstacle avoidance problems with quadrocopters. In most cases, these problems are related to either obstacle detection or the attempt to find an optimal

flight path. However, in most of these papers, there is a notable gap, as the studies do not evaluate the safety area around the obstacle. Taking into consideration the regulations on UAV flights, the accuracy of the flight path to avoid a ground obstacle should be evaluated as well. Therefore, in this study, Dubins paths are applied for safe UAV ground obstacle avoidance while complying with UAV regulations. We propose to generate additional waypoints, assuming that the ground obstacle coordinates are known (could be found by using camera sensing or other discussed methods) and are given to a UAV. One of the methods to test UAV navigation performance is to use some simulation technique like "MAV3DSim" [43]. In our case the research is done by using Software-in-the-Loop (SITL) simulation and a low cost GNSS sensor, which was attached to a flying-wing UAV. A new UAV navigation and guidance algorithm is derived by using GNSS sensor coordinates and simulated ground obstacle coordinates. In the next section, the standard methodology of Dubins paths for UAV flights is applied to show the mathematics of our obstacle avoidance algorithm.

3. Research Methods for Ground Obstacle Avoidance

In this section, the derivation of a mathematical model for an autonomous UAV flight which can be used to safely avoid ground obstacles is introduced.

It is assumed that a triangular shaped trajectory, which starts at waypoint $\mathbf{w}_i$, is to be flown by using Dubins paths near the turns. Also, all ground obstacles need to be automatically avoided by a safe distance. This scenario is represented in Figure 2.

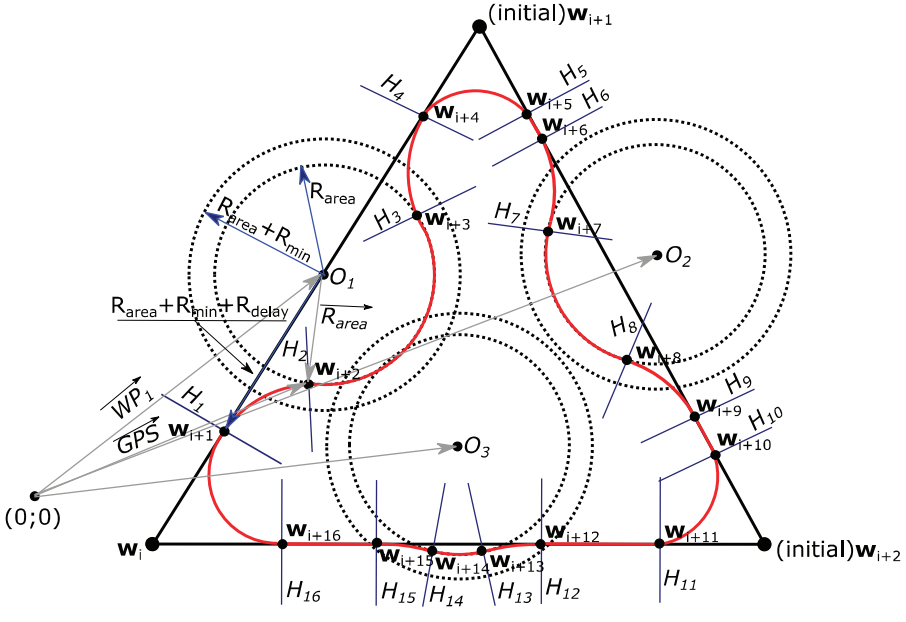


Figure 2. Flight mission with three ground obstacles that should be automatically avoided.

In Figure 2, the coordinates of the ground obstacles are represented by O_1, O_2 and O_3. In order to safely fly around the ground obstacles, at least a radius R_{area} must be flown, which is the minimum safe distance defined by national or international legal organizations. However, during the turn, the UAV will need some additional distance to get to the boundary of the minimum safe distance from the

obstacle. In this case, the distance from the obstacle must be increased to the distance R_{area} + R_{min}, where R_{min} can be found from Equation (8):

$$\mathbf{R}_{\min} = \frac{V^2}{g \times tg\theta_{max}} \tag{8}$$

The R_{min} depends on true air speed, V, and the maximum roll angle, θ_{max}. Finally, the UAV will experience a certain time lag before starting the turn, so the final distance from the obstacle where the turn action must be initiated is $\mathbf{R}_{area}$ + $\mathbf{R}_{min}$ + $\mathbf{R}_{delay}$. The approximate delay of the UAV decision is found by Equation (9), but can be adjusted for tuning purposes:

$$\mathbf{R}_{delay} = 2.5 \cdot V \tag{9}$$

To fly around the ground obstacle safely, four additional waypoints are generated, and their exact coordinates are needed. At these points, the planes from H_1 to H_{16} must be located to indicate a UAVs transition from one flight segment to another flight segment. As the current position of the UAV is measured by using a GNSS sensor, with the ground obstacle location at a constant value, the coordinate at which one of the H planes must be located is found by using Equation (10):

$$\mathbf{GPS} = \mathbf{O}_i - \mathbf{R} \tag{10}$$

In Equation (10), vector $\mathbf{R}$ is expressed by using Equations (11) and (12).

$$\mathbf{R} = \mathbf{R}_{area} + \mathbf{R}_{min} + \mathbf{R}_{delay} \tag{11}$$

$$\mathbf{R}_{area} = \mathbf{R}_1 + \mathbf{R}_{err} \tag{12}$$

where $\mathbf{R}_{err}$ is the radius calculation error, as in Equations (13) and (14). In this case, the $\mathbf{R}_{err}$ value inside of the length $\mathbf{R}_1$ is found initially, and this geometrical calculation uncertainty is compensated later by adding $\mathbf{R}_{err}$ to $\mathbf{R}_1$, as shown in Equation (12):

$$\mathbf{R}_{err} = \mathbf{R}_1 - \sqrt{\mathbf{R}_1^2 - \left(\frac{k}{2}\right)^2} \tag{13}$$

$$\mathbf{R}_{err} = \mathbf{R}_1 - \sqrt{\mathbf{R}_1^2 - \left(\frac{\frac{V^2}{\cos\left(tg\left(\frac{|GPS_x - O_x|}{|GPS_y - O_y|}\right)\right) g \cdot tg\theta_{max}}}{2}\right)^2} \tag{14}$$

A more detailed analysis of the application of Dubins paths for one general straight path flight segment and one ground obstacle is illustrated in Figure 3.

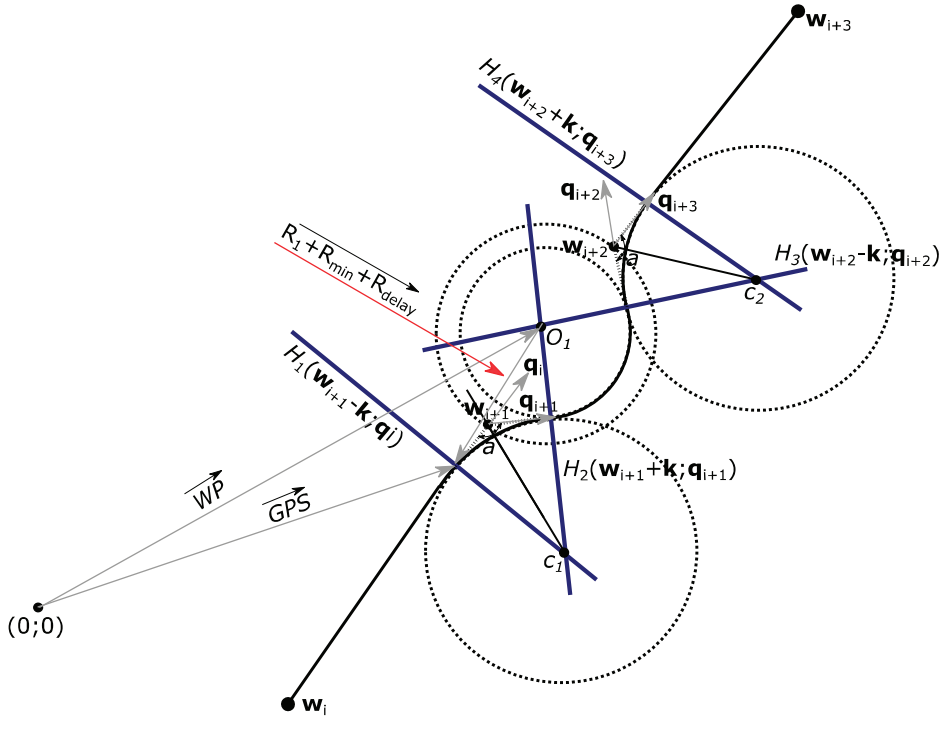


Figure 3. Additional waypoint and planes for one flight mission segment to safely fly around ground obstacle O_1.

As shown in Figure 3, plane H_1 is not exactly on the position of waypoint $\mathbf{w}_{i+1}$ which is found from Equation (3), but is in the same direction where the unit vector $\mathbf{q}_i$ points to by distance $\mathbf{k}$ from this plane. According to Dubins paths theory, Distance $\mathbf{k}$ can be found by using Equation (15):

$$\mathbf{k} = \left(\frac{r}{\tan \frac{\angle a}{2}} \right) \mathbf{q}_i \tag{15}$$

In Equation (6), however, r—the radius of the Dubins circle with the center C_1 and $\angle a$, the size of which depends on how much the UAV changes its course—is still unknown. For this reason, flight, trajectory geometrical calculations, which are visualized in Figure 4, are applied.

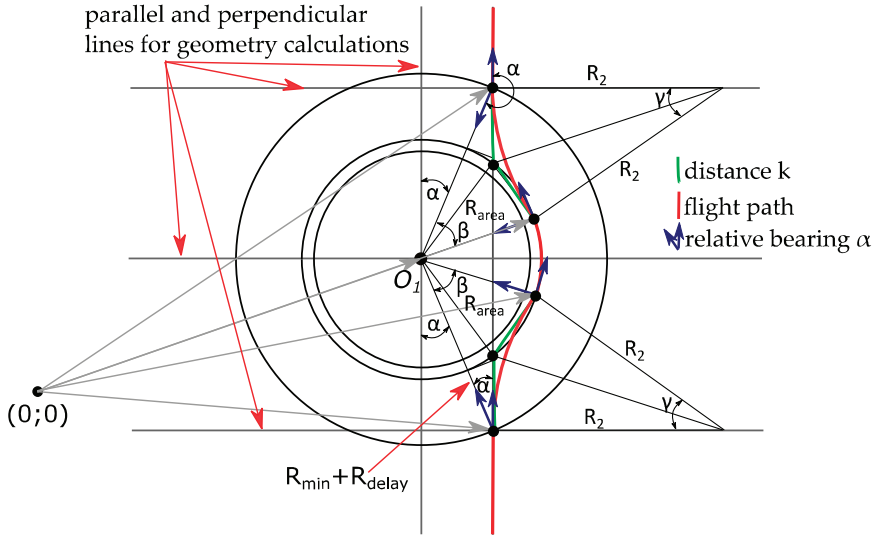


Figure 4. Geometry of ground obstacle avoidance.

To begin with, an equation is used to find the unit vector $\mathbf{q}_i$, which shows the flight path direction from the initial waypoint to the next waypoint, without paying attention to ground obstacles. According to Dubins trajectory theory, vectors $\mathbf{q}_i$ can be found from Equation (16):

$$\mathbf{q}_i = \frac{\mathbf{w}_{i+3} - \mathbf{w}_i}{\|\mathbf{w}_{i+3} - \mathbf{w}_i\|} \tag{16}$$

As shown in Figure 4, distance k can be found by using Equation (17), and the UAV relative bearing angle can be calculated by using the live GNSS on-board sensor data for Equation (18).

$$k \approx \frac{\mathbf{R}_{\min} + \mathbf{R}_{\text{delay}}}{\cos(\alpha)} \tag{17}$$

$$\angle\alpha = \text{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|}\right) \tag{18}$$

By inserting Equations (8) and (18) into Equation (17), the calculation of distance k based on the known parameters is obtained, as in Equation (19):

$$k \approx \frac{1}{\cos\left(\text{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|}\right)\right)} \times \left(\frac{\text{V}^2}{\text{g} \times \text{tg}\theta_{\max}} + \text{R}_{\text{delay}}\right) \tag{19}$$

Distance k should only be calculated when the coordinate values received from the GNSS sensor meet the requirements for value **R**, found from Equations (10)–(12). This condition can be expressed by Equation (20):

$$\mathbf{WP} - \mathbf{GNSS} \leq \mathbf{R}_1 + \mathbf{R}_{\text{err}} + \mathbf{R}_{\min} + \mathbf{R}_{\text{delay}} \tag{20}$$

When the condition in Equation (20) is met, distance k is calculated from Equation (19) and is converted into a vector form by using either Equation (21) or Equation (22):

$$\mathbf{k} = k \cdot \mathbf{q}_i \tag{21}$$

$$\mathbf{k} = k \cdot \left(\frac{\mathbf{w}_{i+3} - \mathbf{w}_i}{\|\mathbf{w}_{i+3} - \mathbf{w}_i\|} \right) \tag{22}$$

By inserting k into Equation (22), a representation of vector $\mathbf{k}$ with the known values is obtained, as shown in Equation (23):

$$\mathbf{k} \approx \left(\frac{1}{\cos\left(\mathrm{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right) \right)} \times \left(\frac{\mathrm{V}^2}{\mathrm{g} \times \mathrm{tg}\theta_{\max}} + 2.5 \times \mathrm{V} \right) \right) \left(\frac{\mathbf{w}_{i+3} - \mathbf{w}_i}{\|\mathbf{w}_{i+3} - \mathbf{w}_i\|} \right) \tag{23}$$

Then, the position of plane H_1 is represented by Equation (24).

$$H_1(\mathbf{w}_{i+1} - \mathbf{k}, \mathbf{q}_i) \tag{24}$$

To find the coordinates of plane H_2, the unknown angles β, γ and the Dubins circle radius R_2 are necessary. From the flight path geometry expressed in Figure 4, we derive the magnitude of angle β. Its value can be calculated either with Equation (25) or, if all known values are inserted—with Equation (26):

$$\angle\beta = \frac{l \cdot 180}{\pi \mathrm{R}_1} + l \sin\alpha \tag{25}$$

$$\angle\beta = \frac{\frac{\mathrm{V}^2}{\mathrm{g} \times \mathrm{tg}\theta_{\max}} (180 + \pi \mathrm{R}_1) \mathrm{tg}^2 \left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right)}{\pi \mathrm{R}_1} \tag{26}$$

As is known form flight geometry, the waypoint relative bearing from an airplane's position during the turn must change from the initial value, angle α, to a new value of 90°, and angle γ can be calculated by either using Equation (27) or Equation (28) if all known values are used:

$$\angle\gamma = 90^\circ - \alpha - \beta \tag{27}$$

$$\angle\gamma = 90 - \mathrm{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right) - \frac{\frac{\mathrm{V}^2}{\mathrm{g} \times \mathrm{tg}\theta_{\max}} (180 + \pi \mathrm{R}_1) \mathrm{tg}^2 \left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right)}{\pi \mathrm{R}_1} \tag{28}$$

Then, R_2 is obtained from Equation (29) or Equation (30):

$$\mathrm{R}_2 = k \cdot \mathrm{tg}\frac{\gamma}{2} \tag{29}$$

$$\begin{aligned} \mathrm{R}_2 = & \left(\frac{1}{\cos\left(\mathrm{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right) \right)} \times \left(\frac{\mathrm{V}^2}{\mathrm{g} \times \mathrm{tg}\theta_{\max}} + \mathbf{R}_{\mathrm{delay}} \right) \right) \cdot \\ & \mathrm{tg}\left(45 - \tfrac{1}{2} \cdot \mathrm{tg}\left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right) - \frac{\frac{\mathrm{V}^2}{\mathrm{g} \times \mathrm{tg}\theta_{\max}} (180 + \pi \mathrm{R}_1) \mathrm{tg}^2 \left(\frac{|GNSS_x - O_x|}{|GNSS_y - O_y|} \right)}{2\pi \mathrm{R}_1} \right) \end{aligned} \tag{30}$$

When the radius R_2 is found, we look for angle a using Equation (31). Then, this value must be used for a UAV to find the Dubins circle center coordinate and the unit vector $\mathbf{q}_{i+1}$:

$$\angle a = \mathrm{arctg}\frac{\mathrm{R}_2}{k} \tag{31}$$

The unit vector $\mathbf{q}_{i+1}$ is found as the vector's $\mathbf{q}_i$ rotation as in Equation (32):

$$\mathbf{q}_i = \begin{bmatrix} \cos(180 - a) & -\sin(180 - a) \\ \sin(180 - a) & \cos(180 - a) \end{bmatrix} \mathbf{q}_{i+1} \tag{32}$$

Having vectors $\mathbf{q}_i$, $\mathbf{q}_{i+1}$ and distance k, we find the Dubins circle center coordinates from Equation (33), and the position of the next plane, H_2—from Equation (34):

$$\mathbf{c}_1 = \mathbf{w}_{i+1} - \left(\frac{R_2}{\sin \frac{\angle a}{2}} \right) \frac{\mathbf{q}_i - \mathbf{q}_{i+1}}{\|\mathbf{q}_i - \mathbf{q}_{i+1}\|} \tag{33}$$

$$H_2(\mathbf{w}_{i+1} + \mathbf{k}, \mathbf{q}_{i+1}) \tag{34}$$

Plane H_2, which is expressed by Equation (34), provides the second point where the UAV makes the transition from flying along the circular arc of one Dubins path to another circular arc, which is used to fly around the obstacle. The two remaining plane positions, H_3 and H_4, and the additional waypoint $\mathbf{w}_{i+2}$ would be symmetrical to the previous H_1, H_2 and $\mathbf{w}_{i+1}$. In this case, the procedure for finding these values is identical and would be repeated for other obstacles.

As the derivation of obstacle avoidance model is an important part, UAV also needs to know the rules how and when to perform these calculations. For this reason, in next section we introduce exact algorithm to avoid ground obstacles.

4. Ground Obstacle Avoidance Algorithm

In order to avoid ground obstacles, UAVs must make correct decisions. It is not enough to find only the required variables and unknowns. For this case, special sectors around the obstacle are defined, as represented in Figure 5.

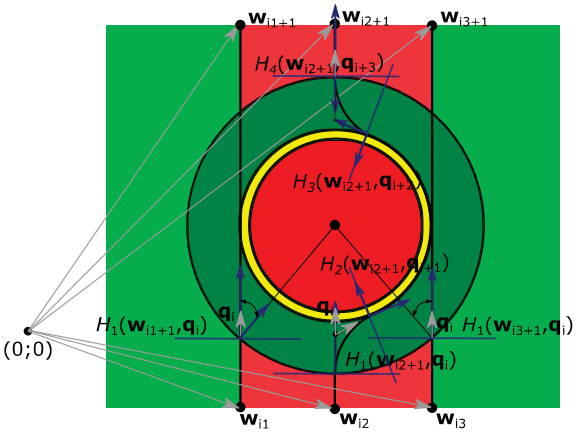


Figure 5. Decision areas for UAV ground obstacle avoidance algorithm: dark red—prohibited to fly area; yellow—additional safety area due to the calculations errors; dark green—area for the UAV to make an initial turn for the ground obstacle avoidance; light red—UAV approaching from there will need to make an obstacle avoidance; light green—UAV approaching from there will not need to make an obstacle avoidance.

As known from previous equations, the red, yellow, and dark green areas represent special areas around the obstacle: ground obstacle safety area, calculations error area and UAV turn area, respectively. There are also two additional areas: light green and light red. If a UAV is approaching an obstacle from the light red area, it will need to make an action to fly around the obstacle. However, if a UAV is arriving from the light green area, no additional action will be needed, and it will continue straight forward.

As shown in Figure 5, if a UAV is flying from waypoint $\mathbf{w}_{i2}$ to waypoint $\mathbf{w}_{i2+1}$, it arrives first at a coordinate of plane $H_1(\mathbf{w}_{i2+1},\mathbf{q}_i)$, which is located within distance $\mathbf{R}$ from ground obstacle O_1,

as in Equation (11). An on-board GNSS sensor determines the UAV's position, but the obstacle's coordinates should be known and kept in the UAV's memory. When the UAV is at a position located in plane $H_1(\mathbf{w}_{i2+1},\mathbf{q}_i)$ (the distance from the ground obstacle is **R**), it has not determined if it is safe to continue in this direction without any change of course. Therefore, the following action for the UAV is to evaluate the relative bearing from the ground obstacle, which is calculated by Equation (35):

$$\angle \mathrm{RB} = 180 - \cos^{-1}(-\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_{O_i}) \tag{35}$$

Then, it must be checked if the $|RB|$ is below the maximum $\mathrm{RB}_{\max}$ for ground obstacle avoidance. $\mathrm{RB}_{\max}$ can be found using Equation (36):

$$\angle \mathrm{RB}_{\max} = \sin\left(\frac{\mathbf{R}_{\mathrm{area}}}{\mathbf{R}}\right) \tag{36}$$

If $|\mathrm{RB}| < \mathrm{RB}_{\max}$, the UAV must generate several Dubins path circles around the ground obstacle. These Dubins paths would be located between planes $H_1(\mathbf{w}_{i2+1},\mathbf{q}_i)$ and $H_2(\mathbf{w}_{i2+1},\mathbf{q}_{i+1})$, $H_1(\mathbf{w}_{i2+1},\mathbf{q}_{i+1})$ and $H_2(\mathbf{w}_{i2+1},\mathbf{q}_{i+2})$, and $H_1(\mathbf{w}_{i2+1},\mathbf{q}_{i+2})$ and $H_2(\mathbf{w}_{i2+1},\mathbf{q}_{i+3})$. The described flight procedure is illustrated in Figure 6 as a UAV actions algorithm.

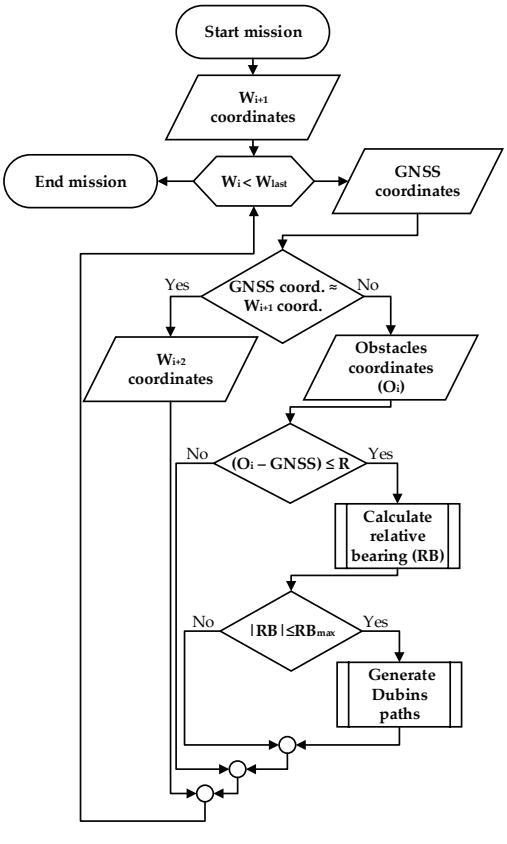


Figure 6. Algorithm for UAV ground obstacle avoidance.

This procedure is repeated until the UAV completes the entire flight mission. If there were no ground obstacles on the path, the UAV would omit this ground obstacle avoidance procedure and continue flying a regular flight mission. In the next section, the hardware and software used to test the new algorithm are presented.

5. Experimental Approach

Two methods have been chosen to test the algorithm: Software-in-the-Loop simulation (SITL) and real flight experiments. The Software-in-the-Loop simulation was done by using the ArduPilot autopilot software, whereas real flights were done with a more advanced autopilot—the Pixhawk 2 (PX4). Both tests were carried out by taking coordinate measurements with either a real or a simulated GNSS receiver, the "Ublox Neo-M8N". The main technical characteristics of used GNSS receiver are presented in Table 1.

Table 1. Main technical characteristics for GNSS receiver "Ublox Neo-M8N".

	Ublox Neo-M8N
Satellite systems	GPS/QZSS; GLONASS; Galileo; BeiDou
Supply	2.7–3.6 V; Lowest power (DC/DC)
Interfaces	UART; USB; SPI; I^2C
Features	Flash; Data logging; Additional SAW and LNA; RTC crystal; Active antenna

As we can see from the Table 2, it is stated that "Ublox Neo-M8N" has horizontal position accuracy of about 2.0–2.5 m. However, this accuracy is only valid if we make data logging in a static position for about 24 h. During UAV flight GNSS receiver accuracy would be much worse as we need to make real time measurements in a dynamic mode. In order to check accuracy of the new algorithm and GNSS receiver influence on that algorithm, we start with SITL simulation.

Table 2. "Ublox Neo-M8N" accuracy and limits for various operational modes.

Accuracy or Type of Limit	**Mode**	**GPS/GLONASS**	**GPS/BeiDou**	**GPS**
Time to first fix	Cold start	26 s	27 s	29 s
	Hot start	1 s	1 s	1 s
	Aided starts	2 s	3 s	2 s
Sensitivity	Navigation	−164 dBm	−162 dBm	−163 dBm
	Reacquisition	−159 dBm	−159 dBm	−159 dBm
	Cold start	−147 dBm	−147 dBm	−147 dBm
	Hot start	−156 dBm	−156 dBm	−156 dBm
Max nav update rate		5 Hz	5 Hz	10 Hz
Velocity accuracy		0.05 m/s	0.05 m/s	0.05 m/s
Heading accuracy		0.3°	0.3°	0.3°
Horizontal accuracy	Autonomous	2.5 m	2.5 m	2.5 m
	SBAS	2.0 m	2.0 m	2.0 m
Operational limits	Dynamics	≤4 g	≤4 g	≤4 g
	Altitude	50,000 m	50,000 m	50,000 m
	Velocity	500 m/s	500 m/s	500 m/s

The SITL configuration is shown in Figure 7.

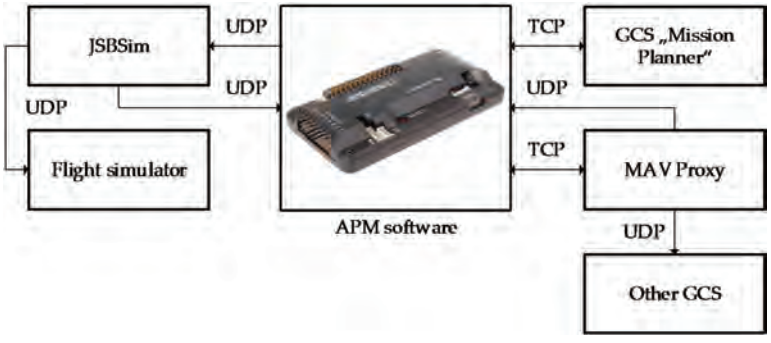


Figure 7. Software-in-the-Loop simulation setup.

For simulation purposes, an open source ArduPilot "Mega" (APM) autopilot software was used. The flights were set up by using two ground stations: the Ground Control Station (GCS), referred to as "Mission planner 1.3.44", and the MAV Proxy on the APM console. The Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) were used as communication protocols. This simulation setup allowed using additional flight simulators to expand visualization, but in our case that was not necessary. In this case, the GNSS data were simulated by using the software and data communication protocols described earlier.

Further real flight experiments were performed with a flying-wing UAV which was designed with "Solidworks". It has a mass of 1.8 kg and a wingspan of 1.6 m. Its maximum flight time is 1 h and 35 min with a maximum flight distance of approximately 30 km. The UAV was equipped with the Pixhawk 2 autopilot, which was used to simulate ground obstacles and to test the new algorithm. UAV model is shown in Figure 8a and its "Soliworks" model is represented in Figure 8b.

(**a**) (**b**)

Figure 8. Research UAV: (**a**) Manufactured UAV; (**b**) UAV prototype in "Solidworks".

The complete setup of the UAV with all its hardware inside the fuselage is shown in Figure 9. The main flight data were collected by using a "Pixhawk 2" autopilot and were recorded into the internal memory for later analysis. Live data were checked by using 433 MHz telemetry units and was sent to ground station "QGroundControl v3.1.3". The most important technical specification of the "Pixhawk 2" autopilot are presented in Table 3.

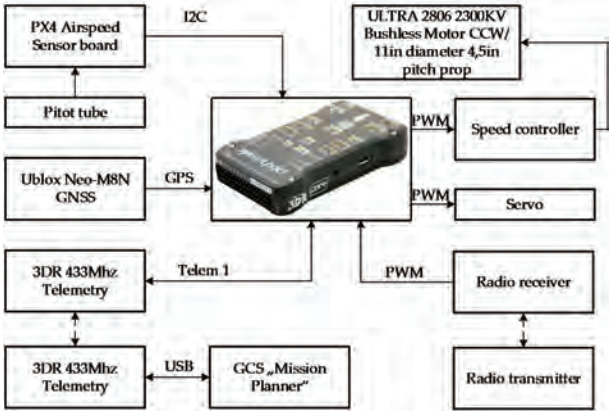


Figure 9. Setup of the aircraft equipment for the experiment.

Table 3. "Pixhawk 2" autopilot technical specification.

Pixhawk 2	
Processor	32bit STM32F427 Cortex M4 core with FPU; 168 MHz; 256 KB RAM; 2 MB Flash; 32 bit STM32F103 failsafe co-processor;
Sensors	ST Micro L3GD20H 16 bit gyroscope; ST Micro LSM303D 14 bit accelerometer/magnetometer; Invensense MPU 6000 3-axis accelerometer/gyroscope; MEAS MS5611 barometer

The Software-in-the-Loop and real flight experiments were set up for a UAV to fly an equilateral triangle with a side length of 500 m (the same mission as in Figure 2). Three imaginable ground obstacles were added to this trajectory. With this setup, after all flights, experiment data were recalculated into the East-North-Up (ENU) coordinate frame by an inversion matrix, as in Equation (37):

$$\begin{bmatrix} E \\ N \\ U \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} \Delta LON \\ \Delta LAT \\ \Delta ALT \end{bmatrix} \quad (37)$$

In Equation (37), λ_r indicates the reference longitude and ϕ_r—the reference latitude. These values are used as the main coordinates at which the conversion to ENU is done. ΔLON, ΔLAT, and ΔALT are the difference between the UAV's position and a reference coordinate. The ENU coordinates of the flight mission and ground obstacles are shown in Table 4. Additional waypoints are calculated by the UAV during the mission.

Table 4. ENU coordinates of the mission points and ground obstacles.

	WP_1	**WP_2**	**WP_3**	**O_1**	**O_2**	**O_3**
E (m)	500	750	1000	625	900.51	750
N (m)	500	933.01	500	716.51	731.24	544.32
U (m)	50	50	50	50	50	50

For experimental purposes, a safe radius of 50 m around the obstacle was chosen and remained constant during the experiments. In the next section, we present the results of the simulation and real flights as well as the comparison of the data.

6. Results

Firstly, the Software-in-the-Loop experiment was carried out. The autopilot was setup to fly at 15 m/s. For this experiment, four attempts with three laps were done for the flight mission illustrated in Figure 2. It was realized that the "ArduPilot" has some delay in decision-making—it tends to undershoot or overshoot the trajectory during the turn. The trajectories with undershoots and overshoots are visible in Figure 10a. In this figure, the circles represent the 50 m radius area around the ground obstacle positions (the minimum safety area around an obstacle, as stated in various UAV flight regulations). Each attempt was done with a different UAV delay configuration. As the results illustrated in Figure 10 show, the UAV was not able to fly around these obstacles safely. Figure 10b illustrates the statistics of the trajectory deviation from the triangular flight path with ground obstacles. In Figure 10c, the same statistics is provided, but the trajectory deviation is only measured from the 50 m ground obstacle boundary area. To avoid negative values, which represent that a UAV is inside of the 50 m ground obstacle boundary area, some corrections were made by tuning the UAV delay. As seen in Figure 10c, during attempt 4, the lowest quartile was closer to 0. It means that for this attempt 25% of the flight data (coordinates) had a slightly greater accuracy. Although, as seen in Figure 10d, the probability density is the highest for a trajectory deviation of 15–20 m (from the 50 m obstacle boundary) and is approximately $6{,}7 \cdot 10^{-3}$–$7{,}2 \cdot 10^{-3}$. Furthermore, the probability density of being exactly where we need to be (0 m deviation) is approximately $5{,}3 \cdot 10^{-3}$–$6{,}0 \cdot 10^{-3}$. If only the flight trajectory is considered, the flight path deviation is smaller and has the greatest probability density that a UAV will be on the path of 0.15. Other important statistical data for this experiment is shown in Table 5.

Table 5. Results of the flight path deviation for the SITL experiment with a 15 m/s UAV speed.

	Min (m)	Max (m)	μ (m)	σ (m)
Attempt 1	−13.29 (0.01)	197.34 (67.39)	79.20 (11.01)	62.27 (15.00)
Attempt 2	−11.03 (0.02)	197.99 (66.63)	77.07 (10.69)	60.78 (14.68)
Attempt 3	−6.99 (0.00)	198.93 (62.41)	74.61 (9.31)	59.45 (13.68)
Attempt 4	−5.93 (0.01)	197.77 (58.89)	72.72 (8.29)	58.50 (12.94)

In Table 5, the minimum and maximum values of the flight path deviation are provided together with a mean μ and standard deviation σ for flight accuracy. The first values in the Table 5 represent the statistics obtained when using our new algorithm (distance is measured from 50 m. boundary area circles) and the values in brackets represent the statistics obtained when the UAV's position is measured from the standard flight mission path. It was realized that the UAV delay would be smaller if the flight speed was reduced. Therefore, the same Software-in-the-Loop experiment was repeated for the second time, with a 10 m/s flight speed of the UAV. The results for this experiment are shown in Figure 11. It is evident from Figure 11a that the UAV flight was more accurate and the GNSS position values were closer to the theoretical calculations of the path coordinates. In Figure 11b,c, the first and even second quartiles were closer to 0 m, which is a good sign for the required flight accuracy. The probability density functions remained approximately the same as for the previous experiment. More Software-in-the-Loop simulation results for a flight speed at 10 m/s are shown in Table 6.

Table 6. Results of the flight path deviation for the SITL experiment with a 10 m/s UAV speed.

	Min (m)	Max (m)	μ (m)	σ (m)
Attempt 1	−8.81 (0.00)	197.61 (63.66)	73.37 (8.22)	59.04 (13.17)
Attempt 2	−6.06 (0.00)	199.69 (60.70)	71.99 (7.25)	58.03 (12.77)
Attempt 3	−5.02 (0.00)	197.94 (58.51)	70.71 (6.81)	57.39 (12.51)
Attempt 4	−4.49 (0.00)	199.26 (56.31)	70.52 (6.69)	57.27 (12.45)

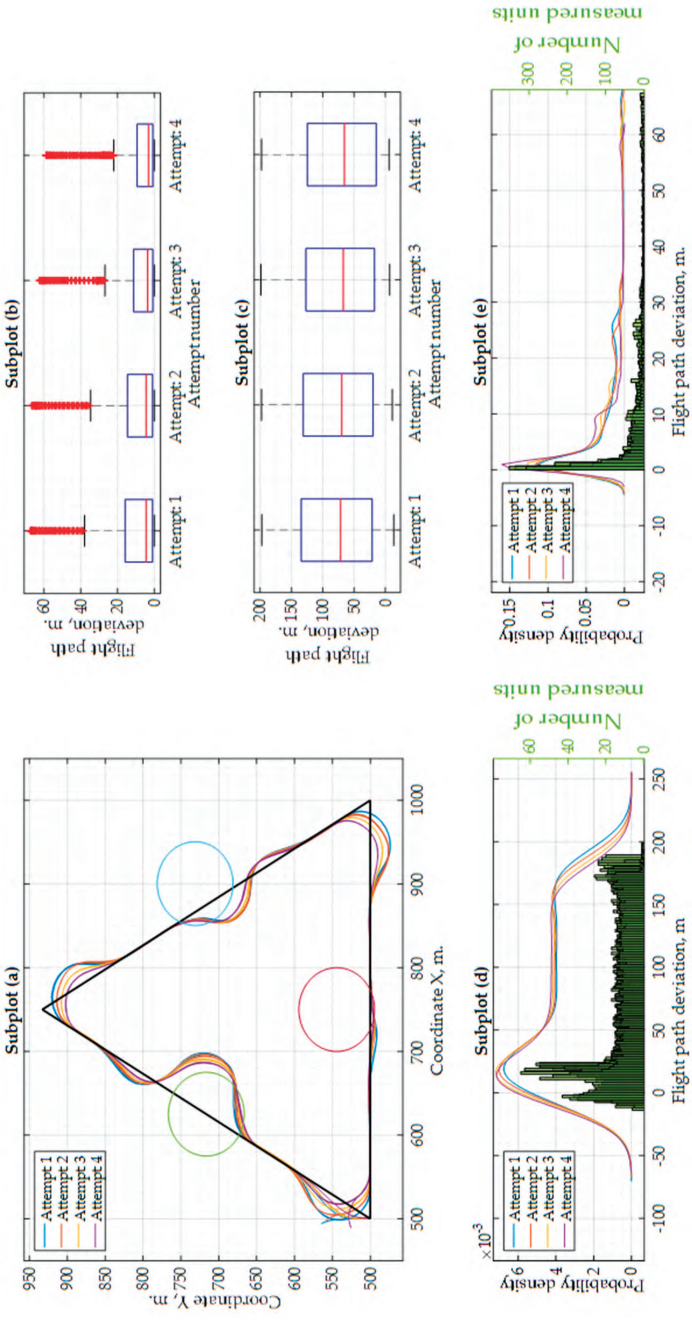


Figure 10. Results of the Software-in-the-Loop simulation with the "ArduPilot" autopilot with the flight speed at 15 m/s: (**a**) flight mission, flight trajectory and ground obstacles; (**b**) flight deviation from the standard flight mission path (when using ground obstacle avoidance); (**c**) flight deviation from the ground obstacle area boundary of 50 m. (when using ground obstacle avoidance); (**d**) probability density values and quantity of measurements for every flight path deviation value (standard mission); (**e**) probability density values and quantity of measurements for every flight path deviation value (from ground obstacle area boundary).

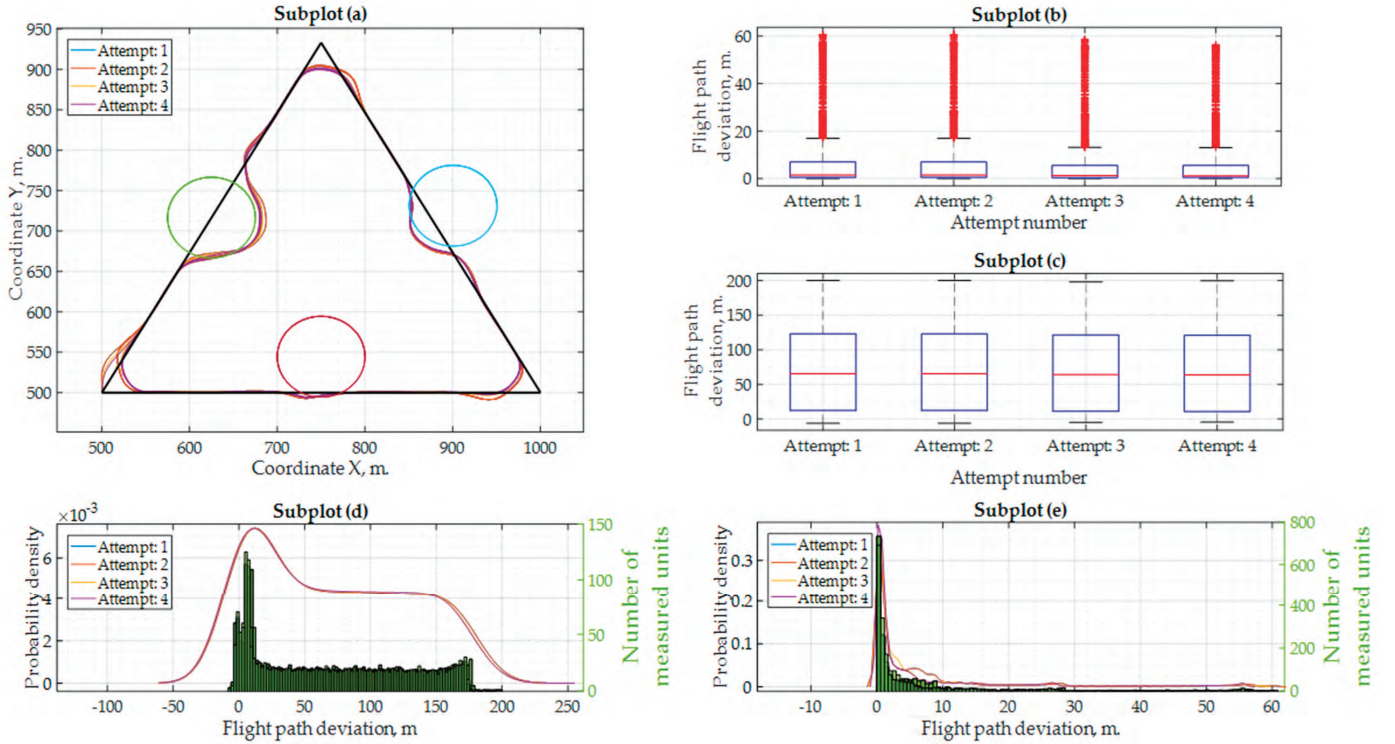


Figure 11. Results of the Software-in-the-Loop simulation with the "ArduPilot" autopilot with the flight speed at 10 m/s: (**a**) flight mission, flight trajectory and ground obstacles; (**b**) flight deviation from the standard flight mission path (when using ground obstacle avoidance); (**c**) flight deviation from the ground obstacle area boundary of 50 m. (when using ground obstacle avoidance); (**d**) probability density values and quantity of measurements for every flight path deviation value (standard mission); (**e**) probability density values and quantity of measurements for every flight path deviation value (from ground obstacle area boundary).

Flight experiments were carried out with a "PX4" autopilot. This autopilot was chosen because of a very short delay time for decision-making. This flight experiment was done with a UAV flying at 15 m/s and experiencing a 2–3 m/s southwest wind. The results are shown in Figure 12 and more data is presented in Table 7.

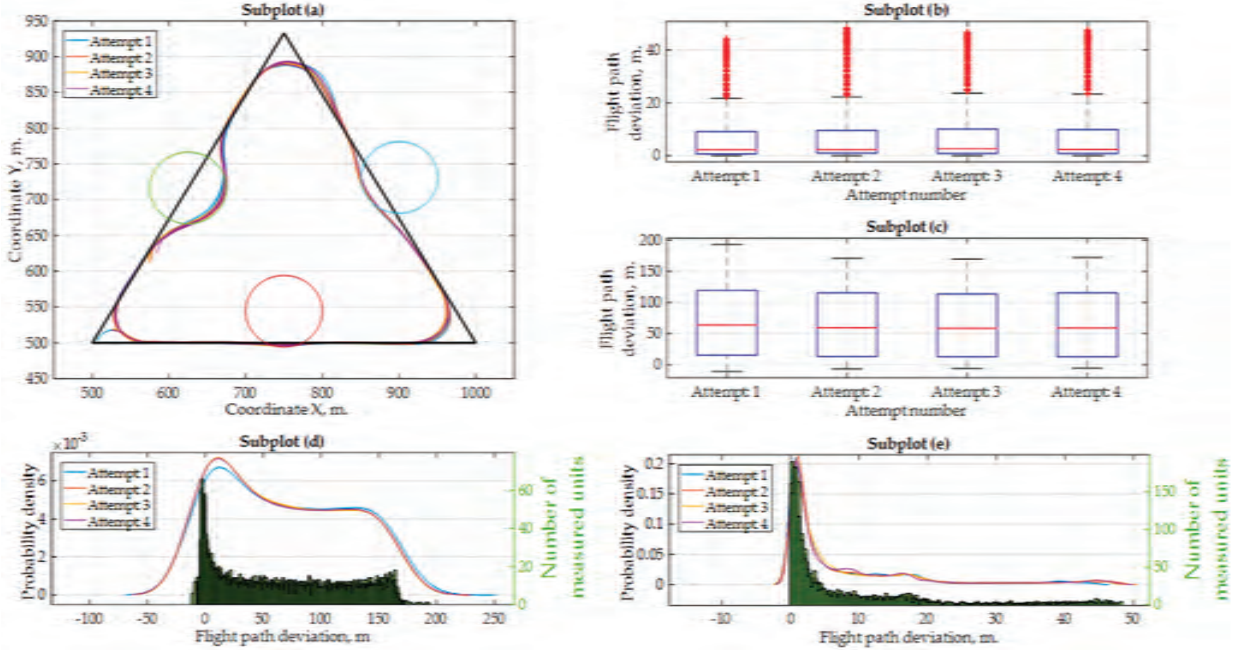


Figure 12. Results of real flight experiments with the "PX4" autopilot with the flight speed at 15 m/s and using the "Ublox Neo-M8N" GNSS receiver: (**a**) flight mission, flight trajectory and ground obstacles; (**b**) flight deviation from the standard flight mission path (when using ground obstacle avoidance); (**c**) flight deviation from the ground obstacle area boundary of 50 m. (when using ground obstacle avoidance); (**d**) probability density values and quantity of measurements for every flight path deviation value (standard mission); (**e**) probability density values and quantity of measurements for every flight path deviation value (from ground obstacle area boundary).

Table 7. Results of the flight path deviation for real flight experiments with a UAV speed of 15 m/s and a 2–3 m/s southwest wind.

	Min (m)	Max (m)	μ (m)	σ (m)
Attempt 1	−10.93 (0.00)	193.27 (44.22)	69.54 (6.67)	55.98 (9.49)
Attempt 2	−6.79 (0.00)	171.22 (48.32)	66.59 (7.43)	54.95 (10.83)
Attempt 3	−5.22 (0.01)	170.00 (46.64)	65.81 (7.59)	55.59 (10.84)
Attempt 4	−3.14 (0.01)	172.56 (47.52)	66.29 (7.54)	55.03 (10.86)

In this scenario, as seen in Figure 12a, the UAV flies around the imaginable ground obstacles much more accurately. The UAV makes instant decisions, and the flight trajectory comes much closer to the theoretically calculated one. For this flight experiment, all four attempts were done with the same flight configuration, since no changes can be made for UAV delay tuning with this autopilot. Some overshoot is still seen near the coordinate WP_2. It is obvious that it was caused by the southwest wind.

Although the accuracy of the GNSS receiver which was used for the coordinate measurements is unknown, an assessment of the received coordinate accuracy was carried out. The GNSS "Ublox Neo-M8N" was put stationary on the geodetic coordinate 54.67192105° N 25.506942669° E. The static reading of this coordinate was measured, and, for comparison, the coordinates were transformed from the geodetic into the ENU frame. The GNSS data results are illustrated in Figure 13. As seen in Figure 13b,c, the common static data drifted from the correct position coordinate up to 4 m. Also, as seen from Figure 13d, the latitude uncertainty varied up to 4 m, whereas the longitude uncertainty was up to 2 m. These values could be subtracted or added to the data provided in Tables 2–4. For example, if we take the GNSS error into consideration and subtract this value from the data in Table 4, where real flight experiment data is provided, it is possible that our new algorithm could be applied for the safe avoidance of ground obstacles.

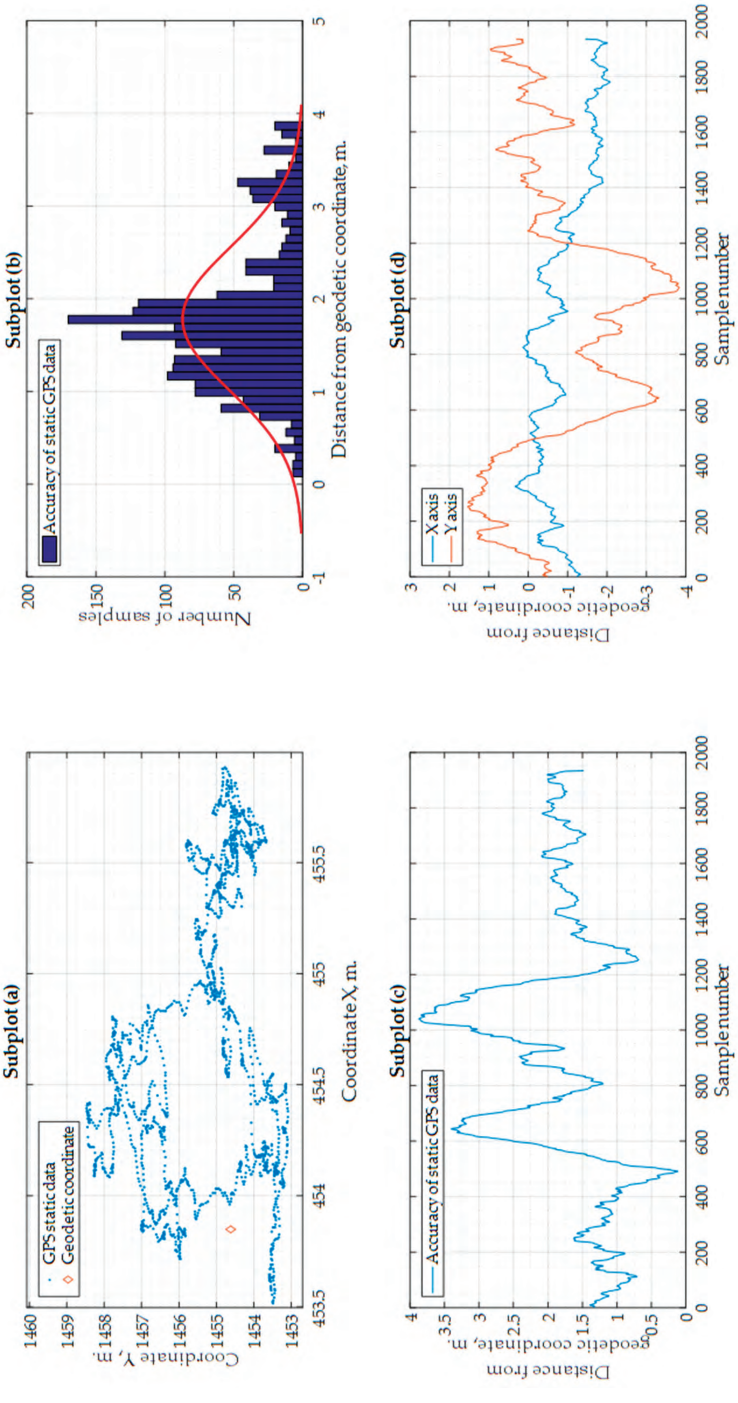


Figure 13. Accuracy of the "Ublox Neo-M8N" GNSS receiver when the data is measured by using a precise geodetic coordinate: (**a**) GNSS receiver measurement drift from the accurately measured geodetic coordinate; (**b**) Gaussian distribution of GNSS samples; (**c**) GNSS receiver measurement error; (**d**) GNSS receiver measurement error in X and Y directions.

7. Conclusions

The results of the experiments showed that the "ArduPilot" autopilot is unable to fly around the obstacles as accurately as the theoretical calculations predict. This problem arises due to this autopilot's decision lag. Due to this lag, the UAV makes some overshoots and undershoots near the turn or in cases when a flight manoeuver has to be made around the obstacle. Since our theoretical calculations only evaluate the initial turn lag when an obstacle is detected, the calculations could be improved by recognizing that this decision delay will appear for all additional waypoints that are required to fly around the obstacle safely. Also, in order to have smaller overshoots or undershoots, it might be possible to find an optimal "ArduPilot" autopilot flight configuration for different turns.

Due to this decision delay of the "ArduPilot" autopilot, for real flight experiments the "PX4" autopilot was chosen due to a greater processing power and a lower decision lag. The flight tests showed that the coordinate results for the flight path were much closer to the theoretically calculated trajectory. The results proved that the decision delay is an important factor for ground obstacle avoidance. Although the 50 m boundary area around the ground obstacle was still crossed and the maximum distance by which the safety rule was violated increased from an average of -6.09 m (Software-in-the-Loop) to -6.52 m ("PX4"). However, the average flight path error, maximum flight path error and standard deviation of the flight path error have decreased.

Real flight experiment results were influenced by the 2–3 m/s variable southwest wind, due to which some overshoots are seen near the second mission waypoint. Also, as a low cost GNSS is used to collect the coordinate data, a flight path error of up to 4 m may be caused by the GNSS data inaccuracy. Nevertheless, in general, the new algorithm works as expected if we would consider that there was a negative influence of the GNSS inaccuracy.

The new algorithm for ground obstacle avoidance can be implemented as an autopilot software for more advanced testing. In further research, more attention should be devoted to wind impact on safe ground obstacle avoidance, the influence of the autopilot's decision delay on flight accuracy and GNSS data adaptation to reduce the influence of the GNSS on flight safety when a UAV is flying around ground obstacles.

Acknowledgments: The research and paper publishing was financially supported by Vilnius Gediminas technical university.

Author Contributions: R.K. designed the ground obstacle avoidance model and algorithm, derived the equations, performed research and calculations, prepared figures and the manuscript. J.S. supervised the research and helped coordinating all of the tasks. D.R. was responsible for the design and manufacture of an unmanned aerial vehicle, prepared all of the hardware, SITL setup, helped with the modelling part. T.M. analyzed statistical data and prepared the results.

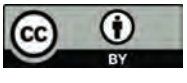
Conflicts of Interest: The authors declare no conflict of interest.

References

1. Claesson, A.; Fredman, D.; Svensson, L.; Ringh, M.; Hollenberg, J.; Nordberg, P.; Rosenqvist, M.; Djarv, T.; Österberg, S.; Lennartsson, J.; et al. Unmanned aerial vehicles (drones) in out-of-hospital-cardiac-arrest. *Scand. J. Trauma Resusc. Emerg. Med.* **2016**, *24*, 124. [CrossRef] [PubMed]
2. Al-Kaff, A.; Moreno, F.M.; Javier, L.; Jos, S.; De Escalera, A.; Nieva, A.; Garc, F.; Mart, D. VBII-UAV: Vision-Based Infrastructure Inspection-UAV. *Recent Adv. Inf. Syst. Technol.* **2017**, *571*. [CrossRef]
3. Nedjati, A.; Vizvari, B.; Izbirak, G. Post-earthquake response by small UAV helicopters. *Nat. Hazards* **2016**, *80*, 1669–1688. [CrossRef]
4. Nex, F.; Remondino, F. UAV for 3D mapping applications: A review. *Appl. Geomat.* **2014**, *6*, 1–15. [CrossRef]
5. Park, K.; Ewing, R. Landscape and Urban Planning The usability of unmanned aerial vehicles (UAVs) for measuring park-based physical activity. *Landsc. Urban Plan.* **2017**, *167*, 157–164. [CrossRef]
6. Harrington, A. Who controls the drones? *Eng. Technol.* **2015**, *10*, 80–83. [CrossRef]

7. Hlas, M.; Straub, J. Autonomous navigation and control of unmanned aerial systems in the national airspace. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–7. [CrossRef]
8. Abouzakhm, P.; Sharf, I. Guidance, Navigation, and Control for Docking of Two Cubic Blimps. *IFAC-PapersOnLine* **2016**, *49*, 260–265. [CrossRef]
9. Chen, H.; Chang, K.; Agate, C.S. UAV path planning with tangent-plus-lyapunov vector field guidance and obstacle avoidance. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 840–856. [CrossRef]
10. Maillot, T.; Boscain, U.; Gauthier, J.P.; Serres, U. Lyapunov and Minimum-Time Path Planning for Drones. *J. Dyn. Control. Syst.* **2014**, *21*, 47–80. [CrossRef]
11. Liu, X.F.; Guan, Z.W.; Song, Y.Q.; Chen, D.S. An optimization model of UAV route planning for road segment surveillance. *J. Cent. South Univ.* **2014**, *21*, 2501–2510. [CrossRef]
12. Li, J.; Deng, G.; Luo, C.; Lin, Q.; Yan, Q.; Ming, Z. A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9585–9596.
13. Gageik, N.; Benz, P.; Montenegro, S. Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors. *IEEE Access* **2015**, *3*, 599–609. [CrossRef]
14. Vanegas, F.; Gonzalez, F. Enabling UAV navigation with sensor and environmental uncertainty in cluttered and GPS-denied environments. *Sensors* **2016**, *16*. [CrossRef] [PubMed]
15. Wang, Y.; Cao, Y. Coordinated Target Tracking via a Hybrid Optimization Approach. *Sensors* **2017**, *17*, 472. [CrossRef] [PubMed]
16. Chauffaut, C.; Defay, F.; Burlion, L.; De Plinval, H. UAV obstacle avoidance scheme using an output to input saturation transformation technique. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 227–234. [CrossRef]
17. Kwon, J.-W.; Seo, J.; Kim, J.H. Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV. *Electron. Lett.* **2014**, *50*, 1431–1432. [CrossRef]
18. Kyristsis, S.; Antonopoulos, A.; Chanialakis, T.; Stefanakis, E.; Linardos, C.; Tripolitsiotis, A.; Partsinevelos, P. Towards autonomous modular UAV missions: The detection, geo-location and landing paradigm. *Sensors* **2016**, *16*. [CrossRef] [PubMed]
19. Liu, C.; Liu, J.; Song, Y.; Liang, H. A novel system for correction of relative angular displacement between airborne platform and UAV in target localization. *Sensors* **2017**, *17*. [CrossRef] [PubMed]
20. Agrawal, P.; Ratnoo, A.; Ghose, D. Inverse optical flow based guidance for UAV navigation through urban canyons. *Aerosp. Sci. Technol.* **2017**, *68*, 163–178. [CrossRef]
21. Wen, N.; Zhao, L.; Su, X.; Ma, P. UAV online path planning algorithm in a low altitude dangerous environment. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 173–185. [CrossRef]
22. Sasongko, R.A.; Rawikara, S.S.; Tampubolon, H.J. UAV Obstacle Avoidance Algorithm Based on Ellipsoid Geometry. *J. Intell. Robot. Syst.* **2017**. [CrossRef]
23. Yang, P.; Tang, K.; Lozano, J.A.; Cao, X. Path Planning for Single Unmanned Aerial Vehicle by Separately Evolving Waypoints. *IEEE Trans. Robot.* **2015**, *31*, 1130–1146. [CrossRef]
24. Arya, S.R.; Ashokkumar, C.R.; Arya, H. Gamma and velocity tracking for UAV obstacle avoidance in pitch plane. In Proceedings of the 2016 Indian Control Conference (ICC), Hyderabad, India, 4–6 January 2016; pp. 362–368.
25. Babel, L. Curvature-constrained traveling salesman tours for aerial surveillance in scenarios with obstacles. *Eur. J. Oper. Res.* **2017**, *262*, 335–346. [CrossRef]
26. Lugo-Cardenas, I.; Flores, G.; Salazar, S.; Lozano, R. Dubins path generation for a fixed wing UAV. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; Volume 2, pp. 339–346. [CrossRef]
27. Owen, M.; Beard, R.W.; McLain, T.W. Implementing Dubins Airplane Paths on Fixed-Wing UAVs. In *Hand book of Unmanned Aerial Vehicles*; Springer: New York City, NY, USA, 2015.
28. Flores, G.; Lugo-Cardenas, I.; Lozano, R. A nonlinear path-following strategy for a fixed-wing MAV. In Proceedings of the 2013 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 28–31 May 2013; pp. 1014–1021. [CrossRef]
29. Yang, C.; Liu, L.; Wu, J. Path planning algorithm for small UAV based on dubins path. In Proceedings of the IEEE International Conference on Aircraft Utility Systems (AUS), Beijing, China, 10–12 October 2016; pp. 1144–1148. [CrossRef]

30. Manyam, S.; Rathinam, S.; Casbeer, D. Dubins paths through a sequence of points: Lower and upper bounds. In Proceedings of the 2016 International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA, USA, 7–10 June 2016; pp. 284–291. [CrossRef]
31. Hota, S.; Ghose, D. Time-optimal convergence to a rectilinear path in the presence of wind. *J. Intell. Robot. Syst. Theory Appl.* **2014**, *74*, 791–815. [CrossRef]
32. Chang, A.J.; Brazil, M.; Rubinstein, J.H.; Thomas, D.A. Optimal curvature-constrained paths for general directional-cost functions. *Optim. Eng.* **2013**, *14*, 395–416. [CrossRef]
33. Bakolas, E.; Tsiotras, P. Optimal Synthesis of the Asymmetric Sinistral/Dextral Markov-Dubins Problem. *J. Optim. Theory Appl.* **2011**, *150*, 233–250. [CrossRef]
34. Bakolas, E.; Tsiotras, P. Optimal Synthesis of the Zermelo-Markov-Dubins Problem in a Constant Drift Field. *J. Optim. Theory Appl.* **2013**, *156*, 469–492. [CrossRef]
35. Nunez, H.E.; Flores, G.; Lozano, R. Robust path following using a small fixed-wing airplane for aerial research. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 1270–1278.
36. Lin, Y.; Saripalli, S. Path planning using 3D Dubins Curve for Unmanned Aerial Vehicles. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 296–304.
37. Aguilar, W.; Casaliglla, V.; Pólit, J. Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics* **2017**, *6*, 10. [CrossRef]
38. Lin, Y.; Saripalli, S. Sampling-Based Path Planning for UAV Collision Avoidance. *IEEE Trans. Intell. Transp. Syst.* **2017**, 1–14. [CrossRef]
39. Wang, H.; Liu, Y.; Li, M.; Huang, H.; Xu, H.M.; Hong, R.J.; Shen, H. Reference Path Generation and Obstacle Avoidance for Autonomous Vehicles Based on Waypoints, Dubins Curves and Virtual Force Field Method. *Optoelectron. Adv. Mater. Rapid Commun.* **2010**, *4*, 1166–1169. [CrossRef]
40. Lai, S.; Wang, K.; Qin, H.; Cui, J.Q.; Chen, B.M. A robust online path planning approach in cluttered environments for micro rotorcraft drones. *Control. Theory Technol.* **2016**, *14*, 83–96. [CrossRef]
41. Upadhyay, S.; Ratnoo, A. Smooth Path Planning for Unmanned Aerial Vehicles with Airspace Restrictions. *J. Guid. Control. Dyn.* **2017**, *40*, 1596–1612. [CrossRef]
42. Zhong, W.; Yan, L. A target visiting path planning algorithm for the fixed-wing UAV in obstacle environment. In Proceedings of the 2014 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC), Yantai, China, 8–10 August 2014; pp. 2774–2778. [CrossRef]
43. Lugo-Cárdenas, I.; Flores, G.; Lozano, R. The MAV3DSim: A simulation platform for research, education and validation of UAV controllers. *IFAC Proc. Vol.* **2014**, *19*, 713–717. [CrossRef]



sensors

MDPI

Article

Observing Spring and Fall Phenology in a Deciduous Forest with Aerial Drone Imagery

Stephen Klosterman [1,*] and Andrew D. Richardson [1,2,3]

[1] Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, MA 02138, USA; Andrew.Richardson@nau.edu

[2] School of Informatics, Computing and Cyber Systems, Northern Arizona University, Flagstaff, AZ 86011, USA

[3] Center for Ecosystem Science and Society, Northern Arizona University, Flagstaff, AZ 86011, USA

* Correspondence: steve.klosterman@gmail.com

Received: 29 September 2017; Accepted: 5 December 2017; Published: 8 December 2017

Abstract: Plant phenology is a sensitive indicator of the effects of global change on terrestrial ecosystems and controls the timing of key ecosystem functions including photosynthesis and transpiration. Aerial drone imagery and photogrammetric techniques promise to advance the study of phenology by enabling the creation of distortion-free orthomosaics of plant canopies at the landscape scale, but with branch-level image resolution. The main goal of this study is to determine the leaf life cycle events corresponding to phenological metrics derived from automated analyses based on color indices calculated from drone imagery. For an oak-dominated, temperate deciduous forest in the northeastern USA, we find that plant area index (PAI) correlates with a canopy greenness index during spring green-up, and a canopy redness index during autumn senescence. Additionally, greenness and redness metrics are significantly correlated with the timing of budburst and leaf expansion on individual trees in spring. However, we note that the specific color index for individual trees must be carefully chosen if new foliage in spring appears red, rather than green—which we observed for some oak trees. In autumn, both decreasing greenness and increasing redness correlate with leaf senescence. Maximum redness indicates the beginning of leaf fall, and the progression of leaf fall correlates with decreasing redness. We also find that cooler air temperature microclimates near a forest edge bordering a wetland advance the onset of senescence. These results demonstrate the use of drones for characterizing the organismic-level variability of phenology in a forested landscape and advance our understanding of which phenophase transitions correspond to color-based metrics derived from digital image analysis.

Keywords: phenology; Harvard Forest; leaf color; plant area index; drone; UAV

1. Introduction

Phenology, the study of recurrent biological events, has been a focus of plant science for centuries [1]. Phenology responds to interannual and spatial variability in environmental conditions, particularly temperature [2,3], and also mediates key ecosystem functions, including carbon assimilation and evapotranspiration [4–8]. As global environmental change becomes an increasingly public issue, the value of plant phenology as a sensitive indicator of the effects of change has kindled interest in creating and interpreting phenology records, such as those from digital repeat photography. Tower- or building-mounted phenocams, typically located at positions just above the canopy [9–11], preserve a valuable visual record of vegetation phenology in forests and other ecosystems. "Near-surface" methods such as phenocams complement the phenology records of satellite remote sensing, which extensively observe entire landscapes, but at a spatial resolution that typically makes it impossible to discern individual plants [12]. Unlike other near-surface methods

based on radiometric sensors, e.g., [13,14], images from phenocams have the advantage that they can be visually interpreted by direct examination.

With phenocam imagery, it is also possible to objectively quantify phenology from time series of vegetation "greenness" or "redness". Discrete phenophase transition dates can be derived from those time series, using techniques developed for analyzing satellite remote sensing data [15,16]. These methods have been successfully applied to the evaluation of satellite remote sensing phenology products [17,18] and exploration of the connections between ecosystem function, e.g., carbon assimilation, and the greenness metrics of phenology [5,6]. However the oblique angle of tower-mounted cameras has been suggested to result in biased estimations of the timing of canopy maturity, compared to the vertically integrated plant area index (PAI), as well as in situ observations of leaf expansion [19]. Phenocam imagery does make it possible to easily integrate across multiple organisms in the camera field of view and thus characterize landscape-level phenology. But, in mixed-species stands, these estimates may be biased because the organisms located closest to the camera dominate the field of view and hence are over-represented, while more distant organisms are under-represented.

Drones, also called UAVs, open up an exciting new possibility for the near-surface remote sensing of plant phenology. With drones, researchers can obtain aerial photography with a similar spatial resolution to tower-mounted phenocams, but at the landscape scale, similar to satellite remote sensing. Compared to traditional aircraft, drones can be operated at a fraction of the cost, making more frequent observations feasible. Additionally, using photogrammetry techniques with drone images facilitates significant advances over tower-mounted cameras. Orthomosaics simulate an undistorted nadir perspective of the canopy, with a consistent spatial resolution over landscapes [20]. Because of this feature, orthomosaics enable the identification and analysis of larger numbers of individual organisms than is typically possible using tower-mounted camera imagery.

Previous research has begun to explore the potential of drones for the study of forest phenology. The first study to use aerial drone imagery to explore forest structural and color properties demonstrated that the changing color of leaves at different times of the growing season can be quantified by applying digital image processing techniques to georeferenced orthomosaics [20]. Subsequently, researchers leveraged the high spatial resolution of near-surface aerial drone photography to delineate crowns and study the phenology of individual trees. It was found that the color of individual tree crowns could be used to identify their species, based on knowledge of their expected phenological status during different times of the season [21]. In another study, high temporal frequency aerial images taken during springtime were used to show that the timing of budburst of individual trees, observed in situ, appears to coincide with the beginning of an increase in a canopy greenness metric calculated from the images of these trees [22]. In our previous work, which used an earlier year of imagery from the same study site reported here, we used drone imagery to show how landscape scale variance in phenology could be attributed to plant species, and explored the nature of the spatial variability of phenology at different spatial resolutions [23].

Our goal in this study is to extend previous efforts to determine the leaf life cycle events of trees that correspond to digital image analysis metrics of those same trees in aerial drone imagery. We use imagery of the complete growing season to go beyond budburst and consider leaf expansion in spring, as well as color change and abscission in autumn. We also compare metrics derived from aerial images with the progression of PAI, to interpret color indices with reference to canopy structural development. Finally, we examine how contrasting air temperature regimes at microsites within the study area correlate with spatial variation in landscape phenology, after accounting for spatial variance induced by differences in tree species. These results demonstrate the use of drones for observing the complete seasonal cycle of deciduous canopy development at the landscape scale, with a high enough spatial resolution to discern organism-level phenomena.

2. Materials and Methods

2.1. Study Site

We conducted our study at Harvard Forest in Petersham, MA, USA (42.5377° N, 72.1715° W). The climate is temperate, with a mean annual precipitation of 110 cm and mean annual temperature of 7.1 °C. The study area is a mixed deciduous-evergreen forest, with some woody wetlands. Deciduous trees in the study area include predominantly red oak (*Quercus rubra*) and red maple (*Acer rubrum*), as well as yellow birch (*Betula alleghaniensis*), American beech (*Fagus grandifolia*), and black oak (*Quercus velutina*). The study area is 2.8 ha in size and contains approximately 1900 trees with a diameter at breast height ≥10 cm (Figure 1).

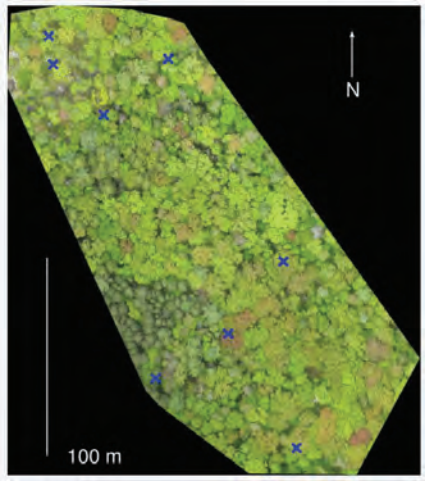


Figure 1. Study area at Harvard Forest on 5/21/17 (DOY 141). Location of microsite temperature loggers indicated as blue "×" symbols.

2.2. Drone Image Acquisition and Processing

We used methods described in an earlier study [23] to obtain and process aerial photography. Briefly, we used a drone (3DR ArduCopter Quad-C Frame, 3D Robotics, Berkeley, CA, USA) equipped with a Canon Powershot A3300 camera (35 mm film equivalent focal length 28 mm, approx. 16 million pixels). We used the same camera and image settings for all flights, and took pictures of a gray reference square (ColorChecker classic, X-rite, Grand Rapids, MI, USA) before each flight. Flight frequency was roughly every five days during spring leaf out and every week during autumn leaf color change, depending on the weather conditions (dates shown in Figure 2). Images were taken at a minimum shutter speed of 1/1000 s, with constant exposure during each flight. The same color balance was used for all acquisition dates, because consistent color balance is necessary for reliable digital camera observations of phenology [11]. We conducted flights at mid-day (between 10 a.m. and 3 p.m.) on either clear or evenly overcast days, and never during periods of variable cloud cover, as exposure was constant during flights. For each imagery acquisition date, we created orthophotos of the study area using about 100 JPEG photos taken with an intervalometer script (Canon Hack Development Kit, http://chdk.wikia.com/wiki/CHDK), with the PhotoScan photogrammetry software (Agisoft, St. Petersburg, Russia), and performed final georeferencing in ERDAS IMAGINE AutoSync (Intergraph, Huntsville, AL, USA). The orthophotos used in this study are available in the Harvard Forest Data Archive [24].

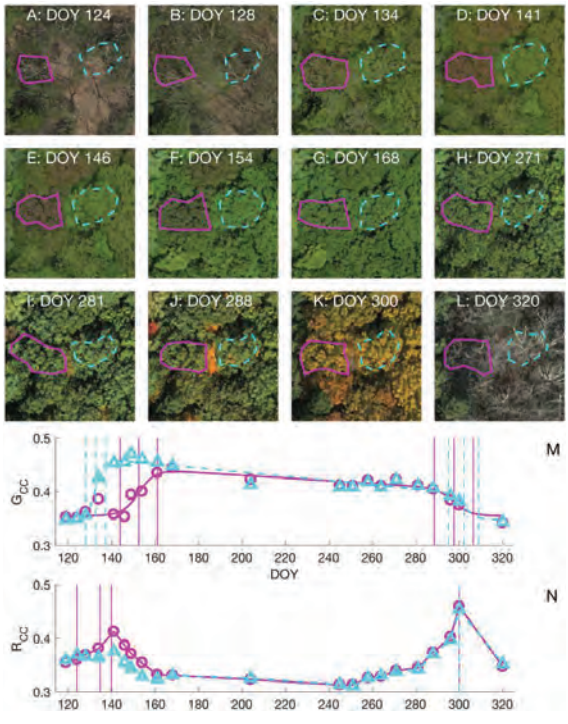


Figure 2. (**A**–**L**): Close-up (30 m by 30 m) of aerial images for a selection of dates from 2015. Each close-up shows the ROIs used to analyze two red oak trees. Solid magenta ROIs were used to calculate solid magenta G_{CC} and R_{CC} values (circles symbols) and curves; similar with the dashed cyan ROIs (triangle symbols.). Images have been equalized to have the same brightness (sum of R, G, and B digital numbers) for visualization purposes in this figure; (**M**,**N**): the resulting G_{CC} and R_{CC} time series (symbols); curve fits or interpolations; and estimated phenophase transition dates (vertical lines). The vertical lines show six annual dates for both G_{CC} time series, while spring dates calculated from R_{CC} series are only shown for the tree with red spring leaves. R_{CC}-EOF occurred on the same date for both trees.

We calculated green chromatic coordinate (G_{CC}) [25] and red chromatic coordinate (R_{CC}) [11] time series from orthophotos using Matlab (R2017a), to account for changes in scene illumination due to differences in atmospheric conditions or times of day between flights:

$$G_{CC} = G/(R + G + B), \quad (1)$$

$$R_{CC} = R/(R + G + B), \quad (2)$$

where R, G, and B are the mean red, green, and blue digital numbers, respectively, in regions of interest from image files. Example G_{CC} and R_{CC} time series and additional processing details are available in Figure S1. We note that the phenology signal observed in the G_{CC} and R_{CC} time series of vegetation was approximately 10 times greater in amplitude than the noise observed in the analysis of the gray reference square, which arose due to factors such as varying illumination conditions and the different times of day of flights. We also explored other indices, including $G_{CC} + R_{CC}$, GRVI, ExG, and Hue, as described in the caption of Figure S2, but found the most reliable results from G_{CC} and R_{CC}, and used them in our analysis.

Color indices were calculated using different regions of interest (ROIs) depending on the goal of the analysis. For comparison between image-based metrics and in situ observations of individual tree phenology, we drew ROIs around the crowns of the trees that we observed, as shown in Figure 2. For comparison to upward photo-based estimates of PAI, we created ROIs representing the canopy projection of the image field of view from the ground perspective. To examine phenology across the landscape, we used a square grid of 10 m ROIs (grid cells).

2.3. Estimating Phenology Dates from Time Series Data

We used curve fitting methods detailed in Klosterman et al. [18] to estimate phenology dates from time series data, including G_{CC} and additional data described below. Sigmoid models are commonly used to approximate the seasonal trajectory of vegetation indices and to facilitate the determination of phenological transition dates. The greendown sigmoid is a variant of the standard dual sigmoid model [16]; a key difference is that the greendown sigmoid allows for a linear decrease in greenness over the course of the growing season. We used the greendown sigmoid model to approximate the seasonal trajectory of G_{CC} for each ROI. Phenology dates were calculated from curve fit parameters by finding the dates of extrema in the curvature change rate (CCR) [15]. We used these methods to calculate dates for the start, middle, and end of canopy development in spring (G_{CC}-SOS, G_{CC}-MOS, G_{CC}-EOS), and similar dates for canopy senescence in fall (G_{CC}-SOF, G_{CC}-MOF, G_{CC}-EOF). We estimated uncertainties using the Jacobian matrix of curve fit parameters to generate Monte Carlo ensembles of phenology dates, and calculated the inner 95% confidence interval of these ensembles.

To estimate phenology dates from R_{CC} time series, we calculated the dates of crossing the 10th, 50th, and 90th percentiles of a linear interpolation of R_{CC} values, while R_{CC} increased to its spring maximum, to determine R_{CC}-SOS, R_{CC}-MOS, and R_{CC}-EOS. We used the date of the maximum redness value in fall to represent R_{CC}-EOF. We determined uncertainties as the interval between the nearest two observation dates, or for transitions that occurred on an observation date, as the interval between the midpoints of that date and the nearest two dates.

In addition to these automated curve fitting and interpolation approaches, we also visually identified the day of the first observable leaves in aerial images of the individual trees discussed below in Section 2.4.1.

2.4. In Situ Measurements

We identified eight microsites within the study area based on earlier results, using imagery from a previous year at this site. We had previously found that species composition explained most of the spatial variance in phenology across the Harvard Forest landscape, using a multiple linear regression [23]. However, we identified locations within the study area where regression residuals were relatively large; these were the same areas in regressions using both the 2013 data from the previous study and the 2015 data from the present study. We located microsites in these places, as well as locations where residuals were relatively small, to better understand the drivers of phenology throughout the study area. To provide context for the image-derived phenological transition dates, we made direct observations of phenology on individual trees, and measured canopy-level PAI using digital cover photography. To characterize the microsite environment, we measured air temperature.

2.4.1. Direct Observation of Trees

We observed the phenological status of 30 trees spread across the eight microsites within the study area during each drone photography acquisition date (Figure 1, three to five trees per microsite), and one additional date when the drone was not flown due to weather constraints (5/17/15, DOY 137). Similar to a protocol of established observations of tree phenology at Harvard Forest [26], we estimated the percentage of buds burst on trees, and average leaf size (length) as a percentage of mature leaf size. In autumn, we estimated the percentage of leaves that had changed color due to senescence, and that had fallen. From these observations, we created time series using linear interpolation, and found the

point in time corresponding to the deciles of phenological status for each phenology event: day of year for 10%, 20%, ... 90% of budburst, leaf size, leaf color, and leaf fall. We calculated uncertainties in the same way used for R_{CC} time series.

We determined the leaf life cycle events corresponding to image-derived metrics of individual tree ROIs by finding the decile of progress in a specific event that had the lowest RMSD with respect to a given metric across the 30 trees under observation. In other words, we compared all deciles (10%, 20%, ...) of all observed events (e.g., budburst, leaf size) to each of the image-derived transitions (SOS, MOS, ...), and identified the decile that had the minimum RMSD across all trees (e.g., 10% budburst was closest to SOS). We also examined correlations of G_{CC} and R_{CC} values on each date with percentages of progress in each life cycle event on the same date, within and across trees. Since color indices and leaf transitions typically trend in the same direction, for example, increasing G_{CC} and leaf size in spring, an ordinary least squares regression may yield spuriously low coefficient standard errors. Therefore, we used an econometric approach to time series regression: heteroscedasticity and autocorrelation consistent (HAC) regression. HAC regression calculates the variances of regression coefficients (i.e., square of standard error, used to calculate regression *p*-value) based on the inferred autocorrelation structure of regression residuals; the coefficient estimates themselves are the same as those of ordinary least squares. We used the 'hac' function in Matlab, with pre-whitening at a lag of one time step and the quadratic spectral weighting scheme, as we found that these options led to the most conservative estimates of coefficient variances in a sensitivity analysis [27,28]. We used *t*-tests to see if there were significant differences in regression parameters between trees. We also calculated ordinary least squares regressions of pooled values across all trees.

2.4.2. Digital Cover Photography (DCP)

Upward photos were taken with the same model of camera used for drone photography (non-fisheye lens). We took images manually for upward photos, on the same dates we acquired drone photography, using the same image settings (i.e., automatic exposure), except for white balance, which was set to "auto" for DCP [29]. We used a level to aim the camera directly upward and positioned the camera on the posts holding the temperature data loggers (1 m height, below most mid-canopy trees), in the same orientation on each date. The projected camera field of view had an area of 533 m^2 in the canopy (at the average Harvard Forest canopy height of 22 m). We estimated PAI from upward photos at each microsite using a freely available DCP software tool obtained from Craig MacFarlane (contact information in references [29,30]). We calculated sigmoid curve fits of PAI values to estimate transition dates as with G_{CC} time series. However, we observed that PAI values were generally stable from June to September, so we used the traditional sigmoid model as opposed to the greendown sigmoid, effectively setting the slope of the summertime greendown to zero. We also directly compared G_{CC} and R_{CC} values with PAI values, by performing HAC regressions at the microsite level, as both PAI and color indices typically trend in the same direction at each microsite. We also performed ordinary least squares regressions of data pooled across microsites for spring and fall, similar to the analysis described in Section 2.4.1.

2.4.3. Air Temperature Measurements and Effects of Microclimate on Phenology

Because the microsite locations included places with the largest residuals in a regression of species composition and phenology dates, we suspected that additional factors may contribute to the timing of microsite phenology. Temperature effects on phenology have been widely studied [18,31–33]. Therefore, we recorded air temperatures half hourly from 11 April 2015, approximately one month before the beginning of leaf out, through December 2015, after completion of leaf fall. To do this, we installed a HOBO U23-004 temperature data logger at a 1 m height with an RS-1 radiation shield (Onset Computer Corp, Bourne, MA, USA) at each microsite.

To determine the effect of microclimate on phenology, we correlated spring and fall temperatures with residuals from a statistical model that accounted for the effects of species variation and land cover

type, but not microclimate, on the aggregated canopy phenology of 10 m grid cells. A 10 m grid was chosen based on previous analysis, taking into account spatial inaccuracies in the orthophotos and the nature of the species data [23]. More specifically, the model used tree species from a map of all woody stems ≥1 cm diameter at breast height [34]. This map covered 89% of the land area monitored in drone imagery. Woody species composition was determined using all species that appeared in at least 10 of 245 grid cells, and had at least a 20% basal area fraction in any one grid cell. The remaining species were lumped into one predictor to ensure that the fractional species composition of each grid cell summed to 1. Any grid cell that had <50% coverage in the aerial images was eliminated from this analysis, removing partially covered grid cells on the edge of the imaged area and ensuring all data points had adequate spatial representation. We used a no-intercept regression since fractional species compositions are mixture components that sum to 1 [35]. We then calculated the average residual from these spring and fall regressions for the three by three windows of grid cells centered on the microsite locations (30 m by 30 m areas), and regressed these microsite-average residuals on the monthly means of daily minimum, mean, and maximum temperatures for April (MOS) and September (MOF), as temperatures preceding phenology events are commonly used to predict the timing of those events [18,31–33]. We note that we used R_{CC}-MOS for grid cells with red spring leaves, following a criterion discussed in the Results Section 3.1, as long as the amplitude in R_{CC} was greater than the range of noise observed in the gray reference square (Figure S1). For all other grid cells, we used G_{CC}-MOS. We used Bonferroni correction to account for making three temperature comparisons with both phenology transition dates.

3. Results

3.1. Choice of Color Index in Spring Time

While changes in canopy greenness have commonly been used to track spring budburst and leaf expansion in deciduous forests, we found that in some instances, these processes appeared to be more associated with changes in redness (Figures S2 and S3). For two of the 30 trees we observed in situ, both red oaks, we saw that leaves were various shades of red (pink, orange) in color during leaf expansion (Figure 1, close up in Figure 2D,E, and Figure S4). Additional oak trees, which were not under in situ observation at the microsites, can also be seen to display red spring leaves in Figure 1. We note that leaves higher in the canopy, i.e., those on the top of the crown and most visible from an aerial view, often appeared to be redder than leaves closer to the ground (Figure S4).

For trees with red spring leaves, the springtime G_{CC} profiles showed a delayed increase compared to trees with green spring leaves, including nearby conspecifics that we observed to have nearly the same leaf expansion phenology (Figure 2M,N). However, the springtime R_{CC} time series exhibited a marked peak for trees with red spring leaves. Then, as R_{CC} decreased, G_{CC} increased as leaves became greener in late spring (Figure 2F,G). We found that for these trees, the springtime amplitude in R_{CC} (the increase from the dormant value to the spring maximum) was more than 45% of the spring amplitude in G_{CC} (46% and 64% for the two trees), while for all other trees, it was less than 35%. The springtime increase in R_{CC} was closer in time to observed leaf expansion than G_{CC}, as well as a range of other color indices we considered (G_{CC} + R_{CC}, GRVI, ExG, Hue, Figure S2). We also note that the leaves of oak trees with red spring leaves had no apparent color difference in autumn from those of trees with green spring leaves (Figure 2K).

3.2. Leaf Life Cycle Events of Trees: Correspondence to Image Metrics

We determined the leaf life cycle events corresponding to image metrics by finding the closest decile of life cycle event (i.e., 10% budburst) to each metric (SOS, MOS, etc.) for the 30 trees under observation. Following our observation that R_{CC} was a better indicator of spring leaf phenology than G_{CC} for trees with red spring leaves, we used R_{CC} metrics for the two trees where the spring amplitude in R_{CC} was greater than 40% of spring amplitude in G_{CC}, and G_{CC} metrics for all other trees.

We found that SOS was closest to 10% budburst (RMSD 4.7 days), although the date of 10% budburst was not significantly correlated to SOS across trees (r = −0.04, $p > 0.05$). This low (and not statistically significant) correlation is likely related to the fact that there was relatively little variability in the timing of budburst across trees: all trees under observation had 10% budburst within a period of about a week (DOY 121-129), while SOS date uncertainties were 11 days on average (inner 95% confidence interval, i.e., ±5.5 days). However, we found that our determinations of the first observable leaves in aerial image orthomosaics were correlated with SOS (r = 0.46, $p < 0.05$), indicating SOS dates derived from color-based vegetation indices were associated with biologically meaningful phenological transitions at the branch-to-canopy level.

We found that MOS was closest to 40% leaf size (RMSD 3.6 days), and EOS was closest to 70% leaf size (RMSD 6.9 days; spring time G_{CC}, budburst, and leaf size observations with associated transition dates shown in Figure S5 for an example tree). Later spring transitions were more variable across trees, with a range of 14 days for 40% leaf size and 18 days for 70% leaf size (Figure 3A,B). Across species, the two most common species, red oak and red maple, had similar leaf expansion phenology. However, American beeches were among the first trees to make the MOS and EOS transitions, and attain a 40% and 70% leaf size. The mean inner 95% confidence intervals across trees were six and 12 days for MOS and EOS metrics, respectively. MOS is typically the most certain metric in sigmoid curve fitting analyses [18] due to the clear timing of the steepest part of the increase in spring time G_{CC} (Figure 2M). Consequently, we found that the MOS from image analysis could represent tree-to-tree variation in the timing of leaf expansion observed in situ with greater statistical significance than EOS (MOS r = 0.52, $p = 0.003$; EOS r = 0.38, $p = 0.04$, Figure 3A,B).

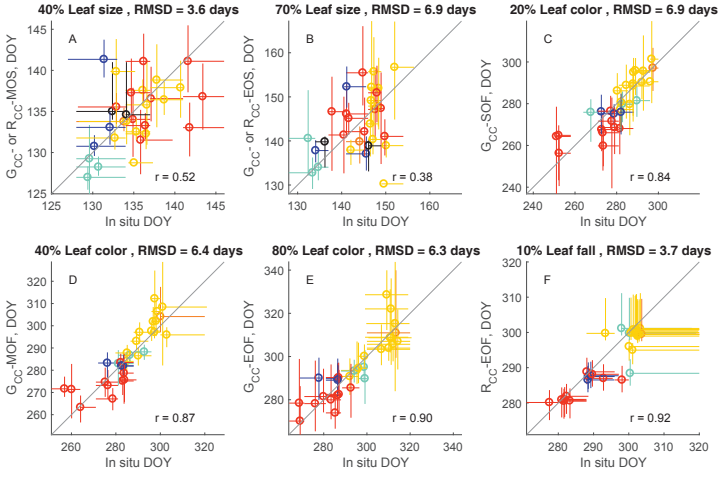


Figure 3. (**A**–**F**) leaf life cycle events corresponding to drone image metrics for 30 trees. Symbols denote transition dates, with confidence intervals shown as colored lines. Yellow = red oak (except trees with red spring leaves, shown in black in (**A**,**B**)), red = red maple, light blue = American beech, dark blue = yellow birch, and orange = black oak. Deciles of in situ observations (i.e., 40% leaf size) were selected as having the minimum RMSD in comparison to drone image metrics. Jitter was added to data points that lay on the same date and the one-one line is shown in black. Correlation coefficients are shown on each panel (all $p < 0.05$).

To examine how representative color indices are of the progression of leaf life cycles at the tree level, we calculated HAC regressions between percentages of life cycle completion and index values across observation dates. Leaf size observations were more highly correlated with G_{CC} values in spring (average r = 0.95, 25 of 28 trees $p < 0.01$, Table S1) than budburst observations (average r = 0.86,

21 of 28 trees $p < 0.01$). We also note that, pooling across all trees with green spring leaves, G_{CC} values were highly correlated with leaf size observations (r = 0.88, $p < 0.001$), and to a similar but lesser extent, with budburst observations (r = 0.80, $p < 0.001$). However, we note that there were significant differences in regression slopes and intercepts between individual trees, even within species, according to *t*-tests ($p < 0.05$). This indicates that although there is a clear, statistically significant relationship between color indices and leaf life cycle progressions when all trees are pooled, there is also significant tree-to-tree variability in these relations when regression parameters are allowed to vary by tree.

For the two trees with red spring leaves, R_{CC} increased, and then declined in spring as leaves continued to expand and change in color from red to green (Figure 2F,G). Therefore, we considered R_{CC} values up to and including the spring maximum when examining correlations. Similar to G_{CC} for trees with green spring leaves, we found a higher correlation of R_{CC} with leaf size (Table S2, average r = 0.95, both trees $p < 0.001$) than budburst (average r = 0.76, neither tree $p < 0.001$). These results indicate that G_{CC} and R_{CC} values are more representative of the progression of leaf expansion than budburst, as these color indices are still increasing after the completion of budburst (e.g., Figures S2, S3 and S5).

The timing of leaf coloration events in autumn varies more among trees than either budburst or leaf expansion (Figure 3C–E) with a range of roughly 50 days in leaf color deciles across the 30 trees under observation. Species differences become more apparent in fall than in spring, with red oaks senescing after red maples, and American beeches and yellow birches in between these two. The SOF, MOF, and EOF transitions calculated from G_{CC} are fairly evenly spaced throughout the period of leaf color change, corresponding to 20% (RMSD 6.9 days), 40% (RMSD 6.4 days), and 80% (RMSD 6.3 days) leaf color, respectively. R_{CC}-EOF, representing the date of the maximum R_{CC} value in autumn, was closest to 10% leaf fall (RMSD 3.7 days), the earliest abscission metric from in situ observations (Figure 3F).

G_{CC} values in autumn were negatively correlated with leaf color percentages (average r = -0.92, 26 of 30 trees $p < 0.01$, Table S1). While G_{CC} for deciduous trees generally declines as leaves senesce in autumn, R_{CC} increases to a maximum before declining to the dormant season level. We distinguished between these increasing and decreasing phases when examining correlations between R_{CC} and autumn phenological processes. R_{CC} values up to and including the fall maximum correlated with leaf color percentages (average r = 0.94, all trees $p < 0.01$, Table S3), while fall R_{CC} values including the maximum and later values were correlated with leaf fall percentages (tree average r = -0.94, 9 of 16 trees with sufficient observations $p < 0.01$). Similar to the springtime analysis, we found significant correlation when pooling all trees: leaf color was correlated to G_{CC}, which decreased in autumn (r = -0.89, $p < 0.001$), as well as R_{CC} as it increased to the fall maximum (r = 0.84, $p < 0.001$). Leaf fall was more highly correlated to R_{CC} as it decreased after the fall maximum (r = -0.89, $p < 0.001$) than G_{CC} (r = -0.56, $p < 0.001$), since G_{CC} decreased due to color change, before leaves began falling. As in spring, we noted significant differences in regression slopes and intercepts between trees of the same species in these comparisons.

3.3. Linking Leaf Area to Color Indices

We derived transition dates (SOS, MOS, etc.) from PAI time series, as well as from drone imagery using ROIs corresponding to the top-of-canopy area analyzed for PAI (example microsite shown in Figure 4). This allowed us to examine the relation between canopy structural development and canopy color development in drone images. The timing of the spring increase in PAI and G_{CC} was similar across the eight microsites, particularly for later spring transitions. At the first spring transition, SOS, the average difference between PAI and G_{CC} derived dates was -4.1 (negative sign indicates G_{CC}-MOS was earlier) ± 3.2 days standard deviation, and was significantly different from zero according to a paired *t*-test ($p < 0.05$). However, the MOS and EOS transitions were closer in time, with -0.6 ± 2.5 days and 2.6 ± 4.6 days differences, respectively. Both of these differences were not significantly different from zero. In autumn, declines in G_{CC} substantially preceded declines in PAI, as leaves changed color, becoming less green, before they were shed, decreasing the canopy PAI. As a

result, PAI and G_{CC} transitions were all significantly different in autumn, with average differences of 12–15 days.

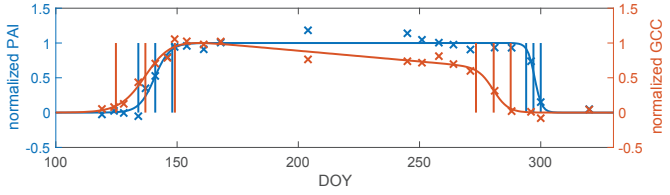


Figure 4. Smoothed (curve) and raw (× symbols) PAI and G_{CC} curves for an example microsite, and the dates derived from each data source (vertical lines show the SOS, MOS, EOS, SOF, MOF, and EOF dates). Values are normalized to have minimum 0 and maximum 1 according to the curve fits, to facilitate the comparison of PAI and G_{CC} trajectories.

To further investigate the connection between color indices and the canopy leaf area, we calculated the correlation of PAI to G_{CC} and R_{CC} values, pooled across microsites (Figure 5), as well as within microsites (Table S4). Spring G_{CC} was highly correlated to PAI ($p < 0.001$) when pooling all the data, as well as at individual microsites ($p < 0.001$ for six of eight microsites). However, there were significant differences in regression slopes and intercepts between microsites (*t*-test with $p < 0.05$). This is partially due to the fact that maximum PAI values vary more than maximum G_{CC} values, spatially throughout the forest. The maximum summertime PAI values we observed at the microsites varied from 3.4 to 6.4, while maximum G_{CC} varied from 0.44 to 0.48 (Table S4).

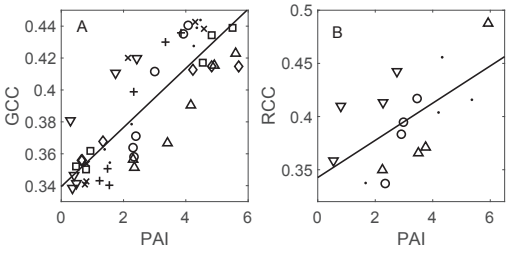


Figure 5. Scatter plots and lines of best fit between PAI values from upward photography and (**A**) G_{CC} values from drone imagery in spring (N = 48, eight microsites); (**B**) R_{CC} values after the maximum R_{CC} in fall, for microsites with one or more observations between the fall maximum of R_{CC} and the last observation (N = 16, four microsites). Different microsites are indicated by distinct symbols. The pooled-data regression equations are $G_{CC} = 0.34 + 0.019$*PAI (r = 0.84, $p < 0.001$) and $R_{CC} = 0.34 + 0.017$*PAI (r = 0.60, $p < 0.05$). Microsite-level correlations reported in Table S4.

In fall, we examined the correlation between PAI and R_{CC} values from the fall R_{CC} maximum onward, following our observation that the peak of autumn R_{CC} indicates the beginning of leaf fall, which also marks the beginning of the decrease in the canopy leaf area. As in spring, we found evidence for linear correlation in pooled data across microsites, although there was a lower statistical significance ($p < 0.05$), likely due in part to a smaller sample size. Similarly, site-level correlations of R_{CC} and PAI were lower than in spring (average r = 0.90 versus 0.96 for springtime G_{CC} and PAI), although half of the microsites (two of four) with sufficient observations for analysis had a statistically significant linear correlation at $p < 0.05$ (based on ordinary least squares regressions, as the HAC adjustment for serial autocorrelation could not be used due to the small sample size).

3.4. Microclimate Effect on Phenology

To examine landscape variability in phenology, we calculated MOS and MOF dates of 10 m grid cells (Figure S6). We used G_{CC}-MOS, except for six grid cells with a spring R_{CC} amplitude greater than 40% of the spring G_{CC} amplitude, based on the results of Section 3.1, and where we observed red spring leaves. We also filtered these grid cells to require the spring amplitude in R_{CC} to be greater than 0.02, to ensure that the color metric reflected a substantial change in redness that represented leaf growth as opposed to background noise. For these six grid cells, we used R_{CC}-MOS. Variability across grid cells (Figure S6) was similar to that observed for these dates in in situ observations of trees (14 days range in MOS, 42 days in MOF, Figure 3), with a localized area of later spring and earlier fall phenology along a forest edge bordering a wetland, in the northwest of the study area (Figure S6).

We found that 29% of variance in MOS dates and 46% in MOF dates of 10 m grid cells was explained by spatial differences in the woody plant species assemblage (regression results in Table S5). Regression coefficients indicated a similar phenological order of species to that observed in situ: American beech was among the earliest species in spring, and red maple senescence preceded that of red oak by 23 days. In a previous analysis, we found evidence of spatial autocorrelation on length scales up to at least 30 m, in residuals from a similar regression. This was presumably due in part to fine-scale temperature variation [23]. Consequently, we examined residuals from the regression we performed here, in the 30 m by 30 m neighborhoods of the microsites where temperature measurements were made. Due to the low spatial variation in spring temperatures and a limited number of microsites, we did not detect significant relationships between microsite temperature and MOS residuals. In autumn, we found that the September mean of daily minimum temperature was correlated to the residuals of MOF dates (Bonferroni-adjusted $p = 0.08$, $r = 0.77$). We found an effect size of a five day delay (1.7 day SE) in MOF date per °C warmer (Figure S7).

4. Discussion

4.1. Leaf Life Cycles According to Image Metrics

4.1.1. Exploration of Budburst, Leaf Growth, Color Change, and Fall

By making in situ observations of leaf life cycle events for individual trees, we were able to determine which phenology events corresponded to transitions identified from the color index (G_{CC} or R_{CC}) analysis of images. Similar to Berra et al. [22], who also compared ROIs of individual trees in drone imagery to ground observations, and Ahrends et al. [9], who used individual tree ROIs from a tower-mounted camera, we found that observed budburst dates corresponded closely to the beginning of the increase in color indices in spring time, i.e., the SOS transition. But, while previous studies only focused on budburst, we also examined the relation of image metrics to in situ observations of leaf expansion, finding statistically significant correlations with this process as well. MOS and EOS were closest in time to 40% and 70% leaf expansion, respectively. Similarly, we found a higher correlation between springtime color indices and the leaf expansion process, than the progression of budburst. This indicates that while the initial increase in greenness (or redness) is a good indicator of the start of budburst, subsequent increases in index values are more representative of the leaf expansion process, as opposed to the completion of budburst.

We found that both G_{CC} and R_{CC} metrics were significantly correlated with leaf color change in autumn. These results varied somewhat by species as suggested by others [36], although we found that both G_{CC} and R_{CC} work reasonably well for all the species we observed. Red maples, which may display a variety of autumn colors from pale yellow to intense scarlet, had leaf color progressions that were more highly correlated to an increase in fall R_{CC} (species average $r = 0.93$) than a decrease in G_{CC} (-0.84). However, leaf color change in red oaks, which turn a yellow-orange color (Figure 2K), was similarly correlated to both G_{CC} and R_{CC} (species averages $r = -0.96, 0.95$, respectively). In terms of

phenophase transition dates, we did not observe species-specific biases between in situ observations and dates calculated from image analysis (Figure 3).

Our results suggest that color indices may be used to describe the trajectories of both leaf color change and leaf fall in autumn, as well as key transition dates associated with these processes. Phenophase transitions calculated from G_{CC} (SOF, MOF, EOF) were highly correlated to a range of points in the progress of leaf coloration (20%, 40%, and 80% leaf coloration, respectively) across individual trees. The EOF metric determined from maximum fall R_{CC} was closest to the beginning of leaf fall, indicating that the greatest "redness" of trees occurs just as leaves are starting to abscise, although the redness of individual leaves may continue to increase after this [37]. Finally, a decline in R_{CC} after the autumn maximum was significantly correlated with the trajectory of leaf fall.

4.1.2. Plant Area Index

Measures of the plant area index from an analysis of upward photos reinforce the results of the comparison of in situ observations with drone image metrics. We found significant correlations of PAI values with drone-derived G_{CC} in spring. This agrees with our result that in springtime, G_{CC} from drone images is associated with leaf expansion, which increases PAI. In general, the PAI time series and G_{CC} time series have similar trajectories in springtime (Figures 4 and 5). Previous research using G_{CC} from tower-mounted cameras showed that the oblique angle of the camera field of view caused an early bias in late spring phenophase transitions relative to leaf area index (LAI, closely related to PAI) measured by the LAI-2000 instrument [19]. However, the nadir perspective of drone orthophotos appears to alleviate this effect; we found evidence for a strong linear relationship between springtime G_{CC} and PAI (Figure 5A).

We also found that the decrease in R_{CC} after the autumn maximum was significantly linearly correlated to the decrease in autumn PAI. This agrees with the earlier result that leaf fall, which decreases PAI, is correlated with R_{CC} at the individual tree level. Overall, our comparisons with in situ observations demonstrate the utility of drone image analysis for a thorough description of spring and autumn leaf phenology, including budburst, leaf expansion, senescence, and abscission.

4.2. Leaf Color and Color Indices: Red Spring Leaves

We observed a subpopulation of red oak trees that expressed reddish colors in young leaves. When isolating these trees for analysis, we found that a greenness index typically used in spring phenology studies of digital camera images (G_{CC}) was not an effective characterization of leaf expansion phenology; however, a redness index was (R_{CC}, Figure 2). Subsequently, we chose to use a combination of R_{CC} to describe the few trees or grid cells with spring amplitude in R_{CC} greater than 40% of spring amplitude in G_{CC}, and G_{CC} for all other ROIs. While this criterion was useful in our identification of trees and grid cells with red spring leaves, we do not recommend it as a general rule since digital image color index values may vary by camera according to different image sensor characteristics [25], and different trees may have varying degrees of springtime redness. Researchers who observe red spring leaves should identify an appropriate criterion for their analysis.

We examined other options for color indices, including GRVI [38], ExG [17], and Hue [39]; however, these all had similar trajectories to G_{CC}, which failed to describe leaf expansion phenology for trees with red spring leaves (Figure S2). We also calculated an index as the sum of G_{CC} and R_{CC} (equivalent to 1 minus blue chromatic coordinate, defined similarly to R_{CC} and G_{CC} [11]), in an attempt to combine the relevant information for trees with green and red spring leaves into a unified approach. However, because of the relatively large amount of noise in this index, it was less correlated to in situ observations of leaf expansion, and was less reliable in the 10 m grid cell analysis (caption to Figure S2).

While we found utility in using R_{CC} to quantify the phenology of trees with red spring leaves, to our knowledge, red leaves have not presented an issue for the many recent phenology studies using greenness indices to study spring phenology at Harvard Forest and elsewhere, e.g., [9,10,19,25]. This is presumably because of the relatively small number of trees with red spring leaves in relation to those with green spring leaves (i.e., only two of the 30 trees we observed in situ), and the fact that most phenology studies using digital images typically use regions of interest that integrate several tree crowns over the canopy. Indeed, we did not experience any redness-related issues with springtime G_{CC} for the larger ROIs we used to integrate over several trees for comparison to PAI measurements (533 m^2 compared to individual tree ROIs, which were 31 m^2 on average).

However, individual tree ROIs have recently been increasingly examined due to the advent of new methods in photographic studies of phenology. In crowd-sourcing [36], the time-intensive task of drawing regions of interest around many individual trees is alleviated by distributing it among many citizen scientists. In photogrammetry studies using drone imagery [22], larger study areas and an undistorted nadir view of the canopy facilitate the delineation of more individual tree crowns in imagery. As greater numbers of individual trees are analyzed, our results indicate that alternatives to greenness indices, such as the redness index we used here, should prove useful in identifying and analyzing trees with red spring leaves. We also note that besides color indices, texture analysis may yield useful approaches to phenological studies of forest canopies using digital images, as suggested in a recent exploratory study [40]. Texture analysis presents the possibility of color-independent image metrics, which may be applicable across trees with different leaf colors.

Red leaf color is more typically associated with autumn than spring in temperate forests, and is known to be associated with increased concentrations of anthocyanin pigments. However, a study on tropical trees indicated that young, immature leaves were more likely than senescent leaves to express a red color and increased anthocyanin concentrations, across different species [41]. Recent research indicated that, across diverse climates including temperate forests, young leaves of many tree and shrub species have a reddish appearance [42], including red oaks [43]. The physiological function of anthocyanins has not been settled; they have been hypothesized to serve a photoprotective role, as senescent leaves resorb nutrients [44], and may also benefit young, developing leaves in this way [41]. However, we observed differences in spring leaf color in conspecific individuals in close physical proximity and with a similar canopy position (Figure 2), suggesting that light microclimate may not be a determining factor. Our results indicate that drone photography would be a useful method for the further study of intraspecific differences in springtime leaf color.

4.3. Spatial Variance in Phenology and Relation to Microclimate

The effect of spatial temperature variation on phenology is most commonly explored over altitudinal gradients in mountainous terrain, in "space for time" studies that seek to use these gradients as a proxy for global change [45], or across large geographic distances [18,46]. However, temperature variation may also act on phenology over smaller spatial scales, due to microclimates induced by topography or canopy cover within forests [47,48]. We explored the effect of microclimate by first accounting for species composition effects, as variability in phenology among tree species is well known [26,49], and examining the relation of the residual phenological variance with air temperature.

Although we did not detect an effect of microsite temperature on spring phenology, we found evidence that warmer daily minimum temperatures in September delay senescence by five days per °C. These results agree with the chilling degree days hypothesis, which states that senescence is triggered when accumulated cold temperatures reach a critical threshold in autumn: colder temperatures advance senescence locally [31,46]. We note that microclimates with lower daily minimum temperatures were located along a forest edge, near a wetland to the northwest of the study area (Figure 1). This result highlights the importance of considering forest structure effects on microclimate and phenology [48], in addition to factors such as canopy position and tree developmental stage [26] when assessing the causes of phenological differences between trees.

5. Conclusions

We presented the first synthesis of in situ observations of tree phenology over a complete growing season with an analysis of drone images of the same trees. Using digital image processing techniques, we found that greenness metrics integrating several trees were correlated with canopy structural development (PAI) in springtime, while redness metrics were correlated with PAI decrease in autumn. At the individual tree level, we determined that the onset of rising greenness in spring corresponded with the start of budburst, while continued increases in greenness were driven by leaf expansion. However, we documented intraspecific variation in spring leaf color, noting that some oak trees displayed reddish (orange, pink) leaves during leaf-out while other trees had green leaves. We found that this affected the color index (greenness versus redness) that could be used to describe the leaf expansion phenology of individual trees. In autumn, we found that the decrease of canopy greenness, as well as the increase of redness, were correlated with the percentage of leaves that had changed color on trees. The time of greatest canopy redness corresponded to the beginning of abscission, and the trajectory of leaf fall correlated with decreasing redness. These results leverage the novel method of drone photogrammetry to advance our understanding of how digital image metrics relate to foliar phenology and canopy structural development throughout the leaf life cycle, from budburst to abscission.

Supplementary Materials: The following are available online at http://www.mdpi.com/1424-8220/17/12/2852/s1.

Acknowledgments: S.K. was supported by NASA Headquarters under the NASA Earth and Space Science Fellowship Program—Grant 14-EARTH14R-23. Additional support was received from the National Science Foundation under the Harvard Forest Long-Term Ecological Research Program (NSF DEB-1237491) and the Harvard Center for Geographic Analysis. We acknowledge John O'Keefe of Harvard Forest for helpful discussions on this research, and three anonymous reviewers for their comments, which resulted in many improvements to the manuscript.

Author Contributions: S.K. and A.D.R. conceived and designed the experiments. S.K. performed the experiments and analyzed the data. S.K. wrote the paper with input from A.D.R.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lieth, H. Phenology in productivity studies. In *Analysis of Temperate Forest Ecosystems*; Reichle, D.E., Ed.; Springer: Berlin/Heidelberg, Germany, 1973; pp. 29–46.
2. Fu, Y.H.; Zhao, H.; Piao, S.; Peaucelle, M.; Peng, S.; Zhou, G.; Ciais, P.; Huang, M.; Menzel, A.; Peñuelas, J.; et al. Declining global warming effects on the phenology of spring leaf unfolding. *Nature* **2015**, *526*, 104–107. [CrossRef]
3. Vitasse, Y.; Bresson, C.C.; Kremer, A.; Michalet, R.; Delzon, S. Quantifying phenological plasticity to temperature in two temperate tree species. *Funct. Ecol.* **2010**, *24*, 1211–1218. [CrossRef]
4. Keenan, T.F.; Gray, J.; Friedl, M.A.; Toomey, M.; Bohrer, G.; Hollinger, D.Y.; Munger, J.W.; O'Keefe, J.; Schmid, H.P.; Wing, I.S.; et al. Net carbon uptake has increased through warming-induced changes in temperate forest phenology. *Nat. Clim. Chang.* **2014**, *4*, 598–604. [CrossRef]
5. Westergaard-Nielsen, A.; Lund, M.; Hansen, B.U.; Tamstorf, M.P. Camera derived vegetation greenness index as proxy for gross primary production in a low Arctic wetland area. *ISPRS J. Photogramm. Remote Sens.* **2013**, *86*, 89–99. [CrossRef]
6. Toomey, M.; Friedl, M.A.; Frolking, S.; Hufkens, K.; Klosterman, S.; Sonnentag, O.; Baldocchi, D.D.; Bernacchi, C.J.; Biraud, S.C.; Bohrer, G.; et al. Greenness indices from digital cameras predict the timing and seasonal dynamics of canopy-scale photosynthesis. *Ecol. Appl.* **2015**, *25*, 99–115. [CrossRef]
7. Fitzjarrald, D.; Acevedo, O. Climatic consequences of leaf presence in the eastern United States. *J. Clim.* **2001**, *14*, 598–614. [CrossRef]

8. Richardson, A.D.; Keenan, T.F.; Migliavacca, M.; Ryu, Y.; Sonnentag, O.; Toomey, M. Climate change, phenology, and phenological control of vegetation feedbacks to the climate system. *Agric. For. Meteorol.* **2013**, *169*, 156–173. [CrossRef]
9. Ahrends, H.E.; Bruegger, R.; Stoeckli, R.; Schenk, J.; Michna, P.; Jeanneret, F.; Wanner, H.; Eugster, W. Quantitative phenological observations of a mixed beech forest in northern Switzerland with digital photography. *J. Geophys. Res.* **2008**, *113*. [CrossRef]
10. Richardson, A.D.; Jenkins, J.P.; Braswell, B.H.; Hollinger, D.Y.; Ollinger, S.V.; Smith, M.-L. Use of digital webcam images to track spring green-up in a deciduous broadleaf forest. *Oecologia* **2007**, *152*, 323–334. [CrossRef]
11. Richardson, A.D.; Braswell, B.H.; Hollinger, D.Y.; Jenkins, J.P.; Ollinger, S.V. Near-surface remote sensing of spatial and temporal variation in canopy phenology. *Ecol. Appl.* **2009**, *19*, 1417–1428. [CrossRef]
12. Reed, B.C.; Schwartz, M.D.; Xiao, X. Remote sensing phenology. In *Phenology of Ecosystem Processes*; Springer: New York, NY, USA, 2009; pp. 231–246.
13. Garrity, S.R.; Bohrer, G.; Maurer, K.D.; Mueller, K.L.; Vogel, C.S.; Curtis, P.S. A comparison of multiple phenology data sources for estimating seasonal transitions in deciduous forest carbon exchange. *Agric. For. Meteorol.* **2011**, *151*, 1741–1752. [CrossRef]
14. Lange, M.; Dechant, B.; Rebmann, C.; Vohland, M.; Cuntz, M.; Doktor, D. Validating MODIS and sentinel-2 NDVI products at a temperate deciduous forest site using two independent ground-based sensors. *Sensors* **2017**, *17*, 1855. [CrossRef]
15. Zhang, X.; Friedl, M.A.; Schaaf, C.B.; Strahler, A.H.; Hodges, J.C.F.; Gao, F.; Reed, B.C.; Huete, A. Monitoring vegetation phenology using MODIS. *Remote Sens. Environ.* **2003**, *84*, 471–475. [CrossRef]
16. Elmore, A.J.; Guinn, S.M.; Minsley, B.J.; Richardson, A.D. Landscape controls on the timing of spring, autumn, and growing season length in mid-Atlantic forests. *Glob. Chang. Biol.* **2012**, *18*, 656–674. [CrossRef]
17. Hufkens, K.; Friedl, M.; Sonnentag, O.; Braswell, B.H.; Milliman, T.; Richardson, A.D. Linking near-surface and satellite remote sensing measurements of deciduous broadleaf forest phenology. *Remote Sens. Environ.* **2012**, *117*, 307–321. [CrossRef]
18. Klosterman, S.T.; Hufkens, K.; Gray, J.M.; Melaas, E.; Sonnentag, O.; Lavine, I.; Mitchell, L.; Norman, R.; Friedl, M.A.; Richardson, A.D. Evaluating remote sensing of deciduous forest phenology at multiple spatial scales using PhenoCam imagery. *Biogeosciences* **2014**, *11*, 4305–4320. [CrossRef]
19. Keenan, T.F.; Darby, B.; Felts, E.; Sonnentag, O.; Friedl, M.A.; Hufkens, K.; O'Keefe, J.; Klosterman, S.; Munger, J.W.; Toomey, M.; et al. Tracking forest phenology and seasonal physiology using digital repeat photography: A critical assessment. *Ecol. Appl.* **2014**, *24*, 1478–1489. [CrossRef]
20. Dandois, J.P.; Ellis, E.C. High spatial resolution three-dimensional mapping of vegetation spectral dynamics using computer vision. *Remote Sens. Environ.* **2013**, *136*, 259–276. [CrossRef]
21. Lisein, J.; Michez, A.; Claessens, H.; Lejeune, P. Discrimination of deciduous tree species from time series of unmanned aerial system imagery. *PLoS ONE* **2015**, *10*, e0141006. [CrossRef]
22. Berra, E.F.; Gaulton, R.; Barr, S. Use of a digital camera onboard a UAV to monitor spring phenology at individual tree level. In Proceedings of the 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Beijing, China, 10–15 July 2016; pp. 3496–3499.
23. Klosterman, S.; Melaas, E.; Wang, J.; Martinez, A.; Frederick, S.; O'Keefe, J.; Orwig, D.A.; Wang, Z.; Sun, Q.; Schaaf, C.; et al. Fine-scale perspectives on landscape phenology from unmanned aerial vehicle (UAV) photography. *Agric. For. Meteorol.* **2018**, *248*, 397–407. [CrossRef]
24. Klosterman, S.; Richardson, A.D. Landscape Phenology from Unmanned Aerial Vehicle Photography at Harvard Forest since 2013. Harvard Forest Data Archive: HF294. Available online: http://harvardforest.fas.harvard.edu:8080/exist/apps/datasets/showData.html?id=hf294 (accessed on 1 December 2017).
25. Sonnentag, O.; Hufkens, K.; Teshera-Sterne, C.; Young, A.M.; Friedl, M.; Braswell, B.H.; Milliman, T.; O'Keefe, J.; Richardson, A.D. Digital repeat photography for phenological research in forest ecosystems. *Agric. For. Meteorol.* **2012**, *152*, 159–177. [CrossRef]
26. Richardson, A.D.; O'Keefe, J. Phenological differences between understory and overstory. In *Phenology of Ecosystem Processes*; Noormets, A., Ed.; Springer: New York, NY, USA, 2009; pp. 87–117.

27. Andrews, D.W.K. Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica* **1991**, *59*, 817–858. [CrossRef]
28. Andrews, D.W.K.; Monahan, J.C. An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica* **1992**, *60*, 953–966. [CrossRef]
29. Macfarlane, C.; Hoffman, M.; Eamus, D.; Kerp, N.; Higginson, S.; Mcmurtrie, R.; Adams, M. Estimation of leaf area index in eucalypt forest using digital photography. *Agric. For. Meteorol.* **2007**, *143*, 176–188. [CrossRef]
30. Macfarlane, C. Classification method of mixed pixels does not affect canopy metrics from digital images of forest overstorey. *Agric. For. Meteorol.* **2011**, *151*, 833–840. [CrossRef]
31. Richardson, A.D.; Bailey, A.S.; Denny, E.G.; Martin, C.W.; O'Keefe, J. Phenology of a northern hardwood forest canopy. *Glob. Chang. Biol.* **2006**, *12*, 1174–1188. [CrossRef]
32. Vitasse, Y.; François, C.; Delpierre, N.; Dufrêne, E.; Kremer, A.; Chuine, I.; Delzon, S. Assessing the effects of climate change on the phenology of European temperate trees. *Agric. For. Meteorol.* **2011**, *151*, 969–980. [CrossRef]
33. Friedl, M.A.; Gray, J.M.; Melaas, E.K.; Richardson, A.D.; Hufkens, K.; Keenan, T.F.; Bailey, A.; O'Keefe, J. A tale of two springs: Using recent climate anomalies to characterize the sensitivity of temperate forest phenology to climate change. *Environ. Res. Lett.* **2014**, *9*. [CrossRef]
34. Orwig, D.; Foster, D.; Ellison, A. Harvard Forest CTFS-ForestGEO Mapped Forest Plot since 2014. Available online: http://harvardforest.fas.harvard.edu:8080/exist/apps/datasets/showData.html?id=hf253 (accessed on 25 August 2017).
35. Draper, N.R.; Smith, H. *Applied Regression Analysis*, 3rd ed.; John Wiley & Sons: New York, NY, USA, 1998.
36. Kosmala, M.; Crall, A.; Cheng, R.; Hufkens, K.; Henderson, S.; Richardson, A. Season Spotter: Using citizen science to validate and scale plant phenology from near-surface remote sensing. *Remote Sens.* **2016**, *8*, 726. [CrossRef]
37. Archetti, M.; Richardson, A.D.; O'Keefe, J.; Delpierre, N. Predicting climate change impacts on the amount and duration of autumn colors in a new england forest. *PLoS ONE* **2013**, *8*, e57373. [CrossRef]
38. Motohka, T.; Nasahara, K.N.; Oguma, H.; Tsuchida, S. Applicability of green-red vegetation index for remote sensing of vegetation phenology. *Remote Sens.* **2010**, *2*, 2369–2387. [CrossRef]
39. Nagai, S.; Maeda, T.; Gamo, M.; Muraoka, H.; Suzuki, R.; Nasahara, K.N. Using digital camera images to detect canopy condition of deciduous broad-leaved trees. *Plant Ecol. Divers.* **2011**, *4*, 79–89. [CrossRef]
40. Almeida, J.; dos Santos, J.A.; Alberton, B.; Torres, R.S.; Morellato, L.P.C. Applying machine learning based on multiscale classifiers to detect remote phenology patterns in Cerrado savanna trees. *Ecol. Inform.* **2014**, *23*, 49–61. [CrossRef]
41. Lee, D.W.; Collins, T.M. Phylogenetic and ontogenetic influences on the distribution of anthocyanins and betacyanins in leaves of tropical plants. *Int. J. Plant Sci.* **2001**, *162*, 1141–1153. [CrossRef]
42. Lev-Yadun, S.; Yamazaki, K.; Holopainen, J.K.; Sinkkonen, A. Spring versus autumn leaf colours: Evidence for different selective agents and evolution in various species and floras. *Flora-Morphol. Distrib. Funct. Ecol. Plants* **2012**, *207*, 80–85. [CrossRef]
43. Key, T.; Warner, T.; McGraw, J.; Fajvan, M. A comparison of multispectral and multitemporal information in high spatial resolution imagery for classification of individual tree species in a temperate hardwood forest. *Remote Sens. Environ.* **2001**, *75*, 100–112. [CrossRef]
44. Lee, D.W.; O'Keefe, J.; Holbrook, N.M.; Feild, T.S. Pigment dynamics and autumn leaf senescence in a New England deciduous forest, eastern USA. *Ecol. Res.* **2003**, *18*, 677–694. [CrossRef]
45. Vitasse, Y.; Porté, A.J.; Kremer, A.; Michalet, R.; Delzon, S. Responses of canopy duration to temperature changes in four temperate tree species: Relative contributions of spring and autumn leaf phenology. *Oecologia* **2009**, *161*, 187–198. [CrossRef]
46. Dragoni, D.; Rahman, A.F. Trends in fall phenology across the deciduous forests of the Eastern USA. *Agric. For. Meteorol.* **2012**, *157*, 96–105. [CrossRef]
47. Fisher, J.; Mustard, J.; Vadeboncoeur, M. Green leaf phenology at Landsat resolution: Scaling from the field to the satellite. *Remote Sens. Environ.* **2006**, *100*, 265–279. [CrossRef]

48. Motzkin, G.; Ciccarello, S.C.; Foster, D.R. Frost pockets on a level sand plain: Does variation in microclimate help maintain persistent vegetation patterns? *J. Torrey Bot. Soc.* **2002**, *129*, 154–163. [CrossRef]
49. Lechowicz, M.J. Why do temperate deciduous trees leaf out at different times? Adaptation and ecology of forest communities. *Am. Nat.* **1984**, *124*, 821–842. [CrossRef]

sensors

MDPI

Article

Localization Framework for Real-Time UAV Autonomous Landing: An On-Ground Deployed Visual Approach

Weiwei Kong [1,2], Tianjiang Hu [1,*], Daibing Zhang [1], Lincheng Shen [1] and Jianwei Zhang [3]

[1] College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, China; kongww.nudt@gmail.com (W.K.); swimsword@163.com (D.Z.); lcshen@nudt.edu.cn (L.S.)
[2] Naval Academy of Armament, Beijing 100161, China
[3] Institute of Technical Aspects of Multimodal Systems (TAMS), Department of Computer Science, University of Hamburg, 22527 Hamburg, Germany; zhang@informatik.uni-hamburg.de
* Correspondence: t.j.hu@nudt.edu.cn

Academic Editors: Felipe Gonzalez Toro and Antonios Tsourdos
Received: 31 March 2017; Accepted: 9 June 2017; Published: 19 June 2017

Abstract: One of the greatest challenges for fixed-wing unmanned aircraft vehicles (UAVs) is safe landing. Hereafter, an on-ground deployed visual approach is developed in this paper. This approach is definitely suitable for landing within the global navigation satellite system (GNSS)-denied environments. As for applications, the deployed guidance system makes full use of the ground computing resource and feedbacks the aircraft's real-time localization to its on-board autopilot. Under such circumstances, a separate long baseline stereo architecture is proposed to possess an extendable baseline and wide-angle field of view (FOV) against the traditional fixed baseline schemes. Furthermore, accuracy evaluation of the new type of architecture is conducted by theoretical modeling and computational analysis. Dataset-driven experimental results demonstrate the feasibility and effectiveness of the developed approach.

Keywords: UAV; stereo vision; localization

1. Introduction

Over the past few decades, the application of unmanned aircraft has increased enormously in both civil and military scenarios. Although aerial robots have successfully been implemented in several applications, there are still new research directions related to them. Floreano [1] and Kumar et al. [2] outlined the opportunities and challenges of this developing field, from the model design to high-level perception capability. All of these issues are concentrating on improving the degree of autonomy, which supports that UAVs continue to be used in novel and surprising ways. No matter whether fixed-wing or rotor-way platforms, a standard fully-unmanned autonomous system (UAS) involves performs takeoffs, waypoint flight and landings. Among them, the landing maneuver is the most delicate and critical phase of UAV flights. Two technical reports [3] argued that nearly 70% of mishaps of Pioneer UAVs were encountered during the landing process caused by human factors. Therefore, a proper assist system is needed to enhance the reliability of the landing task. Generally, two main capabilities of the system are required. The first one is localization and navigation of UAVs, and the second one is generating the appropriate guidance command to guide UAVs for a safe landing.

For manned aircraft, the traditional landing system uses a radio beam directed upward from the ground [4,5]. By measuring the angular deviation from the beam through onboard equipment, the pilot knows the perpendicular displacement of the aircraft in the vertical channel. For the azimuth

information, additional equipment is required. However, due to the size, weight and power (SWaP) constraints, it is impossible to equip these instruments in UAV. Thanks to the GNSS technology, we have seen many successful practical applications of autonomous UAVs in outdoor environments such as transportation, aerial photography and intelligent farming. Unfortunately, in some circumstances, such as urban or low altitude operations, the GNSS receiver antenna is prone to lose line-of-sight with satellites, making GNSS unable to deliver high quality position information [6]. Therefore, autonomous landing in an unknown or global navigation satellite system (GNSS)-denied environment is still an open problem.

The visual-based approach is an obvious way to achieve the autonomous landing by estimating flight speed and distance to the landing area, in a moment-to-moment fashion. Generally, two types of visual methods can be considered. The first category is the vision-based onboard system, which has been widely studied. The other is to guide the aircraft using a ground-based camera system. Once the aircraft is detected by the camera during the landing process, its characteristics, such as type, location, heading and velocity, can be derived by the guidance system. Based on this information, the UAV could align itself carefully towards the landing area and adapt its velocity and acceleration to achieve safe landing. In summary, two key elements of the landing problem are detecting the UAV and its motion, calculating the location of the UAV relative to the landing field.

To achieve better performance in GNSS-denied environments, some other types of sensors, such as laser range finders, millimeter wavelength radar, have been explored for UAV autonomous landing. Swiss company RUAG (Bern, Switzerland) solved the landing task by the OPATS (object position and tracking sensor) [7]. Figure 1a presents this laser-based automatic landing system, the infrared laser beam of which is echoed back from a passive and optionally heated retro reflector on the aircraft. This system could measure the position of approaching aircraft around 4000 m. Moreover, the Sierra Nevada Corporation provides an alternative to the laser-based method. They developed the UAS common automatic recovery system (UCARS) [8] based on millimeter wavelength ground radar for MQ-8B Fire Scout autonomous landing, as shown in Figure 1b. Benefiting from the short bandwidth, UCARS provides precision approach (within 2.5 cm) in adverse weather condition. While those solutions are effective, they require the use of radar or laser emissions, which can be undesirable in a tactical situation. Furthermore, the limited payload of a small UAV constrains the onboard modules.

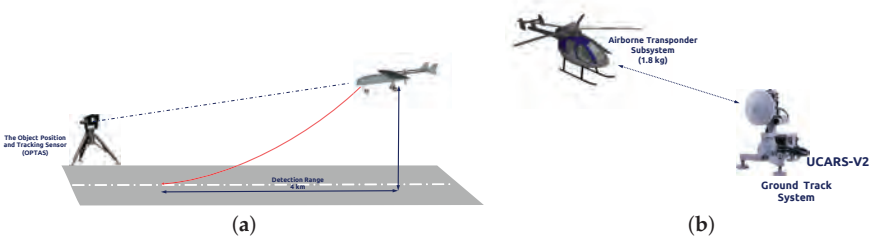


Figure 1. (**a**) The object position and tracking sensor (OPTAS) system is mounted on the tripod. Adapted from [7]; (**b**) UAS common automatic recovery system (UCARS) for MQ-8B autonomous landing. Adapted from [8].

Motivated by these mentioned challenges, we propose and develop a novel on-ground deployment of the visual landing system. In this paper, we mainly focus on the localization and navigation issue and try to improve the navigation accuracy and robustness. The essential contributions of this work are as follows: (1) an extendable baseline and wide-angle field of view (FOV) vision guidance system is developed by using a physically-separated and informationally-connected deployment of the two PTUs on both sides of the runway; (2) localization error and its transferring mechanism in practical situations are unveiled with both theoretical and computational analyses.

In particular, the developed approach is experimentally validated with fair accuracy and better performance in timeliness, as well as practicality against the previous works.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related works. In Section 3, the architecture of the on-ground deployed stereo system is proposed and designed. Section 4 conducts the accuracy evaluation, and its transferring mechanism is conducted through theoretical and computational analysis. Dataset-driven validation is followed in Section 5. Finally, concluding remarks are presented in Section 6.

2. Related Works

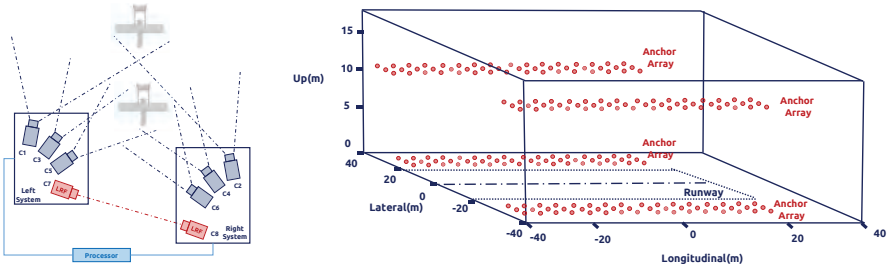
While several techniques have been applied for onboard vision-based control of UAVs, few have shown landing of a fixed-wing guiding by a ground-based system. In 2006, Wang [9] proposed a system using a step motor controlling a web camera to track and guide a micro-aircraft. This camera rotation platform expands the recognition area from 60 cm $\times$ 60 cm–140 cm $\times$ 140 cm, but the range of the recognition is only 1 m. This configuration cannot be used to determine the position of a fixed-wing in the field.

At Chiba University [10], a ground-based Bumblebee stereo vision system was used to calculate the 3D position of a quadrotor at the altitude of 6 m. The Bumblebee has a 15.7-cm baseline with a 66° horizontal field of view. The sensor was mounted on a tripod with the height of 45 cm, and the drawback of this system is the limited baseline leading to a narrow field of view (FOV).

To increase the camera FOV, multi-camera systems are considered attractive. This kind of system could solve the common vision problems and track objects to compute their 3D locations. In addition, Martinez [11] introduced a trinocular on-ground system, which is composed of three or more cameras for extracting key features of the UAV to obtain robust 3D position estimation. The lenses of the FireWire cameras are 3.4 mm and capture images of a 320 $\times$ 240 size at 30 fps. They employed the continuously-adaptive mean shift (CamShift) algorithm to track the four cooperation markers with independent color, which were distributed on the bottom of the helicopter. The precision of this system in the vertical and horizontal direction is around 5 cm and in depth estimation is 10 cm with a 3-m recognition range. The maximum range for depth estimation is still not sufficient for fixed-wing UAV. Additionally, another drawback of the multi-camera system is the calibration process, whose parameters are nontrivial to obtain.

A state-of-the-art study from Guan et al. [12] proposed a multi-camera network with laser rangefinders to estimate an aircraft's motion. This system is composed of two sets of measurement units that are installed on both sides of the runway. Each unit has three high-speed cameras with different focal lengths and FOV to captures the target in the near-filed (20 m–100 m), middle-field (100 m–500 m) and far-field (500 m–1000 m), respectively. A series of field experiments shows that the RMS error of the distance is 1.32 m. Due to the configuration of the system, they have to apply a octocopter UAV equipped with a prism to calibrate the whole measurement system.

Except the camera-based ground navigation system, the ultra-wide band (UWB) positioning network is also discussed in the community. Kim and Choi [13] deployed the passive UWB anchors by the runway, which listen for the UWB signals emitted from the UAV. The ground system computes the position of the target based on the geometry of the UWB anchors and sends it back to the UAV through the aviation communication channel. There are a total of 240 anchor possible locations, as shown in Figure 2b, distributed at each side of the runway, and the longitudinal range is up to 300 m with a positioning accuracy of 40 cm.



(**a**) (**b**)

Figure 2. (**a**) Architecture of the multi-camera network. (**b**) In the longitudinal direction, there are 20 anchors separated by 3 m. In the vertical direction, the anchors are either located on the ground or at the 10-m height antennas. The red points show the 240 possible anchor locations.

Our group first developed the traditional stereo ground-based system with infrared cameras [14], while this system has limited detection distance. For short-baseline configuration, cameras were setup on one PTU, and the system should be mounted on the center line of the runway. However, the short-baseline limits the maximum range for UAV depth estimation. To enhance the operating capability, we conducted the triangular geometry localization method for the PTU-based system [15]. As shown in Figure 3, we fixed the cameras with separate PTUs on the both sides of the runway. Therefore, the landing aircraft can be locked by our system around 1 km. According to the previous work, the localization accuracy largely depends on the aircraft detection precision in the camera image plane. Therefore, we implemented the Chan–Vese method [16] and the saliency-inspired method [17] to detect and track the vehicle more accurately; however, these approaches are not suitable for real-time requirements.

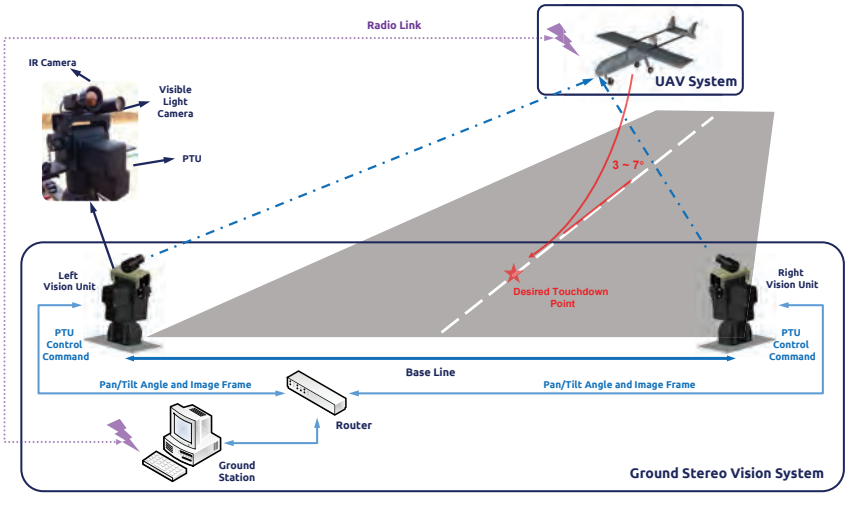


Figure 3. Architecture of the ground stereo vision system.

For more information, we also reviewed various vision-based landing approaches performed on different platforms [18], and Gautam provides another general review of the autonomous landing techniques for UAVs [19].

3. System Architecture and Deployment

In this section, we introduce the theoretical model for the ground-to-air visual system. We first recap the traditional stereo vision model, which has a limited baseline, restraining the detection distance. To enlarge the system working boundary, we setup the camera and other sensor modules on the two separated PTUs and then calculate the target according to the image information and rotation angle from PTU. Each vision unit works independently and transfers the results of image processing and PTU status to the navigation computer, which calculates the estimated relative position of the UAV. The architecture of the ground stereo vision system is shown in Figure 3.

3.1. Fundamental Principles of Ground-Based Stereo Systems

The standard camera model is a pin-hole camera model. The coordinate of the target M is (x, y, z), and its position on the image plane is (u, v). The camera focus is f; then, the relationship of the coordinate between the 3D world and 2D image plane can be calculated by:

$$\lambda \begin{bmatrix} u \\ v \\ f \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{1}$$

where λ is the scale factor.

Although the above model is simple, it could be helpful to estimate the theoretical camera lens according to the expected distance and resolution or to measure the target size roughly based on the pixel length on the image plane. Let the width and height of the target be W and H; the distance between the camera and target be L; the target projection on image plane be w and h; the relationship between them is:

$$f = \frac{wL}{W} \qquad f = \frac{hL}{H} \tag{2}$$

We define the coordinates of the left and right navigation module as shown in Figure 4b. When the optical axes of these two cameras are parallel, we could calculate the target in 3D space by:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{b}{d} \begin{bmatrix} u_l \\ u_r \\ f \end{bmatrix} \tag{3}$$

where b is the baseline and $d = u_l - u_r$ is the pixel disparity, as shown in Figure 4a. Even though some calibration methods could manage the axes' nonparallel situation, it is still difficult to calculate the system correctly, as the baseline is large.

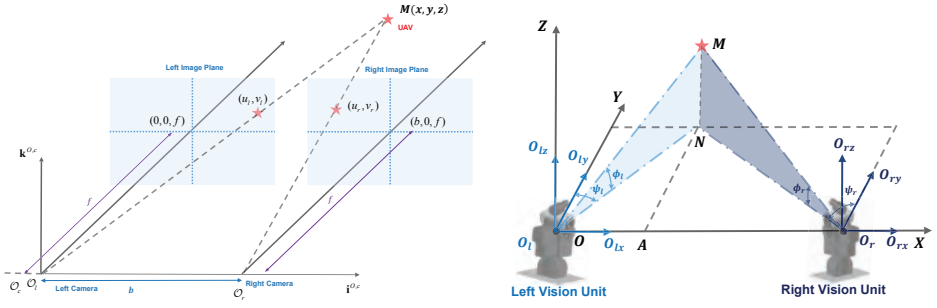


Figure 4. (**a**) Theoretical stereo vision model; (**b**) theoretical optical model. The target M is projected at the center of the optical axis.

3.2. Separated Long Baseline Deployment

In order to detect the target at long distance, a large baseline, more than 5 m, is required. Benefiting the camera assembled on the PTU separately, we could switch the baseline freely according to the expected detection distance and target size.

In this paper, we assumed that the world coordinate system (X, Y, Z) is located on the origin of the left vision unit, the rotation center of the PTU. For the sake of simplicity, the camera is installed on the PTU in the way that the axes of the camera frame are parallel to those of the PTU frame. The origins of these two frames are close. Therefore, it can be assumed that the camera frame coincides with the body frame. Figure 4b reveals the theoretical model for visual measurement. After installing the right camera system on the X-axis, the left and right optical center can be expressed as O_l and O_r, respectively. Then, the baseline of the optical system is O_lO_r, whose distance is D. Considering the center of mass of the UAV as a point M, O_lM and O_rM illustrate the connections between the each optical center and the UAV. In addition, ϕ_l, ϕ_r, ψ_l, ψ_r denote the tilt and pan angle on both sides. Therefore, we define $\phi_l = 0$, $\phi_r = 0$, $\psi_l = 0$ and $\psi_r = 0$, as the PTU is set to the initial state, i.e., the optical axis parallel to the runway; the measurement of the counterclockwise direction is positive.

Since the point M does not coincide with the principle point, which is the center of the image plane, the pixel deviation compensation in the longitudinal and horizontal direction should be considered. As shown in Figure 5, we calculate pixel deviation compensation on the left side by:

$$\begin{cases} \psi_{cl} = \arctan \dfrac{(u - u_0)du}{f} \\ \phi_{cl} = \arctan \dfrac{(v - v_0)\cos\psi_{cl}dv}{f} \end{cases} \tag{4}$$

where the optical point is $o(u_o, v_o)$, du and dv are the pixel length of the u- and v-axis in image plane and f is the focus. The current PTU rotation angle can be directly obtained through the serial ports during the experiments. Let ϕ_{pl} and ψ_{pl} be the left pan and tilt angle separately. Then, the total pan and tilt angle on the left side can be detailed as:

$$\begin{cases} \phi_l = \phi_{cl} + \phi_{pl} \\ \psi_l = \psi_{cr} + \psi_{pr} \end{cases} \tag{5}$$

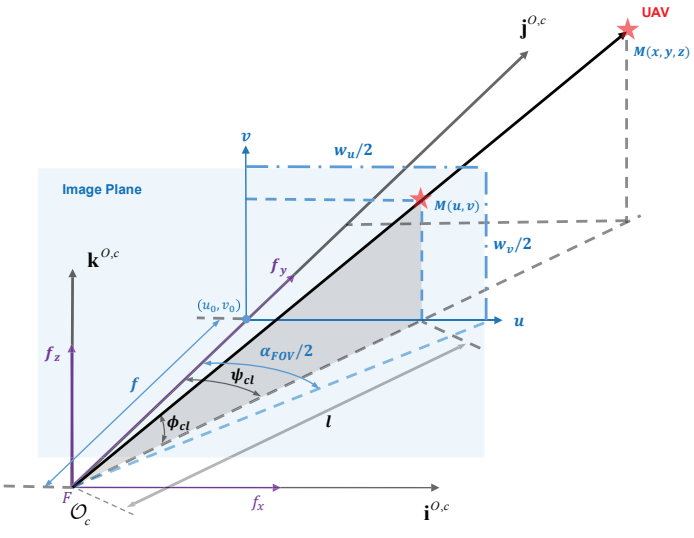


Figure 5. The geometry of one PTUwith respect to the optical center and the image plane.

For the other side, we could also calculate the angle in the same way.

The world coordinates of point M is $(x_M, y_M, z_M) \in \mathbf{R}^3$. Point N is the vertical projection of point M on the XOY plane, and NA is perpendicular to the X-axis. If we define $NA = h$, the following navigation parameters can be obtained:

$$\begin{cases} x_M = h \tan \psi_l = \dfrac{D \tan \psi_l}{\tan \psi_l - \tan \psi_r} \\ y_M = h = \dfrac{D}{\tan \psi_l - \tan \psi_r} \\ z_M = \dfrac{h \tan \phi_l}{\cos \psi_l} = \dfrac{D \tan \phi_l}{\cos \psi_l (\tan \psi_l - \tan \psi_r)} \end{cases} \tag{6}$$

Furthermore, errors in the internal and external camera calibration parameters marginally affect some of the estimates: the x-position and z-position, in particular.

4. Accuracy Evaluation of the On-Ground Stereo System

4.1. Theoretical Modeling

We are now in the position to analyze the error related to the PTU rotation angle. The discussion was first presented in our previous works [15]. According to Equation (6), the partial derivatives of each equation with respect to the pan angle and the tilt angle are denoted in the following way,

$$\begin{cases} \dfrac{\partial x_M}{\partial \psi_l} = \dfrac{D \tan \psi_r}{\cos^2 \psi_l (\tan \psi_l - \tan \psi_r)^2} \\ \dfrac{\partial x_M}{\partial \psi_r} = \dfrac{D \tan \psi_l}{\cos^2 \psi_r (\tan \psi_l - \tan \psi_r)^2} \end{cases} \tag{7}$$

$$\begin{cases} \dfrac{\partial y_M}{\partial \psi_l} = \dfrac{D}{\cos^2 \psi_l (\tan \psi_l - \tan \psi_r)^2} \\ \dfrac{\partial y_M}{\partial \psi_r} = \dfrac{D}{\cos^2 \psi_r (\tan \psi_l - \tan \psi_r)^2} \end{cases} \tag{8}$$

$$\begin{cases} \dfrac{\partial z_M}{\partial \phi_l} = \dfrac{D}{\cos \psi_l \cos^2 \phi_l (\tan \psi_l - \tan \psi_r)} \\ \dfrac{\partial z_M}{\partial \psi_l} = \dfrac{D \tan \phi_l (\cos \psi_l + \sin \psi_l \tan \psi_r)}{\cos^2 \psi_r (\tan \psi_l - \tan \psi_r)^2} \\ \dfrac{\partial z_M}{\partial \psi_r} = \dfrac{D \tan \phi_l}{\cos \psi_l \cos^2 \psi_r (\tan \psi_l - \tan \psi_r)^2} \end{cases} \tag{9}$$

To analyze the influence of the error from the angle, we define the gradient of the world coordinate as:

$$\nabla_{x_M}(\psi_l, \psi_r) := \left(\frac{\partial x_M}{\partial \psi_l}(\psi_l, \psi_r), \frac{\partial x_M}{\partial \psi_r}(\psi_l, \psi_r) \right) \tag{10}$$

$$\nabla_{y_M}(\psi_l, \psi_r) := \left(\frac{\partial y_M}{\partial \psi_l}(\psi_l, \psi_r), \frac{\partial y_M}{\partial \psi_r}(\psi_l, \psi_r) \right) \tag{11}$$

$$\nabla_{z_M}(\psi_l, \psi_r) := \left(\frac{\partial z_M}{\partial \psi_l}(\psi_l, \psi_r), \frac{\partial z_M}{\partial \psi_r}(\psi_l, \psi_r) \right) \tag{12}$$

In this case, simulation is needed to evaluate the behavior of our visual system. Figure 6a–c is the vector field distribution of $\nabla_{x_M}(\psi_l, \psi_r)$, $\nabla_{y_M}(\psi_l, \psi_r)$ and $\nabla_{z_M}(\psi_l, \psi_r)$, which give us an intuitive result under different types of errors. The length of each vector describes the strength at a specific point; the direction along the vector points to the direction of the fastest error increase. However, only when $y_M \geq 0$ (the aircraft is in front of two cameras), the area $\psi_l - \psi_r > 0$ has the physics meaning. Figure 7a shows that x_M has a significant variation when ψ_l is approximate to ψ_r, namely the optical axes are nearly parallel. Further, y_M and z_M have similar variations. Considering the general working status of the ground-based system, we mainly focus on the second quadrant of the aforementioned vector fields as shown in Figure 7a–c. In these areas, there are slight variations that theoretically demonstrate the feasibility of the system.

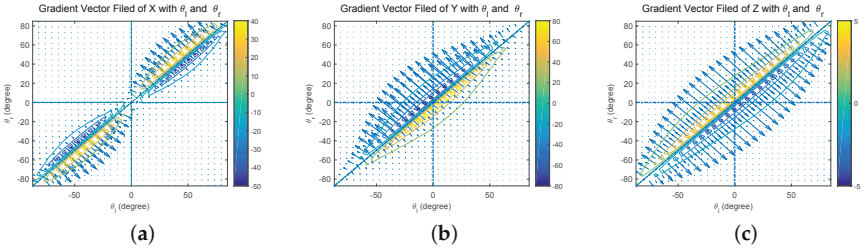


Figure 6. Vector field distribution of (**a**) $\nabla_{x_M}(\psi_l, \psi_r)$, (**b**) $\nabla_{y_M}(\psi_l, \psi_r)$ and (**c**) $\nabla_{z_M}(\psi_l, \psi_r)$.

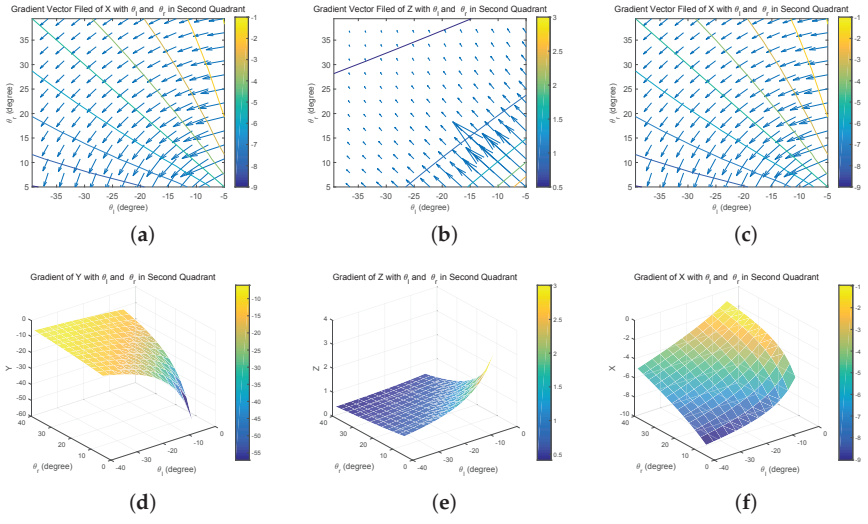


Figure 7. (**a**–**c**) Vector field distribution of $\nabla_{x_M}(\psi_l, \psi_r)$, $\nabla_{y_M}(\psi_l, \psi_r)$, and $\nabla_{z_M}(\psi_l, \psi_r)$ in the second quadrant; (**d**–**f**) gradient of X, Y, Z with θ_l and θ_r in the second quadrant.

4.2. Computational Analysis

In theory, O_lM and O_rM should intersect perfectly at one point all of the time, as shown in Figure 4b. Due to the inevitable errors from PTU rotation and tracking algorithms, we estimate the intersecting point by combing the vertical line of two different planes in space.

(1) We set $(x_{ol}, y_{ol}, z_{ol}) = (0, 0, 0)$, and $(x_{or}, y_{or}, z_{or}) = (D, 0, 0)$ is the optical center of each camera. Assuming that $a_l \neq 0$, $b_l \neq 0$, $c_l \neq 0$ and $a_r \neq 0$, $b_r \neq 0$, $c_r \neq 0$, we obtain the parametric equations of lines O_lM and O_rM:

$$\begin{cases} \dfrac{x - x_{ol}}{a_l} = \dfrac{y - y_{ol}}{b_l} = \dfrac{z - z_{ol}}{c_l} = t_l, \\ \dfrac{x - x_{or}}{a_r} = \dfrac{y - y_{or}}{b_r} = \dfrac{z - z_{or}}{c_r} = t_r, \end{cases} \tag{13}$$

$$\begin{cases} a_l = \cos\phi_l \sin\psi_l \\ b_l = \cos\phi_l \cos\psi_l \\ c_l = \sin\phi_l \end{cases} \quad \begin{cases} a_r = \cos\phi_r \sin\psi_r \\ b_r = \cos\phi_r \cos\psi_r \\ c_r = \sin\phi_r \end{cases} \tag{14}$$

where t_l, t_r are the parameters for the line O_lM and O_rM separately. Therefore, any point (x, y, z) on each line is usually written parametrically as a function of t_l and t_r:

$$\begin{cases} x_l = a_l t_l + x_{ol} \\ y_l = b_l t_l + y_{ol} \\ z_l = c_l t_l + z_{ol} \end{cases} \quad \begin{cases} x_r = a_r l_r + x_{or} \\ y_r = b_r t_r + y_{or} \\ z_r = c_r t_r + z_{or} \end{cases} \tag{15}$$

(2) In our situation, O_lM and O_rM are skew lines, such that these two lines are no parallel and do not intersect in 3D. Generally, the shortest distance between the two skew lines lies along the line that

is perpendicular to both of them. By defining the intersection points of the shortest segment line for each line by (x_{lp}, y_{lp}, z_{lp}) and (x_{rp}, y_{rp}, z_{rp}), we get the parametric equations:

$$\begin{cases} x_{lp} = a_l t_l + x_{ol} \\ y_{lp} = b_l t_l + y_{ol} \\ z_{lp} = c_l t_l + z_{ol} \end{cases} \quad \begin{cases} x_{rp} = a_r l_r + x_{or} \\ y_{rp} = b_r t_r + y_{or} \\ z_{rp} = c_r t_r + z_{or} \end{cases} \tag{16}$$

(3) Knowing the position of the intersection points on each line, the distance is calculated by the square Euclidean norm:

$$J = \|(x_{lp}, y_{lp}, z_{lp}) - (x_{rp}, y_{rp}, z_{rp})\|_2^2 \tag{17}$$

(4) By deriving the function J, we achieved the minimum distance when $\frac{\partial J}{\partial t_l} = 0$ and $\frac{\partial J}{\partial t_r} = 0$. Then, the above functions derive the following equation:

$$\begin{bmatrix} a_l^2 + b_l^2 + c_l^2 & -(a_l a_r + b_l b_r + c_l c_r) \\ -(a_l a_r + b_l b_r + c_l c_r) & a_l^2 + b_l^2 + c_l^2 \end{bmatrix} \begin{bmatrix} t_l \\ t_r \end{bmatrix} = (x_{ol} - x_{or}) \begin{bmatrix} -a_l \\ a_r \end{bmatrix} + (y_{ol} - y_{or}) \begin{bmatrix} -b_l \\ b_r \end{bmatrix} + (z_{ol} - z_{or}) \begin{bmatrix} -c_l \\ c_r \end{bmatrix}.$$

We could define the matrix on the left side as:

$$\mathbf{H} = \begin{bmatrix} a_l^2 + b_l^2 + c_l^2 & -(a_l a_r + b_l b_r + c_l c_r) \\ -(a_l a_r + b_l b_r + c_l c_r) & a_l^2 + b_l^2 + c_l^2 \end{bmatrix} \tag{18}$$

Considering that there is a uniqueness vertical line, so $\det \mathbf{H} \neq 0$, and the position of the target point Min the world coordinate is:

$$\begin{bmatrix} x_M \\ y_M \\ z_M \end{bmatrix} = w \begin{bmatrix} x_{lp} \\ y_{lp} \\ z_{lp} \end{bmatrix} + (1 - w) \begin{bmatrix} x_{rp} \\ y_{rp} \\ z_{rp} \end{bmatrix}, w \in [0, 1]. \tag{19}$$

where w is weight, and the other parameters are:

$$\begin{cases} x_{lp} = a_l D \dfrac{a_l(a_l^2 + b_l^2 + c_l^2) - a_r(a_l a_r + b_l b_r + c_l c_r)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} \\ y_{lp} = b_l D \dfrac{a_l(a_l^2 + b_l^2 + c_l^2) - a_r(a_l a_r + b_l b_r + c_l c_r)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} \\ z_{lp} = c_l D \dfrac{a_l(a_l^2 + b_l^2 + c_l^2) - a_r(a_l a_r + b_l b_r + c_l c_r)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} \end{cases} \tag{20}$$

and:

$$\begin{cases} x_{rp} = D \left[a_r \dfrac{a_l(a_l a_r + b_l b_r + c_l c_r) - a_r(a_l^2 + b_l^2 + c_l^2)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} + 1 \right] \\ y_{rp} = b_r D \dfrac{a_l(a_l a_r + b_l b_r + c_l c_r) - a_r(a_l^2 + b_l^2 + c_l^2)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} \\ z_{rp} = c_r D \dfrac{a_l(a_l a_r + b_l b_r + c_l c_r) - a_r(a_l^2 + b_l^2 + c_l^2)}{(a_l b_r - b_l a_r)^2 + (b_l c_r - c_l b_r)^2 + (a_l c_r - c_l a_r)^2} \end{cases} \tag{21}$$

The angle between the UAV landing trajectory and the runway area is usually between 3° and 7°. By considering 1 mrad normal distributed disturbance (the accuracy of the PTU is 0.006°), Figure 8 illustrates measurement errors of x_M, y_M and z_M in the case of different points $(x, y) \in S$, where $S = \{(x, y) | -50 \leq x \leq 50, 20 \leq y \leq 1000\}$.

Obviously, the errors at a considerable distance are notable, but their incidence declines while the aircraft is close to the runway. When the UAV is only 100 m to the landing area, the error of altitude is about 0.02 m, which is dependable for the landing task, as shown in Figure 9. Figure 10 shows that the navigation could be improved at the same distance with a large baseline configuration (20 m).

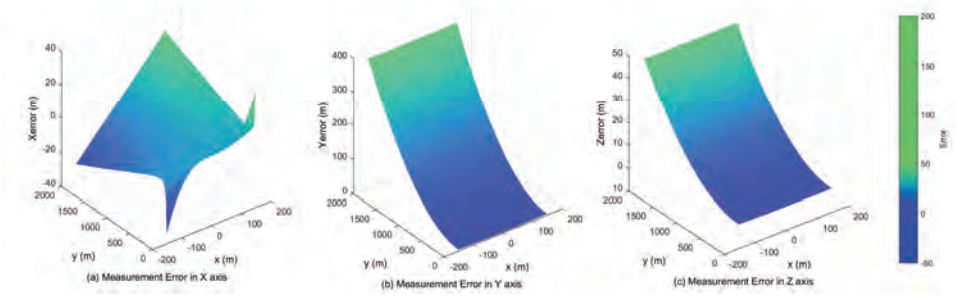


Figure 8. The baseline is 15 m, and the focus is 100 mm. The measurement errors in the *X*, *Z* and *Y* axis with 2000 m.

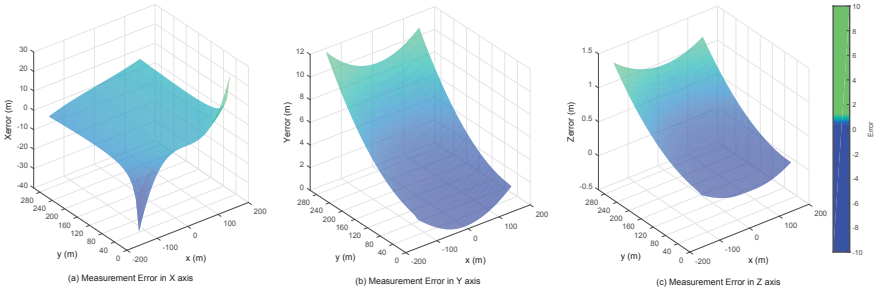


Figure 9. The baseline is 15 m, and the focus is 100 mm. The measurement errors in the *X*, *Z* and *Y* axis with 280 m.

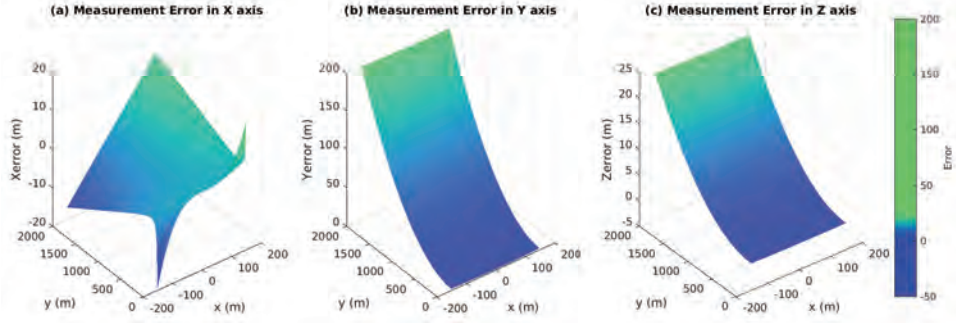


Figure 10. The baseline is 20 m, and the focus is 100 mm. The measurement errors in the *X*, *Z* and *Y* axis with 2000 m.

Furthermore, the errors are much smaller as the UAV lands aligned with the center line of the runway. Tables 1 and 2 illustrate that the error varies non-linearly and decreases significantly as the target approaches the touch down point. The smaller the disturbance of the system is, the better the accuracy in each axis will be.

Table 1. Errors when the center extraction of the aircraft with 1-pixel disturbance, f = 100 mm and D = 10 m.

Error (m)/Distance (m)	4000	3000	2000	1000	500	200	100	50
Xerror	−1.44	−1.16	−0.63	−0.65	−0.25	−0.10	−0.05	−0.05
Yerror	1141.92	692.31	333.44	195.65	23.82	3.92	1.00	0.25
Zerror	133.42	82.62	39.24	22.70	2.43	0.28	0.03	−0.02

Table 2. Errors when the center extraction of the aircraft with 5-pixel disturbance, f = 100 mm and D = 10 m.

Error (m)/Distance (m)	4000	3000	2000	1000	500	200	100	50
Xerror	−3.33	−3.00	−2.53	−2.14	−1.02	−0.46	−0.23	−0.13
Yerror	2663.28	1800.31	1000.11	642.87	100.23	18.19	4.17	1.23
Zerror	320.66	214.93	117.73	74.57	10.24	1.32	0.09	−0.07

Different from the traditional binocular vision system, the optical axes of each vision unit are not parallel during the operation, and there is an initial offset between the camera optical center and the rotation axes of PTU. Therefore, the traditional checkerboard pattern calibration method is not sufficient and convenient to obtain the stereo system parameters for our large baseline system. To solve the calibration issue, we firstly chose the intrinsic camera model, which includes the principal point displacement, optical distortions, skew angle, etc. Each camera should be calibrated separately by the classical black-white chessboard method with the help of the calibration module from OpenCV. Secondly, we setup the setting points with the help of the differential Global Positioning System (DGPS) module and calibrate the system based on the PTU rotation angle, coordinates and the ground-truth position of the setting points.

4.3. Rotation Compensation

According to the above discussion, the 3D location estimation depends largely on the precision of the target center position in the image plane. Our previous work [16,17,20] introduced saliency-inspired and the Chan–Vese model method to track the UAV during the landing progress. Both of these approaches predict and extract the center of the UAV position without considering the PTU rotation. However, in the practical situation, the PTU might jump suddenly due to the disturbance of the control signal and the unexpected maneuver of the UAV. We define the target center position as (x_t, y_t) in the image frame, which can be predicted iteratively by:

$$x_t = f\frac{x_{t-1} - f\psi}{x_{t-1}\psi - \phi y_{t-1} + f} \tag{22}$$

$$y_t = f\frac{y_{t-1} - f\phi}{x_{t-1}\psi - \phi y_{t-1} + f} \tag{23}$$

where ψ and ϕ are the PTU rotation angles and f is the camera focal length. The precision of the bounding box prediction (BBP) could be improved by the PTU rotation compensation.

4.4. Localization Framework

In the ensemble configuration, we separate the vehicle guidance and control into an inner loop and an outer loop, because it is a much simpler and well-tested design approach. As the inner loop controller already exists in the autopilot, we developed an efficient and robust outer navigation loop, which manages the visual information with the on-board sensors. Figure 11 presents the separated long baseline stereo localization frame.

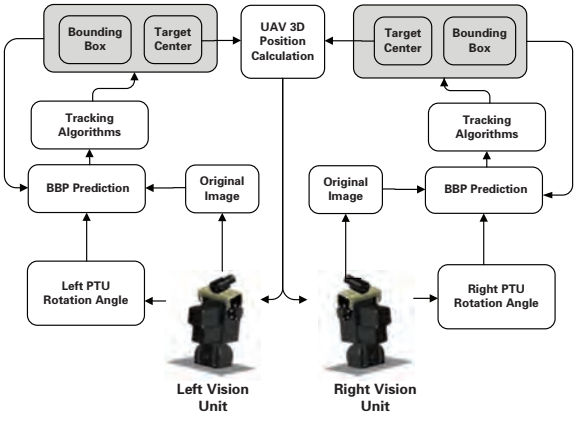


Figure 11. PTU-based UAV localization framework.

5. Experiments and Discussion

5.1. Experiments' Setup

For visible light camera, we selected DFK 23G445, which was developed by Imaging Source GmbH. The sensor of this camera is the Sony ICX445AQA equipped with the GigE interface, which has high data transfer rates, typically up to 1000 Mbit/s. This camera has an image resolution of 1280 × 960 with the RGB32 color model and a maximum frame rate of 30 fps. The lens of the vision system we adopted is 100 mm, and the baseline is 10 m. To extend the field of view, we adopted precision PTU to actuate the camera. PTU-D300E is a high performance product from FLIR. Its pan/tilt speeds up to 50°/s with the position resolution of 0.006°. Moreover, it is a user-programmable product integrating Ethernet and RS-232 interface. The real-time command interface supports advanced applications such as video tracking. We set up the camera on the top bracketing, and the assembled individual vision system is illustrated in Figure 3.

This experimental test-bed is a customized fixed-wing aircraft, which is a gasoline-powered radio-controlled model aircraft. The on-board autopilot allowed for the aircraft to perform simple commanded maneuvers. Our autopilot module is iFLY-F1A, and the navigation module is iFLY-G2 [21], which is a small six-DOF (degree of freedom) navigation system. This module supports real-time 3D information including attitude angle, angular rate, position, speed, acceleration, true air speed and calibrated air speed. F1A is connected with G2 through the RS-232 serial port. Table 3 lists the other technical specifications of the UAV platform. Communication is crucial in the landing navigation framework, because the relative localization is broadcast through the radio. The navigation data are sent using an advanced radio modem that transmits and receives on the 900-MHz band. The XTend RF modems support up to 22 km outdoor communication with the interface data rates from 10 bps–23,000 bps, which is sufficient to transfer GNSS data and predicted position from the ground station to the on-board navigation modem.

Table 3. The technical specifications of Pioneer.

Items	Description
Vehicle mass	9000 g
Maximum Payload mass	5000 g
Diameter	2900 mm
Flight duration	up to 180 min
Cruising speed	30.0 m/s

The landing procedure was divided into four sections: (1) the UAV takeoff from the runway; (2) cruise near the landing area in a large range to test the control system; (3) cruise near the landing area in a small range; and after the UAV is locked by the visual system and received the visual references, the UAV control system was using

vision-based localization data, and the GPS data was only recorded as the benchmark; (4) safely landing back on the runway.

5.2. Results and Discussion

Based on the results of the simulation, eight sets of experimental results are conduced to establish the feasibility of the proposed approach. The results are shown in Table 4. Considering the real-time capability and the precision of the target detection, we modified the original discriminative scale space tracker (DSST) [22], which additionally calculates a one-dimensional discriminative scale filter to evaluate the target size. In the realistic application, it is a very critical requirement that the lateral deviation error from the middle line of the runway and the lateral acceleration of the vehicle be perfectly eliminated to minimize the damage of the vehicle. Figure 12 illustrates the approach results. The left image shows the landing waypoints projecting on a satellite map where A is the locking point of the ground landing and B is the desired touch down point on the runway. In addition, the three 3D landing curves represent the calculated results from the Chan–Vese, DDST and DDST with BBP methods. To compare with the ground truth, recording during the landing process by DGPS, the location errors of each axis are lists on the right side. In X and Z-axis, the location error decreases while the vehicle approaches the landing area. The error in Y-axis has larger error compared with X and Z-axis, and the disturbance is significant.

As the theoretical and simulation result discussed, the localization errors in each axis are large when the UAV is far way from the ground visual system. To illustrate the result more clearly, we compared the localization results with DGPS at separate intervals, which are shown in Table 5. Previously, the average errors of each axis at a large distance (more than 400 m) are large, especially in the depth dimension. The comparison of the average frame rate of different kinds of tracking algorithms is listed in Table 6. The DSST results have the best real-time performance, which reaches 21.345 fps, and has better accuracy compared with the mean shift method, which has similar process speed. We also calculated the errors in $\mathcal{O}_c(\mathbf{i}^{O,c}, \mathbf{j}^{O,c}, \mathbf{k}^{O,c})$ coordinates by implementing those tracking methods with the identical landing imaging streaming and PTU status. The results are shown at separate interval in Tables 7–9. In the accuracy measurement, the DSST with BBP calculates the 3D position more precisely at the cost of a slower frame rate.

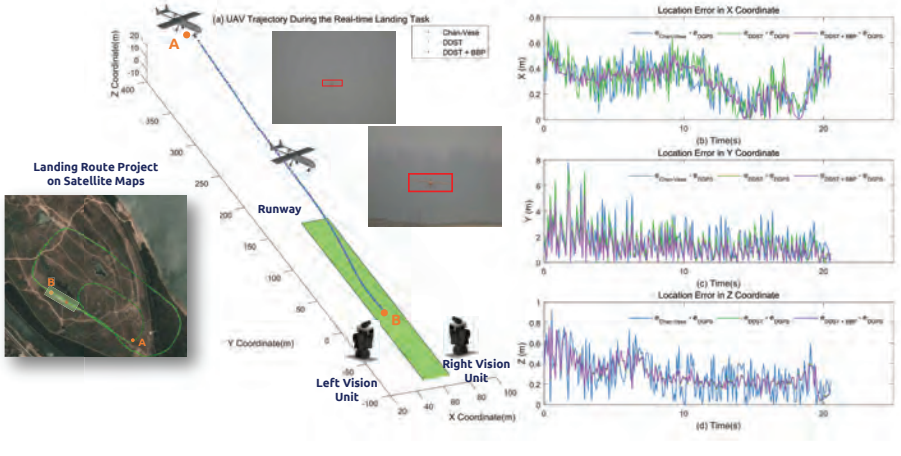


Figure 12. Final approach of the fixed-wing UAV.

Table 4. Eight experiment results in different weather condition.

No.	Weather Condition	Detection Distance	RMSE $i^{O,c}$ (m)	RMSE $j^{O,c}$ (m)	RMSE $k^{O,c}$ (m)
1	Clear	848.735	0.330	1.465	0.291
2	Clear	**892.134**	0.241	1.292	**0.229**
3	Clear	872.311	0.389	1.322	0.293
4	Clear	847.373	0.261	1.413	0.245
5	Clear	857.117	**0.251**	**1.252**	0.312
6	Overcast	491.193	0.503	1.689	0.602
7	Overcast	503.175	0.495	**1.353**	**0.587**
8	Overcast	**534.238**	**0.482**	1.781	0.592

Table 5. Localization error in each axis at separated intervals with the DSST and bounding box prediction (BBP) algorithms.

Interval (m)	$i^{O,c}$ (m)	$j^{O,c}$ (m)	$k^{O,c}$ (m)	Interval (m)	$i^{O,c}$ (m)	$j^{O,c}$ (m)	$k^{O,c}$ (m)
600∼580	0.348	1.862	0.441	**300∼280**	0.138	1.526	0.153
580∼560	0.322	1.444	0.362	**280∼260**	0.142	1.413	0.178
560∼540	0.218	1.557	0.327	**260∼240**	0.103	1.114	0.124
540∼520	0.197	1.558	0.284	**240∼220**	0.094	0.711	0.132
520∼500	0.228	1.841	0.183	**220∼200**	0.105	0.898	0.143
500∼480	0.229	1.430	0.226	**200∼180**	0.151	0.831	0.163
480∼460	0.192	1.483	0.233	**180∼160**	0.163	0.842	0.134
460∼440	0.183	1.472	0.239	**160∼140**	0.157	0.913	0.192
440∼420	0.192	1.431	0.121	**140∼120**	0.142	0.725	0.160
420∼400	0.191	1.663	0.199	**120∼100**	0.169	0.922	0.149
400∼380	0.169	1.662	0.193	**100∼80**	0.147	0.797	0.069
380∼360	0.171	1.542	0.185	**80∼60**	0.133	0.697	0.079
360∼340	0.173	1.541	0.183	**60∼40**	0.114	0.441	0.068
340∼320	0.153	1.333	0.161	**40∼20**	0.124	0.312	0.064
320∼300	0.156	1.311	0.163	**20∼00**	0.082	0.284	0.103

Table 6. Average frame rate of the tracking algorithms.

	Previous Methods				**Our Methods**	
	Mean Shift [23]	**AdaBoost** [24]	**Chan–Vese** [25]	**Saliency-Inspired** [26]	**DSST**	**DSST + BBP**
Average Frame Rate (fps)	20.131	13.152	7.131	8.013	**28.335**	27.867

Table 7. Errors (m) in the $i^{O,c}$-axis with tracking algorithms at separate intervals (m).

Interval	**Previous Methods**				**Proposed Methods**	
	Mean Shift [23]	**AdaBoost** [24]	**Chan–Vese** [25]	**Saliency-Inspired** [26]	**DSST**	**DSST + BBP**
500–400	1.113	0.911	1.201	1.181	0.034	0.332
400–300	0.932	0.891	0.921	0.943	0.029	0.027
300–200	0.562	0.416	0.713	0.765	0.017	0.016
200–100	0.329	0.287	0.512	0.447	0.013	0.013
<100	0.213	0.120	0.341	0.281	0.013	0.011

Table 8. Errors (m) in the $\mathbf{j}^{O,c}$-axis with tracking algorithms at separate intervals (m).

Interval	Previous Methods				Proposed Methods	
	Mean Shift [23]	**AdaBoost** [24]	**Chan–Vese** [25]	**Saliency-Inspired** [26]	**DSST**	**DSST + BBP**
500–400	11.923	6.238	3.661	3.231	2.218	1.761
400–300	7.317	5.721	2.887	2.902	1.876	1.632
300–200	5.365	2.576	1.799	1.767	1.703	1.371
200–100	3.897	1.739	1.310	1.134	0.981	1.131
<100	1.762	0.780	0.737	0.692	0.763	0.541

Table 9. Errors (m) in the $\mathbf{k}^{O,c}$-axis with tracking algorithms at separate intervals (m).

Interval	Previous Methods				Proposed Methods	
	Mean Shift [23]	**AdaBoost** [24]	**Chan–Vese** [25]	**Saliency-Inspired** [26]	**DSST**	**DSST + BBP**
500–400	1.231	1.087	1.387	1.299	0.487	0.484
400–300	0.976	0.901	0.762	0.876	0.370	0.325
300–200	0.812	0.557	0.337	0.612	0.459	0.313
200–100	0.438	0.312	0.489	0.532	0.141	0.179
<100	0.301	0.211	0.401	0.321	0.102	0.103

6. Concluding Remarks

This paper presents a complete localization framework of the ground-based stereo guidance system. This system could be used to pilot the UAV for landing autonomously and safely in the GNSS-denied scenario. Compared with the onboard solutions and other state-of-the-art ground-based approaches, this ground-based system profited enormously from the computation capacity and flexible configuration with the baseline and sensors. The separate deployed configurations did not improve the detection distance, which was discussed in our previous works [15]; however, they enhance the maximum range for depth estimation. Although the system has some pitfalls, such as the low accuracy at a long distance in the depth axis and not supporting the attitude measurement, this low-cost system could be arranged quickly for any proposed environment. Additional future work will focus on estimate errors over time and investigate methods to improve inevitable error propagation through the inclusion of additional sensors, such as GNSS and on-board sensors.

Acknowledgments: The work is supported by Major Application Basic Research Project of NUDT with Granted No. ZDYYJCYJ20140601. The authors would like to thank Dianle Zhou and Zhiwei Zhong for their contribution on the experimental prototype development. Tengxiang Li, Dou Hu, Shao Hua and Yexun Xi made a great contribution on the practical experiment configuration.

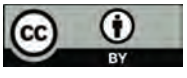
Author Contributions: W.K. proposed and designed the ground-based navigation system architecture. W.K. and T.H. implemented the vision algorithms to test the capability of the system. D.Z., L.S., and J.Z. participated in the development of the guidance system and conceive the experiments. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Floreano, D.; Wood, R.J. Science, technology and the future of small autonomous drones. *Nature* **2015**, *521*, 460–466.
2. Kushleyev, A.; Mellinger, D.; Powers, C.; Kumar, V. Towards a swarm of agile micro quadrotors. *Auton. Robot.* **2013**, *35*, 287–300.
3. Manning, S.D.; Rash, C.E.; LeDuc, P.A.; Noback, R.K.; McKeon, J. *The Role of Human Causal Factors in US Army Unmanned Aerial Vehicle Accidents*; Army Aeromedical Research Lab: Fort Rucker, AL, USA, 2004.
4. McLean, D. *Automatic Flight Control Systems*; Prentice Hall: Englewood Cliffs, NJ, USA, 1990; p. 606
5. Stevens, B.L.; Lewis, F.L. *Aircraft Control and Simulation*; John Wiley & Sons: Hoboken, NJ, USA, 2003.
6. Farrell, J.; Barth, M.; Galijan, R.; Sinko, J. *GPS/INS Based Lateral and Longitudinal Control Demonstration: Final Report*; California Partners for Advanced Transit and Highways (PATH): Richmond, CA, USA, 1998.

7. RUAG. Available online: https://www.ruag.com/de/node/307 (accessed on 8 June 2017).
8. Northrop Grumman. *MQ-8B Fire Scout Vertical Takeoff and Landing Tactical Unmanned Aerial Vehicle System*; Northrop Grumman: Falls Church, VA, USA.
9. Wang, W.; Song, G.; Nonami, K.; Hirata, M.; Miyazawa, O. Autonomous Control for Micro-Flying Robot and Small Wireless Helicopter X.R.B. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 2906–2911.
10. Pebrianti, D.; Kendoul, F.; Azrad, S.; Wang, W.; Nonami, K. Autonomous hovering and landing of a quad-rotor micro aerial vehicle by means of on ground stereo vision system. *J. Syst. Des. Dyn.* **2010**, *4*, 269–284.
11. Martinez, C.; Campoy, P.; Mondragon, I.; Olivares-Mendez, M.A. Trinocular ground system to control UAVs. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 3361–3367.
12. Guan, B.; Sun, X.; Shang, Y.; Zhang, X.; Hofer, M. Multi-camera networks for motion parameter estimation of an aircraft. *Int. J. Adv. Robot. Syst.* **2017**, *14*, doi:10.1177/1729881417692312.
13. Kim, E.; Choi, D. A UWB positioning network enabling unmanned aircraft systems auto land. *Aerosp. Sci. Technol.* **2016**, *58*, 418–426.
14. Kong, W.; Zhang, D.; Wang, X.; Xian, Z.; Zhang, J. Autonomous landing of an UAV with a ground-based actuated infrared stereo vision system. In Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 2963–2970.
15. Kong, W.; Zhou, D.; Zhang, Y.; Zhang, D.; Wang, X.; Zhao, B.; Yan, C.; Shen, L.; Zhang, J. A ground-based optical system for autonomous landing of a fixed wing UAV. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 4797–4804.
16. Tang, D.; Hu, T.; Shen, L.; Zhang, D.; Kong, W.; Low, K.H. Ground Stereo Vision-based Navigation for Autonomous Take-off and Landing of UAVs: A Chan-Vese Model Approach. *Int. J. Adv. Robot. Syst.* **2016**, *13*, doi:10.5772/62027.
17. Ma, Z.; Hu, T.; Shen, L. Stereo vision guiding for the autonomous landing of fixed-wing UAVs: A saliency-inspired approach. *Int. J. Adv. Robot. Syst.* **2016**, *13*, 43.
18. Kong, W.; Zhou, D.; Zhang, D.; Zhang, J. Vision-based autonomous landing system for unmanned aerial vehicle: A survey. In Proceedings of the 2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI), Beijing, China, 28–29 September 2014; pp. 1–8.
19. Gautam, A.; Sujit, P.; Saripalli, S. A survey of autonomous landing techniques for UAVs. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 1210–1218.
20. Hu, T.; Zhao, B.; Tang, D.; Zhang, D.; Kong, W.; Shen, L. ROS-based ground stereo vision detection: Implementation and experiments. *Robot. Biomim.* **2016**, *3*, 14.
21. Beijing Universal Pioneering Technology Co., Ltd. *iFLYUAS*; Beijing Universal Pioneering Technology Co., Ltd.: Beijing, China, 2016.
22. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, 1–5 September 2014*; BMVA Press: Durham, UK, 2014.
23. Oza, N.C. Online bagging and boosting. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 10–12 October 2005; Volume 3, pp. 2340–2345.
24. Bradski, G.R. Real time face and object tracking as a component of a perceptual user interface. In Proceedings of the IEEE Workshop on Applications of Computer Vision (WACV'98), Princeton, NJ, USA, 19–21 October 1998; pp. 214–219.
25. Chan, T.F.; Sandberg, B.Y.; Vese, L.A. Active contours without edges for vector-valued images. *J. Vis. Commun. Image Represent.* **2000**, *11*, 130–141.
26. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259.



sensors

MDPI

Article

A Stereo Dual-Channel Dynamic Programming Algorithm for UAV Image Stitching

Ming Li [1,2,3], Ruizhi Chen [1,2,*], Weilong Zhang [1,*], Deren Li [1,2], Xuan Liao [1], Lei Wang [1], Yuanjin Pan [1] and Peng Zhang [1]

1 State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; lisouming@whu.edu.cn (M.L.); drli@whu.edu.cn (D.L.); sdliaoxuan@163.com (X.L.); lei.wang@whu.edu.cn (L.W.); pan_yuanjin@163.com (Y.P.); fenix@whu.edu.cn (P.Z.)

2 Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430079, China

3 School of Resource and Environmental Science, Wuhan University, Wuhan 430079, China

* Correspondence: ruizhi.chen@whu.edu.cn (R.C.); zhangweilong@whu.edu.cn (W.Z.); Tel.: +86-136-3863-6532 (R.C.); +86-134-8819-9034 (W.Z.)

Received: 6 July 2017; Accepted: 6 September 2017; Published: 8 September 2017

Abstract: Dislocation is one of the major challenges in unmanned aerial vehicle (UAV) image stitching. In this paper, we propose a new algorithm for seamlessly stitching UAV images based on a dynamic programming approach. Our solution consists of two steps: Firstly, an image matching algorithm is used to correct the images so that they are in the same coordinate system. Secondly, a new dynamic programming algorithm is developed based on the concept of a stereo dual-channel energy accumulation. A new energy aggregation and traversal strategy is adopted in our solution, which can find a more optimal seam line for image stitching. Our algorithm overcomes the theoretical limitation of the classical Duplaquet algorithm. Experiments show that the algorithm can effectively solve the dislocation problem in UAV image stitching, especially for the cases in dense urban areas. Our solution is also direction-independent, which has better adaptability and robustness for stitching images.

Keywords: UAV image; dual-channel; dynamic programming; seam line; energy function

1. Introduction

Unmanned aerial vehicle (UAV) remote sensing systems have been widely used in environmental monitoring, ecological farming, disaster emergency management, navigation map production, and 3D urban reconstruction because of its advantages of low cost, fast data collection, and easy operation [1–5]. Due to its low flight altitude and the camera perspective constraints, the coverage area of a single UAV image is small. In order to expand the image coverage area to capture more information from the target area, multiple UAV images are collected, leading to the need of stitching multiple images to form a mosaic image. Because high-altitude wind has a significant impact on the UAV platform due to its light-weight [6], problems such as irregular image overlapping and uneven image exposure are introduced into the adjacent images. Furthermore, images captured from an unstable UAV platform will lead to a vulnerable stitched image with ghosting, blur, dislocation, and color inconsistence.

Various methods for seamless stitching of UAV remote sensing images have been investigated [7–17]. These methods can generally be classified into two types: weighted fusion algorithms and seam line-based algorithms [7–14]. The weighted fusion algorithms focus on overlapping area pixels on the adjacent images. They adopted algorithms to eliminate the seams, which can be simply and easily implemented and can effectively adjust the exposure difference [7–10]. These weighted fusion algorithms process the pixels directly into the spatial domain of the images.

However, when there are matching errors or abundant image texture structures in the overlapping area, the stitched images will produce ghosting, dislocation, and fuzzy phenomena and weak texture details [7–10]. Seam line-based algorithms are intended to reduce grayscale and geometric differences, and consist of energy function-based algorithms (including the dynamic programming algorithm, Graph cut-based algorithm) and geometric information-based algorithms [11–14]. The latter have the advantages of being fast and having simple calculation, and do not have the requirement of constructing an energy function. In contrast, the former focusing on the energy difference between the images and its effect is superior. Additionally, energy function-based algorithms [11–13] adopt Dijkstra's shortest path algorithm to search for the optimal seam line, which addresses the ghosting and dislocation problems because of the movements of the objects and registration errors, but it suffers from low search efficiency and complex weight determination. The seam line-based algorithms—more specifically, the ant colony algorithm—can evade the area where the color contrast is larger on the image, while it will easily lead the search processing to the local optimum due to its sensitivity to ants' number [14]. Moreover, there are some algorithms based on the snake model [15], and some based on morphological model [16,17]. Although these algorithms can almost guarantee the consistence of the geometric structure and evade the phenomenon of ghosting in overlapping areas under certain conditions, they are unable to guarantee that ghosting and seams will be overcome at the same time—especially when there is a significant brightness difference in the adjacent images. Similarly, these algorithms are unable to achieve satisfactory results when there are rich texture structures in image pairs, registration errors, and radiation brightness differences. Furthermore, most of the current seam line generation methods do not adopt the dynamic programming algorithm. This is because the dynamic programming algorithm is relatively mature in theory, and not easy to develop further. Furthermore, it relies strongly on image direction that leads to a low robustness with modified energy functions.

This paper proposes a new algorithm to realize the seamless stitching of UAV images through a comprehensive theoretical analysis of the dynamic programming algorithm. The algorithm especially solves the dislocation problem caused by the seam line in the building-intensive areas. The basic idea is to determine the geometric errors introduced by perspective errors, camera distortions, and radiation errors by analyzing the mapping relationships between the left and right images. The determination of these errors can aid the seam line searching process. Thus, it is an optimization problem. There is no unique (that is a non-ground-truth problem) solution for this image seam line searching problem. Fortunately, we only require a close optimum solution that meets the requirement of stitching the images.

2. Methodology

2.1. Duplaquet Algorithm

In 1958, Bellman proposed an optimization solution that transforms the multi-stage process into a series of single-stage processes, and thus he invented the dynamic programming algorithm [18]. Based on this, Duplaquet proposed a classic algorithm for searching image seam lines. The algorithm ensures that the lengths of all alternative seam lines are equal, and the seam line with the smallest accumulated energy value is the optimal seam line. Equation (1) shows the energy criterion defined in the algorithm [19]:

$$C(x, y) = C_{dif}(x, y) - \lambda C_{edge}(x, y), \quad (1)$$

where $C_{dif}(x, y)$ is the mean value of the gray level difference of the pixel in the overlapping areas between two adjacent images, $C_{edge}(x, y)$ is the minimum gradient value of the pixel in the overlapping areas of the image pair, and λ is a weighing factor, which can be used to adjust the proportion of gray difference and structure difference in energy function.

2.2. Problems Analysis

The energy criterion in the Duplaquet algorithm only considers the horizontal and vertical gradients, and compares the pixels in three adjacent directions near the current pixel, as shown in Figure 1. When the overlapping area has dense high-rise buildings and tall trees, the seam lines output from the Duplaquet algorithm are likely across the edge of the buildings due to the inconsistent deformation from image point to the roof point. Therefore, it is easy to find the dislocations in the stitched image leading to a large matching error. Furthermore, it is not guaranteed to obtain the best seam line by using the classical Sobel operator to calculate the approximate gradient of the pixel based on the horizontal and vertical templates (see Equation (2)) without considering diagonal directions in the calculation process [20]. As shown in Figure 2, the red seam-line not only crosses the edge of the houses, but also deviates from the ideal seam-line.

$$D_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad D_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2}$$

Specifically, the algorithm has the following main problems: (1) The gradient guidance direction of the energy function does not support omnidirectional searching. (2) The energy function is direction-dependent; the energy aggregation takes only three directions into consideration, and the direction of energy traversal is also limited from left to right, as well as from top to bottom. (3) The energy function is prone to local optimal solution due to the impact of the two factors mentioned above. This will directly lead to the situation where the optimal seam line can be easily affected by buildings.

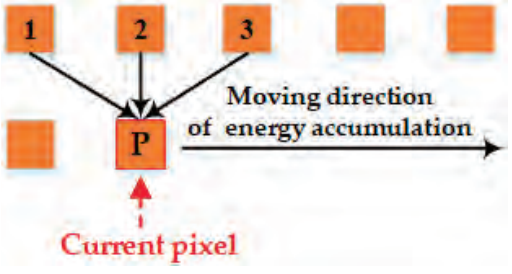


Figure 1. The schematic diagram of Duplaquet's energy criterion.

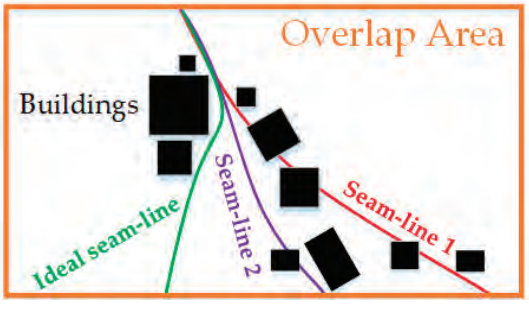


Figure 2. The sketch map of search results for seam lines.

There are two experimental results based on the Duplaquet algorithm. As shown in Figure 3, the seam lines across the houses are prone to ghosting and dislocation in stitched images. In addition,

some researchers believe that the energy function is poorly fitted, making it difficult to find the optimal seam line. For this reason, these researchers attempt to modify the energy function in dynamic programming; however, they overlook the optimality of the corresponding model. This also happens in the algorithms included in the OpenCV library.

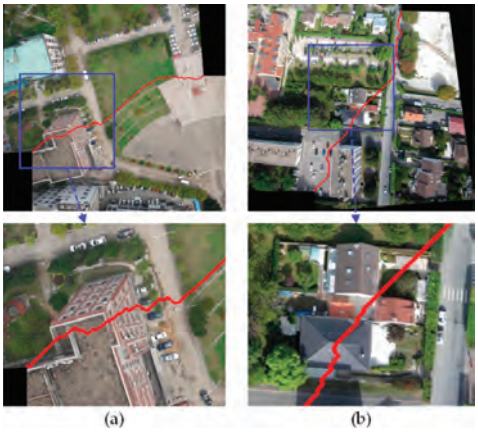


Figure 3. The stitching results using the Duplaquet algorithm for two datasets. (**a**) The experimental result with UAV images shot in Wuhan; (**b**) The experimental result with UAV images shot in Paris.

2.3. Our Algorithm

We developed a new algorithm for searching seam line by taking gradient guidance direction, energy accumulation directions (energy aggregation direction, energy traversal direction), and energy local optima into account.

2.3.1. Gradient Calculation

The Duplaquet algorithm only considers the horizontal and vertical gradient in the energy criterion; it fails to obtain the optimal seam line in some cases. In order to solve this problem, we develop a new algorithm on the basis of the classical Sobel operator, by considering eight directional neighborhood information of current pixel and the similarity of its surrounding structure [21]. The new approach of gradient calculation is as follows:

$$\begin{aligned} D_{0^\circ} &= \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} D_{\pi/4} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix} D_{\pi/2} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} D_{3\pi/4} = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \\ D_{\pi} &= \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} D_{5\pi/4} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix} D_{3\pi/2} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} D_{7\pi/4} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \end{aligned} \quad (3)$$

2.3.2. Directionality of Energy Accumulation

In order to solve the direction-dependent problem in energy accumulation, we introduce a new direction in our algorithm, as shown in Figure 4. The algorithm resolves the problem of seam line deviation from the true seam line with an improved strategy of energy aggregation. As shown in Figure 2, the improved purple seam-line 2 is closer to the ideal seam line than the red seam line 1, despite its obvious insufficiency.

The optimal solution of the energy function is affected by not only the directions of energy aggregation, but also the directions of energy traversal. Therefore, we redefined the energy criterion

and added the aggregate directions to our new dynamic programming algorithm with a stereo dual-channel energy accumulation strategy. It improves the searching scheme of optimal seam line based on energy accumulation. As shown in Figure 5, there is a schematic diagram of our optimal seam line search strategy, which optimizes the seam line search criteria by detecting the eight pixels (contain the horizontal direction) surrounding the current pixel.

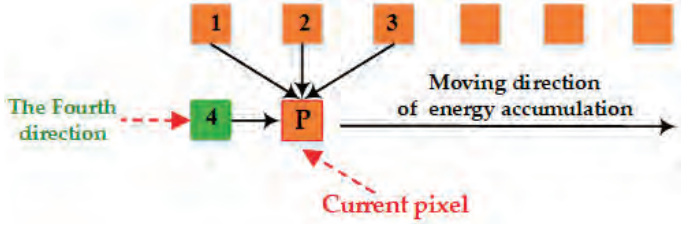


Figure 4. Schematic diagram of energy accumulation by improving energy guidelines.

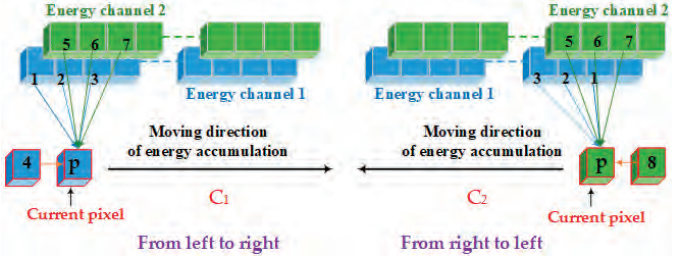


Figure 5. Schematic diagram of our search strategy.

In Figure 5, **P** is the current pixel. We redefine the nine related directions surrounding **P** as follows: 0 (initial invalid direction), 1 (top-left of the current pixel for energy aggregation channel 1), 2 (top of the current pixel for energy aggregation channel 1), 3 (top-right of the current pixel for energy aggregation channel 1), 4 (left of the current pixel for energy aggregation channel 1), 5 (top-left of the current pixel for energy aggregation channel 2), 6 (top of the current pixel for energy aggregation channel 2), 7 (top-right of the current pixel for energy aggregation channel 2), 8 (right of the current pixel for energy aggregation channel 2). Seam line searching is an aggregation process of minimum energy. Each seam line consists of neighborhood pixels with smallest energy value. In our approach, the longest seam line is the optimal seam line. Furthermore, it requires the shape of the overlapping area and close distance to be adapted to the center line of adjacent images.

2.3.3. The Energy Function

Based on the analysis of the theoretical model, we constructed a mathematical abstract expression of the theoretical model. Assuming that image 1 and image 2 are an original image pair to be stitched, the energy function is defined as follows:

$$E = \sum_{(x,y)\in O} B(x,y)\sigma(|f_1(x,y) - f_2(x,y)|) + \sum_{(x,y)\in O} \sigma\left(\max\left(|\underset{0\le k\le 7}{d_k}(f_1(x,y)) - \underset{0\le k\le 7}{d_k}(f_2(x,y))|\right)\right) + \sum_{(x,y)\notin O} N(x,y) \quad (4)$$

In Equation (4), $B(x, y)$ determines whether the current pixel (x, y) is in the boundary of overlapping area of the adjacent images; when $B(x, y) = 1$, it means that it is not in the boundary region,

and when $B(x, y) = 10$, it is in the boundary region. $\sigma(*)$ is the Gaussian smoothing term, which uses the information in the local window to enhance the local influence of the current pixel; $f_1(\cdot), f_2(\cdot)$ are pending images to be stitched; $\boldsymbol{O}$ is the overlapping area; $d(*)$ represents the gradient function of one of the eight directions; $N(x, y)$ is the energy value of the invalid area, which is the constant term, and the value is 100 times larger than the maximum value of $\boldsymbol{O}$.

2.3.4. Computation Procedure

The image size of the overlapping area is set as $m \times n$, and if the overlapping area is irregular, it can be extended to a regular area by using the minimum exterior rectangle of the overlapping area as shown in Figure 6.

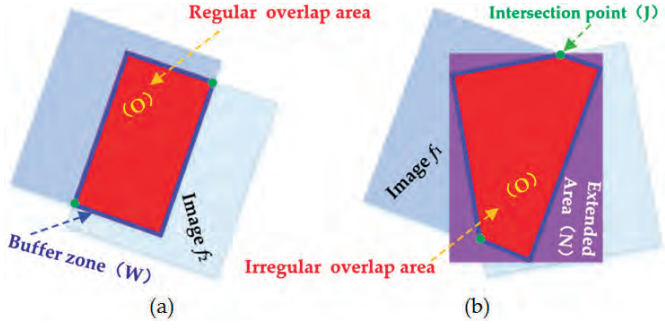


Figure 6. Processing principle of different images overlap area: (**a**) Regular overlap area; (**b**) Irregular overlap area.

The image energy matrix $\boldsymbol{E}$ can be obtained by Equation (4). The algorithm procedure is as follows:

(1) Extract the corresponding points from the adjacent images in order to correct the left and right pending matching images to the virtual unified reference image, so that the images are in the same coordinate system.

(2) Define the overlapping area of the adjacent images $\boldsymbol{O}$, the boundary buffer area $\boldsymbol{W}$ (set its width value is 20 pixels), and $\boldsymbol{W}$ is an empirical value, the invalid area $\boldsymbol{N}$ (extend area), and the boundary intersection $\boldsymbol{J}$.

(3) Calculate the matrices $\boldsymbol{O}$, $\boldsymbol{W}$, $\boldsymbol{N}$ according to Equation (4), where $\boldsymbol{W} \in [1, 10]$, the closer to the boundary, the larger the value is, and set $\boldsymbol{N} = 100 \times \max(\boldsymbol{O})$; $\boldsymbol{J} = -1000 \times \max(\boldsymbol{O})$.

(4) Fill the energy matrix $\boldsymbol{E}$ according to the results of (3).

(5) Reestablish two energy aggregation channels: the $\boldsymbol{C1}$ and $\boldsymbol{C2}$ matrices, which have the same size as $\boldsymbol{E}$; each pair of corresponding elements in these two matrices hold two scalar numbers representing the current aggregation value and the current path direction of the seam line.

(6) For the first row of the energy aggregation channel matrix assigned with the first row of $\boldsymbol{E}$ as the initial value, set its corresponding direction as 0.

(7) The energy aggregation channel matrix starts to make a difference from the second row, which is divided into two aggregation processes from left to right and from right to left. For the energy aggregation channel $\boldsymbol{C1}$, its aggregation process is from the left to the right; the current pixel only considers the directions of 1, 2, 3, 4, 5, 6, and 7. For the energy aggregation channel $\boldsymbol{C2}$, its aggregation process is from the right to the left, and the current pixel only considers the directions of 1, 2, 3, 5, 6, 7, and 8.

(8) When the aggregation is completed, the minimum energy value is found from the last row in $\boldsymbol{C1}$ and $\boldsymbol{C2}$ respectively; an optimal stitching path is then found based on the direction information stored in the matrixes.

In order to ensure that the seam line starts and ends at the intersection points, we select two special intersection points that have the smallest energy value, so that the seam line can be guided and adsorbed.

3. Experiments and Analysis

3.1. Experimental Environment and Data

To verify the effectiveness of our algorithm, we not only utilized the UAV images from different regions with different flight altitudes, but also compared the processing results with the Duplaquet algorithm and the algorithms from open source in OpenCV. In this paper, we used Visual C++ based on OpenCV open source library to program the proposed improvement algorithm. The experimental images are divided into three groups; among them, the data in Figure 7a were acquired by Canon IXUS 220HS (Canon, Oita, Japan) in Paris, its focal length is 4 mm, and the height of the UAV was approximately 250 m. The data in Figure 7b were acquired by DMC-GF3 (Panasonic, Xiamen, China) at Wuhan University Square, its focal length is 14 mm, and the height of the UAV was approximately 200. The data in Figure 7c are a sequence of images of a single-strip; they were acquired by ILCE-600 (Sony, Chonburi, Thailand) in Shaxi town, China, its focal length is 35 mm, and the height of the UAV was approximately 410–440 m. The experimental computer environment is Windows 7 operating system, CPU: Intel (*R*) Core (*TM*) i7-4790, RAM: 32 GB. Figure 8 is a flow chart of our algorithm, listing the key steps of the algorithm.

Figure 7. Three groups of experimental unmanned aerial vehicle (UAV) images. (**a**) The first image pair; (**b**) The second image pair; (**c**) The sequence images of a single-strip.

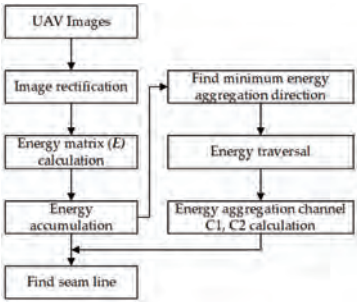


Figure 8. A flow chart of our algorithm.

3.2. Results Analysis

3.2.1. Comparison of Three Algorithms under the Condition of Image Rotation

Firstly, the vertical images in Figure 7a were rotated to be horizontal. Then, we used the Duplaquet algorithm, the OpenCV algorithm, and the algorithm proposed in this paper to search the seam lines. Figure 9 shows the results. The partially enlarged pictures illustrated that the Duplaquet algorithm changed the search results of the optimal seam lines before and after rotation, and the seam lines passed through the edge of buildings in two cases. The optimal seam lines searched by the OpenCV algorithm had slight changes before and after rotation. The seam lines searched by the Duplaquet algorithm and the OpenCV algorithm crossed the edge of the houses in different locations. Furthermore, there was also a problem of house information loss in stitched images. The seam lines searched by our algorithm had basically no change before and after rotation; they still went along the road, and were very good at avoiding the ground buildings. This shows that the traditional methods are sensitive to the direction of images, and the direction of seam lines change with the direction of images. That is to say, the minimum value of the energy function is related to the direction of energy aggregation and traversal, and they may also lead to house information loss. Therefore, due to the specific improvements to the above issues, our algorithm has advantages in adaptability and robustness for different UAV images.

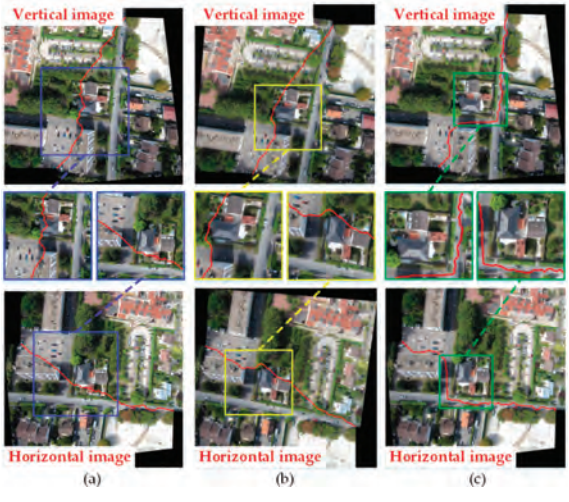


Figure 9. The seam lines of different searching algorithms under different situations. (**a**) Duplaquet algorithm; (**b**) OpenCV algorithm; (**c**) Our algorithm.

3.2.2. Comparison of Stitching Results with Two Groups of Image Pairs

In order to further verify the superiority of our algorithm, the Duplaquet algorithm, the OpenCV algorithm, and our algorithm were used to search the optimal seam lines of image pairs in Figure 7a,b with irregular overlapping areas. Figures 10 and 11 are the respective results. It can be seen from Figures 10 and 11 that the seam lines are obviously different with the three test algorithms. From the local zoom view of Figures 10 and 11, we can find that the Duplaquet algorithm not only places the seam lines across the edge of houses, but also presents a ghosting phenomenon. The OpenCV algorithm often focuses on making the seam line follow along the edge of buildings as much as possible to conceal the seam lines, but the seam lines are still prone to cross the edge of ground buildings. Besides, the house corner information is lost in Figures 10 and 11. In this paper, the optimal seam lines searched by our algorithm are basically following along the road direction, which avoids the ground buildings; this will greatly reduce the probability of dislocation and the seams reason of image geometric errors. The other two algorithms are prone to stitching dislocation, ghosting, and image information loss around the edge.

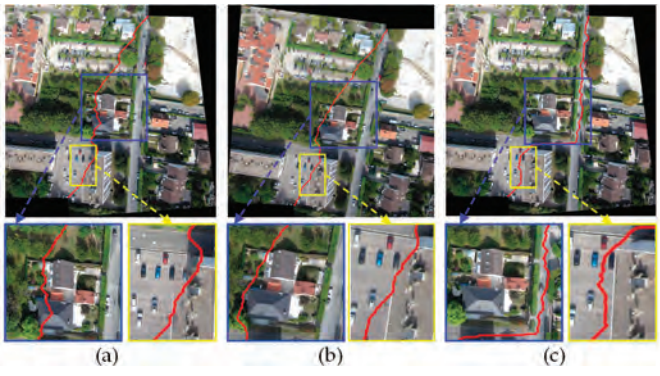


Figure 10. The seam lines of different searching algorithms for Figure 7a: (**a**) Duplaquet algorithm; (**b**) OpenCV algorithm; (**c**) Our algorithm.

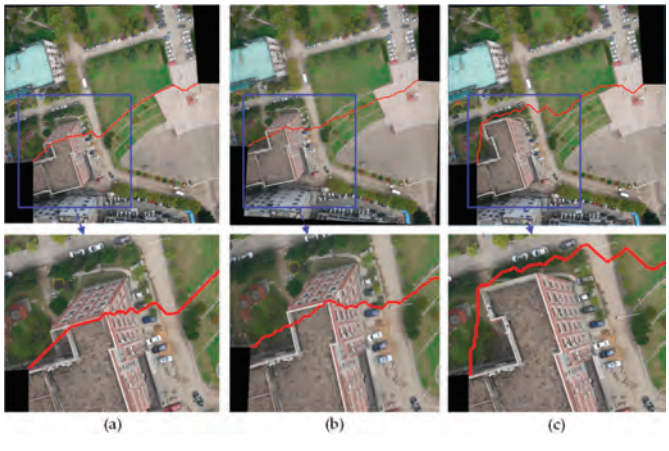


Figure 11. The seam lines of different searching algorithms for Figure 7b: (**a**) Duplaquet algorithm; (**b**) OpenCV algorithm; (**c**) Our algorithm.

3.2.3. Stitching with an Image Sequence

In this experiment, we selected eight UAV images in a single-strip image sequence. Figure 12 showed the results of the single-strip image sequence by the algorithm proposed in this paper. We can see that the algorithm achieved a good result for the stitching of multiple UAV images. There is basically no ghosting, seams, or dislocation in the stitched panorama image. This further illustrates that a strategy based on finding an optimal seam line can avoid crossing the ground buildings, so it is a good solution to UAV image stitching.

Figure 12. The stitching result of a single-strip image sequence.

3.2.4. Comparing the Efficiency of Energy Accumulation Processing

In the previous experimental results, our algorithm found the most satisfying seam lines. Since this algorithm is based on the classical Duplaquet algorithm, this section will compare the energy accumulation time efficiency of these two algorithms. Firstly, we assumed that our algorithm and the Duplaquet algorithm could find the same optimal seam line. Their time efficiency difference can be quantitatively analyzed from the algorithms' complexity. In this paper, the direction of energy aggregation from three aggregation directions increased to eight is mainly an improvement. Assuming that the time complexity of the Duplaquet algorithm is $O(m^3)$ and the time complexity of our algorithm is $O(m^8)$, where m is the total number of pixels within the minimum exterior rectangle of overlapping image area, m can be expressed as the product of w and h, where w is the width of the minimum exterior rectangle and h is the height of the minimum exterior rectangle. Both w and h are measured by unit pixel. However, because the local energy minima exists in the energy function of the Duplaquet algorithm, it results in a lot of time consumption. So, the above assumption is invalid.

We selected five experimental image pairs to verify the above conclusions. Image pair 1 came from Paris, image pair 2 came from Wuhan University, image pairs 3–5 came from Shaxi town. In order to speed up the calculation, it is generally necessary to zoom the image at a certain scale. So, the size of the overlapping area is not the size of the original image overlap area. The experimental results can be seen in Table 1. The efficiency of our algorithm is more than 40 times that of the Duplaquet algorithm. The convergence speed of the energy function is faster. It is necessary to further point out that the theory and the results of the proposed algorithm are obviously different than the classical Duplaquet algorithm. The theoretical improvement and experimental comparisons has proven that the Duplaquet algorithm has a serious local optimal problem and a direction-sensitive problem; they only have the same observation conditions and common goal. The results of the Duplaquet algorithm are a pseudo-solution, and the results of the two algorithms are the same only in the intersection of their feasible solutions, but this is a small probability event. In addition, this paper proposed a global and

non-direction optimization algorithm, which not only has the optimal seam line, but also has better time efficiency.

Table 1. Compared time efficiency of energy accumulation processing.

Image Pair	1	2	3	4	5
$m = w \times h$	957 × 523	909 × 421	919 × 317	923 × 384	919 × 335
Location	(Paris)	(Wuhan University)	(Shaxi town)	(Shaxi town)	(Shaxi town)
Duplaquet algorithm	4898 ms	3572 ms	2735 ms	3368 ms	2854 ms
Our algorithm	91 ms	69 ms	54 ms	68 ms	54 ms
Multiple	53	51	50	49	52

4. Conclusions

This paper analyzed and evaluated the current mainstream stitching algorithms for the problems of ghosting, dislocation, and seams in UAV image stitching. Then, it selected the essential problems of dynamic programming algorithms for seam line searching, and carried out a detailed theoretical study and a large number of UAV image stitching experimental verifications. At last, this paper proposed a stereo dual-channel dynamic programming algorithm to search the optimal seam lines through several improved key problems of the classical Duplaquet algorithm. Meanwhile, the superiority and efficiency of the algorithm proposed in the paper are verified by the credible experiments of two image pairs and an image sequence with irregular overlapping area. The stitched results are better than the Duplaquet algorithm and the OpenCV algorithm. The proposed algorithm is proved to be direction-independent, more efficient, and more robust. However, this paper did not consider the role of image structure information in stitching too much, and that is our next research direction to consider the optical flow in dynamic programming algorithm for UAV image stitching.

Acknowledgments: This study is supported by the National Key Research and development Program of China (2016YFB0502201 and 2016YFB0502202), the NSFC (91638203), the State Key Laboratory Research Expenses of LIESMARS.

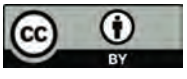
Author Contributions: M.L. and W.Z. proposed the idea and wrote the manuscript; R.C. and D.L. conceived and designed the experiments; X.L., L.W., Y.P. and P.Z. performed the experiments.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, S.; Laefer, D.; Mangina, E. State of technology review of civilian UAVs. *Recent Pat. Eng.* **2016**, *10*, 160–174. [CrossRef]
2. Byrne, J.; Keeffe, E.; Lenon, D.; Laefer, D. 3D reconstructions using unstabilized video footage from an unmanned aerial vehicle. *J. Imaging* **2017**, *3*, 15. [CrossRef]
3. Li, D.; Li, M. Research advance and application prospect of unmanned aerial vehicle remote sensing system. *Geomat. Inf. Sci. Wuhan Univ.* **2014**, *39*, 505–513.
4. Chen, R.; Chu, T.; Landivar, J.; Yang, C.; Maeda, M. Monitoring cotton (*Gossypium hirsutum* L.) germination using ultrahigh-resolution UAS images. *Precis. Agric.* **2017**, *18*, 1–17. [CrossRef]
5. Zhang, W.; Li, M.; Guo, B.; Li, D.; Guo, G. Rapid texture optimization of three-dimensional urban model based on oblique images. *Sensors* **2017**, *17*, 911. [CrossRef] [PubMed]
6. Li, M.; Li, D.; Fan, D. A study on automatic UAV image mosaic method for paroxysmal disaster. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *XXXIX-B6*, 123–128. [CrossRef]
7. Wang, W.; Michale, K. A variational approach for image stitching. *SIAM J. Imaging Sci.* **2013**, *6*, 1318–1344. [CrossRef]
8. Tao, M.; Johnson, M.; Paris, S. Error tolerant image compositing. *Int. J. Comput. Vis.* **2013**, *103*, 178–189. [CrossRef]
9. Levin, A.; Zomet, A.; Peleg, S. Seamless image stitching in the gradient domain. In Proceedings of the European Conference on Computer Vision, Prague, Czech Republic, 11–14 May 2004.

10. Zomet, A.; Levin, A.; Peleg, S.; Weiss, Y. Seamless image stitching by minimizing false edges. *IEEE Trans. Image Process.* **2006**, *15*, 969–977. [CrossRef] [PubMed]
11. Dijkstra, E. A note on two problems in connexion with graphs. *Numer. Math.* **1995**, *1*, 269–271. [CrossRef]
12. Davis, J. Mosaics of scenes with moving objects. In Proceedings of the IEEE Computer Society Conference on Computer Vision & Pattern Recognition, Santa Barbara, CA, USA, 23–25 June 1998.
13. Chon, J.; Kim, H.; Lin, C. Seam-line determination for image mosaicking: A mismatch and the global cost. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 86–92. [CrossRef]
14. Zhang, J.; Sun, M.; Zhang, Z. Automated seamline detection for orthophoto mosaicking based on ant colony algorithm. *Geomat. Inf. Sci. Wuhan Univ.* **2009**, *34*, 675–678.
15. Gracias, N.; Mahoor, M.; Negahdaripour, S.; Gleason, A. Fast image blending using watersheds and graph cuts. *Image Vis. Comput.* **2009**, *27*, 597–607. [CrossRef]
16. Bielski, C.; Grazzini, J.; Soille, P. Automated morphological image composition for mosaicking large image data sets. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Barcelona, Spain, 23–28 July 2007.
17. Soille, P. Morphological image compositin. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 673–683. [CrossRef] [PubMed]
18. Bellman, R. On a routing problem. *Q. Appl. Math.* **1958**, *16*, 87–90. [CrossRef]
19. Duplaquet, L. Building large images mosaics with invisible seam-lines. *Proc. SPIE* **1998**, *3387*, 369–377.
20. Gonzalez, R.; Woods, R. *Digital Image Processing*, 3rd ed.; Publishing House of Electronics Industry: Beijing, China, 2011.
21. Cheng, X. *Research on Fast Produce of Orthophoto with UAV Sequence Images*; Wuhan University: Wuhan, China, 2011.



sensors

MDPI

Article

The Development of an Open Hardware and Software System Onboard Unmanned Aerial Vehicles to Monitor Concentrated Solar Power Plants

Francisco Javier Mesas-Carrascosa *, Daniel Verdú Santano, Fernando Pérez Porras, José Emilio Meroño-Larriva and Alfonso García-Ferrer

Department of Graphic Engineering and Geomatics, Campus de Rabanales, University of Córdoba, Córdoba 14071, Spain; p12vesad@uco.es (D.V.S.); perezporras.fernando@gmail.com (F.P.P.); ir1melaj@uco.es (J.E.M.-L.); agferrer@uco.es (A.G.-F.)

* Correspondence: fjmesas@uco.es; Tel.: +34-957-218-537

Academic Editors: Felipe Gonzalez Toro and Antonios Tsourdos

Received: 20 April 2017; Accepted: 5 June 2017; Published: 8 June 2017

Abstract: Concentrated solar power (CSP) plants are increasingly gaining interest as a source of renewable energy. These plants face several technical problems and the inspection of components such as absorber tubes in parabolic trough concentrators (PTC), which are widely deployed, is necessary to guarantee plant efficiency. This article presents a system for real-time industrial inspection of CSP plants using low-cost, open-source components in conjunction with a thermographic sensor and an unmanned aerial vehicle (UAV). The system, available in open-source hardware and software, is designed to be employed independently of the type of device used for inspection (laptop, smartphone, tablet or smartglasses) and its operating system. Several UAV flight missions were programmed as follows: flight altitudes at 20, 40, 60, 80, 100 and 120 m above ground level; and three cruising speeds: 5, 7 and 10 m/s. These settings were chosen and analyzed in order to optimize inspection time. The results indicate that it is possible to perform inspections by an UAV in real time at CSP plants as a means of detecting anomalous absorber tubes and improving the effectiveness of methodologies currently being utilized. Moreover, aside from thermographic sensors, this contribution can be applied to other sensors and can be used in a broad range of applications where real-time georeferenced data visualization is necessary.

Keywords: UAV; monitoring; open hardware; concentrated solar power

1. Introduction

The constantly growing global energy demand and increasing effects of climate change have resulted in the promotion of renewable energy sources to reduce CO_2 levels [1]. Solar technologies are a promising renewable resource because of their ever-increasing output efficiencies and ability to be used in a variety of locations [2]. Solar energy can be converted into electrical energy with different technologies such as photovoltaic (PV) panels [3], concentrated solar power (CSP) [4] and concentrator photovoltaics (CPV) [5]. Among them, CSP plants are a promising source of renewable energy to be used for predictable utility-scale power generation [6]. CSP plants utilize heat generated by solar irradiation concentrated on a small area. Currently, there are four available CSP technologies: parabolic trough collectors (PTC), solar power towers (SPT), linear Fresnel reflectors (LFR) and parabolic dish systems (PDS). Of the four, PTC is the most commercialized CSP technology to date [7], focusing direct solar radiation onto a focal line of the collector's axis [8]. In a PTC CSP plant, the mirrors concentrate the sun's rays on absorber tubes and the working fluid, flowing inside the tubes, absorbs solar energy by convection heat transfer. To reduce heat losses, the absorber tubes are enclosed

by an anti-reflective coated borosilicate glass envelope [9] (Figure 1). Owing to weather conditions and extremely non-uniform heat flux, absorber tubes are subjected to thermal stress resulting in the rupturing of the glass envelopes, causing heat loss [10]; this presents a problematic challenge. Papaelias et al. (2016) [6] analyzed and evaluated non-destructive inspection techniques for CSP plants. These techniques are grouped into liquid penetrant inspection (LPI), magnetic particle inspection (MPI), magnetic flux leakage (MFL) and visual inspection (VI). LPI is a visual technique based on spreading special dyes over the area for inspection. It is very sensitive to small defects and the inspection is fast, but due to the large quantity of components in a plant, a considerable amount of time is needed for thorough inspection. The MPI technique magnetizes the surface area of the tubes, which has been previously sprayed with ferrous particles for inspection. The MFL technique is based on magnetizing the ferrous components for inspection with a magnetic field. Finally, VI of structural components is performed by personnel walking through the plant or by pipe-crawling inspection robots. LPI and MPI techniques cannot be used for absorber tube inspection due to the presence of the glass envelope and MFL is not suitable for inspecting absorber tubes of austenitic stainless steel alloy. VI is also ineffective as there are several kilometers of absorber tubes in a plant to be inspected. However, as the thermal losses are correlated directly with the glass of the absorber tubes, which can be measured using a thermal sensor [11,12], a more efficient alternative to VI inspection is to use robots. In CSP plants, depending on the inspection requirements, these robots can be either ground or climbing robots [13]. VI inspections use mainly RGB sensors to evaluate geometrical characteristics [14,15] and thermal sensors to detect heat loss [16,17].

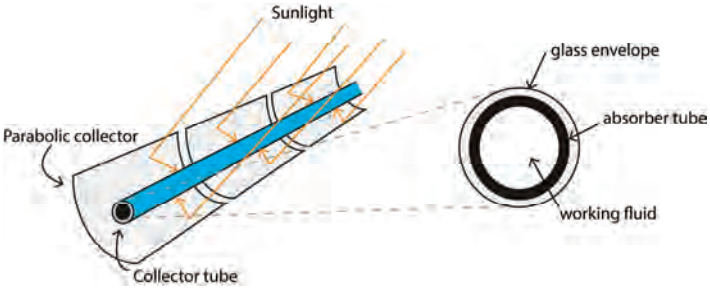


Figure 1. Illustration of functioning parabolic collector and absorber tube.

In the last decade, professionals and researchers' interest in using unmanned aerial vehicles (UAVs) to replace human activities of a dirty, dangerous and/or dull nature has increased [18]. In this context, the use of UAVs has become widespread in civil applications such as precision agriculture [19,20], emergencies [21], archaeology [22,23], traffic monitoring, [24,25] and visual inspection, [26,27] among others. Relative to industrial inspection, UAVs have been successfully used in monitoring power lines [28], gas pipelines [29] and photovoltaic plants [30]. To our knowledge, no detailed investigation has been conducted regarding the use of UAVs to monitor absorber tube heat loss in CSP plants using PTC technology.

To date, the main problems of current implementations of UAVs are the limited flight autonomy and the size-to-payload ratio [31]. Recent progress in electronics, wireless communications and the production of small-sized sensors offer new opportunities to monitor and control crops, cities, homes and the environment [32]. In this context, the design and implementation of a light-weight payload for UAVs can facilitate CSP plant inspections [33,34]. While there are complete thermographic systems already integrated with UAVs, they are not only expensive, but also closed systems that have little capacity for modification and variation [35]. Open source hardware (OSH) can be an alternative to closed systems when customization is necessary. Although OSH is quite new compared to open source software (OSS) [36], it is already being used for different applications in agriculture [37], management sensors [38], monitoring heritage buildings [39] and chemical instrumentation [40], among others.

In reference to UAVs, some researchers have developed unmanned aerial platforms based on OSH autopilots [41,42] and others have used OSH to integrate and use sensors for data-collection needs [43].

This manuscript describes a system that integrates information registered by a thermographic sensor and geolocation information from OSH components to be accessible in real time by users independently of the type of device used to monitor the inspection (laptop, smartglasses, smartphone or tablet) and its operating system. Its practicality is demonstrated in its application of monitoring absorber tube heat loss in a PTC CSP plant.

The article is divided in the following sections: in Section 2, we present the system; in Section 3, we describe the technical implementation; in Section 4, some field tests and their results are presented, followed by a conclusion.

2. System Overview

Figure 2 shows the conceptual model of the system. It has two main components (Figure 2a): the main system, and a camera sensor connected to the developed device, called Khepri, using OSH. In this project, a thermal sensor was used, but it is possible to use Red-Green-Blue (RGB) or multispectral sensors for other applications. Meter accuracy is sufficient enough to geolocate the thermographic video, as it allows the thermal sensor to be placed over the Khepri. Both components work synchronously to generate two principal data packages (Figure 2b). The first package is the thermal video registered by the sensor, and the second is the georeferenced information generated by the Khepri. These two data packages are processed to display three windows on a device screen (Figure 2c). The first window shows the information registered by the thermal sensor, the second georeferences the system on a cartograph, and the third shows an artificial horizon using orientation information. With this information, users will be able to locate incidents at the plant once they are detected through the position and orientation provided by the Khepri.

The information presented in the windows (Figure 2c) is generated and diffused in real time and accessible by various devices (laptop, tablet, smartphone, smartglasses) regardless of the operating system (Figure 2d). Finally, the main system can be used as the payload of an UAV, as well as installed on a terrestrial vehicle or used by personnel (Figure 2e). The electronic components of the Khepri were chosen after considering the restrictive load capacity of the three aforementioned platforms in both weight and dimensions. All selected components were based on OSH.

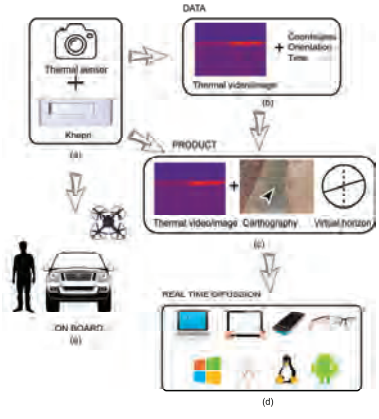


Figure 2. Conceptual model of the system: (**a**) Thermal sensor connected to the Khepri; (**b**) Data package generated by the system; (**c**) Thermal information and georeferentiation of the system over cartograph; (**d**) Real-time diffusion regardless of device and operating system and (**e**) Working onboard multiple platforms.

2.1. Selection of Sensors

Table 1 lists the electronic components and Figure 3 shows a diagram of all the electronic components used in this project. In this study, the Raspberry Pi 2 Model B (Raspberry Pi Foundation, Cambridge, United Kingdom) [44] was used as a single-board computer and the Arduino UNO R3 (Arduino AG, Ivrea, Italy) [45] was used as the microcontroller board. The Raspberry Pi 2 Model B contains a system-on-a-chip (SoC) Broadcom BCM2836 (Broadcom, Irvine, CA, USA), which includes a 900 MHz ARM Cortex-A7, 1 Gb synchronous dynamic random access memory (SDRAM), four Universal Serial Bus (USB) ports, one high-definition multimedia interface port (HDMI), one micro secure digital (SD) card to store information, and one Ethernet port. The Raspberry Pi stores the video registered by the connected sensor. The Arduino UNO R3 is a microcontroller board based on the ATmega328 (Atmel, San José, CA, USA) microcontroller, which manages and arranges the onboard sensor data. It has the minimal requirements for the study, containing 14 digital input/output pins, six of which can be used as pulse width modulation outputs, six analogue inputs, an ATmega328 microcontroller with an 18 MHz crystal oscillator, a USB connection, a power jack, an In-Circuit Serial Programming (ICSP) header, 32 KB of flash memory, 2 KB of SRAM and 1 KB of EEPROM (electrically erasable programmable read-only memory).

Table 1. Khepri electronic components.

Title 1	Weight (gr)	Dimension (mm)
Raspberry Pi 2 Model B	45	85.6 × 56.5
Arduino UNO R3	25	68.6 × 53.4
GPS Shield ublox NEO-6M	32	61.4 × 53.3 × 16
Accelerometer ADXL 345	5	3 × 5 × 1
DS1302 clock	10	9.91 × 7.87 × 4.45
Modem USB	24	88 × 28 × 10
LCD Screen 1602	50	80 × 36 × 13.5
Lipo Battery	79	125 × 7 × 21

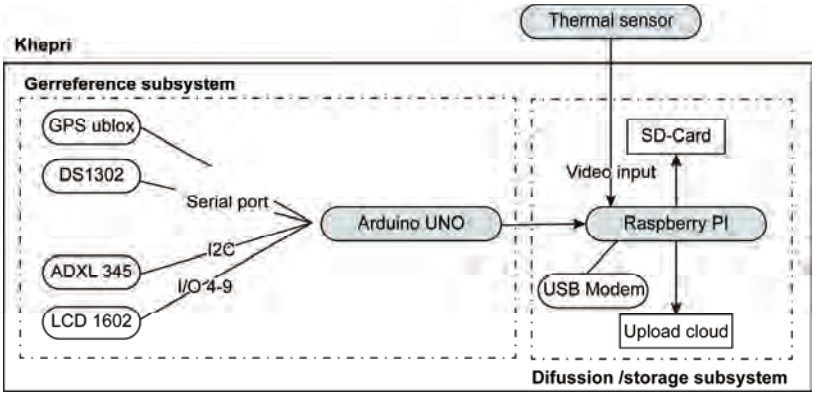


Figure 3. Conceptual model.

A GPS Shield based on the ublox NEO-6M (Ublox, Thalwil, Switzerland) receiver module [46] is used to obtain the coordinates of the device. The positioning engine uses 50 channels and measures time based on Time To First Fix (TTFF) of less than 1 second, searching in parallel both time and frequency space, allowing instant satellite discovery. The horizontal position accuracy is 2.5 m, which is sufficient for this study. The 3-axis digital accelerometer ADXL 345 (SparkFun electronics, Boulder, CO, USA) [47] was used to measure static acceleration as tilt and dynamic acceleration to determine

the attitude of the system. It has an output resolution equal to 10 bits and a sensitivity deviation from ideal equal to ±1%. Each data package was temporally referenced with a DS1302 (Maxim Integrated, San José, CA, United States) real-time clock [48]. It operates with a supply of 2 to 5.5 V and can operate between 0 and +70 °C. It contains a real-time clock/calendar and 31 bytes of static RAM.

A USB modem was used to upload information generated by the system to the cloud to be diffused to other users. Finally, a LCD Screen 1602 (Shenzhen Eone Electronics CO., Ltd., Shenzhen, China) [49] was used to display battery percentage and the verification of all the components. It operates with a supply of 5 V. A rechargeable 7.4 V LiPo Battery with 1400 mAh capacity was used to power the system.

All the electronic components were distributed inside a polylactic acid (PLA) case printed by a 3D printer. PLA material is a biodegradable polymer harder than other materials such as acrylonitrile–butadiene–styrene (ABS), melting at lower temperatures (from 180 to 220 °C), with a glass transition temperature around 60 °C. The PLA case dimensions are 132 × 132 × 47 mm and its weight is equal to 50 gr. The total weight of the device is 375 gr in addition to the thermal sensor weight, in this case, 263 gr.

2.2. Technical Implementation

As Figure 3 shows, the Khepri is composed of two main subsystems: the Arduino and the Raspberry Pi connected to an external sensor. The Arduino subsystem manages geo-information data packages generated by GPS and Inertial Measurement Unit (IMU). The GPS Shield is connected to digital port 0 and 1 of the Arduino UNO board. From the NMEA (National Marine Electronics Association) string, received latitude and longitude are extracted. This information is used to locate the system on a cartograph in real time. An accelerometer ADXL 345 is connected to digital port 2 and analogical port 4 and 5 of the Arduino UNO board. Data from ADXL 345 are converted to yaw, pitch and roll angles, allowing the orientation of the system to be known. Both the thermal sensor and the Khepri work simultaneously, displaying their respective data packages on screen. Using the coordinates obtained by GPS and orientation by IMUs, the Khepri's location is approximately (≈ 1m) registered to indicate the approximate location of the thermal sensor. As such, a user will be able to see, with an external device, live onscreen thermal video in a window along with its geolocation on a cartograph in another window.

As the data is being rendered, the Raspberry Pi collects the information from the thermal sensor and from the Arduino subsystem and makes a backup copy on a SD card. From the thermal sensor, a thermographic video is stored while the Arduino system creates a log file with geolocation information, which allows the video to be georeferenced for later use if needed.

Two methods have been adopted for the cartographic display: API Google Maps and Leaflet. Public API Google Maps is a set of object classes, functions and data structures applied using JavaScript or other scripting languages [50]. The latest version is supported by traditional web browsers or those implemented on mobile devices. API Google Maps uses cartography from Google Maps. On occasion, the resolution of this cartography is insufficient, either in temporal or spatial terms. For this reason, a web mapping application was developed using the JavaScript library Leaflet [51]. In this research project, an orthomosaic was produced using RGB images registered by an UAV flight over a CSP plant with a spatial resolution equal to 5 cm. This orthomosaic was published through a Web Map Service (WMS) using Geoserver [52] and displayed by Leaflet. With the availability of these options, users will be able to choose the most suitable mapping source.

In short, the Khepri will display three windows: one with thermographic information, the second with a map viewer and the third with a virtual horizon. These windows will be distributed on the screen.

To diffuse information from sensors or devices onboard UAVs in real time, most applications use a radio frequency transmission system that connects the UAV with a Ground Control Station (GCS) [53,54]. Images or video registered by the onboard sensor are transmitted to the GCS and further

relayed to other users. On occasion, a lost transmission data link can occur and result in the loss of information sent from the UAV to the GCS [55]. In this developed solution, the system itself has the function of sending the information in real time without needing a GCS to communicate with the users as the Khepri transmits the information of interest over the Internet via USB modem. Therefore, it is necessary to have a solid mobile data link to work in real time, which is maintained in these plants for security reasons. In other cases, the inspection can be analyzed later using the information registered in the SD card.

Two user profiles have been defined for the system: administrator and invited user. The administrative user has system configuration privileges, for example, to control how the thermal sensor works or to interrupt data transmission. These profiles can be applied in two different scenarios. In the first (Figure 4a), while the system is on terrain, the administrator can connect a screen to the Khepri through the HDMI port, accessing the windows and functions of the system. In the second scenario, the administrator can connect remotely, accessing the Khepri through Chrome Remote Desktop using a Chrome browser with an Internet connection (Figure 4b). In this case, if there are invited users, they will be able to access the thermal inspection. This is because the system, when the administrator chooses, shares the desktop screen live using the application Screenleap (Figure 4b). Invited users are linked via Uniform Resource Locator (URL) for view-only access, which allows the system to be accessible regardless of the type of device and/or operating system. This solution guarantees the interoperability of devices and operating systems. It is also worth noting that the number of invited users is unlimited. This multi-access to the Khepri allows users to work at different locations and with different objectives simultaneously. For example, the UAV operator can perform the flight with focus on guaranteeing the security of the UAV mission while, simultaneously, other users are managing the thermal inspection, whether at the same plant or in another country. Moreover, this activity can be performed by more than one user to guarantee the quality of the visual analysis.

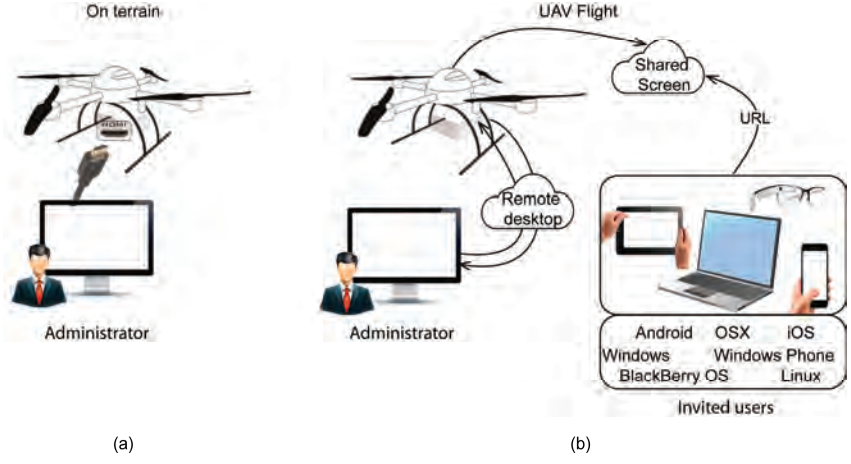


Figure 4. User profiles and cases of use: (**a**) The administrator sets the system on terrain and (**b**) The system onboard an unmanned aerial vehicle (UAV) flight being accessed by the administrator through a remote desktop and invited users with different operating systems accessing inspection via the URL.

3. Results

The prototype PLA case measures 132 × 122 × 45 mm, which allows it to be installed in an UAV gimbal (Figure 5a). It was designed to be ergonomic, resistant to deformation and low weight. All the components of the Khepri were distributed in such a way that the center of mass was close to the center of the PLA case (Figure 5b).

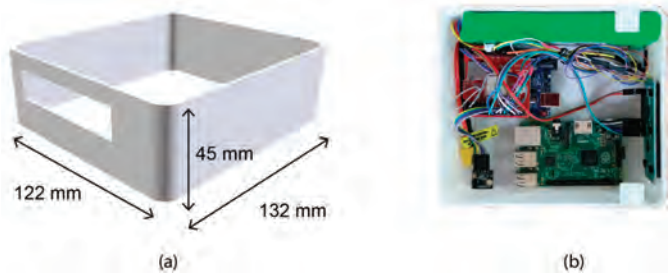


Figure 5. Polylactic acid (PLA) case: (**a**) Design of prototype and (**b**) Distribution of components.

An UAV flight campaign over a CSP plant was executed with two objectives: to verify that the system works properly, and to test different altitudes Above Ground Level (AGL) and cruising speeds to define the specifications for use. A quadrocopter, model MD4-1000 (Microdrones GmbH, Siegen, Germany), was used as the UAV to perform the flights above the La Africana PTC CSP plant, located in Córdoba, Spain (CSP plant details in [56]) (Figure 6a). This CSP plant has 104 km of absorber tubes distributed on the surface. Currently, at this CSP plant, a thermographic inspection is performed by a technician with a thermal gun walking through the solar field and inspecting the absorber tubes. Once a problem is detected, the incident and its location are noted. With this methodology, an inspection takes one week for an expert technician and two if the inspection is done by a junior technician, and is only performed once a month or every two months. These numbers highlight the ineffectiveness of the method.

Figure 6. (**a**) The UAV performing an inspection on a concentrated solar power (CSP) plant; (**b**) Front view of the system onboard the UAV; (**c**) Back view of the system onboard.

MD4-1000 is a vertical take-off and landing platform that is entirely carbon designed and equipped with 4 × 250 W gearless brushless motors powered by a 22.2 V battery. Maximum cruising and climb speeds are 12 and 7.5 m/s, respectively. MD4-1000 can operate in remote control mode or automatically. The payload was the thermographic sensor Gobi 640 GiGe (Xenics nv, Leuvem, Belgium) connected to the Khepri (Figure 6b,c). The Gobi 640 GiGe is an uncooled micro-bolometer sensor that operates in the 8–14 μm spectral band with an image size of 640 × 480 pixels and a focal length of 18 mm. It is calibrated to operate from −25 to 125 °C with a resolution equal to 50 mK. Figure 6b,c show a front and back view of the Gobi 640 GiGe and the Khepri onboard the UAV.

Figure 7 shows a screen capture from Khepri during a thermal inspection on the PTC CSP plant. The left window contains the information from the thermal sensor, which shows an anomaly on an absorber tube. Simultaneously, in the right window, the system is located on an orthomosaic of the CSP plant. In this example, an orthomosaic produced by an RGB sensor onboard an UAV and published by Leaflet is used to position the system.

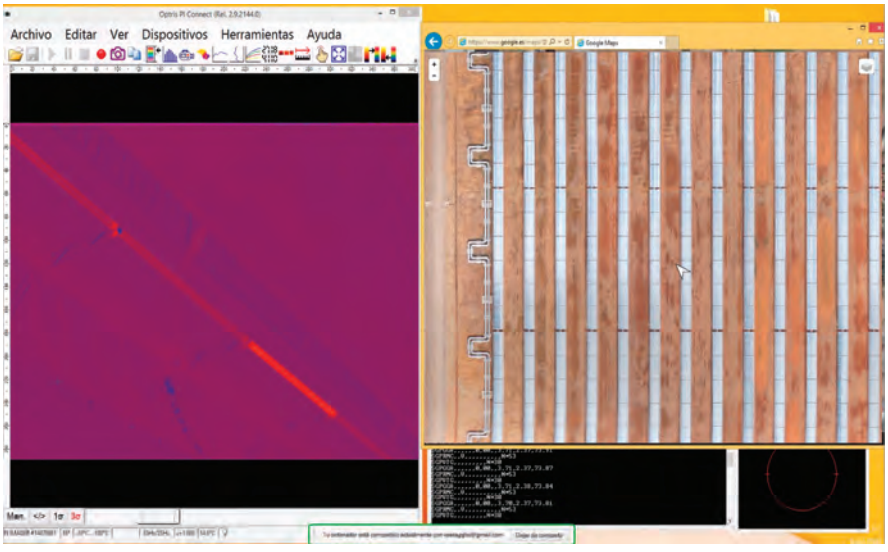


Figure 7. Screen details from Khepri during a thermal inspection on a CSP.

In this study, a set of flight missions were flown at altitudes of 20, 40, 60, 80, 100 and 120 m AGL. The UAV flew in autonomous mode with each flight mission uploaded previously. Each altitude AGL is linked to a specific ground sample distance (GSD) value (Table 2). Additionally, three different cruising speed settings were used: 5, 7 and 10 m/s. Maximum velocity was set to ensure the video did not undergo blurring caused by platform movements. All flights were performed under the same conditions, the wind speed being equal to 1 m/s. Table 2 summarizes, for each altitude AGL, the GSD of the video frame and, for each cruising speed, the time, in hours, taken to inspect the entire plant. Duration is expressed apart from the time spent taking off and landing and time to change batteries. In this study, the GSD ranged from 1.9 cm × pixel^{-1} at 20 m AGL, to 11.3 cm × pixel^{-1} at 120 m AGL. Inspection duration ranged from 0.8 h flying at 120 m AGL and cruising speed equal to 10 m/s, to 5.8 h flying at 20 m AGL and cruising speed equal to 5 m/s. Table 2 demonstrates that, as altitude AGL increases, inspection duration decreases because each video frame covers more area, monitoring several lines of absorber tubes simultaneously in each lap. This occurs regardless of the cruising speed of the UAV. Altitude AGL and duration inspection show a logarithmic correlation for all cruising speeds equal to 0.869 (Figure 8). Therefore, as altitude AGL was increased, inspection duration

logarithmically decreased. In addition, cruising speed was not relevant in detecting anomalies during the inspection due to the great temperature differential of a broken absorber tube.

Table 2. Summary of the unmanned aerial vehicle (UAV) flights.

		UAV Inspection Time (Hours)		
Altitude AGL [1] (m)	**GSD [2] (cm)**	**5 m/s**	**7 m/s**	**10 m/s**
20	1.9	5.8	4.1	2.9
40	3.8	5.8	4.1	2.9
60	5.7	3.0	2.1	1.5
80	7.6	3.0	2.1	1.5
100	9.4	2.0	1.4	1.0
120	11.3	1.6	1.1	0.8

[1] Altitude AGL: Altitude above ground level. [2] GSD: Ground sample distance.

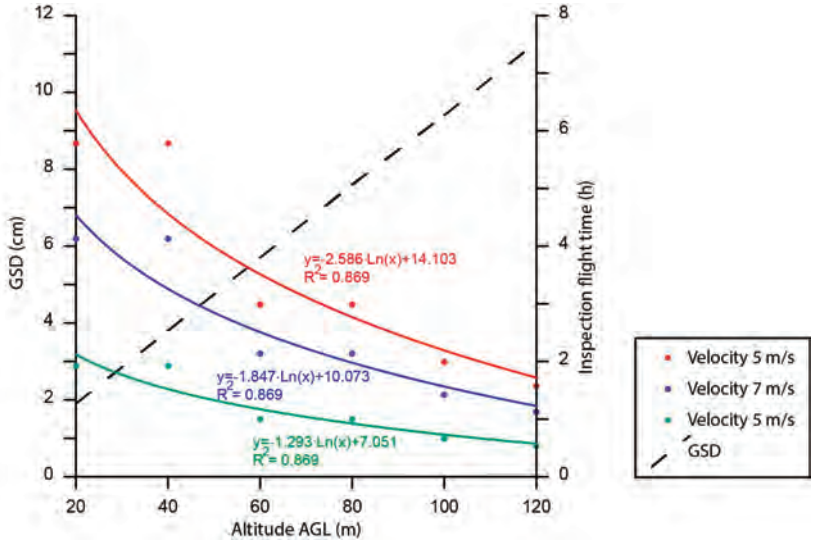


Figure 8. Inspection flight time considering cruising speed and altitude AGL.

As for visual inspection, Figure 9 shows an example of a video frame at 20 m AGL (Figure 9(a1,a2)) and 100 m AGL (Figure 9(b1,b2)). All considered altitudes AGL detected heat losses from absorber tubes satisfactorily. However, due to the wide range of inspection times, it is necessary to optimize the flight parameters for best results. All positive inspections were caused by the high temperature around the broken absorber tubes. Flying at 20 m AGL (Figure 9(a1)), the thermographic video showed bellows between absorber tubes, making it possible to identify the thermal section along the tube. This high level of detail requires longer UAV flight times at low altitude AGL. Increasing altitude AGL (Figure 9(b1)) diffuses the image, but the anomaly is still detected, and inspection time is decreased. In both cases, if the histograms of the frames are stretched, the anomalies in absorber tubes are clearly shown independently of altitude AGL (Figure 9(a2,b2)).

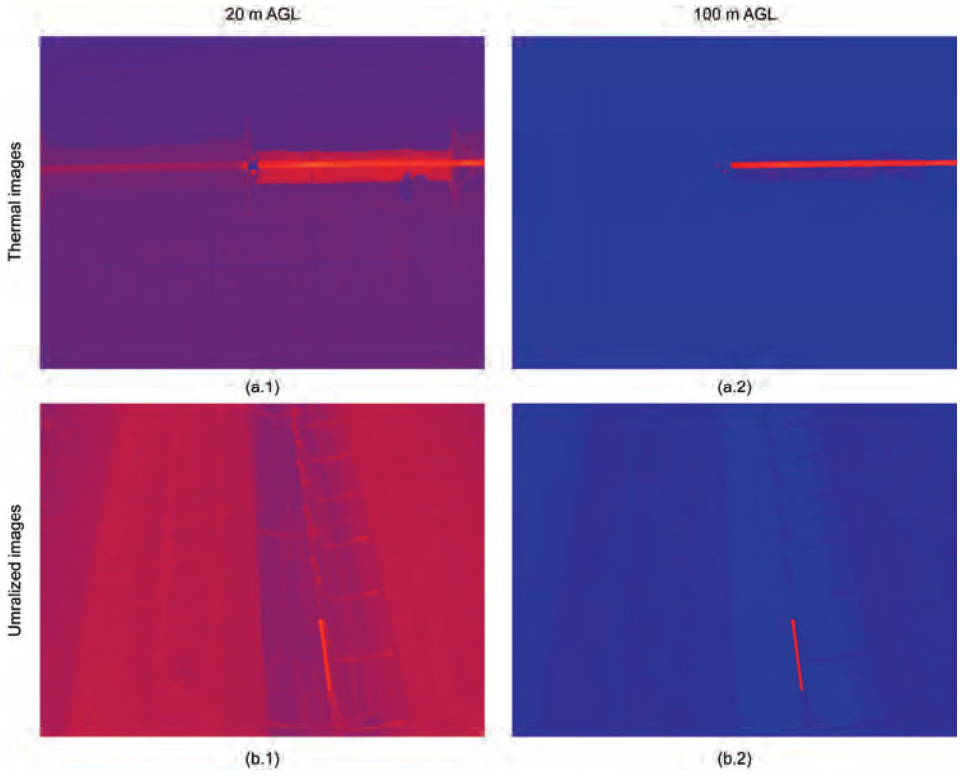


Figure 9. Examples of the UAV inspection at (**a**) 20 m and (**b**) 100 m AGL; (**1**) un-stretched and (**2**) stretched histograms.

In summary, as a strategy in a PTC CSP plant inspection, the system can fly at low and high altitude AGL jointly, optimizing UAV flights and therefore time inspection. Firstly, a UAV flight can be executed at a high altitude AGL, detecting absorber tubes which show anomalies, as well as obtaining the percentage of anomalous tubes of the entire CSP plant. Once these are located, they can be viewed at very high spatial resolution and analyzed with precision by flying at low-altitude AGL in specific areas. Using this strategy, it is possible to optimize the number and duration of UAV flights without losing information.

All tests were observed in real-time by different users in different locations and with different devices. The invited users were located both at the CSP plant and our university department. The invited users followed the inspection simultaneously using tablets and smartphones with android and iOS operating systems, laptops with Windows, Linux and OSX and android smartglasses. To connect to the inspection using mobile devices, a QR code was provided with the URL where the Khepri windows were being displayed. Connection tests were successful, with no interruptions occurring during the connection. The administrative profile interacted with the Khepri both directly, before the UAV flights started, and from our university department through a connection by remote desktop.

Considering how van Blyenburgh [18] established that UAVs ought to be used for activities which are "dull, dirty and dangerous", and walking inspections are categorically dull, recurring mode UAV flights at a PTC CSP plant fits van Blyenburgh's defined use while at the same time intensifying the cadence of thermal inspection at the plant. Table 3 shows the percentage of improvement in inspection time with the developed system due to its ultra-high resolution and speed compared to

current methods based on walking inspections at the CSP plant. The improvement of time inspection ranges from 85.6% flying at 20 m AGL and a cruising speed of 5 m/s to 98.0% flying at 120 m AGL and cruising speed equal to 10 m/s. Independently of altitude AGL and cruising speed used, the developed system allows a more productive method without losing information compared to current methods.

Table 3. Improved productivity of UAV inspection work versus manual inspection.

Altitude AGL (m)	5 m/s (%)	7 m/s (%)	10 m/s (%)
20	85.6	89.7	92.8
40	85.6	89.7	92.8
60	92.5	94.7	96.3
80	92.5	94.7	96.3
100	95.0	96.4	97.5
120	96.1	97.2	98.0

4. Conclusions

A system onboard an UAV to monitor CSP plants using OSH and OSS was developed with customization capabilities depending on use. The thermal inspection of a CSP plant was successful at different altitudes AGL and cruising speeds, and allowed the possibility to observe the inspection using different devices with different operating systems. As a result, we propose flights at two altitudes AGL. Firstly, UAV flights at 100–120 m AGL can locate incidents on absorber tubes around the plant. Once they are located, a focused inspection at low-altitude AGL, 20 m AGL, can be performed to analyze the problem in detail. With this strategy, inspection time is optimized.

The proposed methodology is more efficient than traditional methods based on walking with a thermal gun through the solar plant. Moreover, the proposed system will allow the numbers of inspections to be increased, improving the efficiency of CSP plants.

The entire system has been developed using OSH and OSS, allowing for customization depending on application. In the future, RGB or multispectral sensors will be able to be used in other scenarios such as precision agriculture. Moreover, board computers with higher capacities will be able to be used on the Khepri to process video or images in real time to detect anomalies and send alerts, among other possible uses. This automation would permit a reduction in the number of experts required to analyze the thermal video to detect leaks.

Acknowledgments: This research was funded by Khepri project supported by CDTI: Centro para el Desarrollo Tecnológico Industrial, FEDER Funds: Fondo Europeo de Desarrollo Regional and CTA: Corporación Tecnológica de Andalucía. The authors would like to thank Magtel Corporation (Spain) for providing access to La Africana CSP plant.

Author Contributions: F.J.M.-C. and D.V.S. conceived and designed the experiments; F.J.M.-C. and D.V.S. performed the experiments; F.J.M.-C., D.V.S., F.P.P., J.E.M.-L. and A.G.-F. analyzed the data; F.J.M.-C. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lewis, N.S.; Nocera, D.G. Powering the planet: Chemical challenges in solar energy utilization. *Proc. Natl. Acad. Sci. USA* **2006**, *103*, 15729–15735. [CrossRef] [PubMed]
2. Devabhaktuni, V.; Alam, M.; Depuru, S.S.R.; Green II, R.C.; Nims, D.; Near, C. Solar energy: Trends and enabling technologies. *Renew. Sustain. Energy Rev.* **2013**, *19*, 555–564. [CrossRef]
3. Bialasiewicz, J.T. Renewable energy systems with photovoltaic power generators: Operation and modeling. *IEEE Trans. Ind. Electron.* **2008**, *55*, 2752–2758. [CrossRef]
4. Zhang, H.L.; Baeyens, J.; Degrève, J.; Cacères, G. Concentrated solar power plants: Review and design methodology. *Renew. Sustain. Energy Rev.* **2013**, *22*, 466–481. [CrossRef]

5. Ho, C.K. A review of high-temperature particle receivers for concentrating solar power. *Appl. Therm. Eng.* **2016**, *109*, 958–969. [CrossRef]
6. Papaelias, M.; Cheng, L.; Kogia, M.; Mohimi, A.; Kappatos, V.; Selcuk, C.; Constantinou, L.; Muñoz, C.Q.G.; Marquez, F.P.G.; Gan, T.-H. Inspection and structural health monitoring techniques for concentrated solar power plants. *Renew. Energy* **2016**, *85*, 1178–1191. [CrossRef]
7. Morin, G.; Dersch, J.; Platzer, W.; Eck, M.; Häberle, A. Comparison of linear fresnel and parabolic trough collector power plants. *Solar Energy* **2012**, *86*, 1–12. [CrossRef]
8. Fernández-García, A.; Zarza, E.; Valenzuela, L.; Pérez, M. Parabolic-trough solar collectors and their applications. *Renew. Sustain. Energy Rev.* **2010**, *14*, 1695–1721. [CrossRef]
9. Almanza, R.; Lentz, A.; Jiménez, G. Receiver behavior in direct steam generation with parabolic troughs. *Solar Energy* **1997**, *61*, 275–278. [CrossRef]
10. Reddy, K.S.; Kumar, K.R.; Satyanarayana, G.V. Numerical investigation of energy-efficient receiver for solar parabolic trough concentrator. *Heat Transf. Eng.* **2008**, *29*, 961–972. [CrossRef]
11. Price, H.; Forristall, R.; Wendelin, T.; Lewandowski, A.; Moss, T.; Gummo, C. Field survey of parabolic trough receiver thermal performance. In Proceedings of the ASME 2006 International Solar Energy Conference, Denver, CO, USA, 8–13 June 2006; pp. 109–116.
12. Yaghoubi, M.; Ahmadi, F.; Bandehee, M. Analysis of heat losses of absorber tubes of parabolic through collector of shiraz (iran) solar power plant. *J. Clean Energy Technol.* **2013**, *1*, 33–37. [CrossRef]
13. Maurtua, I.; Susperregi, L.; Fernández, A.; Tubío, C.; Perez, C.; Rodríguez, J.; Felsch, T.; Ghrissi, M. Mainbot—Mobile robots for inspection and maintenance in extensive industrial plants. *Energy Procedia* **2014**, *49*, 1810–1819. [CrossRef]
14. García-Cortés, S.; Bello-García, A.; Ordóñez, C. Estimating intercept factor of a parabolic solar trough collector with new supporting structure using off-the-shelf photogrammetric equipment. *Appl. Energy* **2012**, *92*, 815–821. [CrossRef]
15. Pottler, K.; Ulmer, S.; Lüpfert, E.; Landmann, M.; Röger, M.; Prahl, C. Ensuring performance by geometric quality control and specifications for parabolic trough solar fields. *Energy Procedia* **2014**, *49*, 2170–2179. [CrossRef]
16. Ibarguren, A.; Molina, J.; Susperregi, L.; Maurtua, I. Thermal tracking in mobile robots for leak inspection activities. *Sensors* **2013**, *13*, 13560–13574. [CrossRef] [PubMed]
17. Abedini-Sanigy, M.H.; Ahmadi, F.; Goshtasbirad, E.; Yaghoubi, M. Thermal stress analysis of absorber tube for a parabolic collector under quasi-steady state condition. *Energy Procedia* **2015**, *69*, 3–13. [CrossRef]
18. Van Blyenburgh, P. UAVs: An overview. *Air Space Eur.* **1999**, *1*, 43–47. [CrossRef]
19. Zhang, C.; Kovacs, J.M. The application of small unmanned aerial systems for precision agriculture: A review. *Precis. Agric.* **2012**, *13*, 693–712. [CrossRef]
20. Mesas-Carrascosa, F.-J.; Torres-Sánchez, J.; Clavero-Rumbao, I.; García-Ferrer, A.; Peña, J.-M.; Borra-Serrano, I.; López-Granados, F. Assessing optimal flight parameters for generating accurate multispectral orthomosaicks by UAV to support site-specific crop management. *Remote Sens.* **2015**, *7*, 12793–12814. [CrossRef]
21. Mohammed, F.; Idries, A.; Mohamed, N.; Al-Jaroodi, J.; Jawhar, I. UAVs for smart cities: Opportunities and challenges. In Proceedings of the 2014 International Conference on Unmanned Aircraft Systems (ICUAS), Orlando, FL, USA, 27–30 May 2014; pp. 267–273.
22. Xu, Z.; Wu, L.; Shen, Y.; Li, F.; Wang, Q.; Wang, R. Tridimensional reconstruction applied to cultural heritage with the use of camera-equipped UAV and terrestrial laser scanner. *Remote Sens.* **2014**, *6*, 10413–10434. [CrossRef]
23. Mesas-Carrascosa, F.-J.; Notario García, M.D.; Meroño de Larriva, J.E.; García-Ferrer, A. An analysis of the influence of flight parameters in the generation of unmanned aerial vehicle (UAV) orthomosaicks to survey archaeological areas. *Sensors* **2016**, *16*, 1838. [CrossRef] [PubMed]
24. Puri, A.; Valavanis, K.P.; Kontitsis, M. Statistical profile generation for traffic monitoring using real-time UAV based video data. In Proceedings of the 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6.
25. Kanistras, K.; Martins, G.; Rutherford, M.J.; Valavanis, K.P. Survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K.P., Vachtsevanos, G.J., Eds.; Springer: Dordrecht, The Netherlands, 2015; pp. 2643–2666.

26. Máthé, K.; Buşoniu, L. Vision and control for UAVs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors* **2015**, *15*, 14887–14916. [CrossRef] [PubMed]
27. Montambault, S.; Beaudry, J.; Toussaint, K.; Pouliot, N. On the application of vtol UAVs to the inspection of power utility assets. In Proceedings of the 2010 1st International Conference on Applied Robotics for the Power Industry, Montreal, QC, Canada, 5–7 October 2010; pp. 1–7.
28. Jones, D. Power line inspection—A UAV concept. In Proceedings of the 2005 The IEE Forum on Autonomous Systems, London, UK, 28–28 November 2005; p. 8.
29. Hausamann, D.; Zirnig, W.; Schreier, G.; Strobl, P. Monitoring of gas pipelines—A civil UAV application. *Aircr. Eng. Aerosp. Technol.* **2005**, *77*, 352–360. [CrossRef]
30. Grimaccia, F.; Aghaei, M.; Mussetta, M.; Leva, S.; Quater, P.B. Planning for PV plant performance monitoring by means of unmanned aerial systems (UAS). *Int. J. Energy Environ. Eng.* **2015**, *6*, 47–54. [CrossRef]
31. Rossi, M.; Brunelli, D.; Adami, A.; Lorenzelli, L.; Menna, F.; Remondino, F. Gas-drone: Portable gas sensing system on UAVs for gas leakage localization. In Proceedings of the IEEE SENSORS 2014 Proceedings, Valencia, Spain, 2–5 November 2014; pp. 1431–1434.
32. Mesas-Carrascosa, F.J.; Verdú Santano, D.; Meroño, J.E.; Sánchez de la Orden, M.; García-Ferrer, A. Open source hardware to monitor environmental parameters in precision agriculture. *Biosyst. Eng.* **2015**, *137*, 73–83. [CrossRef]
33. Leira, F.S.; Trnka, K.; Fossen, T.I.; Johansen, T.A. A ligth-weight thermal camera payload with georeferencing capabilities for small fixed-wing UAVs. In Proceedings of the 2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO, USA, 9–12 June 2015; pp. 485–494.
34. Bologna, F.; Mahatho, N.; Hoch, D.A. Infra-red and ultra-violet imaging techniques applied to the inspection of outdoor transmission voltage insulators. In Proceedings of the IEEE AFRICON, 6th Africon Conference in Africa, George, South Africa, 2–4 October 2002; Volume 592, pp. 593–598.
35. Kockara, S.; Halic, T.; Hudson, C.; Loney, A.; Crawford, A. Portable malignant lesion detection with low cost mobile infrared thermography. In Proceedings of the 2014 IEEE Innovations in Technology Conference, Warwick, RI, USA, 16–16 May 2014; pp. 1–5.
36. Faugel, H.; Bobkov, V. Open source hard- and software: Using Arduino boards to keep old hardware running. *Fusion Eng. Des.* **2013**, *88*, 1276–1279. [CrossRef]
37. Bitella, G.; Rossi, R.; Bochicchio, R.; Perniola, M.; Amato, M. A novel low-cost open-hardware platform for monitoring soil water content and multiple soil-air-vegetation parameters. *Sensors* **2014**, *14*, 19639–19659. [CrossRef] [PubMed]
38. Di Gennaro, S.F.; Matese, A.; Mancin, M.; Primicerio, J.; Palliotti, A. An open-source and low-cost monitoring system for precision enology. *Sensors* **2014**, *14*, 23388–23397. [CrossRef] [PubMed]
39. Mesas-Carrascosa, F.J.; Verdú Santano, D.; Meroño de Larriva, J.E.; Ortíz Cordero, R.; Hidalgo Fernández, R.E.; García-Ferrer, A. Monitoring heritage buildings with open source hardware sensors: A case study of the mosque-cathedral of córdoba. *Sensors* **2016**, *16*, 1620. [CrossRef] [PubMed]
40. Grinias, J.P.; Whitfield, J.T.; Guetschow, E.D.; Kennedy, R.T. An inexpensive, open-source usb arduino data acquisition device for chemical instrumentation. *J. Chem. Educ.* **2016**, *93*, 1316–1319. [CrossRef] [PubMed]
41. Doneus, M.; Wieser, M.; Verhoeven, G.; Karel, W.; Fera, M.; Pfeifer, N. Automated archiving of archaeological aerial images. *Remote Sens.* **2016**, *8*, 209. [CrossRef]
42. Sun, J.; Li, B.; Jiang, Y.; Wen, C.-Y. A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. *Sensors* **2016**, *16*, 1778. [CrossRef] [PubMed]
43. Sørensen, L.Y.; Jacobsen, L.T.; Hansen, J.P. Low cost and flexible UAV deployment of sensors. *Sensors* **2017**, *17*, 154. [CrossRef] [PubMed]
44. Raspberry Pi 2 Model B Datasheet. Available online: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/ (accessed on 16 February 2017).
45. Arduino Uno R3 Datasheet. Available online: https://www.arduino.cc/en/Main/ArduinoBoardUno (accessed on 16 February 2017).
46. Ublox Neo-6m Datasheet. Available online: https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf?utm_source=en%2Fimages%2Fdownloads%2FProduct_Docs%2FNEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf (accessed on 16 February 2017).
47. Accelerometer Adxl 345 Datasheet. Available online: http://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf (accessed 17 February 2017).

48. Ds1302 Clock Datasheet. Available online: https://datasheets.maximintegrated.com/en/ds/DS1302.pdf (accessed on 17 February 2017).
49. Lcd Screen 1602 Datasheet. Available online: https://cdn-shop.adafruit.com/datasheets/TC1602A-01T.pdf (accessed on 17 February 2017).
50. Udell, J. *Beginning Google Maps Mashups with Mapplets, KML, and Georss: From Novice to Professional*; Apress: New York, NY, USA, 2008.
51. Crickard, P. *Leaflet.js Essentials*; Packt Publishing Ltd.: Birmingham, UK, 2014; p. 165.
52. Iacovella, S. *Geoserver Cookbook*; Packt Publishing Ltd.: Birmingham, UK, 2014.
53. Taro, S.; Daichi, M.; Jun-ichi, M.; Yoshiharu, A.; Takumi, H.; Koich, S.; Jun-ich, T. Real-time hazard map generation using small unmanned aerial vehicle. In Proceedings of the 2008 SICE Annual Conference, Tokyo, Japan, 20–22 August 2008; pp. 443–446.
54. Chen, Y.M.; Dong, L.; Oh, J.S. Real-time video relay for UAV traffic surveillance systems through available communication networks. In Proceedings of the 2007 IEEE Wireless Communications and Networking Conference, Kowloon, China, 11–15 March 2007; pp. 2608–2612.
55. Chen, J.; Fei, Q.; Geng, Q. The design of data link for multi-UAVs. In Proceedings of the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, Nanchang, China, 26–27 August 2012; pp. 106–109.
56. Barcia, L.A.; Peón Menéndez, R.; Martínez Esteban, J.Á.; José Prieto, M.A.; Martín Ramos, J.A.; de Cos Juez, F.J.; Nevado Reviriego, A. Dynamic modeling of the solar field in parabolic trough solar power plants. *Energies* **2015**, *8*, 13361–13377. [CrossRef]

sensors

MDPI

Article

Automatic Hotspot and Sun Glint Detection in UAV Multispectral Images

Damian Ortega-Terol [1], David Hernandez-Lopez [2], Rocio Ballesteros [3] and Diego Gonzalez-Aguilera [1,*]

[1] Higher Polytechnic School of Ávila, University of Salamanca, 05003 Ávila, Spain; dortegat@gmail.com
[2] Institute for Regional Development (IDR), University of Castilla-La Mancha, Campus Universitario s/n, 02071 Albacete, Spain; david.hernandez@uclm.es
[3] Regional Centre of Water Research (CREA), University of Castilla-La Mancha, Carretera de las Peñas km 3.2, 02071 Albacete, Spain; Rocio.Ballesteros@uclm.es
* Correspondence: daguilera@usal.es; Tel.: +34-920-353-500

Received: 13 August 2017; Accepted: 13 October 2017; Published: 15 October 2017

Abstract: Last advances in sensors, photogrammetry and computer vision have led to high-automation levels of 3D reconstruction processes for generating dense models and multispectral orthoimages from Unmanned Aerial Vehicle (UAV) images. However, these cartographic products are sometimes blurred and degraded due to sun reflection effects which reduce the image contrast and colour fidelity in photogrammetry and the quality of radiometric values in remote sensing applications. This paper proposes an automatic approach for detecting sun reflections problems (hotspot and sun glint) in multispectral images acquired with an Unmanned Aerial Vehicle (UAV), based on a photogrammetric strategy included in a flight planning and control software developed by the authors. In particular, two main consequences are derived from the approach developed: (i) different areas of the images can be excluded since they contain sun reflection problems; (ii) the cartographic products obtained (e.g., digital terrain model, orthoimages) and the agronomical parameters computed (e.g., normalized vegetation index-NVDI) are improved since radiometric defects in pixels are not considered. Finally, an accuracy assessment was performed in order to analyse the error in the detection process, getting errors around 10 pixels for a ground sample distance (GSD) of 5 cm which is perfectly valid for agricultural applications. This error confirms that the precision in the detection of sun reflections can be guaranteed using this approach and the current low-cost UAV technology.

Keywords: UAV; hotspot; sun glint; image preprocessing; photogrammetry; remote sensing; flight planning and control; software development

1. Introduction

Nowadays, the proliferation of unmanned aerial systems, popularly known as "Unmanned Aerial Vehicle" (UAV), is a reality for local policy makers, regulatory bodies, mapping authorities, start-ups and consolidated companies. The number of developed UAVs has increased threefold from 2005 to present and, additionally, a relevant increase has been observed in the civil/commercial type of platforms, especially in 2012 and 2013 [1]. There are many uses and benefits of UAVs based on their own pilot system (autonomous or remotely controlled) and sensory to achieve accurate positioning and to acquire a great variety of data. By this binomial, UAVs are an efficient solution for the observation, inspection, measurement and monitoring of territory; ensuring better spatial, radiometric, temporal and spectral resolution than any manned aerial vehicle and satellite platform [2]. In particular, the spectral information acquired from UAV is really important for different applications such as precision agriculture [3,4], recognition of objects [5–7], environmental studies [8,9] and water analysis [10]. However, the images acquired from UAVs with small-format cameras usually suffer

from sun reflections problems such as sun glint or hotspot [11,12]. The sun glint effect is produced in specular surfaces such as water, glass or metal when the angle of incidence of the sun equals the angle of reflection, whereas the hotspot effect is produced due to a direct alignment between the sun, the camera and the position on the ground, that is, the point on the ground opposite the sun in relation to the camera. Both effects can generate serious problems in the modeling of terrain and in the radiometric exploitation of multispectral images. Avoid these radiometric anomalies requires of applying specific strategies for detecting these problems in UAV flight missions. At present, existing UAV software do not cope with this issue, just applying some basic preprocessing strategies based on brightness or contrast adjustment and some vignetting correction in the best case [13]. Even less, these problems are considered during flight planning and control missions. As a result, cartographic products by means of 3D models and multispectral orthoimages can lose quality and its radiometric analysis can provide wrong results in non-supervised classification procedures.

Although there are several investigations focused on image analysis for correcting similar defects on images (e.g., blur, haze, shadows), most of them are applied to satellite images [14,15]. Some approaches have been developed for aerial images acquired in manned photogrammetric flights to cope with shadows problems in mountainous landscapes [16] and even in urban areas [17,18]. Another important defect analysed in aerial images has been the blur motion [19–21]. However, the main problem of these aerial approaches is its high-computational cost and complexity, unfeasible when a lot of images have been acquired. In fact, some authors have analyzed the economic cost that these problems can generate in large projects [22], whereas others have put more emphasis on its scientific cost [23,24], but again applied to satellite missions.

Regarding the topic of this paper, sun reflections problems, some authors have developed specific approaches for its removal, but most of them focused on satellite images. With respect to the sun glint effect, the principle is to estimate the glint contribution to the radiance reaching by the sensor and then subtract it from the received signal. All the methods developed are applied for satellite missions under marine environments and use to fall into two main categories: (i) open ocean imagery with spatial resolutions between 100 and 1000 m [25–27]; (ii) coastal images with spatial resolution less than 10 m [28,29]. A more detailed description of sun glint correction methods for satellite missions is described in [22]. However, there is not a special treatment for sun glint effects over interior water mass such as lakes, reservoirs or ponds. Furthermore, all these sun glint approaches are based on probability distributions of water surfaces and are not valid for high-resolution images acquired from UAV. Concerning hotspot effect, also known as the opposition effect or the shadow point [30], several authors have proposed different solutions: [31] propose to use homomorphic filter for removing hotspot effects in orthophoto mosaic; [32] analyse hotspot directional signatures of different agricultural crops; [33] use different kernel-driven models for analysing hotspot effects in satellite images. In [15] reflectance measurements from the spaceborne Polarization and Directionality of Earth Reflectances (POLDER) instrument are used to analyse the so-called hotspot directional signature in the backscattering direction. However, again all these developments are based on satellite images and are far from the requirements of high-resolution multispectral images acquired from UAV.

Considering the main contributions remarked above, it seems clear that up to date there is not any UAV photogrammetric software solution that copes with the detection of sun glint and hotspot effects. To this end, this paper aims to describe the algorithms developed to deal with these problems and its integration in a UAV flight planning and control software developed by the authors [34].

This paper has been structured as follows: after this introduction, Section 2 describes in detail the method developed; Section 3 outlines the new functionalities of the flight planning and control software focused on the detection of sun reflections (hotspot and sun glint); Section 4 shows and validates the method with a study case; a final section is devoted to depict the main conclusions and future lines.

2. Methodology

The following figure (Figure 1) outlines the workflow developed for an automatic detection of hotspot and sun glint effects in UAV multispectral images.

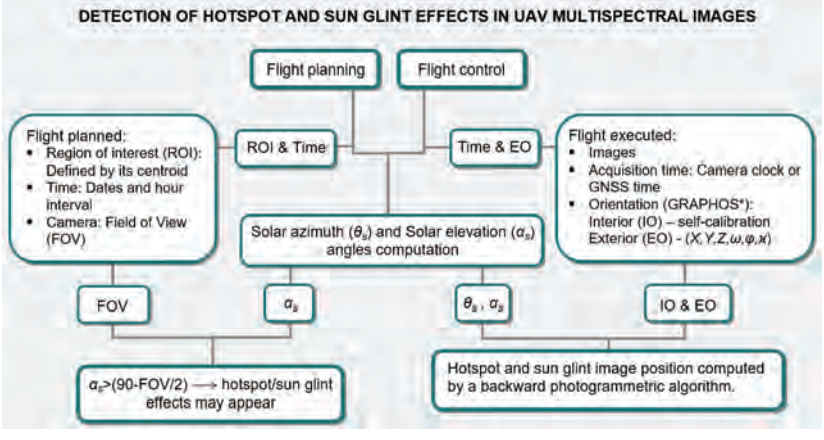


Figure 1. Workflow of the method developed for hotspot and sun glint detection in Unmanned Aerial Vehicle (UAV) multispectral images. * GRAPHOS (inteGRAted PHOtogrammetric Suite) is an open source software for photogrammetric applications developed by the authors.

Next, the main steps for the detection of sun glint and hotspot effects are described:

2.1. Analysis of UAV's Metadata and Data Registration

A specific script has been developed with a twofold function: (i) read and convert the time provided by the digital camera clock or the Global Navigation Satellite System (GNSS) solution in Universal Time Coordinated (UTC) (UTC is the bases of most radio time signals and the legal time systems. It is kept to within 0.9 s of UT (Universal Time) by introducing one second steps to its value (leap second). UT or Greenwich civil time is based on the Earth's rotation and counted from 0-h at midnight; the unit is mean solar day. UT is the time used to calculate the solar position in the described algorithm.); (ii) register the approximate position of the GNSS from the instant acquisition time of each image using the Exchangeable image file (Exif) configuration as input data. Camera clock is the quartz clock of digital cameras which is used for establishing the date and time acquisition for images in the local time of the zone. There is not a specific format requirement and the information is read from the Exif format. One of the most important aspect of these metadata is the synchronization between the time of capture and the time of registration for both sensors (camera clock or GNSS). In our approach we have obtained differences of 200 ms. This value is insignificant for our purpose, since an error of 2 s would correspond to a displacement of 0.5 pixels for the detection of sun reflections effects.

2.2. Images Orientation

Once we have solved the acquisition time with the camera clock or the GNSS, as well as an approximate GNSS position of each image, GRAPHOS [35], an open source photogrammetric software, was used for the orientation of images. GRAPHOS uses an internal geodetic local space rectangular (LSR) system (In this LSR system, the *Z* axis represents the zenith direction, the *X* axis the east direction and the *Y* axis completes a right handed system pointing to the north. The origin of this system has been defined as anchor point using the barycentre of the different images), which avoids the problems of dealing with scale and absolute coordinates provided by the GNSS solution. Absolute coordinates

deal with two different reference systems: one for planimetric coordinates (usually, an ellipsoid with cartographic UTM projection); and other for altimetric coordinates (usually, a geoid with orthometric altitudes). Since both coordinates are supported by different reference systems, the scale definition will be different along the planimetric and altimetric axes. Furthermore, those problems related with coordinate reference systems such as the passing from terrain to ellipsoid and the passing from the ellipsoid to Universal Transverse Mercator (UTM) projection can be solved with higher precision in the orientation of images and thus in the photogrammetric products obtained.

It should be remarked that this internal geodetic LSR approach is not considered in any of the most well-known commercial image-based modeling software, being really useful when we deal with different sensors (e.g., GNSS, multispectral camera, etc.).

Images orientation is solved through a combination between computer vision and photogrammetry. First, this combination is fed by the resulting keypoints extracted and matched using Maximal Self Dissimilarity (MSD) [36] and Scale Invariant Feature Transform (SIFT) [37] algorithms, both included in GRAPHOS. In particular, keypoints are detected with MSD and described with SIFT. The descriptor is mandatory to assign attributes to the point and thus for performing the matching (search of homologous points) between images. Next, an approximation of the external orientation of the cameras is calculated following a fundamental matrix approach [38]. The external orientation of images include the spatial (X, Y, Z) and angular (omega, phi y kappa) positions. Finally, these positions are refined by a bundle adjustment complemented with the collinearity condition based on the Gauss–Newton method [39], obtaining the final external orientation of images. GRAPHOS solves the external orientation based on the combination of two open source solutions such as Bundler [40] and Apero [13]. In addition, the external orientation of images allows to integrate as unknowns several internal camera parameters (i.e., focal length, principal point and lens distortions), allowing the use of non-calibrated cameras and guarantying acceptable results. For the present study case, a self-calibration strategy supported by Fraser calibration model which encloses seven internal parameters (focal length, principal point, two radial distortion parameters and two tangential distortion parameters) is used [41]. This final orientation can be performed with internal constraints (i.e., free-network) or with external constraints (e.g., Ground Control Points—GCPs or known distances) using the collinearity condition (Equation (1)):

$$\begin{aligned}(x - x_0) + \Delta x &= -f\frac{r_{11}(X-S_X)+r_{21}(Y-S_Y)+r_{31}(Z-S_Z)}{r_{13}(X-S_X)+r_{23}(Y-S_Y)+r_{33}(Z-S_Z)}\\ (y - y_0) + \Delta y &= -f\frac{r_{12}(X-S_X)+r_{22}(Y-S_Y)+r_{32}(Z-S_Z)}{r_{13}(X-S_X)+r_{23}(Y-S_Y)+r_{33}(Z-S_Z)}\end{aligned} \quad (1)$$

where x and y are the known image coordinates coming from the matching of keypoints; X_i, Y_i and Z_i are the corresponding known GCPs or known distances coming from topographic surveying or existing cartography; r_{ij} are the unknown 3×3 rotation matrix elements; S_X, S_Y and S_Z represent the unknown camera position; f is the principal distance or focal length; x_0 and y_0 are the principal point coordinates and Δx and Δy are the lens distortion parameters. These internal camera parameters may be known or unknown by the user and thus are introduced as equations or unknowns (self-calibration), respectively. This equation is not linear and has to be solved iteratively based on initial approximations.

It should be noted that in those cases where the UAV can use a GNSS with enough quality (phase solution with monofrequency L1 or bifrequency L1–L2), direct orientation of images can be solved and thus no GCPs would be required.

2.3. Solar Positioning

Once images have been oriented with a robust photogrammetric approach, the relative positioning of the sun (solar azimuth and elevation) is computed based on the UTC time and using geodetic coordinates (latitude, longitude and ellipsoidal altitude) of each image. In particular, solar azimuth (θ_S) and solar elevation (α_S) angles are computed for each image using the following equations (Equations (2) and (3)). In this step the open source solar library SPA is used (http://rredc.nrel.gov/solar/codesandalgorithms/spa/).

$$\theta_S = \Theta + \Delta\psi + \Delta\tau \tag{2}$$

where θ_S is the solar azimuth angle, $\Delta\tau$ is the aberration correction (in degrees), $\Delta\psi$ is the nutation in longitude (in degrees) and Θ is the geocentric (Geocentric means that the sun position is calculated with respect to the Earth center) longitude (in degrees).

$$\alpha_s = \arctan 2\left(\frac{\sin\ \theta_S \times \cos\ \varepsilon - \tan\ \beta \times \sin\ \varepsilon}{\cos\ \lambda s}\right) \tag{3}$$

where α_S is the solar elevation angle (in radians), θ_S is the solar azimuth angle, ε is the true obliquity of the ecliptic (in degrees) and β is the geocentric latitude (in degrees).

2.4. Hotspot and Sun Glint Detection

Computed the relative solar positioning (azimuth and elevation solar angles) for each image, the theoretical position of the possible hotspot and sun glint effects is provided for each image following the next photogrammetric steps. Figure 2 outlines both effects and its geometrical basis.

- Determination of hotspot/sun glint direction angles (azimuth and elevation). In the case of hotspot, the azimuth angle (θ_{hs}) is computed as the solar azimuth $\theta_S \pm 180$ degrees. In the case of sun glint, the azimuth angle (θ_{sg}) is computed using directly the solar azimuth (θ_S). The elevation for both effects (α_{hs}, α_{sg}) corresponds with the relative sun elevation for each image (α_S).
- Transformation between coordinate reference systems. In order to guarantee better accuracy in the photogrammetric process, a transformation from geodesic coordinates (latitude, longitude and ellipsoidal altitude) to LSR coordinates is performed for each image (see Section 2.2).
- Hotspot/sun glint direction vector. Using an arbitrary distance (e.g., 150 m), the sun elevation and azimuth angles (α_S, θ_S) and the image orientation in LSR coordinates, a vector is defined in the internal geodetic LSR system for each image. This arbitrary distance is defined with the length of a vector with origin in the projection center of the camera (S_X, S_Y, S_Z) and with the direction of the optical axis of the camera (r_{ij}). Flight height can be a reference for establishing the length of this arbitrary distance. As a result, a direction vector for the possible hotspot (θ_{hs}, α_{hs}) and sun glint (θ_{sg}, α_{sg}) effect is defined. It should be noted, that all the points of the vector are projected in the same image point, so the distance chosen is completely arbitrary.
- Hotspot/sun glint ground coordinates. Using the hotspot/sun glint direction vector the hotspot/sun glint ground coordinates (X, Y, $Z_{hs/sg}$) are computed in the internal geodetic LSR system.
- Hotspot/sun glint image coordinates. A backward photogrammetric process is applied to detect both effects in the images, using the external and internal orientation of each image and based on the collinearity condition (Equation (1)). If the image coordinates computed (x, $y_{hs/sg}$) (in pixels) are within the format of the camera, hotspot or sun glint effects will appear in the images.
- Masks definition. Since all the process developed accumulate errors (e.g., from the acquisition time to the inner and exterior orientation parameters) and the own effects enclose a size, a buffer area definition of 50 × 50 pixels is defined around these hotspot and sun glint coordinates to isolate those parts that can be affected by sun effects in the images. This buffer size is related with the ground sample distance (GSD) or pixel size. In our case, the GSD is 5 cm, so a buffer area of 50 × 50 pixels corresponds to a ground area of 2.5 m × 2.5 m. Since the sun glint and hotspot effects can show different sizes and shapes, this value is considered enough for enclosing the effect, including also the own propagation error of the process. Masks were used in the photogrammetric process as exclusion areas, that is, nor keypoints used in the orientation of images, neither the image points sampled during the orthophoto generation were taken from these exclusion areas. Alternatively, these exclusion areas were completed taking advantage of the high number of images acquired in UAV flights and the high overlap between images (70%).

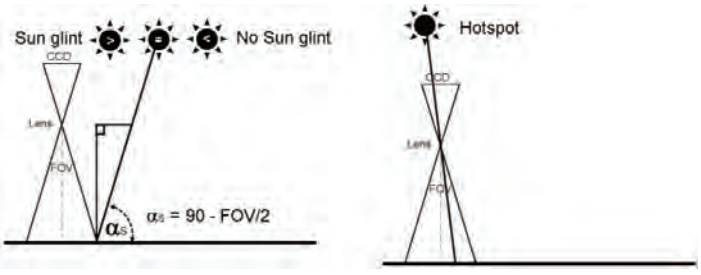


Figure 2. (**Left**) *Sun glint effect*: the sun glint effect is produced from specular surfaces such as water, glass or metal when the angle of incidence of the sun equals the angle of reflection and this angle is lower than the field of view (FOV) of the camera. In particular, if the sun elevation (α_S) is greater than 90 minus 1/2 the FOV of the camera, rays from the sun can be reflected directly onto the charge-coupled device (CCD) causing a sun glint defect on the photo; (**Right**) *Hotspot effect*: the hotspot is produced due to a direct alignment between the sun, the camera and the position on the ground, that is, the point on the ground opposite the sun in relation to the camera. The hotspot produces a change of brightness in this point of the image and its surroundings.

3. UAV Flight Planning and Control Software: MFlip-v2

A specific module for dealing with hotspot and sun glint effects has been added to the software MFliP-v1 [34], which can be used as an interface applicable to a wide variety of UAVs. The application was implemented using the C++ programming language and Extensible Markup Language (XML) for data management. This section provides an overview of the specific module for detecting hotspot and sun glint effects. It should be noted that sometimes the flight must be planned when the sun provides maximum irradiance to the crops (e.g., analysis of crop water stress), so there is more probability to suffer these effects. Although under these circumstances the flight cannot be changed, it will be very useful to detect these problems in order to isolate them in the photogrammetric or remote sensing processing.

Figure 3 shows a screen capture of the software application and its specific module for hotspot and sun glint effects. In this overview we show a simple case of use step-by-step focused on detecting hotspot and sun glint effects during flight planning mission.

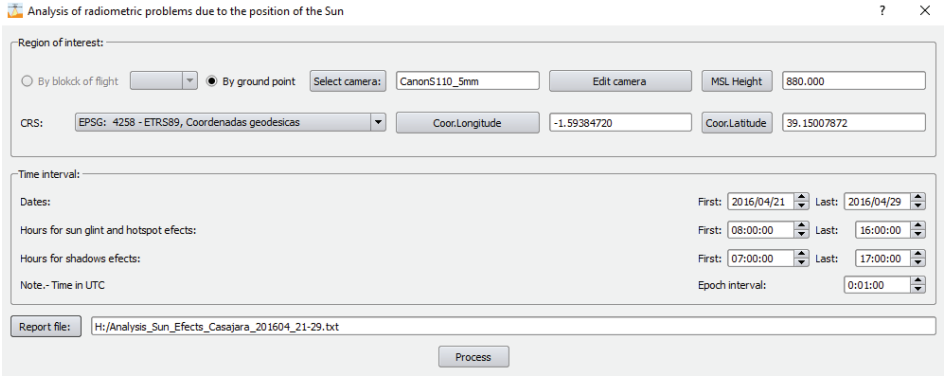


Figure 3. Layout of the developed software, MFliP-v2, for hotspot and sun glint detection in UAV flights.

3.1. Project Definition

When users run the program, they first create a new project. The user is asked to fill in a form with some necessary data for the program to work. All this information is stored in database. This data describes the settings of the UAV flight: coordinate system, region of interest (ROI), flight direction and the date and time (in UTC) of the flight. In addition to the settings of the flight, the settings of the camera must also be entered into the form.

In the following figure (Figure 4) the different steps from project definition to flight planning and control are outlined, highlighting (in red colour) the new module for dealing with hotspot and sun glint effects.

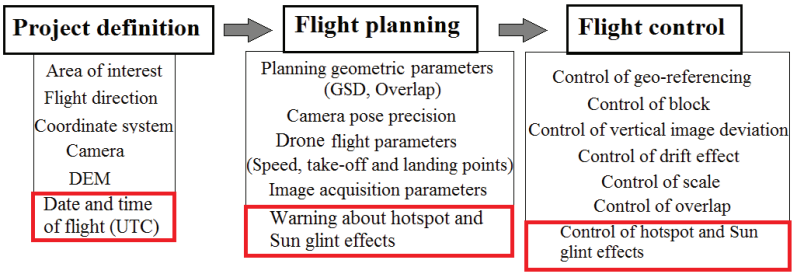


Figure 4. Flowchart for flight planning and control with special attention to hotspot and sun glint effects (highlighted in red colour). The rest of functions in flight planning and control can be analysed in detail in [34].

3.2. Flight Planning

Once we have an active project, we must plan the flight. The first step is to delimitate the ROI together with the flight direction as well as the date and time of the flight. Based on the centroid coordinates of the ROI and the theoretical position of the planned images along the flight direction, a planned sun position (azimuth and elevation) for each image are computed. If the sun elevation angle (α_s) is greater than 90 minus 1/2 the FOV of the camera (Figure 1), the software will alert about the possible presence of hotspot and sun glint effects to the user, giving information about the time interval in which we can suffer these problems.

As a result, a flight planning file is automatically generated by the program and directly uploaded to the UAV firmware.

3.3. Flight Control

Once the flight was executed, the control of the flight is performed based on the validation of different geometric controls: geo-referencing, vertical image deviation, drift effect, scale and overlap. More detailed information about these geometric controls can be found in [34]. Regarding hotspot and sun glint effects, during the control step both effects are detected in those images where exist (see Section 2) and then are isolated with a mask for the next photogrammetric and remote sensing processes.

4. Experimental Results

In this section we check the algorithms developed using a UAV flight executed with a multispectral camera. The area over which images were acquired covered an area of 820,000 m^2 with an approximately rectangular shape of 1050 m $\times$ 780 m. The site was located in Tarazona de la Mancha (Albacete, Spain). The main reason to perform this test was the application of this methodology to photogrammetric and remote sensing applications. In particular, a near infrared orthophoto and normalized digital vegetation

index (NDVI) were computed for agronomical studies. The following figure (Figure 5) outlines the main problems of hotspot and sun glint effects over the region of interest.

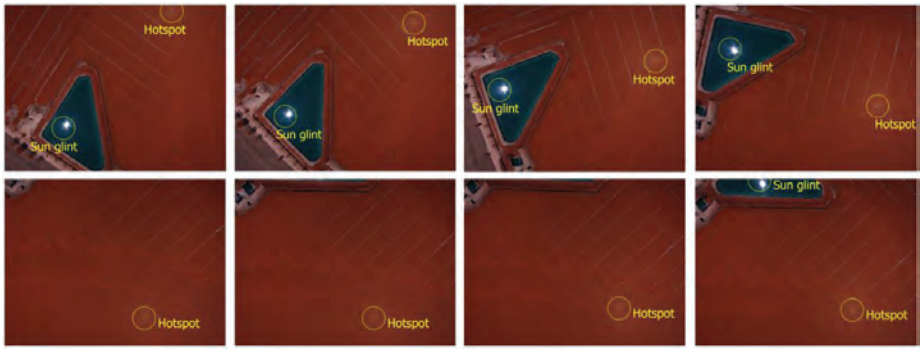


Figure 5. Hotspot and sun glint effects in the region of interest. It should be noted how the position of both effects varies through the different images.

Regarding the platforms and sensor used, a Carabo S3 UAV was used (Table 1) together with a Canon PowerShot S110 compact digital camera. This camera allows us to include a specific near infrared filter [42], recording near infrared, green and blue channels, at the same time. The specific operational conditions (Table 1) were integrated through the autopilot-flight control based on Pixhawk with a processor (168 MHz/252 MIPS Cortex-M4F) and a firmware (APC Copter 3.4.6). Regarding Canon PowerShot S110 and its synchronization, an electronic switch connected to the multiport entry through a micro-USB connector was used. This switch is activated by one of the outputs of the Pixhawk autopilot based on the commands included in the flight mission. In addition, Cannon incorporates a particular firmware, CHDK, which allows us to shoot camera remotely.

Table 1. Technical specifications and operational conditions: UAV Carabo S3.

Technical Specifications: UAV Carabo S3	
Climb rate	2.5 m/s
Cruising speed	6.0 m/s
Vehicle mass	2.6 kg
Recommended payload mass	450 g
Maximum payload mass	900 g
Dimensions	690 mm from rotor shaft to rotor shaft
Flight duration	up to 40 min (Full HD smart sport camera)
Battery	6 S/22.2 V
Accelerometer/magnetometer	ST Micro LSM303D 14 bit
Accelerometer/gyroscope	Invensense MPU 6000
Gyroscope	ST Micro L3GD20H 16 bit
Barometer	MEAS MS5611
GPS	uBlox LEA 6H
Autopilot	Pixhawk
Operational Conditions: UAV Carabo S3	
Temperature	−10 °C to 45 °C
Humidity	max. 90% (rain or snow is no problem)
Wind of tolerance	up to 12 m/s
Flight radius	500 m on RC (Remote Control), with WP (Way Point navigation) up to 2 km
Ceiling altitude	up to 2500 m

Last but not least, the position and time provided by the GPS are stored in a log file and inserted in the heading of each image. This operation is performed using one of the tools provided by the flight

planning (Geotag), which receives as input the own images and the log file remarked before, writing in the exif file of each image the approximate coordinates and GPS time. It should be noted that this procedure is valid for any UAV based on the same autopilot (Pixhawk).

The flight was planned using the region of interest, the flight direction and the date/time interval as input (Figure 6). In addition, aspects such as the GSD, flight height, flight speed, waypoints and overlap between adjacent images were also considered. Particularly, a GSD of 5 cm which corresponds to a flight height of 140 m was considered. Overlaps between images of 70% and 25% were planned for forward and side overlaps, respectively. Regarding the flight speed, two flight speeds were considered: (i) a cruising speed usually of 6 m/s and (ii) an acquisition speed close to 0 m/s when the UAV is in the shooting point. This last issue is crucial in order to minimize the inertial effects of the own UAV movement. Since the precision of navigational sensors (GPS, accelerometers, gyroscope, etc.) is not high, better results will be obtained if images are acquired with the UAV stable and without too much speed. Regarding the shooting rate of the camera, it was setup according to the waypoints of the flight planning. Therefore, the camera acquired images when it is in the waypoint planned, both for planimetric coordinates (X, Y) and altimetric coordinates (Z). As result, we can guarantee the overlaps, scale and the different geometrical constraints for the flight planned. Finally, the camera setup parameters were fixed before the flight. In particular, in our flight these setup parameters were: manual focus fixed to infinity, exposure time: 1/1000, aperture of diaphragm: $\pm$5.6 and automatic ISO.

On the other hand, a previous analysis of sun reflections was performed during flight planning using the field of view (FOV) of the camera and the planned sun elevation angle. In particular, if the sun elevation angle (α_S) is greater than 90 minus 1/2 the FOV of the camera, hotspots and/or sun glint effects may appear. A detailed report about the interval time (in UTC) for both effects is provided, warning the user about possible hotspot and sun glint effects and their time intervals (Figure 7).

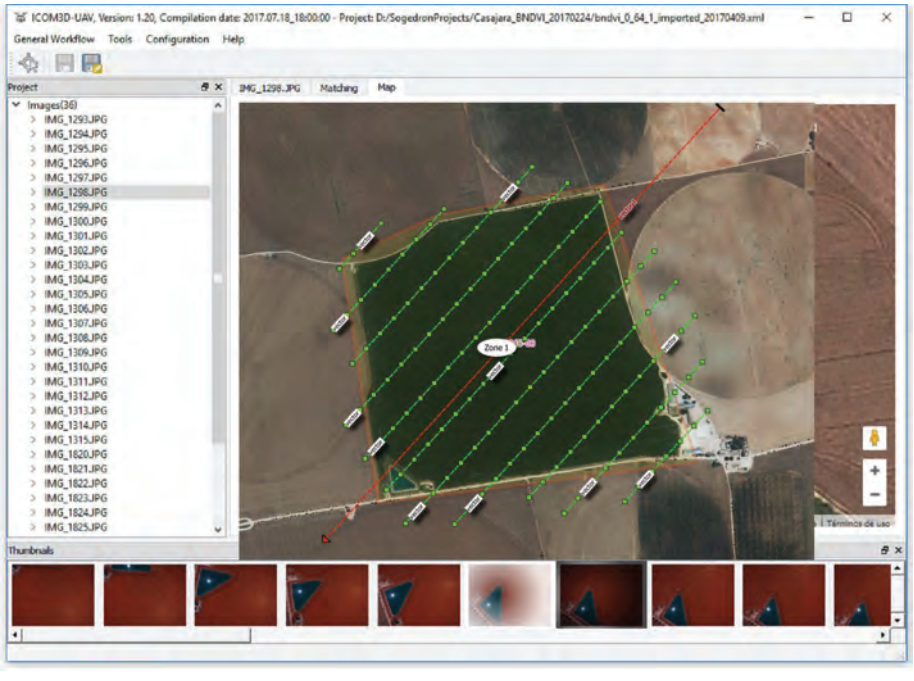


Figure 6. Flight planning over the region of interest (ROI). This planning was defined using the centroid X, Y, Z coordinates of the ROI together with the flight direction and the date/time of the flight.

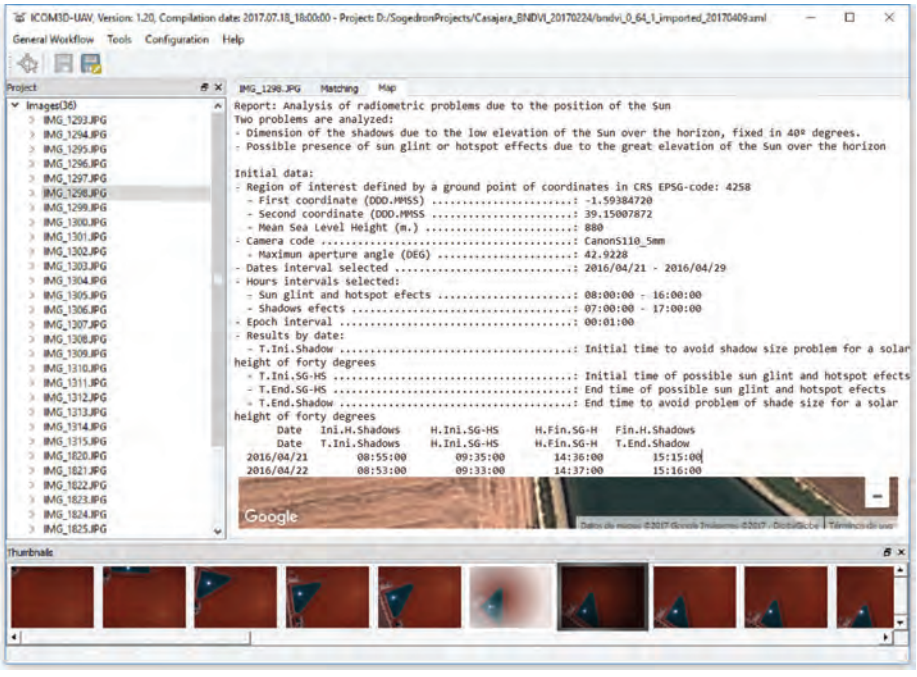


Figure 7. Detailed report about the interval time for possible hotspot/sun glint effects. Note that the time interval for sun glint and hotspot effects are broad since we are using a digital camera with a large field of view.

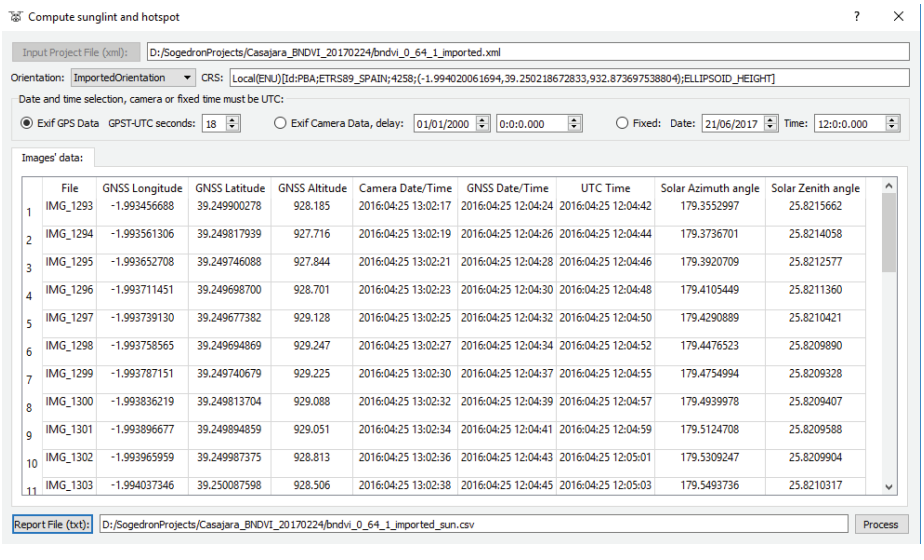
Since it was required to fly at these time intervals due to agronomical reasons, a control of the flight was mandatory to detect possible sun reflections problems. To this end, a proper interior and exterior orientation of the camera is crucial in order to guarantee quality in the detection of hotspot and sun glint effects and thus accurate products in photogrammetry and remote sensing applications.

In our case, the Canon PowerShot s110 digital camera (Canon Inc., Tokyo, Japan) was calibrated based on a bundle adjustment using the self-calibration approach included in GRAPHOS software. The self-calibration results and other camera parameters are shown in Table 2. The orientation information of the images was computed using the keypoints extracted and matched and the GCPs. In particular, these GCPs were extracted from public cartographic products (In Spain, the National Geographical Institute (IGN) periodically provides free cartographic products by means of orthoimages and LiDAR flights. Both products belong to a National Program for Earth Observation (PNOT) supported by the directive INSPIRE (Infrastructure for Spatial Information in Europe). More information can be found in: http://www.ign.es/web/ign/portal/obs-area-observacion-territorio.): X, Y coordinates coming from a public orthophoto (pixel size of 25 cm) and Z coordinate coming from a public Light Detection and Ranging (LiDAR) flight (resolution 0.5 points/m^2). The accuracy for keypoints was subpixel, the accuracy for GCPs was 0.5 m and the accuracy for orientation was 0.2°.

Table 2. Technical specifications and camera self-calibration parameters: Canon Powershot S110.

Technical Specifications: Canon Powershot S110	
Sensor Type	1/1.7″ CMOS sensor
Sensor size (w, h)	(7.2, 5.4) mm
Effective Pixels	4000 × 3000 pixels
Focal length	5.2 mm
Weight	198 g (inc battery)
Self-Calibration Parameters: Canon Powershot S110	
Calibrated focal length	5.054 mm
Principal point (x, y)	(−0.034, −0.126) mm
Radial distortion (k_1, k_2)	(−0.0370017, −0.00429136)
Tangential distortion (P_1, P_2)	(−0.00116555, −0.00518746)

Once images were oriented using the self-calibration approach, the relative solar positioning (sun azimuth-θ_S and sun elevation-α_S) was computed for each image. Figure 8 shows the relative solar positioning (sun azimuth-θ_S, and sun elevation-α_S) for each image based on the camera clock or GNSS time, as well as the orientation for each image.



	File	GNSS Longitude	GNSS Latitude	GNSS Altitude	Camera Date/Time	GNSS Date/Time	UTC Time	Solar Azimuth angle	Solar Zenith angle
1	IMG_1293	-1.993456688	39.249900278	928.185	2016:04:25 13:02:17	2016:04:25 12:04:24	2016:04:25 12:04:42	179.3552997	25.8215662
2	IMG_1294	-1.993561306	39.249817939	927.716	2016:04:25 13:02:19	2016:04:25 12:04:26	2016:04:25 12:04:44	179.3736701	25.8214058
3	IMG_1295	-1.993652708	39.249746088	927.844	2016:04:25 13:02:21	2016:04:25 12:04:28	2016:04:25 12:04:46	179.3920709	25.8212577
4	IMG_1296	-1.993711451	39.249698700	928.701	2016:04:25 13:02:23	2016:04:25 12:04:30	2016:04:25 12:04:48	179.4105449	25.8211360
5	IMG_1297	-1.993739130	39.249677382	929.128	2016:04:25 13:02:25	2016:04:25 12:04:32	2016:04:25 12:04:50	179.4290889	25.8210421
6	IMG_1298	-1.993758565	39.249694869	929.247	2016:04:25 13:02:27	2016:04:25 12:04:34	2016:04:25 12:04:52	179.4476523	25.8209890
7	IMG_1299	-1.993787151	39.249740679	929.225	2016:04:25 13:02:30	2016:04:25 12:04:37	2016:04:25 12:04:55	179.4754994	25.8209328
8	IMG_1300	-1.993836219	39.249813704	929.088	2016:04:25 13:02:32	2016:04:25 12:04:39	2016:04:25 12:04:57	179.4939978	25.8209407
9	IMG_1301	-1.993896677	39.249894859	929.051	2016:04:25 13:02:34	2016:04:25 12:04:41	2016:04:25 12:04:59	179.5124708	25.8209588
10	IMG_1302	-1.993965959	39.249987375	928.813	2016:04:25 13:02:36	2016:04:25 12:04:43	2016:04:25 12:05:01	179.5309247	25.8209904
11	IMG_1303	-1.994037346	39.250087598	928.506	2016:04:25 13:02:38	2016:04:25 12:04:45	2016:04:25 12:05:03	179.5493736	25.8210317

Figure 8. Solar positioning (azimuth and elevation) for the study case analyzed.

Later, hotspot and sun glint effects were detected (in pixel coordinates) based on a backward photogrammetric approach (Section 2.4) using the relative position of sun and the photogrammetric orientation of the images. If these coordinates are within the camera format the effects are detected and isolated with a specific mask. Table 3 shows the main results in pixel coordinates.

Table 3. Example of hotspot and sun glint detection in image coordinates.

Image	Sun Glint Detection (x, y in Pixels)	Hotspot Detection (x, y in Pixels)
IMG_1293	n.a.	(2804, 2327)
IMG_1294	(1160, 16)	(2833, 2355)
… … …	… … … … … … … … .	… … … … … … … … .
IMG_1303	(1153, 2215)	n.a.
IMG_1304	(1098, 2243)	(2962, 47)
IMG_1305	(1109, 2237)	(2960, 26)
… … … … … ..	… … … … … … … .. … .	… … … … … … … .. … .

Last but not least, considering the heterogeneity of the precision of input data, a simulation study for analyzing the a priori precision based on an error analysis was performed using the following simulated errors:

- GNSS positioning errors (meters): 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100;
- Time errors (seconds): 0.1, 0.2, 0.5, 1, 2, 5, 10, 20, 50, 100;
- Orientation (degrees): 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10, 20.

Positive and negative errors were considered for each parameter.

The following figure (Figure 9) outlines the results of this simulation of errors. It is important to note that only those very high errors coming from the orientation parameters have representation at this scale.

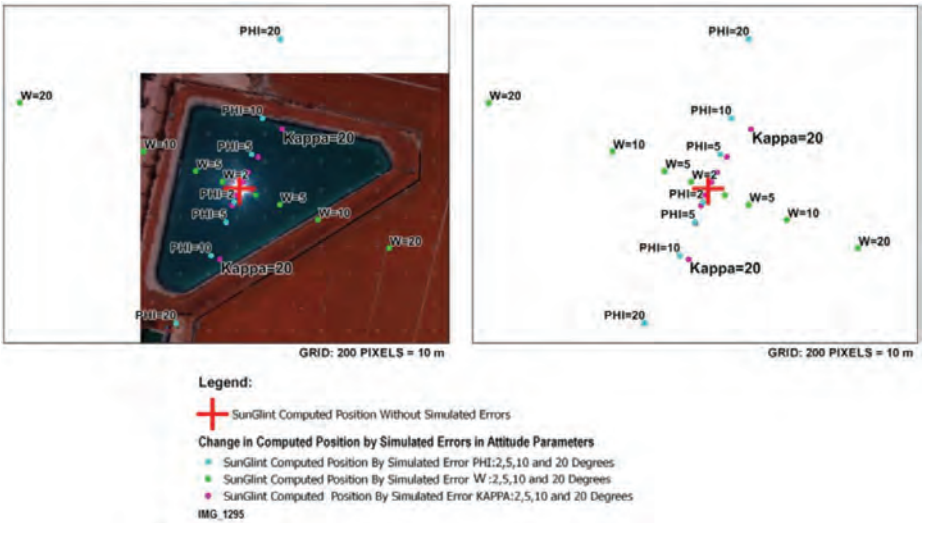


Figure 9. Analysis of errors for the detection of sun glint effects simulating different errors in angular position (omega, phi, kappa).

From the analysis of Figure 9 we can see that an error of 2° in phi or omega angles causes a variation in the position of 100 pixels; an error of 2° in kappa angle causes a variation in the detection of 50 pixels. Regarding time, although the representation scale does not allow us to visualize it, an error of 2 s in time would cause a variation in the detection of 0.5 pixels. This time error of 2 s does not take part in the computation of the orientation parameters of the images. Instead, this orientation of images (spatial and angular) is solved based on photogrammetry with collinearity condition and the GCPs, as was explained in Section 2.2. As a result, the error of 2 s which provides a variation of 0.5 pixels is

related with the Sun position. Therefore, a delay of 2 s in the variation of sun position is insignificant regarding the sun reflection effect in the images. Regarding GNSS, although the representation scale does not allow us to visualize it, an error of 30 m in the position (latitude, longitude y altitude), would provide variations of detection lower than 0.01 pixels.

The performed analysis of the errors aims to highlight two main issues:

(a) The main errors in detection of sun reflections are related with the camera orientation parameters. Therefore, it is crucial to apply a photogrammetric approach to solve this orientation (Section 2.2.).

(b) Simulated spots correspond to errors in orientation which are far from the errors obtained in our photogrammetric approach, even when low-cost UAV sensors are used.

Therefore, this error analysis supports that the precision in the detection of sun reflections is enough and is guaranteed using this photogrammetric approach and the low-cost UAV technology.

In our case of study, an error in the georeferencing based on GCPs of 0.5 m provides subpixel variations, an error for the orientation of 0.2° provides a variation of 10 pixels and a time error of 1 s provides a variation of 0.25 pixels. As a result, an error of approximately 10 pixels can be considered for the detection of sun glint and hotspot effects. Therefore, the most relevant conclusion about this error analysis is that the orientation of images is the most important step for the detection of sun reflections effects, so the photogrammetric approach is crucial.

Finally, a near infrared (NIR) orthophoto (Figure 10) and NDVI images (Figure 11) were computed once the different sun reflection effects were detected and excluded. Note the difference between the NIR orthophoto without applying the process (Figure 10 left down) and the NIR orthophoto detecting and correcting sun reflections effects (Figure 10 right down). Regarding NDVI images, these images were computed using the custom vegetation-sensing filter included in this camera by the manufacturer (https://event38.com/product/custom-ngb-filter-glass-for-diy-camera-conversion/). This filter allows an off-the-shelf camera to be used to collect NIR, Green and Blue channels. With this data, a pseudo NDVI can be calculated as follows (Equation (4)):

$$pNVDI = \frac{NIR - Blue}{NIR + Blue} \tag{4}$$

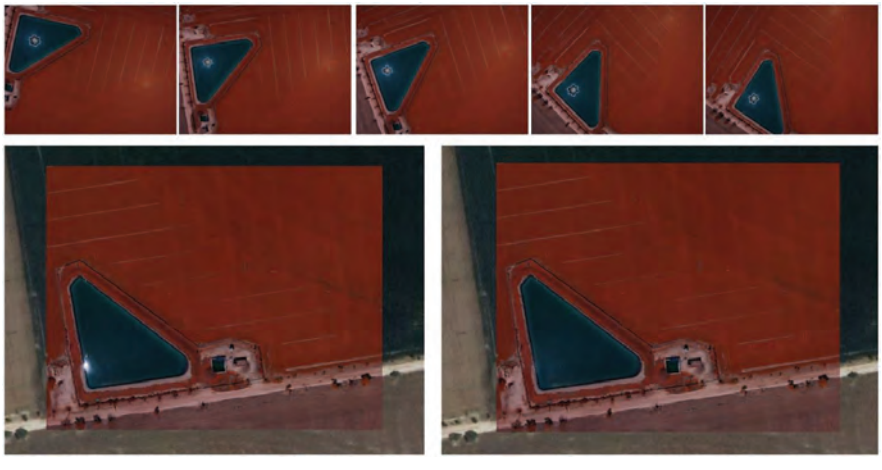


Figure 10. Near infrared orthophoto. (**Up**) detection and clustering with a mask of sun glint effects over near infrared images; (**Down**) comparison between resulting near infrared orthophoto with sun glint correction (**right**) and without sun glint correction (**left**).

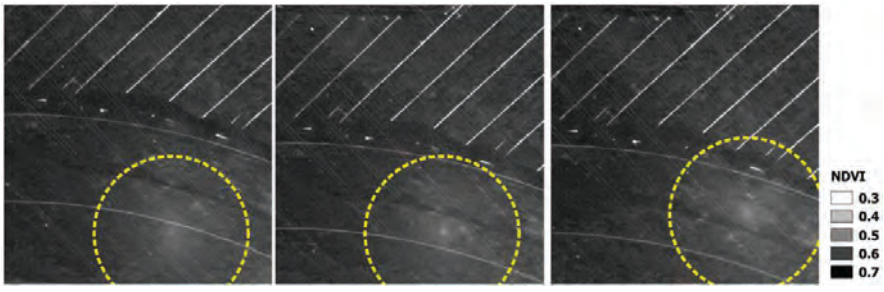


Figure 11. Normalized digital vegetation index (NDVI) images and the corresponding NDVI values in the hotspot areas detected (yellow circle).

Figure 11 shows how affects the hotspot effect to the pseudo NDVI values in a qualitative and quantitative (scale bar) way. Particularly, differences higher than 20% were obtained for pseudo NDVI in those hotspot areas where pseudo NVDI values clearly decrease and whose values could directly affect the different agronomical analysis (e.g., determination of irrigation crops). Figure 11 describes that the same area on the ground can show different values of NDVI, even for those images taken under very low time lapse and very close point of views due to the hotspot effect. That means that this portion of the image under hotspot effects should not be used for vegetation indices estimation, especially if other images which show this area are not under hotspot effects.

5. Conclusions

Through this article has been confirmed the necessity and importance of developing procedures that automate the detection of solar reflection problems in high resolution images acquired from UAV. In fact, currently there is not any commercial or scientific solution in the field of UAV that can deal with this problem, which could have serious consequences both in photogrammetric and remote sensing processes, especially if there is a large number of images. The process developed allows to solve the problem from a twofold perspective: from its possible prevention during flight planning to its detection once the flight was executed applying a flight control protocol, especially for those situations in which is mandatory to fly during the hours where the sun reaches its maximum height, being inevitable that these defects can appear.

The developed process has been implemented in a software for planning and control of UAV flights previously developed by the authors [34], allowing an easy handling of this type of problems.

The experimental results have showed the efficiency of the method to obtain photogrammetric products of higher quality and lower noise (e.g., near infrared orthophoto), as well as to obtain a much more accurate normalized vegetation indexes (NVDI) which are crucial in agronomical analysis. The error analysis performed aims to outline two main aspects: (i) the main errors that could affect the spots detected are related with the camera orientation parameters; (ii) the precision in the detection of sun reflections is around 10 pixels, which is guaranteed with our photogrammetric approach and using the current low-cost UAV technology.

The line developed so far opens new possibilities of improvement associated to the own method and focused on being able to implement strategies that allow not only to detect and isolate the sun reflection problems but also to correct these areas using other acquired images. Furthermore, since sun glint can only occur on reflective surfaces, it would be interesting to develop a strategy that allows to supervise automatically those areas which could not contain reflective surfaces without user intervention.

Acknowledgments: Author want to thank the Spanish Geographical Institute (IGN) for the support provided.

Author Contributions: D.H.L. and D.O.T. conceived and developed the software; R.B. designed and performed the experiments; D.G.A. and D.H.L. analyzed the data; D.G.A. and D.O.T. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [CrossRef]
2. Gonzalez-Aguilera, D.; Rodriguez-Gonzalvez, P. Drones—An Open Access Journal. *Drones* **2017**, *1*, 1. [CrossRef]
3. Hoffmann, H.; Nieto, H.; Jensen, R.; Guzinski, R.; Zarco-Tejada, P.J.; Friborg, T. Estimating evaporation with thermal UAV data and two-source energy balance models. *Hydrol. Earth Syst. Sci. Discuss.* **2016**, *20*, 697–713. [CrossRef]
4. Díaz-Varela, R.A.; de la Rosa, R.; León, L.; Zarco-Tejada, P.J. High-Resolution airborne UAV imagery to assess olive tree crown parameters using 3D photo reconstruction: Application in breeding trials. *Remote Sens.* **2015**, *7*, 4213–4232. [CrossRef]
5. Valero, S.; Salembier, P.; Chanussot, J. Object recognition in hyperspectral images using binary partition tree representation. *Pattern Recognit. Lett.* **2015**, *56*, 45–51. [CrossRef]
6. Balali, V.; Jahangiri, A.; Ghanipoor, S. Multi-class US traffic signs 3D recognition and localization via image-based point cloud model using color candidate extraction and texture-based recognition. *Adv. Eng. Inform.* **2017**, *32*, 263–274. [CrossRef]
7. Yuqing, H.; Wenrui, D.; Hongguang, L. Haze removal for UAV reconnaissance images using layered scattering model. *Chin. J. Aeronaut.* **2016**, *29*, 502–511.
8. Ribeiro-Gomes, K.; Hernandez-Lopez, D.; Ballesteros, R.; Moreno, M.A. Approximate georeferencing and automatic blurred image detection to reduce the costs of UAV use in environmental and agricultural applications. *Biosyst. Eng.* **2016**, *151*, 308–327. [CrossRef]
9. Bhardwaj, A.; Sam, L.; Akanksha; Javier Martín-Torres, F.; Kumar, R. UAVs as remote sensing platform in glaciology: Present applications and future prospects. *Remote Sens. Environ.* **2016**, *175*, 196–204. [CrossRef]
10. Su, T.C. A study of a matching pixel by pixel (MPP) algorithm to establish an empirical model of water quality mapping, as based on unmanned aerial vehicle (UAV) images. *Int. J. Appl. Earth Obs. Geoinf.* **2017**, *58*, 213–224. [CrossRef]
11. Teng, W.L.; Loew, E.R.; Ross, D.I.; Zsilinsky, V.G.; Lo, C.; Philipson, W.R.; Philpot, W.D.; Morain, S.A. *Fundamentals of Photographic Interpretation, Manual of Photographic Interpretation*, 2nd ed.; Philipson, W.R., Ed.; American Society for Photogrammetry and Remote Sensing: Bethesda, MD, USA, 1997; pp. 49–113.
12. Chen, J.M.; Cihlar, J. A hotspot function in a simple bidirectional reflectance model for satellite applications. *Geophys. Res.* **1997**, *102*, 907–925. [CrossRef]
13. Pierrot-Deseilligny, M.; Clery, I. APERO, an open source bundle adjustment software for automatic calibration and orientation of set of images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *38*, 269–276.
14. Harmel, T.; Chami, M. Estimation of the sunglint radiance field from optical satellite imagery over open ocean: Multidirectional approach and polarization aspects. *Geophys. Res. Oceans* **2013**, *118*, 76–90. [CrossRef]
15. Bréon, F.-M.; Maignan, F.; Leroy, M.; Grant, I. Analysis of hot spot directional signatures measured from space. *Geophys. Res.* **2002**, *107*, 4282. [CrossRef]
16. Giles, P. Remote sensing and cast shadows in mountainous terrain. *Photogramm. Eng. Remote Sens.* **2001**, *67*, 833–840.
17. Dare, P.M. Shadow analysis in high-resolution satellite imagery of urban areas. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 169–177. [CrossRef]
18. Adeline, K.R.M.; Chen, M.; Briottet, X.; Pang, S.K.; Paparoditis, N. Shadow detection in very high spatial resolution aerial images: A comparative study. *ISPRS J. Photogramm. Remote Sens.* **2013**, *80*, 21–38. [CrossRef]
19. Krahmer, F.; Lin, Y.; McAdoo, B.; Ott, K.; Wang, J.; Widemannk, D. *Blind Image Deconvolution: Motion Blur Estimation*; Technical Report; Institute of Mathematics and its Applications, University of Minnesota: Minneapolis, MN, USA, 2006.
20. Shan, Q.; Jia, J.; Agarwala, A. High-quality Motion Deblurring from a Single Image. *ACM Trans. Gr.* **2008**, *27*, 1–10.

21. Lelégard, L.; Brédif, M.; Vallet, B.; Boldo, D. Motion blur detection in aerial images shot with channel-dependent exposure time. In *IAPRS*; Part 3A; ISPRS Archives: Saint-Mandé, France, 2010; Volume XXXVIII, pp. 180–185.
22. Kay, S.; Hedley, J.D.; Lavender, S. Sun Glint Correction of High and Low Spatial Resolution Images of Aquatic Scenes: A Review of Methods for Visible and Near-Infrared Wavelengths. *Remote Sens.* **2009**, *1*, 697–730. [CrossRef]
23. Goodman, J.A.; Lee, Z.; Ustin, S.L. Influence of Atmospheric and Sea-Surface Corrections on Retrieval of Bottom Depth and Reflectance Using a Semi-Analytical Model: A Case Study in Kaneohe Bay. *Hawaii Appl.* **2008**, *47*, 1–11. [CrossRef]
24. Hochberg, E.; Andrefouet, S.; Tyler, M. Sea Surface Correction of High Spatial Resolution Ikonos Images to Improve Bottom Mapping in Near-Shore Environments. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1724–1729. [CrossRef]
25. Wang, M.; Bailey, S. Correction of Sun Glint Contamination on the SeaWiFS Ocean and Atmosphere Products. *Appl. Opt.* **2001**, *40*, 4790–4798. [CrossRef] [PubMed]
26. Montagner, F.; Billat, V.; Belanger, S. *MERIS ATBD 2.13 Sun Glint Flag Algorithm*; ACRI-ST: Biot, France, 2011.
27. Gordon, H.; Voss, K. *MODIS Normalized Water-Leaving Radiance Algorithm Theoretical Basis Document (MOD 18)*; version 5; NASA: Washington, DC, USA, 2004.
28. Lyzenga, D.; Malinas, N.; Tanis, F. Multispectral Bathymetry Using a Simple Physically Based Algorithm. *IEEE Trans. Geosci. Remote Sens.* **2006**, *44*, 2251–2259. [CrossRef]
29. Hedley, J.; Harborne, A.; Mumby, P. Simple and Robust Removal of Sun Glint for Mapping Shallow-Water Benthos. *Int. J. Remote Sens.* **2005**, *26*, 2107–2112. [CrossRef]
30. Murtha, P.A.; Deering, D.W.; Olson, C.E., Jr.; Bracher, G.A. Vegetation. In *Manual of Photographic Inter*, 2nd ed.; Philipson, W.R., Ed.; Aamerican Society for Photogrammetry and Remote Sensing: Bethesda, MD, USA, 1997.
31. Sun, M.W.; Zhang, J.Q. Dodging research for digital aerial images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 349–353.
32. Camacho-de Coca, F.; Bréon, F.M.; Leroy, M.; Garcia-Haro, F.J. Airborne measurement of hot spot reflectance signatures. *Remote Sens. Environ.* **2004**, *90*, 63–75. [CrossRef]
33. Huang, X.; Jiao, Z.; Dong, Y.; Zhang, H.; Li, X. Analysis of BRDF and albedo retrieved by kernel-driven models using field measurements. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *6*, 149–161. [CrossRef]
34. Hernandez-Lopez, D.; Felipe-Garcia, B.; Gonzalez-Aguilera, D.; Arias-Perez, B. An Automatic Approach to UAV Flight Planning and Control for Photogrammetric Applications: A Test Case in the Asturias Region (Spain). *Photogramm. Eng. Remote Sens.* **2013**, *1*, 87–98. [CrossRef]
35. González-Aguilera, D.; López-Fernández, L.; Rodriguez-Gonzalvez, P.; Guerrero, D.; Hernandez-Lopez, D.; Remondino, F.; Menna, F.; Nocerino, E.; Toschi, I.; Ballabeni, A.; et al. Development of an all-purpose free photogrammetric tool. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B6*, 31–38.
36. Tombari, F.; Di Stefano, L. Interest points via maximal self-dissimilarities. In Proceedings of the Asian Conference on Computer Vision, Singapore, Singapore, 1–5 November 2014; pp. 586–600.
37. Lowe, D.G. Object recognition from local scale-invariant features. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*; IEEE Computer Society: Washington, DC, USA, 1999; Volume 2, pp. 1150–1157.
38. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: New York, NY, USA, 2003; p. 655.
39. Kraus, K.; Jansa, J.; Kager, H. *Advanced Methods and Applications Volume 2. Fundamentals and Standard Processes Volume 1*; Institute for Photogrammetry Vienna University of Technology: Bonn, Germany, 1997.
40. Snavely, N.; Seitz, S.M.; Szeliski, R. Modeling the world from internet photo collections. *Int. J. Comput. Vis.* **2008**, *80*, 189–210. [CrossRef]
41. Kukelova, Z.; Pajdla, T. A minimal solution to the autocalibration of radial distortion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; p. 7.
42. NDVI Camera—NGB Converted Canon S110 Camera. Available online: https://event38.com/product/ndvi-camera-ngb-converted-canon-s110-camera/ (accessed on 14 October 2017).

sensors

MDPI

Article

UAVs, Hyperspectral Remote Sensing, and Machine Learning Revolutionizing Reef Monitoring

Mark Parsons [1,*], Dmitry Bratanov [2], Kevin J. Gaston [3,4] and Felipe Gonzalez [5]

[1] Queensland University of Technology, 2 George St, Brisbane, QLD 4000, Australia
[2] Research Engineering Facility, Institute for Future Environments, Queensland University of Technology, 2 George St, Brisbane, QLD 4000, Australia; Dmitry.Bratanov@qut.edu.au
[3] Environment and Sustainability Institute, University of Exeter, Penryn, Cornwall TR10 9FE, UK; k.j.gaston@exeter.ac.uk
[4] Institute for Advanced Study, Wissenschaftskolleg zu Berlin, Wallotstrasse 19, 14193 Berlin, Germany
[5] Institute for Future Environments, Robotics and Autonomous Systems, Queensland University of Technology, 2 George St, Brisbane, QLD 4000, Australia; felipe.gonzalez@qut.edu.au
* Correspondence: Mark.Parsons@qut.edu.au; Tel.: +61-0421-238-873

Received: 18 April 2018; Accepted: 13 June 2018; Published: 25 June 2018

Abstract: Recent advances in unmanned aerial system (UAS) sensed imagery, sensor quality/size, and geospatial image processing can enable UASs to rapidly and continually monitor coral reefs, to determine the type of coral and signs of coral bleaching. This paper describes an unmanned aerial vehicle (UAV) remote sensing methodology to increase the efficiency and accuracy of existing surveillance practices. The methodology uses a UAV integrated with advanced digital hyperspectral, ultra HD colour (RGB) sensors, and machine learning algorithms. This paper describes the combination of airborne RGB and hyperspectral imagery with in-water survey data of several types in-water survey of coral under diverse levels of bleaching. The paper also describes the technology used, the sensors, the UAS, the flight operations, the processing workflow of the datasets, the methods for combining multiple airborne and in-water datasets, and finally presents relevant results of material classification. The development of the methodology for the collection and analysis of airborne hyperspectral and RGB imagery would provide coral reef researchers, other scientists, and UAV practitioners with reliable data collection protocols and faster processing techniques to achieve remote sensing objectives.

Keywords: in-water survey; UAS; hyperspectral camera; machine learning; image segmentation; support vector machines (SVM); drones

1. Introduction

Coral reefs are under pressure, and as of 2016, 34.8% of the Great Barrier Reef has been affected by coral bleaching, and other reefs worldwide are also experiencing degradation due to industrial and environmental events (e.g., cyclones) [1]. Large scale monitoring of coral reefs is an expensive, time-consuming, and challenging task, which utilises various mediums (satellite, airplane, boat and both manned and unmanned in-water surveys). Current surveillance practice for reef monitoring is to use satellites, ship-based multibeam sonar (MBES), plane based airborne surveying, or visually inspecting for signs of bleaching damage [2,3]. Over the last decades, the commercial satellite-based remote sensing capability has been significantly improved [4]. Now, with the best ground sampling distance (GSD) of 30 cm for panchromatic, and 1.2 m for multispectral data, satellite imagery remains the best-suited option for large-scale monitoring of entire reef. The satellite surveys, however, have some significant limitations, due to cloud cover and to camera resolution hindering the ability

to distinguish finer details of reef ecosystems. Indeed, the approach struggles to provide marine researchers and biosecurity managers with often required benthic conditions on the sub-centimetre level GSD.

Recent advances in remote sensed imagery and geospatial image processing using unmanned aerial systems (UASs) can enable rapid and ongoing monitoring tools for coral bleaching detection and surveillance. This paper describes a UAS-based remote sensing methodology to increase the efficiency of existing surveillance practices, to detect coral type and coral bleaching, by combining in-water spectral observations and automated classification techniques to determine levels of benthic cover and coral growth in shallow water reef regions. The methodology is evaluated on its ability to identify differences between coral genera, bleaching level, and water depth, with an in-depth accuracy assessment to give insights for further refinements. This shows that UAS-based hyperspectral remote sensing techniques can offer an efficient and cost-effective approach to mapping and monitoring reef habitats over large, remote, and inaccessible areas.

The technology used, the sensors, the UAV, and the flight operations are presented in Section 2 of the paper. Later, in Section 3, we discuss the processing workflow for each imagery dataset, and methods for combining multiple airborne with ground-based datasets. Section 4 presents results of correlation between the different processed datasets.

2. Materials and Methods

Our approach to generate a predictive reef monitoring system employs multiple stages (Figure 1). The first is data collection. This includes collection of airborne RGB and hyperspectral imagery, as well as in-water expert visual assessment of each coral specimen.

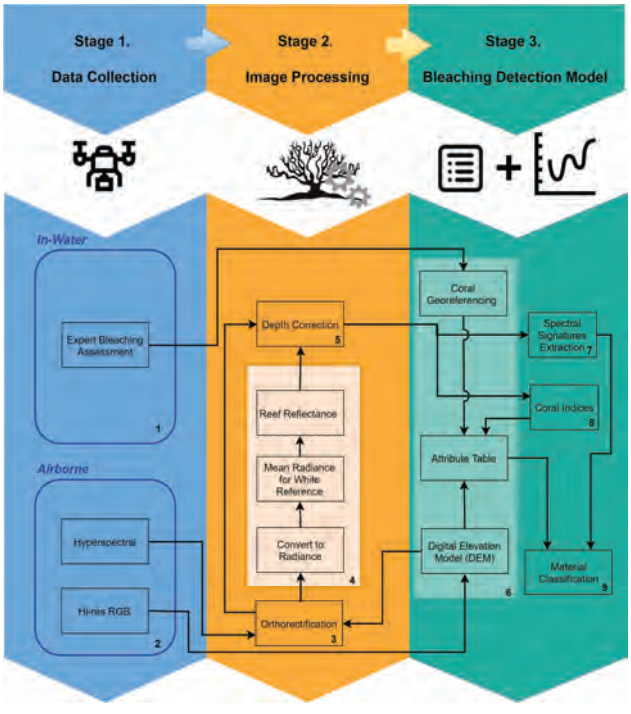


Figure 1. Predictive detection model workflow.

In the second stage, images are processed (using data in Table 1) to obtain the coral type, bleaching assessment, and reef orthomosaics. Hyperspectral scans are then orthorectified using multiple datacubes to produce a georeferenced hyperspectral snapshot of the reef. These orthorectified datacubes are processed to obtain radiance and reflectance, that are used in combination with expert visual coral type and bleaching assessment for signature extraction and calculation of indices of coral types and bleaching levels.

Table 1. Coral bleaching indices used in the analysis of reef type and bleaching.

Coral Index	Equation
Genus Classification Bleaching Classification	$(R_{540} - R_{575}/(R_{450} + R_{586})$
Acropora	
Bleaching Level 1 (Alv1)	$Alv1.1 = (R_{395} - R_{404})/(R_{395} + R_{404})$ $Alv1.2 = (R_{575} - R_{604})/(R_{575} + R_{604})$ $Alv1.3 = (R_{711} - R_{732})/(R_{711} + R_{732})$
Bleaching Level 2 (Alv2)	$Alv2.1 = (R_{404} - R_{489})/(R_{404} + R_{489})$ $Alv2.2 = (R_{595} - R_{662})/(R_{595} + R_{662})$
Bleaching Level 3 (Alv3)	$Alv3.1 = (R_{446} - R_{473})/(R_{446} + R_{473})$ $Alv3.2 = (R_{531} - R_{555})/(R_{531} + R_{555})$ $Alv3.3 = (R_{586} - R_{622})/(R_{586} + R_{622})$
Bleaching Level 4 (Alv4)	$Alv4.1 = (R_{446} - R_{489})/(R_{446} + R_{489})$ $Alv4.2 = (R_{569} - R_{600})/(R_{569} + R_{600})$ $Alv4.3 = (R_{611} - R_{671})/(R_{611} + R_{671})$
Bleaching Level 5 (Alv5)	$Alv5.1 = (R_{484} - R_{522})/(R_{484} + R_{522})$ $Alv5.2 = (R_{695} - R_{720})/(R_{695} + R_{720})$
Bleaching Level 6 (Alv6)	$Alv6.1 = (R_{400} - R_{418})/(R_{400} + R_{418})$ $Alv6.2 = (R_{460} - R_{484})/(R_{460} + R_{484})$ $Alv6.3 = (R_{724} - R_{768})/(R_{724} + R_{768})$
Porites Massive	
Bleaching Level 1 (PLv1)	$PLv1.1 = (R_{437} - R_{473})/(R_{437} + R_{473})$ $PLv1.2 = (R_{680} - R_{737})/(R_{680} + R_{737})$
Bleaching Level 2 (PLv2)	$PLv2.1 = (R_{411} - R_{473})/(R_{411} + R_{473})$ $PLv2.2 = (R_{640} - R_{671})/(R_{640} + R_{671})$
Bleaching Level 3 (PLv3)	$PLv3.1 = (R_{429} - R_{473})/(R_{429} + R_{473})$ $PLv3.2 = (R_{576} - R_{640})/(R_{576} + R_{640})$
Bleaching Level 4 (PLv4)	$PLv4.1 = (R_{406} - R_{418})/(R_{406} + R_{418})$ $PLv4.2 = (R_{533} - R_{582})/(R_{533} + R_{582})$
Gonipora	
Bleaching Level 3 (GLv3)	$GLv3.1 = (R_{409} - R_{477})/(R_{409} + R_{477})$ $GLv3.2 = (R_{640} - R_{722})/(R_{640} + R_{722})$
Turbinaria	
Bleaching Level 5 (TLv5)	$TLv5.1 = (R_{415} - R_{442})/(R_{415} + R_{442})$ $TLv5.2 = (R_{471} - R_{486})/(R_{471} + R_{486})$ $TLv5.3 = (R_{500} - R_{544})/(R_{500} + R_{544})$ $TLv5.4 = (R_{675} - R_{717})/(R_{675} + R_{717})$
Soft Coral	
Bleaching Level 5 (SLv5)	$SLv5.1 = (R_{429} - R_{444})/(R_{429} + R_{444})$ $SLv5.2 = (R_{506} - R_{544})/(R_{506} + R_{544})$ $SLv5.3 = (R_{577} - R_{604})/(R_{577} + R_{604})$ $SLv5.4 = (R_{662} - R_{708})/(R_{662} + R_{708})$

The third stage combines all the multiple sources of data into a single information system. To do this, individual coral genera are segmented, with the aim of creating a table (Table 2) that contains different attributes for each coral specimen within the reef.

1. Coral genus
2. Expert bleaching assessment (1 to 6 where 1 relates to severely bleached coral, 6—unbleached)
3. Depth
4. Notes about surroundings (e.g., coral size, proximity to other coral)
5. Latitude
6. Longitude

Table 2. Fragment of the attribute table at Pandora Reef (14 of 64 in-water data collected shown).

Photo ID	Coral Type	Lv Bleached	Bleached	Depth	Notes	Latitude	Longitude	Pixel x	Pixel y
76	*Porites massive*	1	Yes		1.5>	18.8129	146.4267	1489	1412
77	*Porites massive*	1	Yes			18.8129	146.4267	1507	1421
78	*Porites massive*	4	No	2.3		18.8130	146.4268	1544	1567
79	*Porites massive*	4	No	2.3		18.8130	146.4268	1529	1548
80	*Porites massive*	4	No	2.3		18.8130	146.4268	1541	1551
81	*Goniopora* sp.	3	No	1.7		18.8132	146.4268	1598	1733
82	*Goniopora* sp.	3	No	1.7		18.8132	146.4268	1604	1729
83	*Goniopora* sp.	3	No	1.7		18.8132	146.4268	1607	1727
84	*Acropora* sp.	2	Yes		Acropora plate.	18.8131	146.4268	1604	1669
85	*Acropora* sp.	2	Yes		Acropora plate.	18.8131	146.4268	1604	1666
86	*Acropora* sp.	3	No	1.4	Acropora plate.	18.8131	146.4269	1659	1684
87	*Acropora* sp.	3	No	1.4		18.8131	146.4269	1658	1675
88	*Porites massive*	3	No		1.5>	18.8131	146.4268	1610	1666
89	*Porites massive*	3	No			18.8131	146.4268	1615	1675
90	*Porites massive*	3	No		1.5>	18.8131	146.4269	1633	1636

Once the table is populated with georeferenced data, the spectral signatures are then extracted to generate a correlation between coral genus and each bleaching level. Results obtained in the correlation analysis are the foundation for the development of a preliminary coral bleaching detection model, which is followed by an evaluation and accuracy assessment to obtain final reef belching detection model.

2.1. In-Water Surveys

Coral is categorised by the Australian Institute of Marine Science (AIMS) into one of six levels of bleaching through visual inspection, where level one is severely bleached and level six an unbleached healthy coral specimen. Due to water's optical properties and algal growth associated with reef ecosystems, the spectral properties of a coral sample can vary drastically based on the current conditions, genus, and depth of the area.

Our approach was to use survey test sites where experts identified various genera of coral at varying levels of bleaching across a segment of reef, as a reference for approximation of optical variances due to water and algae. Figures 2 and 3 show examples of coral bleaching data obtained at different in-water survey locations, and after each point is identified, they are then classified based on genus, location (longitude and latitude), depth, and bleaching level.

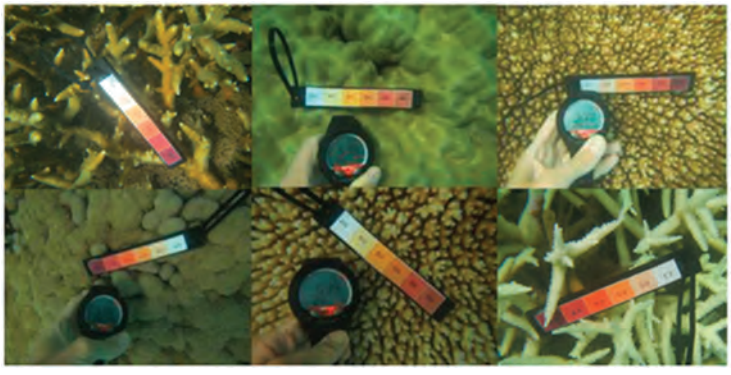


Figure 2. Example of in-water reference data with relevant bleaching level: 1 (severely bleached) to 6 (unbleached). Bleaching levels and genus from top left to bottom right: 5 *Acropora*, 4 *Porites* massive, 3 *Acropora*, 3 *Goniopora*, 2 *Acropora*, 1 *Acropora*. Courtesy of Sam Noonan (Australian Institute of Marine Science (AIMS)).

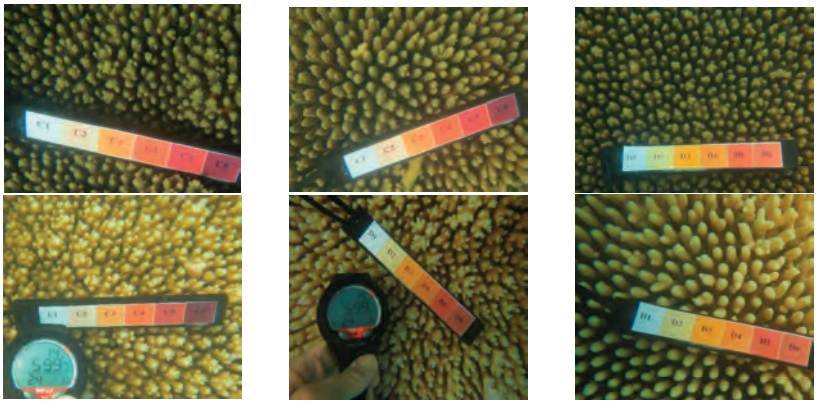


Figure 3. Example of in-water reference data of *Acropora* with relevant bleaching level: 1 (severely bleached) to 6 (unbleached). **Top left**—6, **bottom right**—1. Courtesy of Sam Noonan (AIMS).

2.2. Airborne Methods, UAV, and Sensors

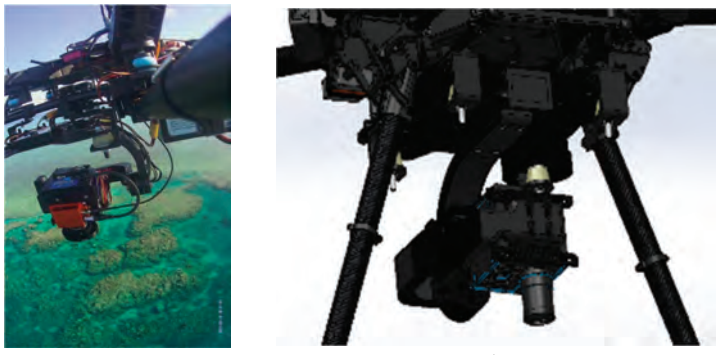
The UAV used was an S800 EVO Hexacopter (DJI-Innovations, Shenzhen, China) that weighs 6.0 kg with motors, propellers, electronic speed controllers (ESCs), control and navigation avionics, and 16,000 mAh battery. The frame is fitted with a retractable undercarriage, providing a sensor field of view clear of obstacles. The UAV has a recommended maximum take-off weight of 8 kg, thus allowing 2 kg for sensor payload. A WooKong-M flight controller provides the navigation and control systems of the UAV, and comes with a stabilization controller and GPS unit with inbuilt compass and inertial management unit (IMU). The flight controller has multiple autopilot modes to enable both remote control by operator and autonomous with position and altitude hold, and auto go home/landing with enhanced fail-safe.

In this work and methodology, we propose the use of high resolution RGB and hyperspectral UAV-based imagery, as they provide the required level of detail in capturing finer differences in coral bleaching signatures. A Canon EOS 5DS R (Canon, Tokyo, Japen) digital camera was used to capture high-resolution RGB images from the mission route, and assist in identification and monitoring of

coral in the studied area using the imagery augmented by the camera's GPS coordinates. The camera specifications include full frame (36 × 24 mm) CMOS 50.6-megapixel sensor, 28-mm Canon lens, and a single-frequency GPS receiver.

Hyperspectral imagery was acquired using a Headwall Nano-Hyperspec (Headwall Photonics Inc., Bolton, MA, USA) hyperspectral sensor. This sensor records datacubes of 274 spectral bands in the VNIR (Visible and Near-Infrared) range (400–1000 nm) with ~2.2 nm spectral interval and 5 nm spectral resolution (fwhm with 20 µm slit). This sensor is equipped with a calibrated f/1.8 4.8 mm Schneider lens, which results in 50.7 deg field of view over 640 pixels. The collected hyperspectral data cubes are synchronized with GPS/INS (Inertial Navigation System) positioning and orientation information to perform datacube orthorectification.

Image stabilization is also an important step, therefore, the hyperspectral sensor was integrated into a S800 UAV by a custom designed gimbal made by Queensland University of Technology (QUT) Research Engineering Facility (REF). This gimbal has 2-axis stabilization, that ensures seamless operation of the push-broom hyperspectral scanner in windy conditions. The gimbal design features carbon reinforcement over 3D printed structure, advanced dampening, brushless motors, and a BaseCam SimpleBGS 32 bit gimbal controller (BaseCam electronics, Riga, Latvia), with the total weight below 1 kg. Mounting the push-broom scanner on the gimbal proved to enhance camera performance in wind gust conditions by ensuring the minimal and consistent required overlaps between consecutive datacubes over the large study area. This leads to an increased overall flight efficiency in real life open environments. Figure 4a shows the S800 UAV with the hyperspectral sensor on the gimbal during one of the missions. The gimbal CAD design model is presented in Figure 4b.



(**a**) (**b**)

Figure 4. (**a**) Headwall Nano hyperspectral sensor onboard S800 UAV; (**b**) Custom-made 2-axis gimbal hosting the hyperspectral camera (SolidWorks 3D model).

2.3. Orthorectification of Hyperspectral Images

Data orthorectification (step 3) was conducted using Headwall SpectralView software, where the individual hyperspectral data cubes are geolocalised and stitched, with verification by the overlay of underwater contours of the reef. Figure 5 shows stitched images overlaid in Google Earth.

The hyperspectral data are then processed with spectral analysis tasks, such as MATLAB Scyllarus open source toolbox, Headwall SpectralView, and Scyven, to obtain radiance values which are then processed with a white reference image to generate the reflectance dataset (step 4). Once obtained, the reflectance dataset is corrected for water depth by running through a noise reduction algorithm (step 5), processed through Scyven using support vector machine automated material discovery, to cluster similar spectral signatures which are geo-referenced (step 6), before being extracted (step 7) for reef indices (step 8) and material classification (step 9).

Figure 5. Multi-orthorectification by Headwall SpectralView overlaid in Google Earth, polygonal regions (in red) indicate visible spume regions in close proximity to coral.

2.4. Image Processing, Radiance, Reflectance, White Reference

Step 4 orthorectified images were processed with Headwall SpectralView to identify the corresponding per pixel illumination value. Once completed, the image then goes through a semi-automated process in which the image is imported into Scyven, matched with a corresponding white reference illumination pattern (generated from white reference file), then lastly processed to generate the reflectance value corresponding to the white reference.

2.5. Depth Correction

Estimating and correcting hyperspectral imagery for water depth (step 5) is a complex process. ENVI offers unique tools for processing characteristics of waterbody features, and this enables quick and easy depth approximation utilizing a bottom albedo-independent bathymetry algorithm that utilizes a log ratio transformer. The log ratio transformer calculates depth independent of the bottom material and brightness (bright sand or dark seabed vegetation). The use of the algorithm means that typical errors caused by varying floor composition [5] (Figure 6) will not affect the overall accuracy of the approximation.

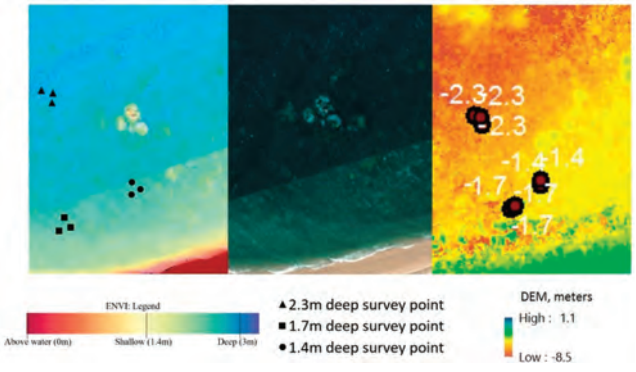


Figure 6. Depth approximation non-homogeneous composition (**left**—ENVI depth approximation; **centre**—Figure 5 extract; **right**—Agisoft Photoscan DEM (Digital Elevation Module) depth approximation).

The use of additional information from in-water surveys is required for refining and generating accurate depth results with a reasonable degree of error. Agisoft Photoscan gives an alternative approach to the water depth estimation by using the reconstruction of a DEM from multiple overlaying images, which allows depth approximation of larger areas with ease, but comes with a more limited result set where smaller depth changes can be missed.

To apply depth correction, the effects of the water column must be ascertained, and to do this, the coral samples at varying water levels are compared; Figure 7 shows the variances in spectral signatures between two samples of massive *Porites* with a bleaching level of four, with one at a depth of 1 m and the other at 2.3 m.

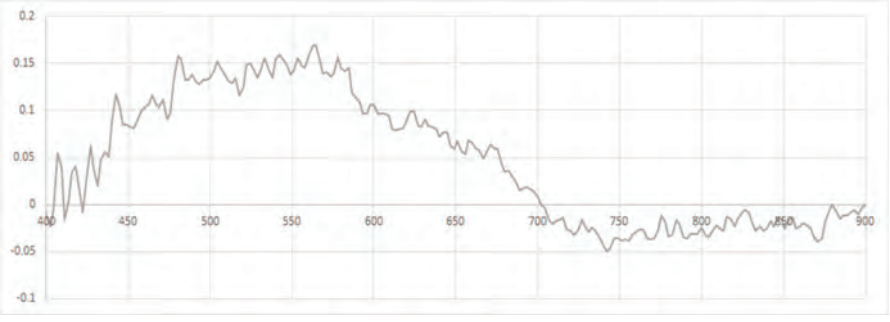


Figure 7. Spectral variance between massive *Porites* coral samples with level 4 bleaching at depths of 1 m and 2.3 m.

To derive the effect of the water column on the data, the log-transformed ratio between the green and blue wavelengths is used, otherwise, the effects of varying benthic cover are also taken into account [6]. Once the data are derived, it can be seen that this is tightly packed (Figure 8).

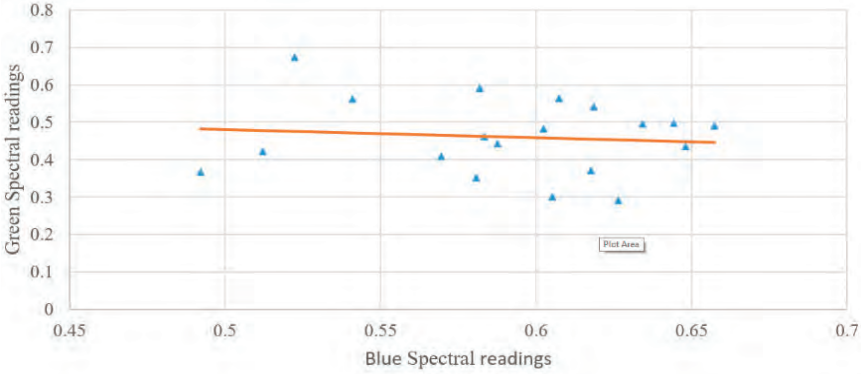


Figure 8. Spectral signature comparison between reflectance in blue and green bands of coral in different locations. Dots represent the ratio values between the green and blue spectral values, the orange line represents the trendline between all the data points.

This data is log transformed, and used as input into the equation identified by Mishra, D. [7] (below), which outputs the percentage variance due to water column effects.

$$y = -18.353x^2 + 10.805x + 0.238 \quad (1)$$

This identified that 6.4% max variation in the spectral signature was due to water column effects, which could be corrected using a noise reduction algorithm, and follows the trend which was identified by Zoffoli, M [8].

2.6. Coral Georeferencing

In-water survey data contain GPS coordinates that are then matched to imagery. This can be completed by using the orthorectified data to overlay GPS coordinates across the image, and then extrapolating the corresponding pixel coordinates; Figure 9 shows an example of extrapolated coral locations and signature extraction. This data is then stored in an attribute table (step 6), along with the depth data and other characteristics (see Table 2 for table extract).

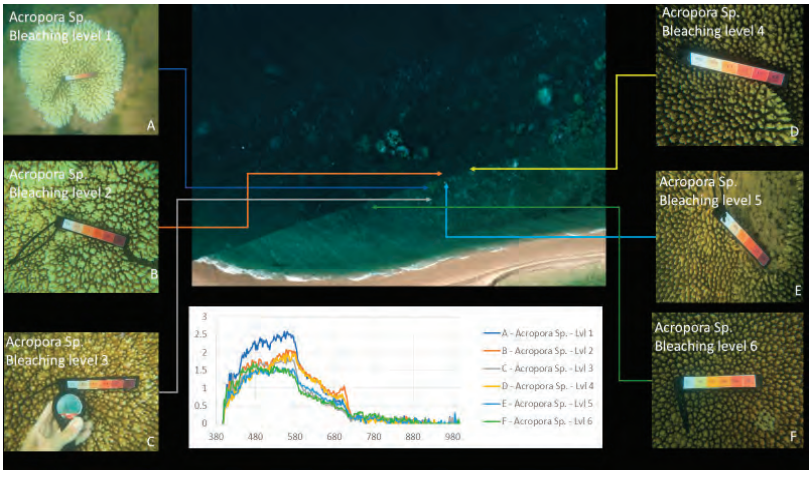


Figure 9. Examples of spectral responses "fingerprints" of bleaching levels 1 to 6 for *Acropora* coral.

2.7. Spectral Signature Extraction

Expert visual assessment and georeferencing (step 6) are used with the depth-corrected reflectance values (step 5) to extract the mean spectral signature for different corals with different levels of bleaching. Headwall SpectralView is used to identify pixels that contain corals of varying bleaching levels, for which the spectral values are then extracted with Scyven. Figure 9 demonstrates the survey points for spectral signature extraction once the images have been converted into reflectance (step 4), then preprocessed with an automated material discovery (part of the material classification phase, step 9) which groups objects of similar spectral signatures.

After each genus of coral has been identified using material discovery, they then need to be compared based on their spectral signatures to identify potential issues with material classification (e.g., spectral signatures of each bleaching level not discernible for other genus samples).

Figure 10 shows an example of the processed results where different regions are identified and grouped based on their spectral signatures. Brighter colours indicate regions with spectral patterns similar to that of the coral, sand, or other material identified in the region.

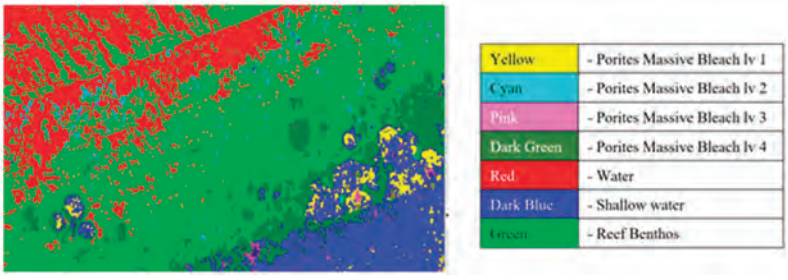


Figure 10. Coral identified with various levels of bleaching.

2.8. Reef Indices

The next step (step 8) is to calculate coral bleaching indices using the spectral signature extraction (step 7), as well as visual assessment (step 1), and georeferencing (step 6). The indices were selected to evaluate symptoms of bleaching such as higher spectral concentrations in the 400 nm and 750 nm regions. A list of the indices used, as well as their equations, is found in Table 1, where each genus of coral has been identified and then classified, based on differences in spectral samples comparing to the next closest spectral sample. Figure 11 shows an example of the spectral signature comparison used for coral bleaching index generation.

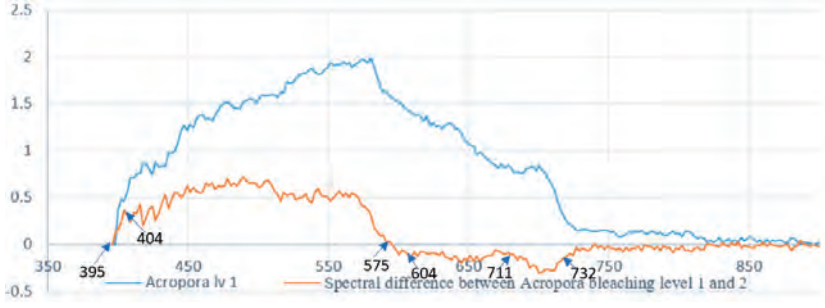


Figure 11. Spectral signature comparison for reef index generation Alv1.

2.9. Classification

Classification is the next step, where the in-water survey data along with the spectral signature extraction are combined using ENVI to develop pixel-by-pixel analysis algorithms, and also, subpixel (spectral mixture analysis) approaches using support vector machine (SVM). SVM was chosen due to the high classification accuracy, ability to handle multiple predictor variables, in-built handling of cases where spectral samples cannot be completely separated, and its ability to handle large datasets. It has also been found that the accuracy of SVM could be further increased if the gamma of images is increased up to a maximum of 75% (variations between captured data brightness and ideal gamma level are expected). This change causes SVMs to be equivalent to naive Bayes classifiers; Zanaty [9] proposed the SVM kernel of Gaussian radial basis polynomials function, which was found to be nearly 10% more accurate than the standard radial bias SVM classification method. Even without this, SVM is equivalent to the multilayer perception neural network in terms of accuracy of result. SVM using radial

bias algorithm (shown below in Equation (2)) is therefore the method used in this paper, within the ENVI system, for material classifications.

$$Radial\ Basis\ Function:\ K(x_i, x_j) = exp\left(-\gamma||x_i - x_j||^2\right), \gamma > 0 \quad (2)$$

Scyven is also used to apply various noise reduction algorithms and data refinement through targeted unsupervised classification, to identify regions requiring additional accuracy refinements.

3. Field Experiments

3.1. Site and In-Water Surveys

The site surveyed on 15 March 2017 is at Pandora Reef located in Queensland, Australia (Figure 12). The Australian Institute of Marine Science (AIMS) monitors this site regularly, and advised its selection due to the diverse range of coral genera, bleaching levels, and varying water depths. This paper demonstrates the method for an area of Pandora Reef in which 64 in-water survey points were collected to act as ground truths for spectral signature extraction (step 7) and training for the material classification (Table 2).

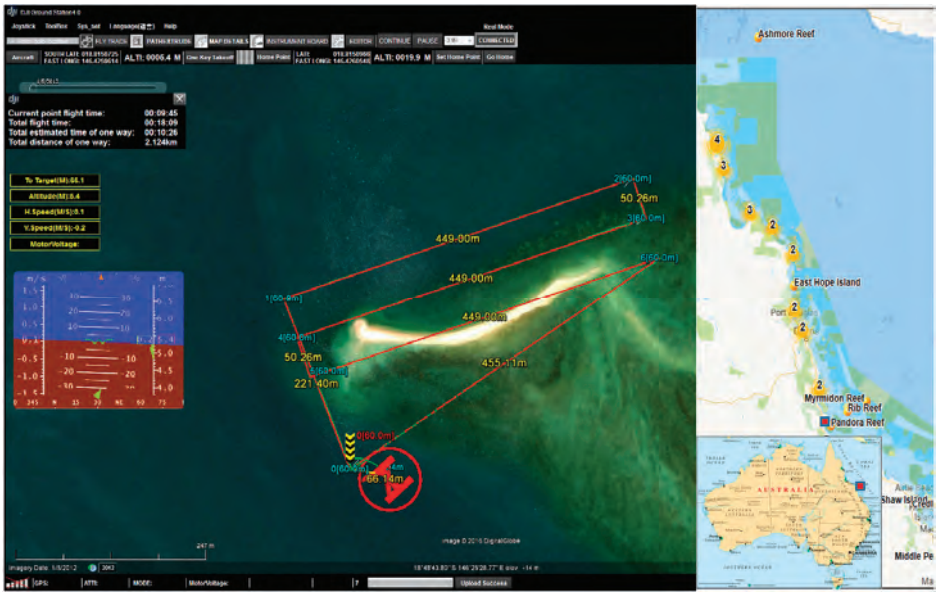


Figure 12. Pandora Reef site, Great Barrier Reef, QLD, Australia.

3.2. In-Water Data Collection Survey

Table 2 summarizes the data which were collected from the in-water surveys on the 15 March 2017 at Pandora Reef. The following attributes were collected: date, reference image id, coral genus, bleaching level, depth, and additional notes about surroundings. These points were then mapped onto the orthorectified data, where its respective location is stored as a set of pixel coordinates. Figure 13 shows all the in-water survey locations mapped in Google Earth.

Figure 13. Numbered in-water survey GPS coordinates at Pandora Reef shown through Google Earth.

4. Results

4.1. Water Depth and Water Depth Extraction Results

Figure 6 shows the result from ENVI's relative water depth tool. Application of the bathymetry method is able to determine relative water depth by utilizing a log ratio transform. Visual assessment shows that the results in Figure 6 are consistent with the researcher's knowledge of the study area and the in-water surveys (Figure 3).

4.1.1. Orthomosaics

The orthomosaic of the entire Pandora Reef (Figure 14) assisted in water depth extraction and generic reef structure. The orthomosaic was generated from Phantom 3 Pro images and is a geospatial RGB aerial view of the reef, with 5.71 cm/px resolution in a format of 234 megapixel image over 62.3 Ha area. The area was covered within a 20 min flight. Approximate areas covered by the underwater surveys are highlighted by red (3 m depth, 0.1 Ha) and white (6 m depth, 0.1 Ha) polygons.

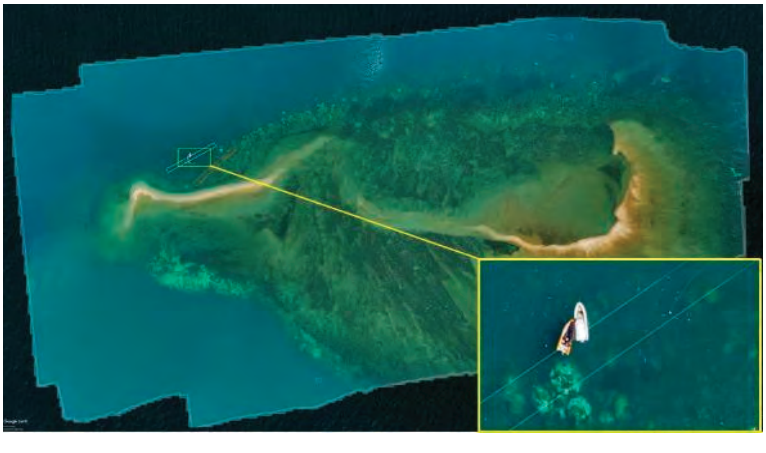
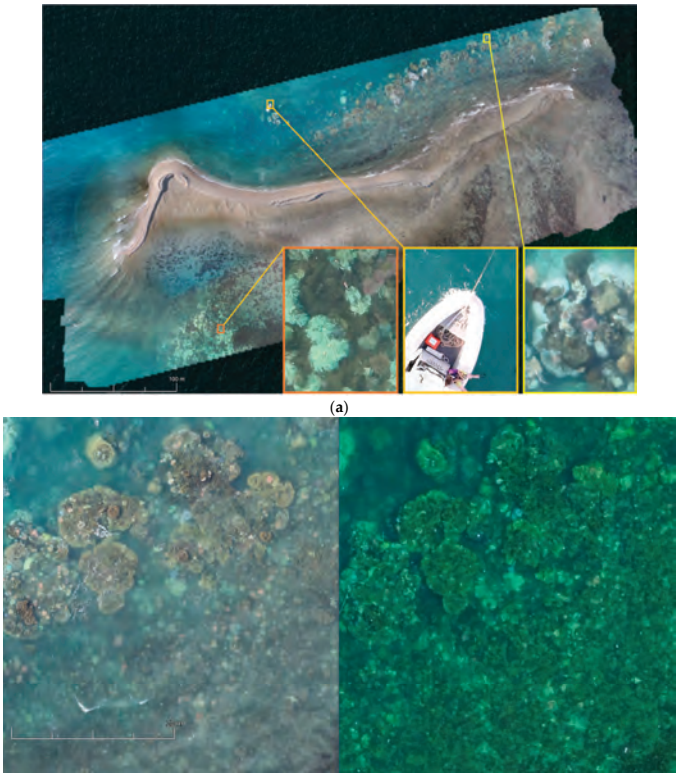


Figure 14. Photogrammetry orthomosaic of Pandora Reef (63 Ha) shows areas of day long in-water surveys performed by AIMS at the depth of 3 m (orange area) and 6 m (white area) 400% zoomed fragment on the bottom right (tide height 2.6 m).

Processing Canon 5DS R images gives significantly better 9.91 mm/px resolution of the orthomosaic/3D mode/DEMl, also bringing larger dynamic range to imagery (Figure 15b), and fully customised lenses and shooting options.

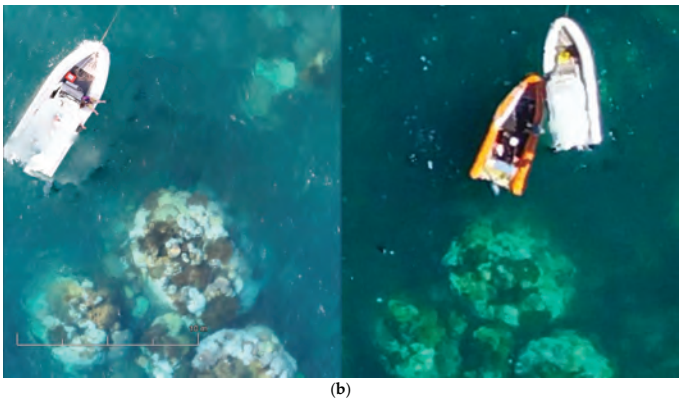
Figure 15a shows an orthomosaic generated out of 395 50,6-megapixel Canon 5DS R images processed on medium quality settings, which resulted in a 1.33 gigapixel image (47,904 × 29,080) over an 8 hectare area of primary AIMS interest. These orthomosaics give researchers a method of monitoring benthic cover and geomorphological variances in benthic algae (Murfitt et al. [10]). Orthomosaics assist in identifying regions that are at greatest risk of degradation, and ascertain what the cause of this degradation is and what systems need to be employed to try and reduce/prevent further degradation in that region. The orthomosaics also give researchers a method for identifying in-water survey data points, and giving approximations of what coral is typically found in this region. For example, Palma et al. [11] found that small rocky outcrops with a regular shape were typically indicative of soft corals whereas large colonies in rocky regions typically indicated *Acropora* coral.

Airborne ultra-HD DSLR cameras provide researchers with a complex and fully adjustable tool resulting in very detailed snapshots of the marine environment. Larger dynamic range, higher resolution, more accurate focusing and white-balancing comes at the cost of a heavier and larger UAV to carry the sensor. We note also that other factors, such as correct camera settings and camera orientation relative to the sun and the object of interest, position of the sun, tide heights, sea state, and water torpidity, define the success, and sometimes are the most important factors for reef monitoring by remote sensing.



(a)

Figure 15. *Cont.*



(**b**)

Figure 15. (**a**) Photogrammetry orthomosaic of western part of Pandora Reef (8 Ha) shows 400% zoomed fragments of bleaching events and the AIMS's Antares research boat on the bottom right (tide height 1.6 m). (**b**) Fragments of the orthomosaics from Canon 5DS R (left) and DJI Phantom 3 Pro imagery (right).

4.1.2. Photogrammetry Ocean Floor Digital Elevation Models

Figure 16 shows the overlaid example of the DEM generated of the ocean floor surface of a fragment of Pandora Reef reconstructed out of Phantom 3 Pro (22 cm/px resolution) and Canon 5DS R (3.96 cm/px resolution) imagery. Despite the presence of minor artefacts and anomalies due to the presence of the moving border between two different mediums (sea and air), we observed adequate level of detail and accuracy of reconstruction for shallow waters (up to 5–8 m deep).

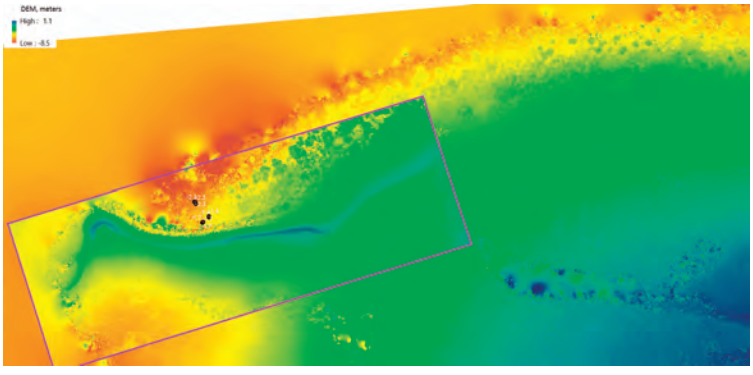


Figure 16. Digital elevation model (DEM) of Pandora Reef generated from Phantom 3 Pro and Canon 5DS R (purple area) and in-water survey depths (black dots).

4.2. Spectral Signature Extraction Results

Figures 10, 17 and 18 show an example of spectral signature extraction for one coral type. Figure 18 shows the results with *Porites* massive bleaching level one and six, Figure 18 with soft corals level 5, *Porites* massive level 4, *Goniopora* level 3, and *Acropora* level 2. These spectral similarities will be one of the greatest potential causes of error in the classification step (step 9) and require processing of each identified point to confirm that the classification is correct. This will be done by solely comparing the classifications in the bands where differences are discernible.

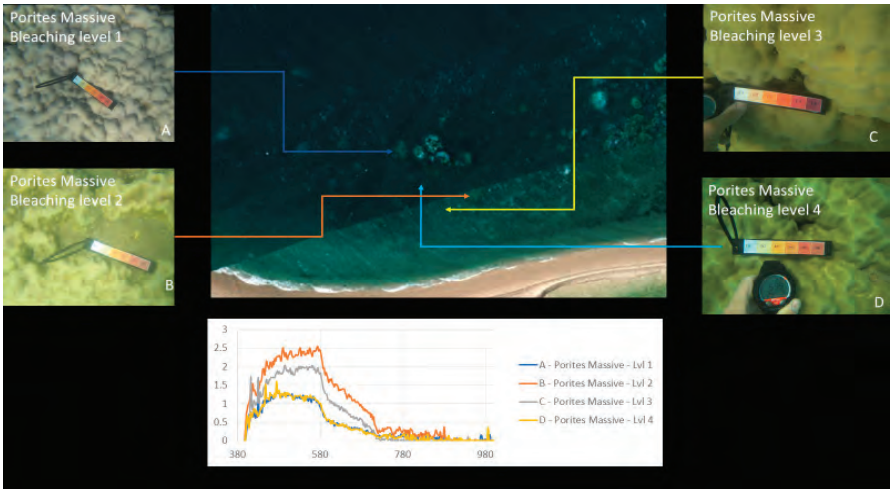


Figure 17. Example of spectral response "fingerprints" of bleaching levels 1 to 4 for *Porites* massive coral.

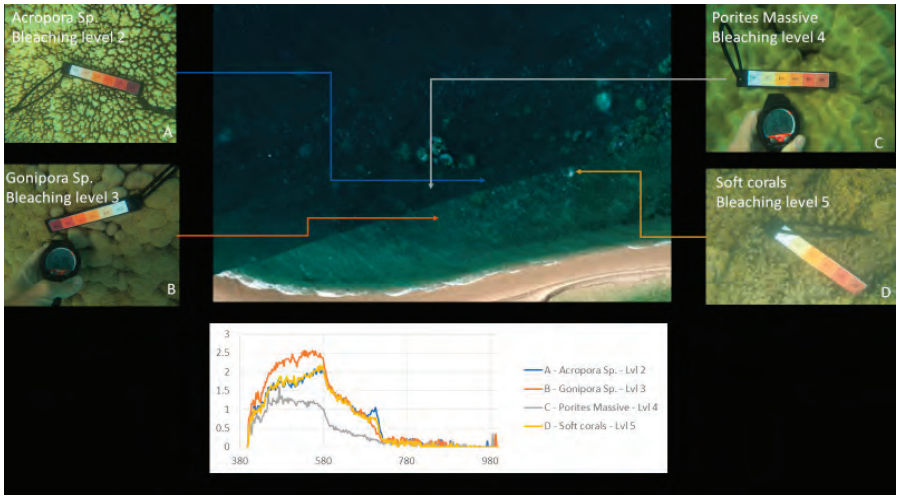


Figure 18. Example of spectral response "fingerprints" of bleaching levels 2, 3, 4 and 5 for *Acropora*, *Gonipora*, *Porites* massive and soft corals, respectively, for spectral comparison.

4.3. Classification Results

Material classification through Scyven, although very similar to ENVI, is done by taking the spectral fingerprint data (Figures 9, 17 and 18) and then running it through an automated process, which matches and groups sets of specific spectral data based on their similarity. Figure 19 shows the results of material classification applied to *Porites* massive coral with the other object classification omitted for clarity.

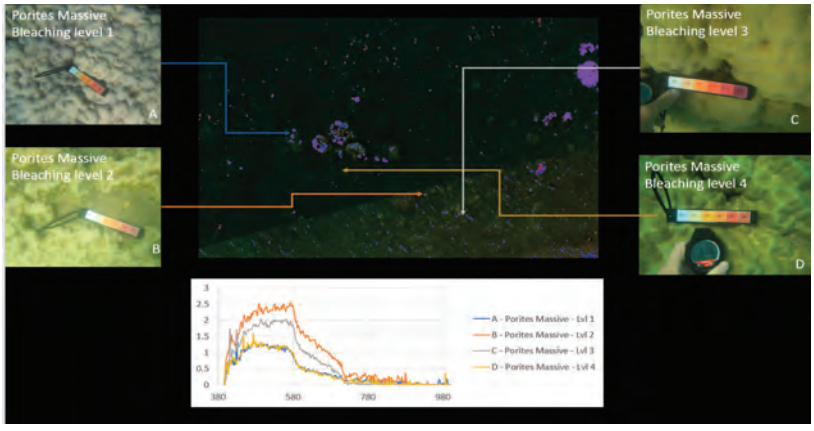


Figure 19. *Porites* massive material classification, with water classification omitted.

Figure 20 shows output of the automatic unsupervised material classification from ENVI, to highlight and show variations which are typically associated with *Porites* massive coral indices (Table 1). These data are utilized to show regions that require exclusion for reduction of false positives.

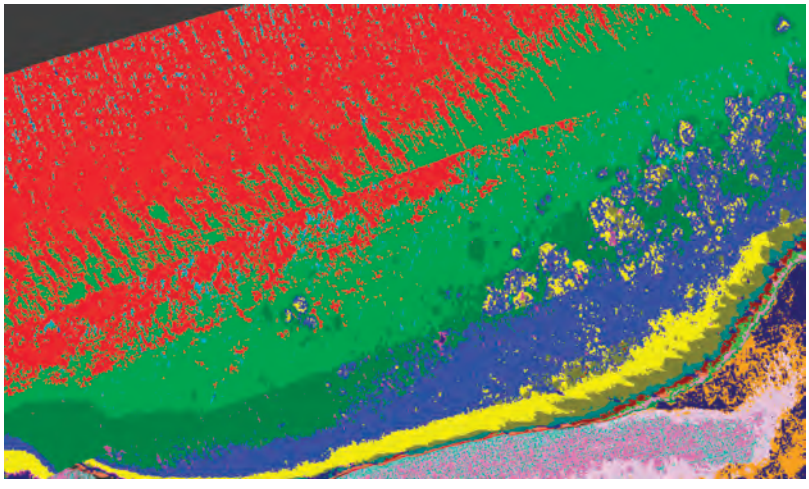


Figure 20. Unsupervised material classification for Pandora Reef.

Once the data has been refined, it can then be classified using the extracted spectral signatures (step 7 Figure 1) and the reef indices generated (step 8 Figure 1), whilst also being careful and taking note of the regions that were highlighted by the unsupervised classifications. For example, in the case of Figure 20, the benthos is shown as light green, spume as light yellow, and the numerous additional misclassifications caused by the ill-classified sandbar. Figure 21 shows the results of this classification after refining the extracted spectral signatures using additional noise reduction algorithms in the form of spectral and spatial filtering, additional ground truth discrimination from unsupervised classification results, and removal of false positives highlighted by coral bleaching indices (Table 1) developed from Figure 20 and Table 2.

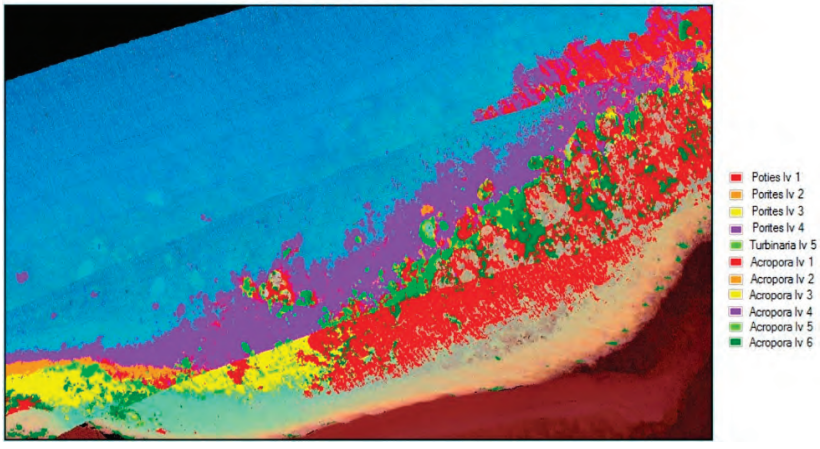


Figure 21. Support vector machine (SVM) material classification for Pandora Reef.

Table 3 Results of the accuracy assessment with labelled data material classification has been completed.

Table 3. Material classification accuracy assessment for Figure 21 (EA = excessive area).

Coral Type	Bleaching Level	Signature Accuracy (%)	Points Found	Accuracy (%)	Found Pixels	Area (%)	Overall Accuracy (%)
Porites massive	1	79.17	2/3	66.667	91,157	2.932	66.67
Porites massive	2	94.44	1/2	50.00	30,007	0.965	50.00
Porites massive	3	68.75	2/7	28.571	55,529	1.786	28.57
Porites massive	4	100.00	4/5	80.00	339,305	10.915	0 (EA)
Acropora sp.	1	100.00	7/13	53.846	173,810	5.591	53.85
Acropora sp.	2	46.67	2/4	50.00	1248	0.040	46.67
Acropora sp.	3	100.00	2/2	100.00	38,729	1.246	100.00
Acropora sp.	4	100.00	2/2	100.00	41,618	1.339	100.00
Acropora sp.	5	100.00	3/14	21.428	32,119	1.033	21.43
Acropora sp.	6	100.00	2/2	100.00	115,879	3.728	0 (EA)
Soft coral	5	44.00	2/2	100.00	392,569	12.628	0 (EA)
Turbinaria sp.	5	73.91	2/2	100.00	2763	0.089	73.91

The classification uses all the information in the image for classifying the different coral types. The accuracy results in Table 3 show the effect of including images with spume during the image processing phase. Classifications which have been greatly affected have been marked with EA (excessive area) in the table. Figure 22 shows the improved results of refining the data by excluding those regions of spume (marked in Figure 6), and this is reflected in Table 4.

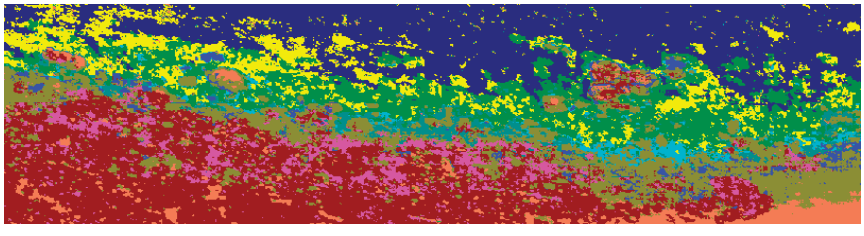


Figure 22. Support vector machine (SVM) material classification for Pandora Reef with spume regions marked in Figure 5 excluded.

Table 4. Material classification accuracy assessment for Figure 22 at Pandora Reef where regions with spume region marker in Figure 5 are excluded.

Coral Type	Bleaching Level	Signature Accuracy (%)	Points Found	Accuracy (%)	Found Pixels	Area (%)	Overall Accuracy (%)
Porites massive	1	88.79	2/2	100.00	238,998	6.636	88.79
Porites massive	2	89.32	2/2	100.00	57,323	1.592	89.32
Porites massive	3	96.32	1/1	100.00	221,203	6.142	96.32
Porites massive	4	88.13	3/3	100.00	45,386	1.260	88.13
Acropora sp.	1	96.54	4/4	100.00	1,130,663	31.396	0 (EA)
Acropora sp.	4	93.54	2/2	100.00	55,085	1.530	93.54
Acropora sp.	5	90.75	3/3	100.00	477,663	13.264	0 (EA)
Acropora sp.	6	90.27	2/2	100.00	214,046	5.944	90.27

4.4. Index Results

Different indices were computed in Section 2.8 in order to highlight the symptoms of bleaching across various genera of coral. Figure 23 shows examples of two NDVI images in the green and yellow wavelengths (identified by indices as bands with greatest variances over multiple bleaching levels), as well as two coral bleaching indices calculated from the hyperspectral data for the *Acropora* and *Turbinaria* genera of coral. This comparison shows the overall difference between the index output and the NDVI, where the NDVI images gives a clear representation, primarily focusing on the healthiest coral. The indices are capable of showing the overall health of the reef by not only showing the healthiest parts, but also all the various genera of coral at different levels of bleaching, and give health approximations based on benthic cover.

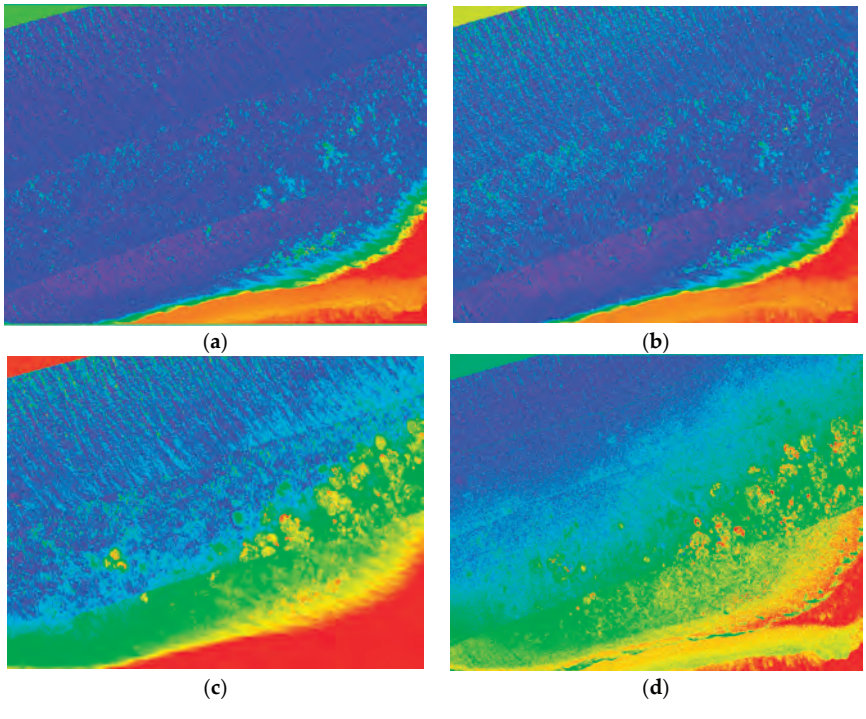


Figure 23. Full colour spectrum images blue to red intensity (red high), (**a**) NDVI yellow; (**b**) NDVI green; (**c**) *Turbinaria* index Tlv5.3; (**d**) *Acropora* index Alv1.2.

The results of the mean spectral signature extraction show a trend between spectral reflectance and severity of bleaching, which is consistent across various genera of coral. Coral bleaching indices developed from Figure 12 allow for analysis of stress caused by bleaching, however, these indices are not always able to distinguish coral genus from bleaching level, and may only be useful in monitoring temporal changes. NDVI images give a more easily interpreted representation, but cannot distinguish between variances of different genera and only show limited reef health data.

4.5. Limitations

The methodology shown in this manuscript has limitations, and requires careful and detailed planning of in-water and airborne surveys that can affect the overall feasibility and accuracy of the classification. The ability to classify objects can be greatly distorted or made near impossible, depending on water depth and the methodology used. Lesser and Mobley [12] found that it is possible to conduct coral classification with hyperspectral imagery to depths of around 10–12 m; this, however, requires the use of the Lambert–Beer law, which requires additional data of measured photoabsorption from testing areas. In this work, we use a band ratio equation which drops the maximum classification depth to around 6–8 m, which is satisfactory for large-scale remote sensing in either inaccessible or near-inaccessible areas, and offers options for health monitoring, however, this is a limitation for remote sensing of reef sections that do not fit these constraints.

Water turbidity, tidal conditions, and weather conditions all play a part in affecting the overall quality of the captured data. Water turbidity can affect the ability to identify coral in regions and, depending on its level, can prevent the ability to identify any coral. Nonetheless, data with moderate levels of turbidity still have some value in showing the current level of benthos/symbiotic algae in the region. Tidal conditions can cause significant amounts of spume to be created, methods can be employed to remove sections affected, but their effectiveness is limited by the spumes' location and, where classification is required, can cause serious issues with the overall accuracy of results.

5. Conclusions

The results show that airborne UAV-based hyperspectral imagery has the potential to detect coral bleaching over large areas across multiple coral genera, where accurate visual inspection may not be possible. The methods and workflow developed during this study enabled material classification using both supervised and unsupervised methods to gain results with an acceptable percentage of error. The results presented and coral indices generated will contribute to the generation of valuable information for reef protection by offering methods that highlight possible symptoms of bleaching in a timely manner. It could also be extrapolated to other areas of research in remote sensing, such as mineral exploration, biodiversity, and ecological assessment.

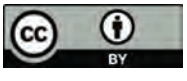
Author Contributions: F.G. and D.B. conceived and designed the field and airborne experiments, guided airborne surveys and quality assessed the collection of the UAV imagery. D.B. undertook the relevant airborne surveys and monitored in-water surveys. M.P., F.G., D.B. conceptualized the methodology in Figure 1. M.P. developed and implemented the methodology, imagery workflow and analysis. K.J.G. guided result conceptualization and project limitation identification. M.P. and D.B processed the data. M.P was the principal author whilst all authors contributed significantly to the whole paper.

Acknowledgments: This work was funded by the Institute for Future Environments at Queensland University of Technology (QUT), with logistics support by The Australian Institute of Marine Science (AIMS). We would also like to acknowledge the High Performance Computing and Research Support Group at QUT for the computational resources and services used in this work, Sam Noolan for his assistance with collecting in-water survey data, Research Engineering Facility (Gavin Broadbent, Dmitry Bratanov) at QUT for operation of the DJI S800 UAV and sensors in the data collection and assistance in the data processing. With special thanks been given to Kaveh Deilami, Kyle Moyle and Juan Sandino Mora for their assistance during this work.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Hughes, T.; Kerry, J.; Baird, A.; Connolly, S.; Dietzel, A.; Eakin, M.; Heron, S.; Hoey, A.; Hoogenboom, M.; Liu, G.; et al. Global warming transforms coral reef assemblages. *Nature* **2018**, *556*, 492–496. [CrossRef] [PubMed]
2. AIMS.gov Resources: Coral Bleaching Events. 2016. Available online: http://www.aims.gov.au/docs/research/climate-change/coral-bleaching/bleaching-events.html (accessed on 10 April 2018).
3. Anderson, D.; Armstrong, R.; Weil, E. Hyperspectral Sensing of Disease Stress in the Caribbean Reef-Building Coral, *Orbicella faveolata*—Perspectives for the Field of Coral Disease Monitoring. *PLoS ONE* **2013**, *8*, e81478. [CrossRef] [PubMed]
4. Goodman, J.A.; Purkis, S.J.; Phinn, S.R. *Coral Reef Remote Sensing. A Guide for Mapping, Monitoring, and Management*; Springer: Dordrecht, The Netherlands, 2013.
5. Hamylton, S.; Hedley, J.; Beaman, R. Derivation of High-Resolution Bathymetry from Multispectral Satellite Imagery: A Comparison of Empirical and Optimisation Methods through Geographical Error Analysis. *Remote Sens.* **2015**, *7*, 16257–16273. [CrossRef]
6. Gordon, H.R.; Brown, O.B. Influence of bottom albedo on the diffuse reflectance of a flat homogeneous ocean. *Appl. Opt.* **1974**, *13*, 2153–2159. [CrossRef] [PubMed]
7. Mishra R, D.; Narumalani, S.; Rundquist, D.; Lawson, M.; Perk, R. Enhancing the detection and classification of coral reef and associated benthic habitats: A hyperspectral remote sensing approach. *J. Geophys. Res. Oceans* **2007**, *112*. [CrossRef]
8. Zoffoli, M.; Frouin, R.; Kampel, M. Water Column Correction for Coral Reef Studies by Remote Sensing. *Sensors* **2004**, *14*, 16881–16931. [CrossRef] [PubMed]
9. Zanaty, E. Support Vector Machines versus Multilayer Perception in data classification. *Egypt. Inform. J.* **2012**, *13*, 177–183. [CrossRef]
10. Grasmuek, M.; Eberli, G.; Viggano, D.; Correa, T.; Rathwel, G.; Lou, J. Autonomous underwater vehicle (AUV) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the Straits of Florida. *Oceans* **2006**, *33*. [CrossRef]
11. Murfitt, S.; Allan, B.; Bellgrove, A.; Rattray, A.; Young, M.; Lerodiaconou, D. Applications of unmanned aerial vehicles in intertidal reef monitoring. *Sci. Rep.* **2017**, *7*, 10259. [CrossRef] [PubMed]
12. Lesser, M.; Mobley, C. Bathymetry, water optical properties, and benthic classification of coral reefs using hyperspectral remote sensing imagery. *Coral Reefs* **2007**, *26*, 819–829. [CrossRef]



sensors

MDPI

Article

Online Aerial Terrain Mapping for Ground Robot Navigation

John Peterson *, Haseeb Chaudhry, Karim Abdelatty, John Bird and Kevin Kochersberger

Virginia Tech Unmanned Systems Laboratory, Blacksburg, VA 24060, USA; haseeb7@vt.edu (H.C.); karim@vt.edu (K.A.); jobird@vt.edu (J.B.); kbk@vt.edu (K.K.)

* Correspondence: jrpeter@vt.edu

Received: 1 December 2017; Accepted: 11 February 2018; Published: 20 February 2018

Abstract: This work presents a collaborative unmanned aerial and ground vehicle system which utilizes the aerial vehicle's overhead view to inform the ground vehicle's path planning in real time. The aerial vehicle acquires imagery which is assembled into a orthomosaic and then classified. These terrain classes are used to estimate relative navigation costs for the ground vehicle so energy-efficient paths may be generated and then executed. The two vehicles are registered in a common coordinate frame using a real-time kinematic global positioning system (RTK GPS) and all image processing is performed onboard the unmanned aerial vehicle, which minimizes the data exchanged between the vehicles. This paper describes the architecture of the system and quantifies the registration errors between the vehicles.

Keywords: unmanned aircraft; air–ground cooperation; multi-robot coordination

1. Introduction

This work presents a multi-robot system for efficient exploration and navigation in outdoor environments. The system consists of a ground vehicle (the Clearpath Jackal) , and a custom multi-rotor. The objective of the mission is to quickly and efficiently drive an unmanned ground vehicle through an unknown outdoor environment. Ground vehicles are typically equipped with a variety of sensors for navigation and obstacle avoidance. Ultrasonic, Light Detection and Ranging (LIDAR), camera, and radar sensors are just a few potential examples. These sensors have different maximum ranges, but they share the requirement for line of sight for obstacle detection, ignoring radar reflections. Many algorithms for efficient path planning exist. Common algorithms, including A*, will plan optimally in discrete environments and there are extensions to continuous coordinates. However, the optimality of these paths is limited by the planner's knowledge of obstacles. The line of sight of sensors on a ground vehicle is limited by the geometry of the environment. In indoor planar environments, line of sight is limited by walls and other obstacles, but in an outdoor environment the ground surface and vegetation also restrict it. For outdoor robots, a common way to improve the range of line of sight is to mount the sensor higher up on the vehicle. This provides the vehicle with a better perspective to see over hills and tall grass.

A logical extension is to use a remote sensor mounted on an aerial vehicle to obtain a completely different perspective on the scene. When using an overhead downwards-facing perspective, there is a compromise between area covered and resolution on the ground. At the extreme, a satellite in orbit can see large portions of the Earth's surface at any given moment, but there are restrictions on the clarity of these images because of the atmosphere and limitations on the time of day that data may be acquired, as dictated by the satellite's orbit. This data may be too out of date to be useful for navigation in dynamic environments. Fixed wing aircraft can circle an area for extended periods of time, but their size and airspeed may require observation to be done from a higher altitude than is ideal. A common

way to increase the spatial resolution from the air is to use higher-resolution cameras. However, in their absence, another approach is to fly at lower altitudes, acquire multiple images of subsets of the mission area, and combine the images into a single high-resolution map.

This system uses a multi-rotor to capture aerial imagery of a mission area to construct a terrain map to efficiently navigate a ground robot to a destination. Planners typically minimize path length which does not necessarily minimize energy consumption. In radiation search operations using scene understanding with autonomous unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) in [1], the authors show the difference in the power consumed by a ground vehicle traversing grass and pavement. This work utilizes overhead imagery to derive terrain classes, infers traversibility costs from these terrain classes, generates an energy-efficient trajectory through this cost map, and then commands the ground vehicle to follow this trajectory.

2. Background

It is common to use aerial maps to augment navigation by ground vehicles. Before its use in robotics, there was clearly a benefit to using a higher vantage point, which afforded a greater and more unobstructed field of view. Within robotics, several different approaches are taken to leverage aerial data for ground robot navigation. Many approaches consider the aerial data collection as a separate mission that might occur days or weeks before the ground vehicle is deployed on the scene. This implies some amount of human intervention in data processing, such as moving files from one vehicle to another, or manually running software on the acquired data. In these surveying missions, an entire area might be imaged or scanned in advance to ensure that aerial data is available for the entire region. Unlike some of these previous works, the approach presented in this paper aims to tightly couple the aerial data collection with the ground level navigation. Rather than blindly scanning an entire area, the aerial vehicle only surveys the area required for the ground vehicle to reach its goal. Given that this work covers aerial mapping, multi-robot systems, computer vision, and ground-level navigation, there are several works which overlap in part or in whole with the system presented here.

In [2], Templeton et al. presented a vision-based terrain mapping system for landing an autonomous helicopter. Their system fuses feature tracking with a global positioning system (GPS) and inertial measurement system (INS) to estimate the global position and orientation of the camera. This estimate of the motion of the camera is used to estimate the 3D position of tracked points. In this case, no ground vehicle is involved at all, but their terrain mapping procedure is online to enable the helicopter to make its autonomous landing.

In [3], Papadakis presented a relatively recent survey covering visual and LIDAR-based terrain traversability estimation methods. The method employed in this paper is meant to be a place holder until a more sophisticated algorithm, perhaps from the set presented in Papadakis' work can be selected and used.

One of the components of this system is an operator control station (OCS) that enables the operator to effectively supervise and control the two vehicles. In [4] Owens et al. created an OCS to enable a single operator to control a large team of robots. Their OCS constructs a visualization of the environment by applying aerial imagery as a texture to a mesh of the environment in real time. However, this mesh is constructed from pre-collected LIDAR scans of the area with 1-m resolution. Their texturing approach is actually quite similar to the method employed in this paper for image assembly.

Quite a few methods take the approach of collecting and preprocessing the aerial data so the ground vehicle is just given a path and follows it. In one of the earlier works in this specific area [5], Stentz et al. treated the aerial perspective as being just a mobile sensor of the ground vehicle. They demonstrated that the aerial perspective of the ground improved the efficiency of planning, even when the maps generated were stale due to changes in the environment. However they were unable to autonomously fly the UAV, so they relied on manual flight of the UAV to collect data from the area before deploying the ground vehicle. The authors did conduct some simulation work to explore the

concept of UAV exploration. In a work the following year by Stentz et al. [6], the authors improved the registration of aerial collected laser scans and improved the stereo algorithm and hardware.

In [7] Vandapel et al. created a terrain mapping system utilizing aerial collected LIDAR scans. Their method, like other LIDAR-based methods, separates vegetation from the ground surface to generate a traversability map. This particular paper focused on evaluating the quality of the generated terrain maps. In a subsequent work [8], Vandapel et al. went into greater detail, describing their multi-robot system that augments the ground navigation of a ground vehicle with pre-collected aerial LIDAR data. In this work, they register a mesh generated from ground-level laser scans with the mesh generated from aerial collected laser scans to localize the ground robot in the prebuilt map, rather than relying on the GPS.

Similarly, the authors of [8,9] examined the registration of LIDAR between aerial and ground vehicles by adapting the iterative closest point (ICP).

In [10] MacArthur et al. used a stereo system on a UAV to identify simulated land mines in a images, localize marked points in the images of an open field, and send a UGV to those locations. In [11] the authors used a combination of satellite-based imagery and data from the Shuttle Radar Topography Mission to generate an aerial terrain map. They planned a risk-sensitive path using A* on this terrain map for a ground vehicle. Naturally all of this space based overhead data was collected long in advance of the ground vehicle's mission.

Chaimowicz et al. in [12] deployed a large multi-robot system consisting of both aerial and ground vehicles. They explored two methods of localization for their aerial vehicles, one using the GPS and INS of the UAV for its localization, and the other method based on image features correspondences. They demonstrated their localization and navigation systems and conducted an experiment where they localized one of the ground vehicles from the air. They ultimately found that both the GPS/INS method and the image feature-based methods had their limitations and neither method was perfect.

In [13], Hudjakov and Tamre applied neural network-based cost generation to pre-constructed orthophotos of the mission area. The authors use negative weights for the output of the neural network to represent easily traversable classes such as road and grass, with positive weights assigned to impassable terrains. They then applied a positive offset to the output for test data to ensure that all costs were positive for path planning with A*. The resulting path however does not appear to have been executed on a ground vehicle.

Giakoumidis et al. presented a more recent take on a small-scale indoor implementation of a multi-robot collaborative UAV–UGV system in [14]. They employed a Hue Saturation Value (HSV) classifier for obstacle detection in an indoor environment on imagery stitched together using Scale-Invariant Feature Transform (SIFT) features. Finally, they generated a slowness map which represents a distance transform to the nearest obstacle at each point in the map, and planned a minimum time trajectory for a ground robot.

Over the past 15 years, a variety of work has been done in autonomous collaborative UAV and UGV systems, but there has been relatively little research specifically on methods that operate both vehicles simultaneously and process data online.

3. Problem Statement

The goal of this project is to construct a multi-robot system to efficiently navigate an environment with a ground robot. Ground robots normally navigate using data collected from onboard sensors, however these sensor have limited range and field of view, and are subject to line of sight restrictions. Because of these limitations, navigation in partially explored environments requires replanning as obstacles and impassible terrain are discovered. Even if the underlying planning algorithm used by the ground vehicle is capable of generating efficient paths, the actual path taken by the vehicle will not be the shortest possible path because it will have to explore and replan. The only way to ensure that the UGV will plan and navigate optimally is to have complete knowledge of the mission area.

By using a UAV to scout ahead without being hindered by terrain, we can gather this data efficiently and plan the path of the ground vehicle with more complete information.

Unlike some of other UAV–UGV collaborative systems, this one has the goal of operating in real time, simultaneously. While the UAV is still flying in the air collecting imagery, the goal is to be able assemble that imagery into an aerial terrain map, plan a path for the ground vehicle, and successfully navigate the ground vehicle to the destination, so that the whole process may be repeated for the same flight.

4. Assumptions

To construct this system in a timely fashion, several simplifying assumptions were made. In principle, these assumptions could be relaxed without changing the architecture of the system, but it would require the substitution of several components in the software. The greatest assumption that this system makes is that the environment is approximately planar. In terms of hardware, a real-time kinematic global positioning system (RTK GPS) base station defines the origin of the common coordinate frame between the robots. The height of this antenna above the ground is measured by hand and used to define the height of the ground plane in the common coordinate frame. This assumption of an approximately 2D ground plane simplifies the obstacle mapping of each vehicle and navigation tasks of the ground vehicle.

For the ground vehicle, this allows the open source package Move Base [15] to be used with a single plane-scanning LIDAR. Move Base operates on 2D obstacle maps, so future 3D reconstructions such as height maps or occupancy voxel grids which can be converted into a 2D obstacle map will still function.

On the aerial vehicle, this simplifies the computer vision task for assembling an orthomosaic and allows the use of a single monocular camera. For the sake of simplifying the obstacle detection with a monocular camera, we assume that obstacles that the UGV is unable to pass through are denoted with a single, known, solid color. In the future, the UAV will be equipped with a stereo camera pair to construct height maps in addition to the colored orthomosaic maps. This will allow obstacles to be naturally detected as discontinuities in the height map.

We also assume that the mission area has been defined in advance and is relatively small, on the order of half a square kilometer. This reduces the requirements in the range of the radios employed and enables modest fixed-sized maps to be used by the software, rather than more sophisticated scrolling implementations.

5. Approach

The system presented here consists of a pair of robots which are supervised and directed from an OCS. The Robot Operating System (ROS) is used to integrate open source and custom software components together providing a framework for organizing the software and interfacing with the hardware.

5.1. Hardware

The system consists of two mobile robots, a Jackal (a small ground robot produced by Clearpath Robotics [16] shown in Figure 1), and a custom hexacopter named the Bogey (shown in Figure 2).

The multi-rotor is a hexacopter equipped with a Pixhawk flight controller running Arducopter firmware [17] and an Nvidia TX2 computer [18]. The Pixhawk flight controller is responsible for the low-level stability of the aircraft, provides a position interface for moving the aircaft, and performs state estimation fusing an inertial measurement unit (IMU), magnetometer, GPS, and barometric pressure into an extended Kalman filter (EKF). MAVROS [19] implements the MAVLink communication protocol enabling the ROS system to interface with the flight controller to receive state estimates and sensor data as well as issue commands. The UAV is equipped with a Basler acA1920–155uc for imaging.

A low-level controller on the Jackal combined with a ROS driver node from Clearpath Robotics offers a ROS-based interface for the Jackal hardware out of the box. The Jackal is equipped with a Hokuyo UTM–30LX mounted parallel to the ground for obstacle detection. The Jackal's integrated magnetometer and IMU are combined with wheel odometry for orientation estimation, while GPS position fixes are combined with wheel odometry for position estimation in an EKF.

Both vehicles are equipped with 5.8 GHz Ubiquiti Bullet M5 radios which both connect to a Rocket M5 radio at the ground station. These radios provide shell access to the onboard computers of the robots for launching and monitoring software at a low level and are used to relay data and commands from the OCS. These radios provide approximately 100 Mbps of bandwidth split between the two connections.

To establish a common coordinate frame between the vehicles, both are equipped with Piksi RTK GPSs. The RTK GPS measures position relative to a fixed base station with more precision than un-augmented GPSs, but it requires a real-time link from the base station to each rover unit for correction signals. This system uses a dedicated telemetry radio at the base station, and one on each vehicle, solely for these corrections.

The UAV has two additional radio links. A telemetry link to a separate computer running Mission Planner [20], enables the pilot in command to monitor the flight controller's state independently of the higher-level ROS-based software system. The other link is a connection to a Radio Control (RC) for the pilot to control the UAV manually during take offs, landings, and to override the autonomous system. Figure 3 displays these radio links and high-level components.

Figure 1. The Jackal unmanned ground vehicle shown with the first-generation Swifnav Piksi real-time kinematic global positioning system (RTK GPS) unit.

Figure 2. A custom unmanned aerial vehicle shown with the first-generation Swifnav Piksi RTK GPS unit.

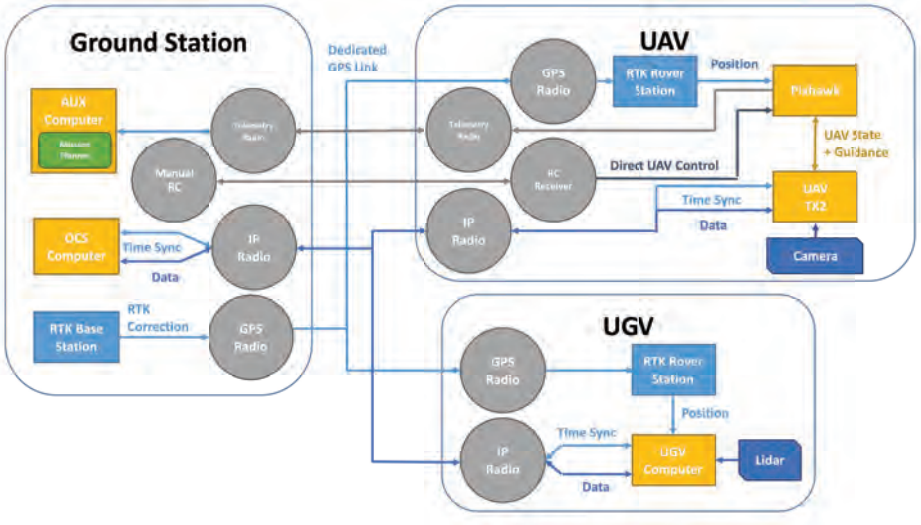


Figure 3. Diagram displaying the wireless links and sensor layout. OCS: operator control station; UAV: unmanned aerial vehicle; UGV: unmanned ground vehicle; IP: Internet protocol.

5.2. System Architecture

The multi-robot system utilizes a ROS to integrate a variety of open source packages. The ROS was chosen because it is a form of well supported robotics middleware which enables the construction of complex systems from individual components, many of which are already widely available. The ROS provides a publisher subscriber architecture for passing data between processes, called nodes, within a single computer and between computers. Service and Action interfaces enable synchronous execution of actions by other processes and provide critical feedback on the state of the execution. Launch files can be written to automate the start up of complex systems.

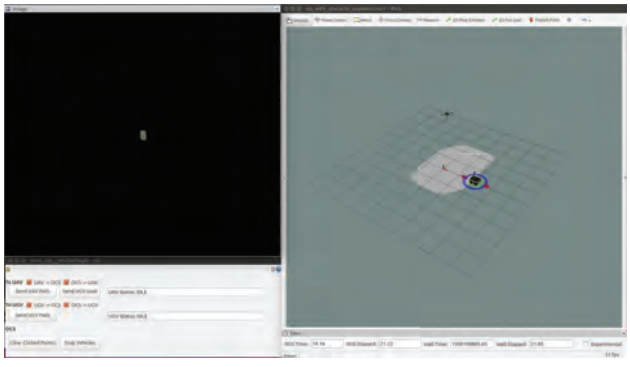


Figure 4. A sample image of the operator control station (OCS) presented to the user. The right hand window is an RViz window displaying the robots, cost maps, and paths. The image window in the upper left-hand corner visualizes the assembled image of the ground plane. The control widget in the lower left-hand corner displays the current status of the vehicles and their comms bridges. This interface allows goals to be specified and planned paths to be visualized before execution.

5.3. Mission and Algorithm

There are four computers in the multi-robot system: two at the base station and one onboard each vehicle. The computer running Mission Planner interfaces directly with the flight controller and is independent of the rest of the software.

The OCS provides a visualization that displays the state of both of the robots as well as an interface for the operator to control the two vehicles. The interface, shown in Figure 4, consists of an RViz window which displays 3D models of the two vehicles depicting relative position of the two vehicles in the common coordinate frame with both the obstacle map constructed by the ground vehicle and synchronized with the UAV, and the terrain-based cost map constructed by the UAV. The state of the vehicles, planning, and moving, as well as error conditions, are also displayed to the operator.

The OCS provides an interface offering several different levels of control of the robots. At the lowest level, the interface enables direct teleoperation the ground vehicle using either a XBox controller or interactive markers. At an intermediate level, a point and click interface for each vehicle allows goals to be specified for each vehicle and invokes the onboard of each planner to autonomously navigate the vehicles to the specified locations. At the highest level, the mission interface enables the operator to execute the full mission of navigation by the UAV, planning for the ground vehicle, then navigation by the ground vehicle. Figure 5 displays an outline of the processes of planning the ground vehicle's path, first onboard the UAV, then on transmitting it to the UGV for execution.

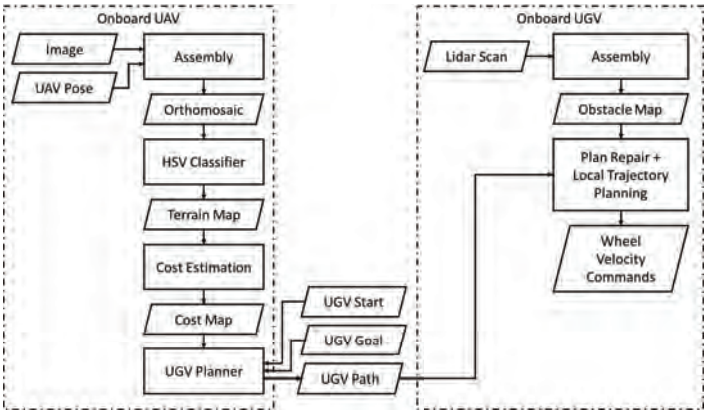


Figure 5. Flow chart displaying the mapping, path planning, and path execution pipelines running onboard the UAVs and UGVs. HSV: Hue Saturation Value.

As a stand-in for more sophisticated exploration algorithms, the operator specifies a sequence of waypoints to control the area covered by the UAV with imagery. Each point specifies the (x, y) coordinate of the waypoints while a slider specifies the altitude of the UAV in the common coordinate frame. The user also specifies a corridor width parameter. The sequence of waypoints, starting from the initial position of the vehicle, defines a sequence of line segments, and these are expanded into rectangular regions by half the width on each side. Camera calibration was performed using the AprilCal [21] calibration toolbox which provides an estimate of the field of view of the camera. The UAV calculates the area covered by a single image on the ground plane using the camera intrinsics and specified altitude, and then assuming continuous capture, computes the amplitude and frequency of a constant altitude sine wave trajectory through the air to image each corridor in the sequence.

The UAV utilizes *MoveIt!* [22] for its trajectory planning and execution. By imagining a four-degree of freedom (4DOF) arm consisting of three perpendicular prismatic joints, ending with a yaw joint, and considering the UAV itself as the end effector, *MoveIt!* can be used to plan and execute the motions of the UAV. The x prismatic joint points east, the y prismatic joint points north, and the z prismatic joint points up. These joints are rooted at the arming position of the UAV. The frame MAVROS accepts commands in, allowing the commanded joint values to be converted directly into position and yaw set-points which the flight controller attempts to achieve. *MoveIt!* is configured to apply velocity and acceleration limits to each of these joints which are conservative estimates of the true capabilities of the vehicle, so the generated time stamped sequence of position and orientation commands for the vehicle is feasible. The flight controller's top priority always remains the stability of the aircraft. In the event that a set-point position is too far away from the vehicle to be achieved in planned amount of time, the vehicle would simply move to the goal at best possible speed and the movement would halted if the vehicle did not reached the goal within the time limit. This allows the system to leverage the planning, control, trajectory execution, and monitoring capabilities that *MoveIt!* already implements. This also provides the capability for performing obstacle avoidance with the UAV if a source of 3D data becomes available in the future. However, this does introduce the additional complexity of converting the state published by MAVROS into joint states of this imagined 4DOF arm, as well as converting the joint state commands output by *MoveIt!* back into set-point position commands for MAVROS to send to the flight controller.

The UAV continuously acquires imagery and projects these images onto the assumed ground plane to construct an orthomosaic. At start-up, user set parameters define the extent and spatial resolution of the orthomosaic to be constructed, defining an image whose pixel coordinates can be

mapped directly to (x, y, z) positions in the common coordinate frame. As each image is acquired, the estimate of the pose of the camera relative to this ground plane image is used to compute a homography between the two. Using this homography, color values for the pixels of the ground plane image can be drawn from the captured images in real time. This process occurs at just over 1 Hz. The current implementation could be sped up by substituting the central processing unit (CPU)-based *warpPerspective* function in OpenCV [23] with the graphical processing unit (GPU) version to leverage the onboard GPU.

Because the homography is computed directly from the UAV's state estimate, the accuracy of the assembled orthomosaic is determined by the accuracy of its state estimate as well as the accuracy of the time stamps assigned to the captured images. An RTK GPS system provides position observations which are fused into an EKF to reduce the error in the estimated position of the UAV. The UAV's orientation estimate is computed from integrated rate gyros and is supplemented by an external compass.

The current terrain classifier is a simplified, pixel by pixel, HSV classifier that classifies the assembled image of the ground plane into four classes: asphalt, grass, obstacle, and unknown. The unknown class explicitly refers to areas of the ground plane that have not yet been observed. Asphalt is recognized as as regions with saturation below a user-defined value which handles both the black surface as well as painted white markings on the runway in the test area. An explicit colored obstacle class is detected as a single solid color with a hue within a defined range, and saturation and value above the minimum values. The remaining area is assigned to the grass class. This classifier used hand-tuned thresholds and was not particularly robust to variability in the conditions, it only serves as a placeholder for more sophisticated terrain classification algorithms, such as the algorithm described in [1], which will be integrated at a later date.

Once classes are assigned to each pixel, this class image is then converted to a cost map. Obstacles are denoted with infinite cost, and unknown areas are denoted with a special value. Asphalt and grass are given cost values matching their ratio in [1] at the Jackal's nominal speed. The cost values are 16 for asphalt and 27 for grass, corresponding to their traversal costs with the Turtle of 0.332 kW for asphalt and 0.543 kW for grass. Once this cost map has been generated, a dilation step must take place to account for the radius of the ground vehicle during planning.

After the UAV has successfully planned and executed each trajectory in the sequence of waypoints, it calls a D* [24] planner running onboard the UAV. This planner retrieves the current ground cost map and plans a path from the UGV's current position to the UGV goal. As a default, this UGV goal defaults to the last point in the sequence of waypoints specified by the user, but the user may also explicitly specify a goal. For efficiency, the classification and cost map generation step described above only happens at the demand of the planner, rather than continuously. The planned path is then transmitted back to the OCS.

By running image assembly, terrain classification, cost map generation, and global path planning all onboard the UAV, the required bandwidth between the UAV, ground station, and UGV is dramatically reduced. Command waypoints, the planned path, and the current UGV state are the only things that must be transmitted to the aerial vehicle, and this requires very little bandwidth compared to sending imagery and cost maps. However, it is important to note that because obstacle maps are not directly synchronized between the aerial vehicle and ground vehicle, there is opportunity for the two planners to generate different plans when they do not detect the same set of obstacles in an environment. This entire half of the mission is managed by a state machine onboard the UAV which uses action and service interfaces for planning, trajectory execution, and data retrieval which improve reliability and correctly handle potential failures in each of the subtasks. The current state of the execution is relayed back to the operator for debugging purposes.

This planned path for the ground vehicle is visualized on the OCS and if the operator is satisfied with the trajectory, he may send it to the UGV. The UGV has a much simpler state machine which reacts to this incoming path and feeds it into Move Base [15]. Move Base is a 2D navigation stack for

ground robots that integrates obstacle mapping, global path planning, local path planning for obstacle avoidance, control, and trajectory execution. A custom global path planning plugin receives the path planned by the UAV, repairs the plan to ensure it is still feasible, and passes it to the rest of the Move Base stack.

The global path plan was generated by planning on terrain based cost map transformed derived from data transformed into the common coordinate frame. That data was generated by sensors attached to a separate robot from the one executing it. This introduces another opportunity for pose estimation error to enter the system. The RTK system equipped to the Jackal keeps its position error down to a few centimeters, but ultimately during execution, the position and orientation error of the UAV, as well as the position error of the UGV within the common coordinate frame will appear as an error between where the UAV thought obstacles were relative to the UGV and where they actually were relative to the UGV. If these errors are large enough, the planned path may pass through an obstacle. There is also the possibility that an obstacle is undetected from the air. Move Base performs obstacle detection using a LIDAR onboard the Jackal and and uses local trajectory planning in an odometry frame to avoid collisions despite these issues. Path following terminates when the vehicle arrives at the destination or if the vehicle is unable to reach the goal.

5.4. Implementation Details

In a system this complex, there are quite a few implementation details that often go unmentioned in academic papers. However, to be useful to other members of the community who might wish to construct a similar system, it is important to mention some of the low level details and design choices made in its construction. This section will touch on a few issues common to all multi-robot systems, time synchronization between computers and communication in multi-robot systems, and issues more specific to ROS-based systems.

5.4.1. Time Sync

Chrony [25] is a an implementation of the Network Time Protocol (NTP) used to synchronize the system clocks of the onboard computers with the system clock of the OCS. ROS utilizes time stamps derived from the current system time on each computer whenever a node looks up what the current time is. For example, the TF [26] package in the ROS maintains a transform tree which utilizes the Universal Robot Description Files (URDF) in conjunction with sensor data to track the transformations between different frames defined for the system as they change with time. Any node may add to this transform tree by publishing transforms between frames at particular times. This system inherently assumes that there is a single consistent source of time for the ROS system, so when nodes are running on multiple computers, or in this case on multiple separate robots, it is necessary to synchronize the system clocks between the computers to avoid violating this assumption.

5.4.2. Comms Bridge

The ROS is a flexible middleware employed in a wide range of robots, but it has some shortcomings in multi-robot systems containing unreliable wireless links. This system implements a custom ROS to ROS bridge which manages bandwidth between the robots, reasons over the state of the wireless link, and improves total bandwidth. The main function of this comms bridge is to split the multi-robot system into several separate ROS masters. Each of the two vehicles and the OCS have their own separate ROS master with their own *roscore* running locally. The comms bridge is implemented as a pair of nodes on each side of the two wireless links. These nodes subscribe to ROS topics, process them, and emit messages which are received by the other node in the pair using ZeroMQ, [27], another message-passing library.

Large messages, such as paths, are broken up into smaller chunks and then re-serialized using a flexible combination of ROS serialization and Message Pack [28]. Conversely, small serialized messages are packed together to hit a 1500 byte payload target. This payload target maximizes throughput

by minimizing the impact of dropped packets on bandwidth. The comms bridge utilizes the User Datagram Protocol (UDP), rather than Transmission Control Protocol (TCP) to transmit between robots which further minimizes overhead. The comms bridge itself handles retransmission at a higher level in ways that are specific to each type of data being sent. For example, when transmitting cost maps across the comms bridge, the two halves of the bridge use a synchronization scheme where the map is broken up into individual tiles. The downstream side has a copy of the map and requests updates from the upstream side. If a tile is dropped in transmission, then the downstream side remains out of date and will request the tile again. Because the upstream side always sends the most recent copy of a tile, it avoids transmitting the unnecessary intermediate states of the tile.

Importantly, from a safety perspective, each side of the comms bridge tracks when the last message was received from the other side. If messages are not being received, then the connection has been lost and both robots will automatically cancel their actions and stop their motions.

5.4.3. UAV State Transitions

On hardware, for safety reasons, the autonomous system does not broadcast commands to transition the flight controller's state into the software-controlled *GUIDED* state. The pilot in command is prompted to change flight modes manually while the system waits to confirm the transition before continuing with trajectory execution. The system switches back to *LOITER* on the completion of a trajectory to ensure that the aircraft is in a safe state. While this approach is more tedious for the pilot, it does give them final authority over the movement of the vehicle and avoids the possibility of any unexpected state transitions being caused by the autonomous system while it is under manual control.

5.4.4. TF Tree and Namespaces

To ease the creation and debugging this software system, it was laid out in a way that keeps the launch files virtually identical between simulation and hardware with both single and separate ROS masters.

Within a single ROS master, all nodes must have unique names. The ROS encourages the use of namespaces to organize nodes. Placing a node into a namespace causes relative topics published and subscribed to by that node to be prepended with the namespace. By launching all nodes within a robot specific namespace regardless of whether they are being run in simulation or hardware, all of the node names and topic names will remain consistent. As long as each robot is given a unique name, multiple copies of the same robot can be run from the same launch file.

Link names within the TF tree must also be uniquely named. The transform tree is a somewhat unique in the ROS in that all nodes that interact with the TF tree publish and subscribe to the same *tf* topic. This means that if a single ROS master is used across multiple identical robots, there is the issue of ensuring that all of the link and joint names remain unique. One possibility would be to maintain multiple separate URDFs for each robot, but this would be redundant. Instead, this system uses a xacro [29] version of the URDF which allows a robot name argument to be prepended to each link and joint name in the URDF. Figure 6 shows an excerpt of the TF tree demonstrating this idea. However, this requires all parts of the system to conform to the convention, which requires modification to the interface software used for the Jackal as well as the implementation of an adapter to modify the names of links referred to in topics being published by an ROS serial interface from a micro-controller built into the Jackal.

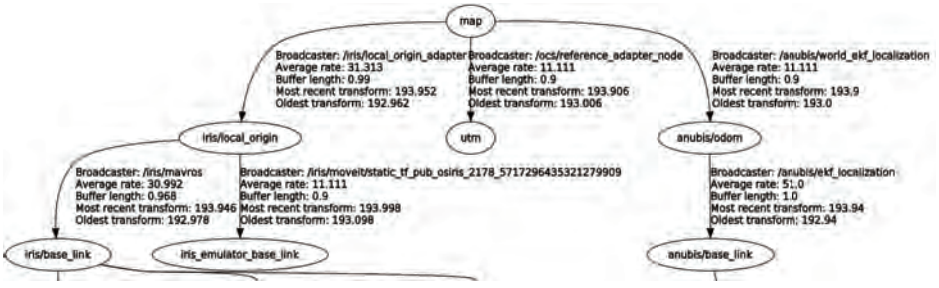


Figure 6. Excerpt of TF tree of the system in simulation. Both vehicles have a $base_link$ frame. Since they are prepended by the vehicle names "iris" and "anubis", the frames are unique.

6. Experiments

Preliminary experiments to investigate the effects of field of view on the UAV's camera on the quality of planned paths for the UGV were initially conducted in Matrix Laboratory (MATLAB). In these trials, the area imaged by the UAV while moving to the goal was simulated by extracting subsets of a pre-collected orthomosaic and the size of this area was varied. Then, a path planner was run on the classified terrains to examine the change in path cost as additional area was made available for planning.

These were followed by experiments in both a simulated Gazebo environment and at Kentland Farm, a Virginia Tech-operated farm that allows the testing of UAVs.

In simulation, a flat ground plane was created and colored obstacles were added. The ground plane was textured with an orthomosaic captured from Kentland Farm and assembled using Agisoft PhotoScan [30] to enable the quality of the image assembly method to be qualitatively evaluated. This simulation environment enabled the majority of the software created for the system to be tested including the planning, navigation, and obstacle detection for both vehicles.

When running on a single computer with a single ROS master, the comms bridge is also being tested. However since messages are passed over the loop back interface, it has much lower latency and higher bandwidth than on hardware. Simulation testing of multiple separate ROS masters on separate computers was conducted to validate the comms bridge before deploying to hardware with the wireless link in the loop.

During initial testing and debugging on hardware, the UAV was not flown, but its software was kept in the loop by substituting the terrain-based cost map with the obstacle map generated by Move Base. For full system testing, a flat area in the vicinity of a small test runway at Kentland Farm was used for several experiments, as shown in Figure 7. End to end system tests were conducted, where the UAV was launched and an operator validated the behavior of the complete system. The imagery assembled by the aerial vehicle was qualitatively evaluated, revealing some issues with the current algorithm.

A color-coded blue trash can was placed on the runway as an obstacle. The position of this obstacle was surveyed with an additional RTK unit to compare its true location with the locations estimated by both the UAV and UGV. After the full system test where it was confirmed that the UAV identified the obstacle from the air and planned a path around it, and it was confirmed that the UGV successfully detected the obstacle and avoided it while executing the path planned by the UAV, an additional test was conducted to for explicit examination of the position errors in the system.

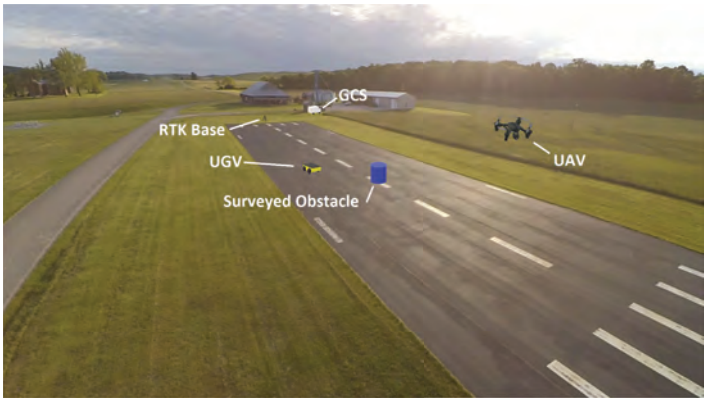


Figure 7. Kentland Farm experimental setup. GCS: Ground Control Station.

7. Results

Figure 8 shows a graphical representation of the effects of varying the effective field of view of the aerial vehicle on the planned paths for the ground vehicle, while Table 1 shows the quantitative effects. As expected, a larger field of view allows the path planner for the ground robot to generate lower cost paths by reducing the distance driven over the more expensive grass class, even if it comes at the cost of a longer overall plan. When the Field of View (FOV) increases from 24 m to 48 m, we see a marked change in the planned path which detours over to the runway. In this same way, we also expect improvements in distance traveled when faced with obstacles rather than just more costly terrain. In more obstacle-dense environments, the aerial vehicle's perspective has the additional benefit of not being blocked. This should allow dead-ends to be more quickly identified from the air and result in less doubling back than would be required of the ground vehicle alone.

Figure 9 shows a typical execution of the mission in simulation. The user clicks on a point on the right hand side of the environment to specify a goal for the system. The UAV generates a path, shown in light blue, ascending to the desired altitude and flies to the destination. The expanding rectangular area of light grey in the right-hand RViz window is the expanding area of observed and classified terrain in the terrain map. The black circle that appears in the terrain map is the detected blue obstacle. Once it has arrived at the end of the path, a trajectory for the ground vehicle, shown in green, is planned. The UGV then executes the planned path avoiding the obstacle by following the plan and using its onboard LIDAR. The light grey circular region in the right hand RViz window is the obstacle map generated by the onboard LIDAR. This onboard obstacle detection enables the ground vehicle to avoid the obstacle despite the parallax error between the aerial constructed terrain map and the ground-level obstacle map.

Table 1. The UGV planned path energy cost vs. aerial imagery field of view. Note that while total path length increases, energy cost decreases because of the reduced distance driven on grass. The paths are shown in Figure 8.

Trial	Field of View (m)	UGV Path Length (m)	UGV Traversal Cost (kJ)	Terrain Type	
				Asphalt (m)	Grass (m)
1	0.6	229.83	58.51	36.85	192.99
2	6	234.51	57.00	63.22	171.29
3	12	235.01	54.59	87.33	147.68
4	24	235.21	52.54	107.25	127.96
5	48	255.04	51.21	170.96	84.09

Figure 8. Paths corresponding to cases presented in Table 1. As the field of view of the simulated camera is expanded, the path cost is reduced despite an increase in distance traveled.

Figure 10 more clearly illustrates the parallax error inherent in the 2D assumption of the ground plane reconstruction. The parallax error is an error in the estimated position of an obstacle from the air if the height of the UAV is small compared to the height of the obstacle.

Hardware testing confirmed that the above mission functioned on hardware with a few caveats. With respect to the hardware, take off and landings are performed manually by the pilot in command. In simulation, the system autonomously makes state transitions on the flight controller automatically. On hardware, the pilot in command is responsible for making this transition at the request of the autonomous system. This procedure works as expected. However, if this transition is not made in a timely fashion, the trajectory execution of *MoveIt!* will timeout since it expects the trajectory to be completed with a certain period of time.

There is an issue of operator immersion in the OCS which makes it difficult to specify goals in unexplored areas. This lack of presence could be addressed by incorporating coarse pre-collected imagery of the area into the visualization similar to other ground station applications, such as Mission Planner.

Overall, the system did function as intended on hardware. The operator was able to specify goals for both vehicles and move them to the specified locations. The online image assembler accepted imagery at 1.37 Hz, which was fast enough to avoid gaps in the map when flying at 10 m above the ground surface and traveling horizontally at around 4 m per second.

The ground vehicle successfully avoided the obstacle in each test, an example of which is shown in Figure 11, however, the error between the aerial constructed terrain map and the ground-level obstacle map was great enough that obstacle avoidance was due to the UGV's onboard replanning, and in the case noted above, the path simply missed the obstacle entirely.

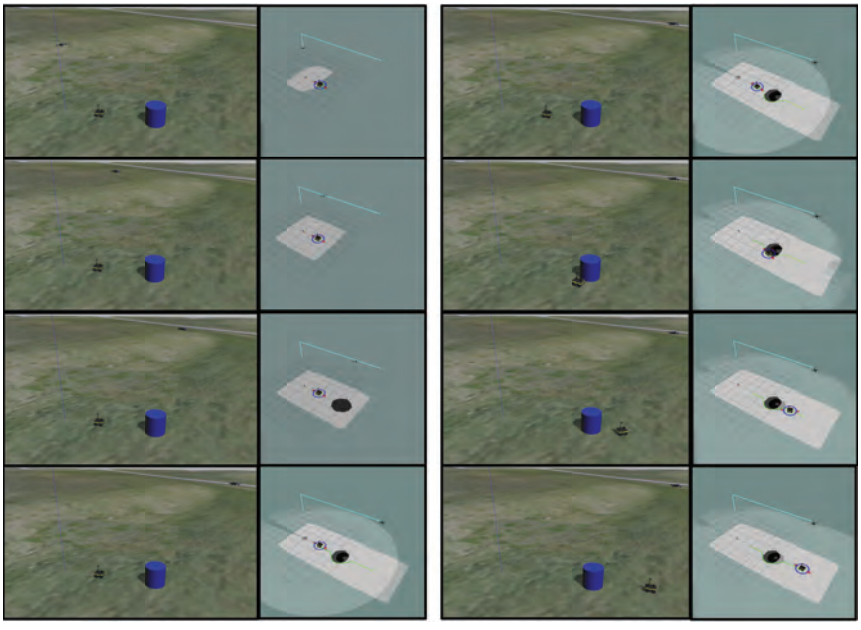


Figure 9. This sequence of images, descending down the left column and then the right column, shows a brief test in simulation. The left window in each pair shows the Gazebo environment, and the right window shows the RViz window. From fourth image on, the ground-based obstacle map is also visualized.

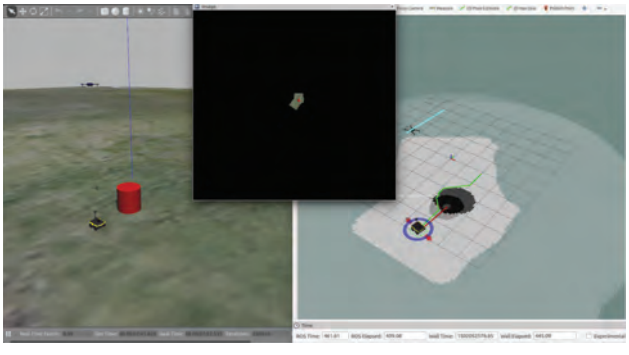


Figure 10. This simulation illustrates the issue of parallax error in the estimated obstacle location. The left half of the figure displays the Gazebo simulation environment, featuring a ground vehicle, an aerial vehicle, and the obstacle. The right half of the figure displays two obstacle maps overlaid on one another. The ring shaped region shows the perimeter of the object as seen by the ground vehicle, while the solid, larger circle displays the obstacle as detected by the aerial vehicle. Note the offset between them as well as the difference in estimated size which is due to the position of the aircraft directly above the ground vehicle and offset from the center of the obstacle.

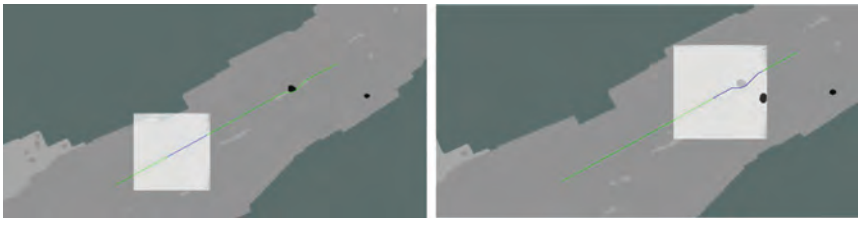


Figure 11. This figure displays the terrain-based cost map, planned ground vehicle path, and local ground-level obstacle map as the vehicle executed the path. The green line shows the planned path of the ground vehicle, while the blue shows the subset of this planned currently being followed. Note the false positive obstacle detected to the right of the planned path. The white square shows the overlaid ground-level obstacle map. Note in the right-hand image, the black circle at the edge of the local cost map does not align with the obstacle as seen from the air.

Two tests to characterize the absolute and relative errors in the estimated locations of the obstacle from the aerial vehicle and from the ground vehicle were conducted. In a more realistic scenario, the UAV was manually flown over the obstacle to gather imagery, and a plan was constructed, sent to the UGV, and then executed. The location of the obstacle in the aerial terrain map and the ground-level obstacle maps were extracted as the centroid of the obstacle cells within the two maps and compared to the ground truth position of the obstacle obtained by surveying its position with an RTK GPS. Table 2 shows the results of these trials. The test was broken up into two flights due to limitations in battery life, but the obstacle was not moved between the first six trials and the second six. Since planning for the UAV is done after the UAV has completely flown past the obstacle, the final estimated position of the obstacle was used. It was observed that while the UGV was driving past the obstacle, the estimated location of the obstacle in the map frame shifted quite substantially. There are two factors at work. The first is that the UGV only observes one side of the obstacle with its LIDAR. Since it does not observe the entire perimeter of the obstacle at once, the centroid of the obstacle pixels in the obstacle map will be the center of this observed arc, rather than the center of the circle that it represents. The second and more dominant source of error is due to error in the estimated orientation of the ground vehicle. This causes the position of the obstacle to shift substantially as the ground vehicle approached the obstacle. To try and quantify this error, both the average and the maximum errors are listed. Because obstacles are only inserted up to 5 m away, the errors introduced by this error in orientation are bounded.

Table 2. Results from passes made by both vehicles at the obstacle. The absolute errors refer to the distance between each vehicles' estimate of the position of the obstacle and the true real-time kinematic global positioning system position. The relative error is a comparison between the estimated positions from each vehicle.

Test	Absolute UAV Error (m)	UGV Average Absolute Error (m)	UGV Max Absolute Error (m)	Average Relative Error (m)	Max Relative Error (m)
1	2.148	0.822	1.898	1.857	2.947
2	0.371	0.242	0.447	0.375	0.628
3	1.819	0.810	1.533	1.818	2.740
4	1.017	0.748	1.597	1.064	2.414
5	1.518	0.395	0.962	1.519	1.829
6	1.583	0.570	1.208	1.714	1.984
7	1.009	0.789	1.721	1.054	2.256
8	1.011	0.465	0.690	1.291	1.475
9	0.776	0.781	1.511	1.296	2.251
10	0.923	0.374	0.729	1.111	1.328
11	0.648	0.869	1.635	1.241	1.618
12	1.169	0.685	1.540	1.411	1.732

One other issue that appeared when taking repeated passes at the obstacle with the UAV was the duplication of the obstacle in the terrain map before the UAV had completed its flyover. If the obstacle is on the edge of the captured image, and there is enough error in the state estimator that the projection of the image does not overlap with the previously projected image of obstacle in the map, it is possible to create duplicate copies of the obstacle in the assembled image.

The second test to quantify the errors between the estimated obstacle locations was to hover the UAV over the obstacle and manually drive the ground vehicle in circles around the obstacle. The ground vehicle was driven around the obstacle to average out the effect of only viewing a portion of the obstacle at a time. Two separate tests were conducted. The first test, shown in Figure 12, was conducted with the first-generation Piksi RTK GPS. We notice the oscillation in estimated position of the obstacle from the UGV related to the path driven by the vehicle. This oscillation was caused by an offset in the orientation of the vehicle and its true orientation due to an issue with the application of magnetic declination correction. This issue was corrected in the second test shown in Figure 13. In the second test, the vehicles were also switched over to the Piksi Multi RTK GPS. This second test shows significantly improved errors except for an unexplained consistent offset in the X position of the location of the obstacle estimated by the UAV. Table 3 more clearly illustrates the improvement in error when the compass error was corrected on the UGV and the general improvement to the system when a more robust RTK GPS was used.

Table 3. Average and standard deviations of errors in hover tests shown in Figures 12 and 13. The results for the second test exclude the data from the UAV during the period of RTK failure.

Hover Test	Average UAV Absolute Error (m)	Standard Deviation UAV Absolute Error (m)	Average UGV Absolute Error (m)	Standard Deviation UGV Absolute Error (m)
1	2.501	0.907	0.330	0.164
2	0.814	0.432	0.236	0.122

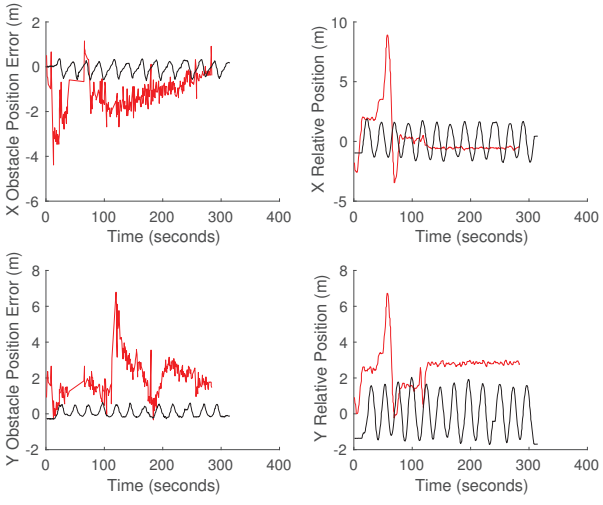


Figure 12. Continuous obstacle position test conducted with first-generation Piksi. Red shows the data associated with the UAV, while black is the data for the UGV.

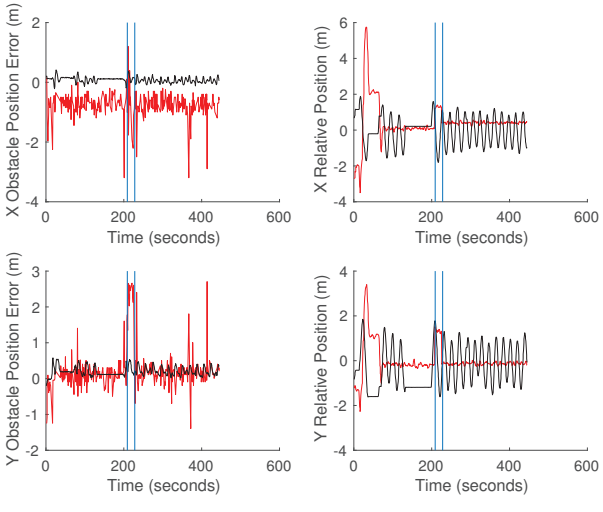


Figure 13. Continuous obstacle position test conducted with the Piksi Multi. Red shows the data associated with the UAV, while black is the data for the UGV. The vertical blue lines indicate the period of time where the RTK GPS on the UAV failed.

One of the biggest failure modes encountered during hardware testing was the RTK GPS onboard one of the vehicles losing its RTK fix. In early testing, this was usually an issue with the UAV which has several other radios and more unshielded electronics, which interfered with the correction radio. The Swifnav RTK units do not send out RTK fix positions unless it has recently received an RTK correction message from base station. The Pixhawk expects GPS updates at a regular rate, and in the event

that the RTK GPS misses an update deadline, it will switch over to a backup GPS receiver onboard. However, because this other GPS receiver position is not augmented, it has slowly time-varying errors in position that appear as an offset between the position reported by the augmented and un-augmented fixes. When the UAV switches GPS units, its position estimate jumps, most obviously appearing as a discontinuity in the assembled imagery as shown in Figure 14. Figure 15 shows a typical result, even with a stable RTK fix, the orientation estimate errors are still visible as small discontinuities.

Despite these large errors between both the estimated obstacle positions and the true position, as the ground vehicle never collided with the obstacle, its onboard obstacle avoidance was sufficient. However, this level of error is great enough that paths taken by the ground vehicle might not remain in the correct class. For example, executing a path planned down the side of a road with 1 m of error might mean that the ground vehicle drives on the grass to the side of the road instead of on the road itself. This issue could be mitigated by dilating the cost map generated by the aerial vehicle by an amount related to this error, but it would be preferable to reduce the positioning error.

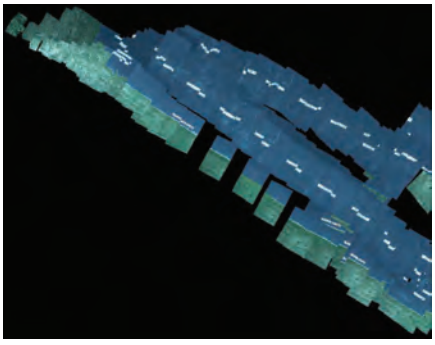


Figure 14. Discontinuity in assembled aerial imagery due to switching from single point position (SPP) fix to RTK fix. *Note that the runway only has a single center white dashed line.*

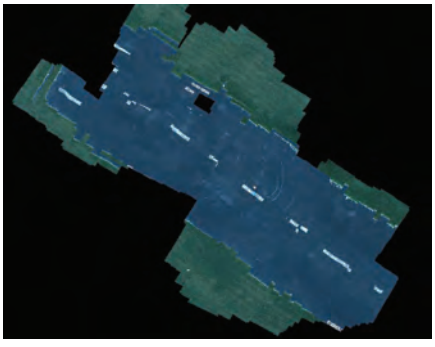


Figure 15. Successful assembly of image of runway showing a more normal result. *Note this figure is not at the same scale as Figure 14.*

8. Future Work

One of the main goals of future work will be to remove the planar environment assumption from the aerial vehicle's mapping. Using a stereo camera pair on the UAV, rather than a monocular camera, will allow the system to construct height maps of the environment using the depth image. This will eliminate parallax errors by explicitly modeling terrain height and allow obstacles to be

detected as discontinuities in height rather than an arbitrary color. This transition will also require the use of feature point tracking to aid in the reconstruction by providing better estimates of the change in position and orientation from one pair of images to the next. Depending on the quality of the reconstruction that can still be achieved in real time, it may also be necessary to implement some smoothing technique to avoid introducing height discontinuities at the edges of the observed depth image due to slight misalignments.

In future work we would also like to more tightly close the loop between the planning and motion of the UAV and UGV.

The current planner for the UAV flies between a sequence of waypoints specified by the user, as a simplified method to collect imagery of the area we expect the ground vehicle to drive through. However, in an environment with a wall or other extended obstacle, this does not guarantee that the aerial imagery will contain a feasible path for the UGV. Instead, the planner and trajectory execution of the UAV should wrap around the ground vehicle planner so that it continues to search the area until a path for the ground vehicle is discovered.

The system should also somehow synchronize obstacle and terrain cost data between the two vehicles. When the UAV plans a path for the ground vehicle, it does not have knowledge of the obstacles that the ground vehicle has discovered. If the ground vehicle observes a large wall that does not appear in the aerial obstacle map, then the ground vehicle will identify that the aerial vehicle's planned path is infeasible because it might pass through an obstacle. The UGV can replan a completely new path that avoids this obstacle to travel to the goal, but because it is planning without the terrain-based cost map on the UAV, its path will no longer minimize energy consumption. Synchronizing the data between the two robots will lead to more consistent planning behavior in challenging environments.

We also wish to use the path execution by the ground vehicle as feedback for the terrain classification. Under a reinforcement learning scheme, the classes of terrain could be corrected as the ground vehicle drives on them. For example, a dark gravel that was mistakenly classified as asphalt might be correctly identified by the higher energy cost to drive over it.

In another scheme, this feedback could be used to update energy costs associated with different terrains. In the current system, the ground vehicle planner utilizes terrain costs estimated offline from previously collected data. However, a wide variety of circumstances, such as recent weather turning dirt to mud, could alter terrain costs substantially from their values in the training set. This could be partly addressed by expanding the terrain classifier to cover more unique classes, distinguishing between dirt, mud, swap, etc. However, another way to address the shortcoming of the original training data is to revise terrain costs as the vehicle traverses it. Without having to split dirt into a separate mud and dry dirt classes, observing the energy cost of driving over mud, so long as the vehicle can do so safely, would let the system adjust the dirt class's cost upwards. Replanning with the new cost will produce a good result without introducing more granularity into the classifier.

Acknowledgments: This work was sponsored by the National Science Foundation through the Industry-University Cooperative Research Centers Program and by the Defense Threat Reduction Agency. The Jackal ground robot was graciously loaned by the Army Research Laboratory for these experiments.

Author Contributions: John Peterson, John Bird, Karim Abdelatty, and Kevin Kochersberger conceived and designed the experiments; John Peterson and Karim Abdelatty implemented and integrated the software; Haseeb Chaudhry designed and constructed the UAV; John Peterson, Haseeb Chaudhry, Karim Abdelatty, and John Bird performed the experiments; John Peterson analyzed the data; Kevin Kochersberger contributed equipment for conducting experiments; John Peterson wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

LIDAR	Light detection and ranging
UAV	Unmanned aerial vehicle
UGV	Unmanned ground vehicle
GPS	Global positioning system
RTK	Real-time kinematic
IMU	Inertial measurement unit
INS	Inertial measurement system
OCS	Operator control station
ICP	Iterative closest point
ROS	Robot operating system
RC	Radio Control
DOF	Degree of freedom
CPU	Central processing unit
GPU	Graphical processing unit
IP	Internet protocol
EKF	Extended Kalman filter
HSV	Hue Saturation Value
SIFT	Scale-Invariant Feature Transform
NTP	Network Time Protocol
UDP	User Datagram Protocol
GCS	Ground Control Station
MATLAB	Matrix Laboratory
FOV	Field of View
TCP	Transmission Control Protocol

References

1. Christie, G.; Shoemaker, A.; Kochersberger, K.; Tokekar, P.; McLean, L.; Leonessa, A. Radiation search operations using scene understanding with autonomous UAV and UGV. *J. Field Robot.* **2017**, *34*, 1450–1468.
2. Templeton, T.; Shim, D.H.; Geyer, C.; Sastry, S.S. Autonomous Vision-based Landing and Terrain Mapping Using an MPC-controlled Unmanned Rotorcraft. In Proceedings the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 1349–1356.
3. Papadakis, P. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Eng. Appl. Artif. Intell.* **2013**, *26*, 1373 – 1385.
4. Owens, S.; Sycara, K.; Scerri, P. Using Immersive 3D Terrain Models For Fusion Of UAV Surveillance Imagery. In Proceedings of the AIAA Infotech@Aerospace Conference, Seattle, WA, USA, 6–9 April 2009.
5. Stentz, T.; Kelly, A.; Rander, P.; Amidi, O.; Mandelbaum, R. *Integrated Air/Ground Vehicle System for Semi-Autonomous Off-Road Navigation*; Robotics Institute: Pittsburgh, PA, USA, 2002.
6. Stentz, A.T.; Kelly, A.; Rander, P.; Herman, H.; Amidi, O.; Mandelbaum, R.; Salgian, G.; Pedersen, J. Real-Time, Multi-Perspective Perception for Unmanned Ground Vehicles. In Proceedings of the AUVSI Unmanned Systems Symposium 2003, Baltimore, MD, USA, 15–17 July 2003.
7. Vandapel, N.; Donamukkala, R.R.; Hebert, M. Quality assessment of traversability maps from aerial LIDAR data for an unmanned ground vehicle. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 1, pp. 305–310.
8. Vandapel, N.; Donamukkala, R.R.; Hebert, M. Unmanned Ground Vehicle Navigation Using Aerial Ladar Data. *Int. J. Robot. Res.* **2006**, *25*, 31–51, doi:10.1177/0278364906061161.
9. Madhavan, R.; Hong, T.; Messina, E. Temporal Range Registration for Unmanned Ground and Aerial Vehicles. *J. Intell. Robot. Syst.* **2005**, *44*, 47–69.

10. Zawodny MacArthur, E.; MacArthur, D.; Crane, C. *Use of Cooperative Unmanned Air and Ground Vehicles for Detection and Disposal of Simulated Mines*; International Society for Optics and Photonics: Washington, DC, USA, 2005; p. 5999.
11. Frederick, P.; Kania, R.; Rose, M.D.; Ward, D.; Benz, U.; Baylot, A.; Willis, M.J.; Yamauchi, H. Spaceborne path planning for unmanned ground vehicles (UGVs). In Proceedings of the MILCOM 2005—2005 IEEE Military Communications Conference, Atlantic City, NJ, USA, 17–20 October 2005; Volume 5, pp. 3134–3141.
12. Chaimowicz, L.; Cowley, A.; Gomez-Ibanez, D.; Grocholsky, B.; Hsieh, M.A.; Hsu, H.; Keller, J.F.; Kumar, V.; Swaminathan, R.; Taylor, C.J. Deploying Air-Ground Multi-Robot Teams in Urban Environments. In *Multi-Robot Systems, From Swarms to Intelligent Automata Volume III, Proceedings of the 2005 International Workshop on Multi-Robot Systems*; Parker, L.E., Schneider, F.E., Schultz, A.C., Eds.; Springer: Dordrecht, The Netherlands, 2005; pp. 223–234.
13. Hudjakov, R.; Tamre, M. Ortophoto analysis for UGV long-range autonomous navigation. *Est. J. Eng.* **2011**. *17*, 17–27.
14. Giakoumidis, N.; Bak, J.U.; Gómez, J.V.; Llenga, A.; Mavridis, N. Pilot-Scale Development of a UAV-UGV Hybrid with Air-Based UGV Path Planning. In Proceedings of the 2012 10th International Conference on Frontiers of Information Technology, Islamabad, India, 17–19 December 2012; pp. 204–208.
15. Marder-Eppstein, E. Move Base. Available online: http://wiki.ros.org/move_base (accessed on 15 September 2017).
16. Clear Path Robotics. Available online: https://www.clearpathrobotics.com (accessed on 23 June 2017).
17. ArduPilot. Available online: http://ardupilot.org/ardupilot/ (accessed on 13 September 2017).
18. Nvidia TX2. Available online: http://www.nvidia.com/object/embedded-systems-dev-kits-modules.html (accessed on 13 September 2017).
19. Ermakov, V. Mavros. Available online: http://wiki.ros.org/mavros (accessed on 19 September 2017).
20. Mission Planner. Available online: http://ardupilot.org/planner/ (accessed on 13 September 2017).
21. Richardson, A.; Strom, J.; Olson, E. AprilCal: Assisted and repeatable camera calibration. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013.
22. Sucan, I.A.; Chitta, S. MoveIt! Available online: http://moveit.ros.org (accessed on 17 September 2017).
23. Itseez. Open Source Computer Vision Library. Available online: https://github.com/itseez/opencv (accessed on 15 September 2017).
24. Likhachev, M.; Ferguson, D.; Gordon, G.; Stentz, A.; Thrun, S. Anytime Dynamic A*: An Anytime, Replanning Algorithm. In Proceedings of the Fifteenth International Conference on International Conference on Automated Planning and Scheduling, Monterey, CA, USA, 5–10 June 2005; AAAI Press: Cambridge, MA, USA, 2005; pp. 262–271.
25. Chrony. Available online: https://chrony.tuxfamily.org/index.html (accessed on 13 September 2017).
26. Foote, T. tf: The transform library. In Proceedings of the 2013 IEEE International Conference on Technologies for Practical Robot Applications (TePRA 2013), Woburn, MA, USA, 22–23 April 2013.
27. iMatrix Corporation. ZeroMQ. Available online: http://zeromq.org (accessed on 24 June 2017).
28. Furuhashi, S. MessagePack. Available online: http://msgpack.org/index.html (accessed on 24 June 2017).
29. Glaser, S.; Woodall, W.; Haschke, R. Xacro. Available online: http://wiki.ros.org/xacro (accessed on 19 September 2017).
30. Agisoft PhotoScan. Available online: http://www.agisoft.com (accessed on 19 September 2017).

sensors

MDPI

Article

Artificial Neural Network to Predict Vine Water Status Spatial Variability Using Multispectral Information Obtained from an Unmanned Aerial Vehicle (UAV)

Tomas Poblete [1], Samuel Ortega-Farías [1,2,*], Miguel Angel Moreno [3] and Matthew Bardeen [2,4]

1 Centro de Investigación y Transferencia en Riego y Agroclimatología (CITRA), Universidad de Talca, Casilla 747, Talca 3460000, Chile; totopoblete@gmail.com
2 Research program on Adaptation of Agriculture to Climate Change (A2C2), Universidad de Talca, Casilla 747, Talca 3460000, Chile; mbardeen@utalca.cl
3 Regional Centre of Water Research, University of Castilla-La Mancha, Campus Universitario s/n, 02071 Albacete, Spain; miguelangel.moreno@uclm.es
4 Facultad de Ingeniería, Universidad de Talca, Curicó 3340000, Chile
* Correspondence: sortega@utalca.cl; Tel.: +56-71-220-0426

Received: 26 September 2017; Accepted: 24 October 2017; Published: 30 October 2017

Abstract: Water stress, which affects yield and wine quality, is often evaluated using the midday stem water potential (Ψ_{stem}). However, this measurement is acquired on a per plant basis and does not account for the assessment of vine water status spatial variability. The use of multispectral cameras mounted on unmanned aerial vehicle (UAV) is capable to capture the variability of vine water stress in a whole field scenario. It has been reported that conventional multispectral indices (CMI) that use information between 500–800 nm, do not accurately predict plant water status since they are not sensitive to water content. The objective of this study was to develop artificial neural network (ANN) models derived from multispectral images to predict the Ψ_{stem} spatial variability of a drip-irrigated Carménère vineyard in Talca, Maule Region, Chile. The coefficient of determination (R^2) obtained between ANN outputs and ground-truth measurements of Ψ_{stem} were between 0.56–0.87, with the best performance observed for the model that included the bands 550, 570, 670, 700 and 800 nm. Validation analysis indicated that the ANN model could estimate Ψ_{stem} with a mean absolute error (MAE) of 0.1 MPa, root mean square error (RMSE) of 0.12 MPa, and relative error (RE) of −9.1%. For the validation of the CMI, the MAE, RMSE and RE values were between 0.26–0.27 MPa, 0.32–0.34 MPa and −24.2–25.6%, respectively.

Keywords: multispectral image processing; artificial neural network; UAV; midday stem water potential

1. Introduction

The largest wine producing and growing regions (France, Spain, Australia, South Africa, parts of USA, Chile and Argentina) have experienced water scarcity during the last years [1,2]. Under this scenario, modern irrigation management is required to improve water productivity (wine production per unit of applied water, $kg \cdot m^{-3}$) of viticultural areas. Traditionally, irrigation management has been based on the monitoring of vine evapotranspiration, soil water content and physiological plant responses. Although some of these methods have been widely used, they do not consider adequately the spatial variability of soil, cultivar and climate conditions to schedule irrigation. In this regard, site-specific irrigation management (SSIM) can be used as a tool to improve water productivity [3,4]. In that case, SSIM characterizes the effect of the intra-vineyard spatial variability of soil and canopy vigor on the estimation of irrigation scheduling (irrigation frequency and timing).

1.1. Monitoring of Evapotranspiration, Soil Water Content and Physiological Plant Responses

Usually, irrigation management relies on the estimation of actual evapotranspiration (ETa) [1]:

$$\text{Eta} = \text{Kc} \times \text{ETo} \quad (1)$$

where ETo is the reference evapotranspiration ($\text{mm}\cdot\text{day}^{-1}$) and Kc is the crop coefficient. ETo is calculated using the Penman–Montieth model which requires air temperature, relative humidity, wind speed, and solar radiation as inputs [1,2]. The Kc is the ratio between ETa and ETo, where ETa can be measured using a lysimeter, soil water balance approach, eddy covariance method, Bowen ratio energy balance system, or surface renewal method [5]. Due to the specificity of Kc for local conditions, it is necessary to calibrate this parameter to the specific conditions of vineyards [6]. To solve this problem, Turner [7] has suggested monitoring the soil water content to schedule irrigation.

Some of the disadvantages of using soil water content monitoring in vineyards are: (1) it does not represent water consumption in vines with extensive root systems and when vines are grown in deep soil; and (2) it does not predict water content in soil with high salinity and a high percentage of stone [8]. In this regard, Granier, et al. [9] indicated that measurement of physiological parameters provides better information about the whole plant, controlling systems to climatic conditions and atmospheric water demands compared with soil water content.

Among the most studied technologies for monitoring physiological plant responses to water stress are sap flow measurement [10,11], dendrometry [12], gas exchange [13,14], chlorophyll (Chl) fluorescence [15], stomatal conductance [15], canopy temperature [16–19] and plant water potential [20,21]. The measurement of plant water potential is a good predictor of vine water status and has been the most commonly used technique to characterize water stress for vineyards under regulated deficit irrigation (RDI) conditions [22–24]. The midday stem water potential (Ψ_{stem}) has been proposed as the most reliable technique to schedule the irrigation of orchards and vineyards [25,26]. However, the measurement of Ψ_{stem} must be carried out in the field for each plant, which incurs high costs and does not capture the spatial variability of water status [27–29]. Based on that, remote sensing platforms have been proposed to replace ground-based measurements and to assess the spatial variability of water status in larger areas [28,30].

1.2. Remote Sensing and Multispectral Indices to Assess Spatial Variability

Different types of remote sensing platforms can be used to carry different types of sensors to assess different spectral wavelength ranges [31]. Unlike manned vehicles and satellite, unmanned aerial vehicles (UAV) have several advantages: they are inexpensive, flexible, more independent of climatic variables and can be flown with minimal training [32]. All these characteristics allow us to obtain high resolution information in an automatic and accurate manner. By using these vehicles it is possible to transport sensors, which provide multispectral information that can be integrated into spectral indices to predict several physiological variables [28,33–35], specifically for the evaluation of intra-vineyard spatial variability [35–38]. Some researchers have suggested that several indices using information between 500 and 800 nm can estimate vine water status indirectly, with coefficients of determination (R^2) ranging between 0.01 and 0.68 (Table 1).

Table 1. Conventional spectral indices used to estimate vine water status of different cultivars of *vitis vinifera*.

Index	Formula	R^2	Reference	Cultivars
GI	$\frac{R_{550}}{R_{670}}$	0.54	[28]	*Vitis vinifera* L. cv tempranillo
GNDVI	$\frac{R_{800}-R_{550}}{R_{800}+R_{550}}$	0.58	[28]	*Vitis vinifera* L. cv tempranillo
MCARI	$[(R_{700}-R_{670})-0.2\times(R_{700}-R_{550})]\times\left(\frac{R_{700}}{R_{670}}\right)$	0.01	[28]	*Vitis vinifera* L. cv tempranillo
MCARI1	$1.2\times[2.5\times(R_{800}-R_{670})-1.3\times(R_{800}-R_{550})]$	0.21	[28]	*Vitis vinifera* L. cv tempranillo
MCARI2	$\frac{1.2\times[2.5\times(R_{800}-R_{670})-1.3\times(R_{800}-R_{550})]}{\sqrt{(2\times R_{800}+1)^2-6\times(R_{800}-5\times R_{670})-0.5}}$	<0.01	[28]	*Vitis vinifera* L. cv tempranillo
MSAVI	$\frac{\left(2\times R_{800}+1-\sqrt{(2\times R_{800}+1)^2-8\times(R_{800}-R_{670})}\right)}{2}$	0.11	[28]	*Vitis vinifera* L. cv tempranillo
MSR	$\frac{\left(\frac{R_{800}}{R_{670}}\right)-1}{\sqrt{\left(\frac{R_{800}}{R_{670}}\right)+1}}$	0.66	[28]	*Vitis vinifera* L. cv tempranillo
MTVI3	$1.2\times[1.2\times(R_{800}-R_{550})-2.5\times(R_{670}-R_{550})]$	0.01	[28]	*Vitis vinifera* L. cv tempranillo
NDVI	$\frac{R_{800}-R_{670}}{R_{800}+R_{670}}$	0.68 0.57 0.03	[28] [36] [37]	*Vitis vinifera* L. cv tempranillo *Vitis vinifera* L. cv chardonnay *Vitis vinifera* L. cv cabernet sauvignon
TCARI/OSAVI	$\frac{3\times\left[(R_{700}-R_{670})-0.2\times(R_{700}-R_{550})\times\left(\frac{R_{700}}{R_{670}}\right)\right]}{(1+0.16)\times(R_{800}-R_{670})/(R_{800}+R_{670}+0.16)}$	0.58 0.01	[28] [38]	*Vitis vinifera* L. cv tempranillo *Vitis vinifera* L. cv thomson seedless
SRI	$\frac{R_{800}}{R_{550}}$	0.64	[28]	*Vitis vinifera* L. cv tempranillo
PRI	$\frac{R_{530}-R_{550}}{R_{530}+R_{550}}$	0.25 0.53 0.19	[28] [38] [37]	*Vitis vinifera* L. cv tempranillo *Vitis vinifera* L. cv thomson seedless *Vitis vinifera* L. cv cabernet sauvignon
RDVI	$\frac{R_{800}-R_{670}}{\sqrt{R_{800}+R_{670}}}$	0.10	[28]	*Vitis vinifera* L. cv tempranillo

GI = Green Index, GNVDI = Green Normalized Difference Vegetation Index, MCARI = Modified Chlorophyll Absorption in Reflectance Index, MSAVI = Improved SAVI Index, MSR = Modified Simple Ratio, MTVI3 = Modified Triangular Vegetation Index, NDVI = Normalized Difference Vegetation Index, TCARI/OSAVI = Transformed Chlorophyll Absorption in Reflectance index/Optimized Soil-adjusted Vegetation Index, SRI = Simple Ratio Index, PRI = Photochemical Reflectance Index, RDVI = Renormalized Difference VI.

For example, values of R^2 for the TCARI/OSAVI ranged from 0.58 in Tempranillo [28] to 0.01 in Thompson Seedless [38], while those for PRI varied from 0.53 in Thompson Seedless [38] to 0.19 in Cabernet Sauvignon [37]. For the NDVI, Baluja, et al. [28] indicated R^2 = 0.68 in Tempranillo while Rapaport, et al. [37] observed R^2 = 0.03 in Cabernet Sauvignon.

In this regard, some studies suggest that spectral indices based on information between 500 and 800 nm are not suitable in all field conditions to estimate vine water potential and suggest that wavelengths greater than 800 nm could better represent water status [37,39]. For example, WI (water Index = R900/R970) has presented good correlations with water status variables in chardonnay with R^2 values of 0.81 and 0.95 in non-stressed and stressed vines, respectively [36]. Several researchers have suggested that hyperspectral information can be adapted to predict water status and physiological parameters. In that context, Rapaport, et al. [37] indicated that WABIs (water balance indices) that use visible (VIS) and short waver infrared (SWIR) information are good predictors to identify water stress in grapevines showing the best correlation (R^2 = 0.89) for the WABI-2 index (R1500 – R538)/(R1500 + R538). Rallo, et al. [40] suggested that information between SWIR and NIR improves the prediction of leaf water potential over the visible spectrum with a R^2 of 0.7 in the validation process. Also, Pôças, et al., in [41,42] showed that developing different modelling techniques using wavelength information of VIS, green, red-edge and NIR can predict water status with good correlations (R^2 = 0.79 and R^2 = 0.78–0.80, respectively). Rodríguez-Pérez, et al. [43] predicted water status using hyperspectral information with R^2 = 0.91 for EWT (equivalent water thickness) and R^2 = 0.619 for water potential. Based on these relationships between spectral information and water status, the use of miniaturized hyperspectral and multispectral sensors has been proposed to be mounted on an UAV [44]. Hyperspectral sensors can provide a measure of spectral response across hundreds of narrowly defined spectral bands simultaneously, however, they sacrifice

spatial resolution and their commercial prices remain high [45]. Hyperspectral sensors also possess inherent image distortion causing geometric errors and limit detection of vegetation stress using red-edge information [5]. On the other hand, multispectral cameras, specifically the MCA-6 camera (Tetracam's miniature camera array), can detect a VIS-NIR range of the electromagnetic spectrum and has been proven in several studies to identify different types of stress in plants [28,33,46–49]. Also, the spectral reflectance of this type of camera has been compared with WorldView-2 satellite, producing similar results [50].

1.3. Machine Learning Techniques and ANN

As conventional multispectral indices present limitations to assess water, artificial neural networks (ANN) could be used to improve the assessment of the spatial variability of vine water status spatial variability. Machine learning techniques and ANN models are applied to perform regression analyses of highly nonlinear problems and find nonlinear relationships between input and output data sets [51]. ANNs have been applied to multispectral information obtained from multiple types of sensors and platforms, for example, for multispectral imagery classification and segmentation [52,53]. Several ANN techniques were preferred over different spectral information capable of predicting firmness and soluble content in apple fruits [54], leaf recognition [55], crop nitrogen stress [56] and vegetation mapping [57]. Moreover, different types of plant stress have been detected using ANN and multispectral information [56,58–60]. Specifically, water stress has been assessed modelling thermal information using ANN [61] showing correlations between 0.89–0.93 in different cultivars.

Considering that conventional spectral indices do not accurately predict the spatial variability of Ψ_{stem}, this study aimed to develop and validate ANN models to improve the prediction of the intra-vineyard spatial variability of Ψ_{stem} using multispectral information between 500–800 nm obtained from an UAV. As a reference, relationships between different conventional spectral indices and Ψ_{stem} were evaluated.

2. Materials and Methods

2.1. Site Description, Experimental Design and Plant Water Status Measurements

The field experiment was conducted in Talca, Maule Region, Chile (35°27′38″ LS 71°29′53″ LW) on Carménère vines grafted on Paulsen-1103. The vines were planted in 2007 in North–South oriented rows at 2.5 m × 1.5 m and trained on a vertical shoot positioned (VSP) system. The field location has a Mediterranean, semi-arid climate with a temperature average of 17.1 °C and annual rainfall of 679 mm. Field collection and flights were carried out in summer, which is predominantly dry and hot (2.2% of annual rainfall). Vineyard soil was classified as Talca series (fine, mixed, thermic Ultic Haploxeralfs) with a clay loam texture and an average bulk density of 1.5 $g \cdot cm^{-3}$. At the effective rooting depth (0–60 cm), the volumetric soil water content at field capacity and wilting point were 0.36 $m^3 \cdot m^{-3}$ and 0.22 $m^3 \cdot m^{-3}$, respectively. The vines were irrigated daily using 4 $L \cdot h^{-1}$ drippers spaced at intervals of 1 m.

The experimental design was completely randomized with four different treatments with four repetitions (see Figure 1) and six plants per repetition. These treatments consisted of four stem water potential thresholds including non-water stress (T_0) ($\Psi_{stem} > -0.8$ MPa), moderate water stress (T_1) (Ψ_{stem} between -0.9 and -1.1 MPa), strong water stress (T_2) (Ψ_{stem} between -1.2 and -1.4 MPa), and severe water stress (T_3) ($\Psi_{stem} < -1.4$ MPa) [27]. A progressive water stress for each treatment was applied by stopping irrigation, and once the specific thresholds were reached, the irrigation was reestablished [62].

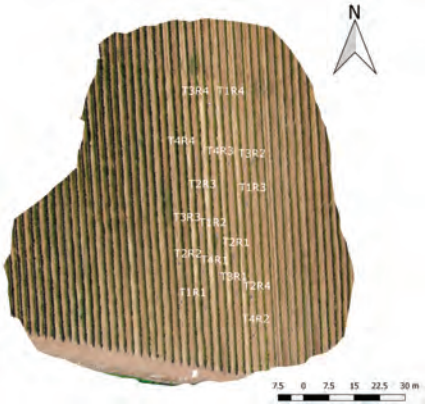


Figure 1. Treatments (T) and Repetitions (R) field distribution.

Ψ_{stem} was measured using a pressure chamber (PMS 600, PMS Instrument Company, Corvallis, OR, USA) from the middle trees of each repetition. A total of 32 leaves were measured corresponding to two mature and healthy sun-exposed leaves, sampled from the middle zone of the canopy which were covered with plastic bags and coated with aluminum foil for at least 1 h before measurements [22]. Ψ_{stem} was measured between 12:00 h and 14:00 h [63].

2.2. UAV Multispectral Image Acquisition

Five flights at an altitude of 60 m high were carried out during two seasons (three flights in 2014 and two in 2015) with the aim of extending the variability of field and plant condition. Meteorological conditions and phenological stages for each day of data collection were recorded (Table 2). All flights and image acquisition were concurrently done with Ψ_{stem} field measurements.

Table 2. Air temperature (Ta), relative humidity (RH), wind speed (u) and phenological stage (PS) at the time of unmanned aerial vehicle (UAV) overpass during the 2014–2015 growing season.

Date	Flight Time (hh:mm)	Ta (°C)	RH (%)	u (Km/h)	PS
04/03/2014	13:00	21.3	52.5	5	Ripening
13/03/2014	12:30	21.6	54.3	3.5	Ripening
19/03/2014	12:45	21.3	51.4	3.5	Berry development
14/01/2015	12:30	25.2	49.7	6.8	Berry development
27/01/2015	12:30	24.4	41.2	7.4	Berry development

Flights in both seasons were carried out between 12:30 and 13:00, to reduce the 'shadow-effect' on the images [64]. Multispectral images were obtained from a MCA-6 camera (Tetracam's miniature camera array), recording wavelengths at 530, 550, 570, 670, 700 and 800 nm. The image reflectance was normalized using a 'white reference' Spectralon panel (Labsphere Inc., Sutton, NH, USA) and compared with a spectroradiometer (SVC HR-1024, Spectra Vista Cooperation, Poughkeepsie, NY, USA) to account for any relative spectral response of each band of the camera as proposed by Laliberte, et al. [65]. All image processing was carried out using Matlab (MATLAB 2013a, The MathWorks, INC., Natick, MA, USA). The MCA-6 sensor was mounted on an octocopter, Mikrokopter Okto XL, equipped with the FlightNav 2.1 flight and navigation controller, MK3638 motors and 12″ × 3.8″ propellers. The sensor was affixed to a servo-controlled gimbal for stability and to ensure that it pointed directly down during flight.

2.3. Soil–Canopy Pixel Distinction

To separate canopy pixels a double normal distribution based on NDVI (normalized difference vegetation index) was built, because this index is related with vegetation structure [66]. From this distribution, one peak corresponded to soil and the other to canopy. The lower percentage of occurrence between both peaks which corresponded to neither canopy nor soil information was calculated. This NDVI value was used to apply a binary mask to the images with the aim of isolating the canopy and to extract pure plant spectral information.

Once the images were obtained and preprocessed, different spectral indices were calculated per pixel (6×6 cm^2). Each experimental plot was isolated from the image, eliminating the border and separating soil from pure canopy information. This information was then correlated with field measurements of Ψ_{stem} to identify the indices that better represented the stem water potential. ANN models were tested to identify the best band combinations to simulate Ψ_{stem}.

2.4. Artificial Neural Network (ANN) Computing

The first model was built including all bands (530, 550, 570, 670, 700 and 800 nm). Each band was then isolated to identify the best relationship between ANN and Ψ_{stem}. A MultiLayer Perceptron (MLP) ANN type was used and a back-propagation process was carried out for weight calculations, in the same manner implemented in related studies [67–69].

To select the best combination of hidden nodes and number of iterations, we implemented the methodology developed by Ballesteros, et al. [70], which also avoided the problem of local convergence of the model. This methodology is based on evaluating the root mean square error (RMSE) with a trial-and-error method that implements the neural network with a wide range of hidden nodes (from two to twenty in one-node increments) and iterations (100–5000 in steps of 100 iterations). For each combination, the ANN was trained 20 times, which avoided the problem of local convergence. This method, although is computationally expensive, permits the clear detection of problems of local convergence that could appear when applying a back-propagation algorithm and overfitting the model [67,71].

2.5. Statistical Analysis

The data set was divided into calibration and validation, where 80% was used for the correlations and the ANN model for the calibration process, while the other 20% was used to validate the models. The validation set was obtained by random selection over a repetition of every treatment. The coefficient of determination (R^2) was calculated to determine the linear correlation of Ψ_{stem} with the conventional indices. For the model validation, comparisons between observed and estimated values of Ψ_{stem} were carried out using the mean absolute error (MAE), root mean square error (RMSE), relative error (RE) and a modified index of agreement (d) [72].

3. Results

3.1. Soil–Canopy Pixel Distinction

The threshold value of NDVI to separate pure vegetation over other information was 0.46 (red line) with 0.029% of occurrence (Figure 2A) which corresponded to the lowest frequency of NDVI occurrence values with 0.45 (Figure 2B). To validate these results, fractional cover (fc) was calculated and compared with the methodology proposed by Ballesteros, et al. [73] who separate vegetation from soil using an ANN. Using the same set of images, the mean values of fc estimated using the methodology suggested by Ballesteros, et al. [73] and proposed in this study were 28.12 ($\pm$ 0.4) and 28.32 ($\pm$ 0.6)%, respectively. The NDVI threshold of 0.46 was selected to build a mask that was applied to the multispectral indices and ANN images. Figure 3 represents an example of pure canopy information for the NDVI and ANN model, based on the built mask.

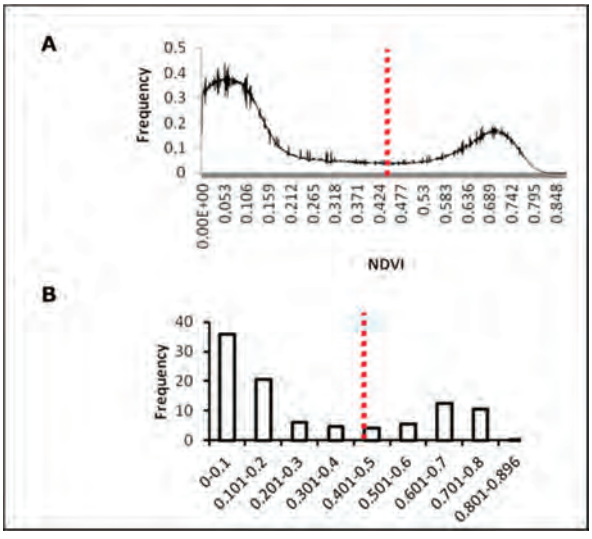


Figure 2. NDVI (Normalized Difference Vegetation Index) values distribution for soil-canopy pixels distinction. (**A**) NDVI frequency graph; (**B**) NDVI-ranged frequencies graph. Red line shows the NDVI threshold to separate canopy from soil.

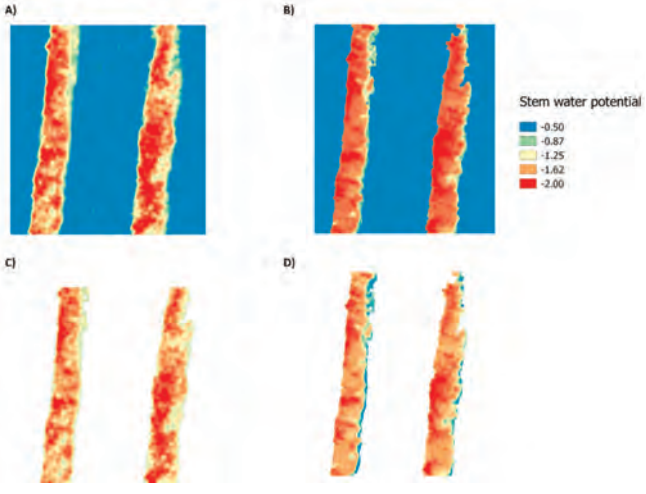


Figure 3. Examples of soil–canopy pixel distinction by the mask application based on the NDVI threshold for NDVI and artificial neural network model ANN-2. (**A**) NDVI soil–canopy information; (**B**) NDVI pure canopy; (**C**) ANN-2 soil–canopy information; (**D**) ANN-2 pure canopy.

3.2. Statistical Analysis for ANN Models and Spectral Indices

Statistical parameters for linear correlations between multispectral indices and midday stem water potential (Ψ_{stem}) are presented in Table 3. There were significant linear correlations between Ψ_{stem} versus the conventional spectral indices NDVI (Normalized Difference Vegetation Index), GNDVI (Green Normalized Difference Vegetation Index) and MSR (Modified Simple Ratio) with values

of R^2 ranging between 0.31–0.35. For other multispectral indices, the regression analysis indicated that values of R^2 were lower than 0.1.

Table 3. Linear correlations between multispectral indices and midday stem water potential (Ψ_{stem}).

Index	a	b	R^2
NDVI *	−4.70	6.19	0.35
GNDVI *	−203.36	−140.75	0.31
PRI	−1.32	1.44	0.09
TCARI-OSAVI	−0.92	−0.74	0.09
GI	−2.03	1.40	0.06
MCARI	−1.27	−0.60	0.02
MCARI1	−1.22	−0.33	0.03
MCARI2	−1.43	0.03	<0.01
MSAVI	−1.31	−0.28	0.00
MSR *	10.78	8.45	0.34
MTVI3	−1.22	−0.33	0.03
SRI	−2.01	0.23	0.06
RDVI	−1.28	−0.35	0.00

* $p < 0.05$, a = intercept, b = slope.

The values of R^2 for the ANN training process were between 0.56–0.87 (Table 4) with the best performance observed for the ANN-2 model, which included the bands 550, 570, 670, 700 and 800 nm.

Table 4. Values of coefficient of determination (R^2) for the artificial neural network (ANN) model training.

ANN Model	Bands	R^2
ANN-1 **	R_{530}, R_{550}, R_{570}, R_{670}, R_{700}, R_{800}	0.87
ANN-2 **	R_{550}, R_{570}, R_{670}, R_{700}, R_{800}	0.87
ANN-3 **	R_{530}, R_{570}, R_{670}, R_{700}, R_{800}	0.84
ANN-4 **	R_{530}, R_{550}, R_{670}, R_{700}, R_{800}	0.78
ANN-5 **	R_{530}, R_{550}, R_{570}, R_{700}, R_{800}	0.78
ANN-6 **	R_{530}, R_{550}, R_{570}, R_{670}, R_{800}	0.68
ANN-7 **	R_{530}, R_{550}, R_{570}, R_{670}, R_{700}	0.56

** $p < 0.01$.

Only the indices that were statistically representative (NDVI, GNDVI and MSR) were used in the model validation. The values of the index of agreement ranged between 0.51–0.54 while those of MAE, RMSE and RE were between 0.26–0.27 MPa, 0.32–0.34 MPa and −24.2–25.6%, respectively. The model validation indicated that ANN models were more accurate than the conventional indices, with indices of agreement ranging between 0.66–0.82. In this case, the ANN-2 model overestimated the values of Ψ_{stem} with values of MAE, RMSE, RE equal to 0.1 MPa, 0.12 MPa and −9.107%, respectively (Table 5).

When the estimated and observed values Ψ_{stem} for the ANN-2 and NDVI models were compared, the ANN-2 model was more accurate with a $R^2 = 0.87$ and $R^2 = 0.35$ respectively and closer to the 1:1 line (Figure 4).

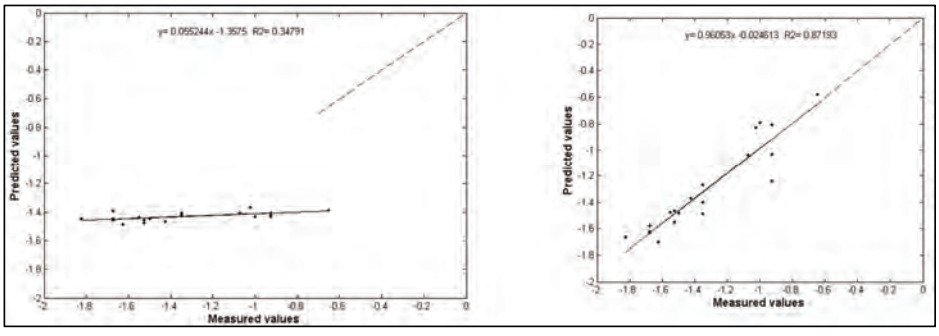
The ANN-2 model that included R550, R570, R670, R700 and R800 was applied to a whole flight and stem water potential was calculated applying the soil–plant filter described in the methodology. Figure 5A shows the prediction of Ψ_{stem} for each canopy pixel that was isolated by the soil–canopy distinction method used in this study. It represents the variability in the field considering a variation as small as 6×6 cm^2. Based on Figure 5A,B, the differences caused by the water status treatments can be identified showing values that ranged between −0.3 and −2 MPa. Figure 5B represents the integration and classification of the information from the individual pixels for the whole treatment and

the high contrast between non-stressed and severely-stressed plants can be analyzed. Although middle stress treatments do not show higher differences among them, they represent the transition between extreme treatments. Figure 6 represents the spatial variability of the field where the zones of different stress levels can be easily identified. The zones with higher levels of stress (T_3 and T_2 treatments) are represented in red and strong orange, while the treatments with lower levels of stress (T_0 and T_1) are represented in green. Based on Figure 6, moderate water stress can be identified for the majority of vines that were not part of the model's construction.

Table 5. Statistical parameters of validation for conventional indices and artificial neural network (ANN) models.

Multispectral Index/ANNModel	MAE (MPa)	RMSE (MPa)	RE (%)	d
Multispectral indices				
NDVI *	0.25	0.32	−24.22	0.54
GNDVI *	0.27	0.34	−25.58	0.51
MSR *	0.26	0.33	−24.57	0.53
ANN models				
ANN-1 **	0.1	0.12	−9.21	0.82
ANN-2 **	0.1	0.12	−9.11	0.82
ANN-3 **	0.11	0.13	−9.68	0.8
ANN-4 **	0.12	0.15	−11.55	0.78
ANN-5 **	0.13	0.15	−11.61	0.77
ANN-6 **	0.15	0.2	−15.2	0.73
ANN-7 **	0.19	0.22	−16.5	0.66

MAE = mean absolute error, RMSE = root mean square error, RE = relative error, d = index of agreement. * $p < 0.05$, ** $p < 0.01$.



(**A**) (**B**)

Figure 4. Comparison between estimated and measured values of midday stem water potential (MPa). (**A**) Normalized difference vegetation index (NDVI); (**B**) ANN-2 model.

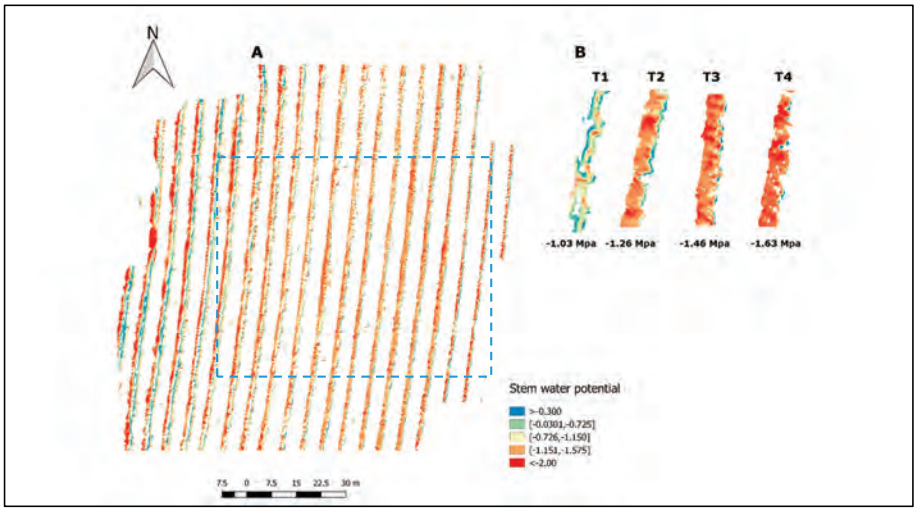


Figure 5. Predicted values of stem water potential (Ψ_{stem}) for a whole flight. (**A**) All pure-canopy pixel information within the vineyard; (**B**) Predicted Ψ_{stem} classification within the four treatments.

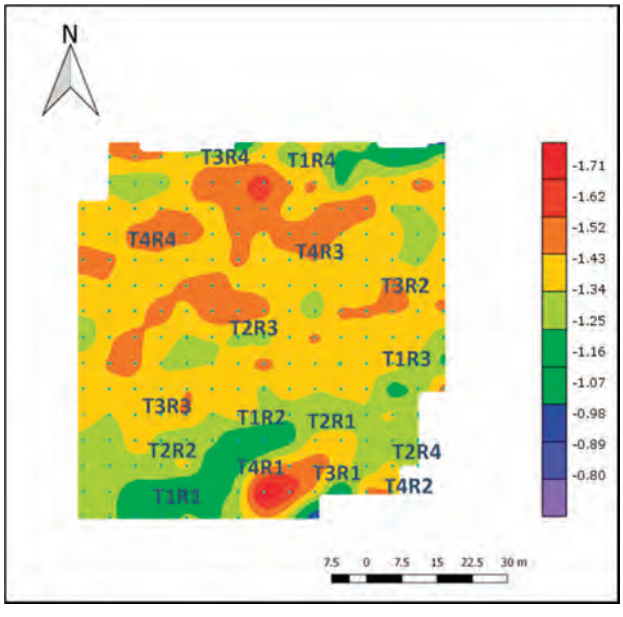


Figure 6. Intra-vineyard spatial variability of predicted midday stem water potential (Ψ_{stem}) using an artificial neural network (ANN) model.

4. Discussion

The results of conventional indices were consistent with the study carried out by Baluja, et al. [28], who indicated that higher correlations were observed for NDVI, GNDVI, TCARI/OSAVI and MSR with R^2 ranging between 0.58 and 0.68. In this study, lower statistical values for correlation and

validation between spectral indices and Ψ_{stem} were found, despite the inclusion of NIR and RED wavelength information, which have high reflectance on plant tissue [74] and a high absorbance by Chl [75], respectively. These indices can only indirectly detect water status differences, because they were developed to represent different physiological variables that can change according to different levels of water status. In this context, NDVI has been reported to be a good indicator of 'vegetative expression' [27] while GNDVI has been reported as a better form to detect Chl pigment concentration [76], which is modified under stress conditions. MSR was developed to improve the relationship of other indices with biophysical parameters in boreal forests [77]. TCARI/OSAVI was developed to make accurate predictions of crop Chl [78]. Furthermore, indices that use wavelength information between 500 and 800 nm have been reported by several studies with a high variation of R^2 among vine cultivars when predicting water status. For example, TCARI/OSAVI showed the best correlation for 0.58 in Tempranillo [28], which decreased to 0.01 in Thompson Seedless [38]. NDVI showed the highest variation, reaching the lowest value of $R^2 = 0.03$ in Cabernet Sauvignon [37]. This variation could be associated with the non-linear effect between water stress on different wavelength reflectances. The relationship between spectral indices and Ψ_{stem} is due to indirect changes produced by different levels of water stress, in contrast to thermal information, where direct effects such as stomatal closure can be assessed by thermal changes [64,68,79].

In this context, ANNs identify complex nonlinear relationships between input and output data sets [51] through input, hidden and output node layers [69]. That is the reason why ANNs have been used in several agricultural studies to analyze complex and non-linear relationships, such as ETo forecasting [80], rainfall–runoff modelling [51], rainfall forecasting [81], fruit firmness prediction [82], nitrogen stress identification [56], leaf recognition [55] and prediction of firmness and soluble content by using multispectral information [54]. ANN models have been proposed to better predict output variables compared with partial least-square (PLS) models [83], especially when NIR information is used [84,85].

Despite the advantages of ANN, several disadvantages and limitations are presented for these models. For example, when compared with linear relationships, building nonlinear models is inherently more difficult than linear ones [86]. Also, when machine learning models are applied to complex and high-dimensionality models, some criteria need to be accounted for. For example, the optimum dimensional reduction of classifiers is needed to improve classification [87–89]. Furthermore, Wu, et al. [81] and Taormina, et al. [90] suggested that inputs, modelling and data processing can be strongly improved if ANN models are coupled with preprocessing techniques [91]. Tu [92], made a comparison between logistic regressions and ANNs suggesting some disadvantage, such as (i) that the 'black-box' nature of ANNs has limited ability to explicitly identify possible causal relationships and cannot easily determine which variables are the most important contributors to a particular output; (ii) ANN development is a computationally intensive procedure that requires greater calculation time, which makes the portability application difficult when applying to the field; and (iii) due to the model interaction and nonlinearity, ANNs may cause overfitting of the training data set and produce poor performance in external sets (which are site-specific). However, Tu [92] suggested that this can be prevented by limiting the number of hidden nodes, adding a penalty term to the objective function for large weights and limiting the amount of training by cross-validation. In this case, dividing the dataset into calibration and validation data avoids overfitting problems. Zhang, et al. [93] suggested that ANN models are stronger than linear models when non-linear problems are studied and give several recommendations and examples of ANN model applications.

For the ANN-2 model, the exclusion of R530 is consistent with Rapaport, et al. [37], who found that at 530 nm the slope in relation to water status started to increase and reached the best correlation with Ψ_{leaf} at 538 nm. It has been proposed that at 531 nm nutrient and water stress can be detected, related to the xanthophyll cycle [94] due to a decrease in reflectance associated with a photoprotective response [95,96]. Despite this response, these changes can not always be identified in severely water stressed conditions [95,97]. In this context, Rapaport, et al [37] indicated that the information of 550,

570, 700 nm could better simulate different plant water statuses, which was included in the ANN-2 model. They also suggested that the NIR and RED wavelengths increased in all plant water statuses (control, light–moderate and severe stress). This can be understood as being due to the fact that 670 nm (RED) and 800 nm (NIR) wavelengths can detect plant tissue and have low and high reflectance in bushy (healthy) plants, respectively [98,99]. Since our study was carried out between two different seasons and in different months, we suggest that these wavelengths are relevant to representing all plant physiology differences presented in the field caused by the treatments. The final prediction for the whole flight (Figures 5 and 6) represents the ability of the ANN-2 model to identify different contrasts of stem water potential present in the field. Also, the spatial variability of the Ψ_{stem} can be assessed and it is consistent with the field experimental design.

5. Conclusions

The utilization of UAV in obtaining high-resolution multispectral images and the use of artificial neural networks improved the assessment of the midday stem water potential spatial variability in a Mediterranean Carménère vineyard located in Talca, Maule Region, Chile. Artificial neural network models using information between 550–800 nm improved the Ψ_{stem} prediction showing values of R^2, MAE, RMSE, RE equal to 0.87, 0.1 MPa, 0.12 MPa and -9.107%, respectively. As a future prospective, this study should be applied to a larger number of cultivars and fitted to different species to predict the spatial variability of water stress. Moreover, ANN-coupled models and different machine learning techniques should be implemented to assess improvements in the prediction of stem water potential spatial variability.

Acknowledgments: This study was supported by the Chilean government through the projects CONICYT-PFCHA (No. 2014-21140229) and FONDECYT (No. 1160997) and by the Universidad de Talca through the research program Adaptation of Agriculture to Climate Change (A2C2). The authors wish to express their gratitude to the Spanish Ministry of Education and Science (MEC), for funding the projects AGL2011-30328-C02-01 and AGL2014-59747-C2-1-R (Co-funded by FEDER).

Author Contributions: Tomas Poblete and Samuel Ortega-Farías conceived and designed this research and data analysis. They prepared the manuscript and consecutive revisions with contribution from all authors. Tomas Poblete also contributed in the image processing, algorithms procedures and artificial neural networks programming. Miguel A. Moreno contributed in the development and implementation of the artificial neural networks models. Matthew Bardeen contributed with the implementation and specialized support on UAV technologies.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Food and Agriculture Organization (FAO); Allen, R.G.; Pereira, L.S.; Raes, D.; Smith, M. *Crop Evapotranspiration-Guidelines for Computing Crop Water Requirements-Fao Irrigation and Drainage Paper 56*; FAO: Rome, Italy, 1998; Volume 300.
2. Ortega-Farias, S.O.; Cuenca, R.H.; English, M. Hourly grass evapotranspiration in modified maritime environment. *J. Irrig. Drain. Eng.* **1995**, *121*, 369–373. [CrossRef]
3. Cohen, Y.; Alchanatis, V.; Meron, M.; Saranga, Y.; Tsipris, J. Estimation of leaf water potential by thermal imagery and spatial analysis. *J. Exp. Bot.* **2005**, *56*, 1843–1852. [CrossRef] [PubMed]
4. Ortega-Farias, S.; Acevedo, C.; Righetti, T.; Matus, F.; Moreno, Y. Irrigation-management decision system (IMDS) for vineyards (regions VI and VII of Chile). FAO Land and Water Bulletin (FAO): Rome, Italy, 2005.
5. Spano, D.; Snyder, R.; Duce, P. Estimating sensible and latent heat flux densities from grapevine canopies using surface renewal. *Agric. For. Meteorol.* **2000**, *104*, 171–183. [CrossRef]
6. Ortega-Farias, S.; Irmak, S.; Cuenca, R. Special issue on evapotranspiration measurement and modeling. *Irrig. Sci.* **2009**, *28*, 1–3. [CrossRef]
7. Turner, N. Plant water relations and irrigation management. *Agric. Water Manag.* **1990**, *17*, 59–73. [CrossRef]
8. Dobriyal, P.; Qureshi, A.; Badola, R.; Hussain, S.A. A review of the methods available for estimating soil moisture and its implications for water resource management. *J. Hydrol.* **2012**, *458*, 110–117. [CrossRef]

9. Granier, C.; Aguirrezabal, L.; Chenu, K.; Cookson, S.J.; Dauzat, M.; Hamard, P.; Thioux, J.J.; Rolland, G.; Bouchier-Combaud, S.; Lebaudy, A. PHENOPSIS, an automated platform for reproducible phenotyping of plant responses to soil water deficit in arabidopsis thaliana permitted the identification of an accession with low sensitivity to soil water deficit. *New Phytol.* **2006**, *169*, 623–635. [CrossRef] [PubMed]
10. Escalona, J.; Flexas, J.; Medrano, H. Drought effects on water flow, photosynthesis and growth of potted grapevines. *Vitis* **2015**, *41*, 57.
11. Poblete-Echeverría, C.A.; Ortega-Farias, S.O. Evaluation of single and dual crop coefficients over a drip-irrigated merlot vineyard (vitis vinifera l.) using combined measurements of sap flow sensors and an eddy covariance system. *Aust. J. Grape Wine Res.* **2013**, *19*, 249–260. [CrossRef]
12. Intrigliolo, D.; Castel, J. Evaluation of grapevine water status from trunk diameter variations. *Irrig. Sci.* **2007**, *26*, 49–59. [CrossRef]
13. Marino, G.; Pallozzi, E.; Cocozza, C.; Tognetti, R.; Giovannelli, A.; Cantini, C.; Centritto, M. Assessing gas exchange, sap flow and water relations using tree canopy spectral reflectance indices in irrigated and rainfed *Olea europaea* L. *Environ. Exp. Bot.* **2014**, *99*, 43–52. [CrossRef]
14. Jara-Rojas, F.; Ortega-Farías, S.; Valdés-Gómez, H.; Acevedo-Opazo, C. Gas exchange relations of ungrafted grapevines (cv. Carménère) growing under irrigated field conditions. *S. Afr. J. Enol. Vitic.* **2015**, *36*, 231–242.
15. Flexas, J.; Escalona, J.M.; Evain, S.; Gulías, J.; Moya, I.; Osmond, C.B.; Medrano, H. Steady-state chlorophyll fluorescence (Fs) measurements as a tool to follow variations of net CO_2 assimilation and stomatal conductance during water-stress in C_3 plants. *Physiol. Plant.* **2002**, *114*, 231–240. [CrossRef] [PubMed]
16. Costa, J.M.; Grant, O.M.; Chaves, M.M. Thermography to explore plant–environment interactions. *J. Exp. Bot.* **2013**, *64*, 3937–3949. [CrossRef] [PubMed]
17. Sepúlveda-Reyes, D.; Ingram, B.; Bardeen, M.; Zúñiga, M.; Ortega-Farías, S.; Poblete-Echeverría, C. Selecting canopy zones and thresholding approaches to assess grapevine water status by using aerial and ground-based thermal imaging. *Remote Sens.* **2016**, *8*, 822. [CrossRef]
18. Webber, H.; Ewert, F.; Kimball, B.; Siebert, S.; White, J.; Wall, G.; Ottman, M.J.; Trawally, D.; Gaiser, T. Simulating canopy temperature for modelling heat stress in cereals. *Environ. Model. Softw.* **2016**, *77*, 143–155. [CrossRef]
19. Deery, D.M.; Rebetzke, G.J.; Jimenez-Berni, J.A.; James, R.A.; Condon, A.G.; Bovill, W.D.; Hutchinson, P.; Scarrow, J.; Davy, R.; Furbank, R.T. Methodology for high-throughput field phenotyping of canopy temperature using airborne thermography. *Front. Plant Sci.* **2016**, *7*. [CrossRef] [PubMed]
20. Girona, J.; Mata, M.; Del Campo, J.; Arbonés, A.; Bartra, E.; Marsal, J. The use of midday leaf water potential for scheduling deficit irrigation in vineyards. *Irrig. Sci.* **2006**, *24*, 115–127. [CrossRef]
21. Düring, H.; Loveys, B. Diurnal changes in water relations and abscisic acid in field grown vitis vinifera cvs. I. Leaf water potential components and leaf conductance under humid temperate and semiarid conditions. *Vitis* **2016**, *21*, 223.
22. Choné, X.; Van Leeuwen, C.; Dubourdieu, D.; Gaudillère, J.P. Stem water potential is a sensitive indicator of grapevine water status. *Ann. Bot.* **2001**, *87*, 477–483. [CrossRef]
23. Romero, P.; García, J.G.; Fernández-Fernández, J.I.; Muñoz, R.G.; del Amor Saavedra, F.; Martínez-Cutillas, A. Improving berry and wine quality attributes and vineyard economic efficiency by long-term deficit irrigation practices under semiarid conditions. *Sci. Hortic.* **2016**, *203*, 69–85. [CrossRef]
24. Balint, G.; Reynolds, A.G. Irrigation level and time of imposition impact vine physiology, yield components, fruit composition and wine quality of ontario chardonnay. *Sci. Hortic.* **2017**, *214*, 252–272. [CrossRef]
25. Williams, L.E.; Trout, T.J. Relationships among vine-and soil-based measures of water status in a thompson seedless vineyard in response to high-frequency drip irrigation. *Am. J. Enol. Vitic.* **2005**, *56*, 357–366.
26. Tognetti, R.; d'Andria, R.; Morelli, G.; Alvino, A. The effect of deficit irrigation on seasonal variations of plant water use in *Olea europaea* L. *Plant Soil* **2005**, *273*, 139–155. [CrossRef]
27. Acevedo-Opazo, C.; Tisseyre, B.; Guillaume, S.; Ojeda, H. The potential of high spatial resolution information to define within-vineyard zones related to vine water status. *Precis. Agric.* **2008**, *9*, 285–302. [CrossRef]
28. Baluja, J.; Diago, M.P.; Balda, P.; Zorer, R.; Meggio, F.; Morales, F.; Tardaguila, J. Assessment of vineyard water status variability by thermal and multispectral imagery using an unmanned aerial vehicle (UAV). *Irrig. Sci.* **2012**, *30*, 511–522. [CrossRef]

29. Park, S.; Ryu, D.; Fuentes, S.; Chung, H.; Hernández-Montes, E.; O'Connell, M. Adaptive estimation of crop water stress in nectarine and peach orchards using high-resolution imagery from an unmanned aerial vehicle (UAV). *Remote Sens.* **2017**, *9*, 828. [CrossRef]
30. Ortega-Farías, S.; Ortega-Salazar, S.; Poblete, T.; Kilic, A.; Allen, R.; Poblete-Echeverría, C.; Ahumada-Orellana, L.; Zuñiga, M.; Sepúlveda, D. Estimation of energy balance components over a drip-irrigated olive orchard using thermal and multispectral cameras placed on a helicopter-based unmanned aerial vehicle (UAV). *Remote Sens.* **2016**, *8*, 638. [CrossRef]
31. Toth, C.; Jóźków, G. Remote sensing platforms and sensors: A survey. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 22–36. [CrossRef]
32. Colomina, I.; Molina, P. Unmanned aerial systems for photogrammetry and remote sensing: A review. *ISPRS J. Photogramm. Remote Sens.* **2014**, *92*, 79–97. [CrossRef]
33. Berni, J.A.; Zarco-Tejada, P.J.; Suárez, L.; Fereres, E. Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 722–738. [CrossRef]
34. Rey, C.; Martin, M.; Lobo, A.; Luna, I.; Diago, M.P.; Millan, B.; Tardáguila, J. Multispectral imagery acquired from a uav to assess the spatial variability of a tempranillo vineyard. In *Precision Agriculture'13*; Springer: Berlin, Germany, 2013; pp. 617–624.
35. Zaman-Allah, M.; Vergara, O.; Araus, J.; Tarekegne, A.; Magorokosho, C.; Zarco-Tejada, P.; Hornero, A.; Albà, A.H.; Das, B.; Craufurd, P. Unmanned aerial platform-based multi-spectral imaging for field phenotyping of maize. *Plant Methods* **2015**, *11*, 1. [CrossRef] [PubMed]
36. Serrano, L.; González-Flor, C.; Gorchs, G. Assessing vineyard water status using the reflectance based water index. *Agric. Ecosyst. Environ.* **2010**, *139*, 490–499. [CrossRef]
37. Rapaport, T.; Hochberg, U.; Shoshany, M.; Karnieli, A.; Rachmilevitch, S. Combining leaf physiology, hyperspectral imaging and partial least squares-regression (PLS-R) for grapevine water status assessment. *ISPRS J. Photogramm. Remote Sens.* **2015**, *109*, 88–97. [CrossRef]
38. Zarco-Tejada, P.J.; González-Dugo, V.; Williams, L.; Suárez, L.; Berni, J.A.; Goldhamer, D.; Fereres, E. A pri-based water stress index combining structural and chlorophyll effects: Assessment using diurnal narrow-band airborne imagery and the cwsi thermal index. *Remote Sens. Environ.* **2013**, *138*, 38–50. [CrossRef]
39. Peñuelas, J.; Pinol, J.; Ogaya, R.; Filella, I. Estimation of plant water concentration by the reflectance water index WI (R900/R970). *Int. J. Remote Sens.* **1997**, *18*, 2869–2875. [CrossRef]
40. Rallo, G.; Minacapilli, M.; Ciraolo, G.; Provenzano, G. Detecting crop water status in mature olive groves using vegetation spectral measurements. *Biosyst. Eng.* **2014**, *128*, 52–68. [CrossRef]
41. Pôças, I.; Rodrigues, A.; Gonçalves, S.; Costa, P.M.; Gonçalves, I.; Pereira, L.S.; Cunha, M. Predicting grapevine water status based on hyperspectral reflectance vegetation indices. *Remote Sens.* **2015**, *7*, 16460–16479. [CrossRef]
42. Pôças, I.; Gonçalves, J.; Costa, P.M.; Gonçalves, I.; Pereira, L.S.; Cunha, M. Hyperspectral-based predictive modelling of grapevine water status in the portuguese douro wine region. *Int. J. Appl. Earth Obs. Geoinform.* **2017**, *58*, 177–190. [CrossRef]
43. Rodríguez-Pérez, J.R.; Riaño, D.; Carlisle, E.; Ustin, S.; Smart, D.R. Evaluation of hyperspectral reflectance indexes to detect grapevine water status in vineyards. *Am. J. Enol. Vitic.* **2007**, *58*, 302–317.
44. Arfaoui, A. Unmanned aerial vehicle: Review of onboard sensors, application fields, open problems and research issues. *Int. J. Image Process.* **2017**, *11*, 12.
45. Uto, K.; Seki, H.; Saito, G.; Kosugi, Y.; Komatsu, T. Development of a low-cost, lightweight hyperspectral imaging system based on a polygon mirror and compact spectrometers. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2016**, *9*, 861–875. [CrossRef]
46. Wójtowicz, M.; Wójtowicz, A.; Piekarczyk, J. Application of remote sensing methods in agriculture. *Commun. Biometry Crop Sci.* **2016**, *11*, 31–50.
47. Reynolds, A.G.; Brown, R.; Kotsaki, E.; Lee, H.-S. Utilization of proximal sensing technology (greenseeker) to map variability in ontario vineyards. In Proceedings of the 19th International Symposium GiESCO, Gruissan, France, 31 May–5 June 2015; pp. 593–597.
48. Stagakis, S.; González-Dugo, V.; Cid, P.; Guillén-Climent, M.L.; Zarco-Tejada, P.J. Monitoring water stress and fruit quality in an orange orchard under regulated deficit irrigation using narrow-band structural and physiological remote sensing indices. *ISPRS J. Photogramm. Remote Sens.* **2012**, *71*, 47–61. [CrossRef]

49. Suárez, L.; Zarco-Tejada, P.J.; González-Dugo, V.; Berni, J.; Sagardoy, R.; Morales, F.; Fereres, E. Detecting water stress effects on fruit quality in orchards with time-series pri airborne imagery. *Remote Sens. Environ.* **2010**, *114*, 286–298. [CrossRef]
50. Laliberte, A.S.; Goforth, M.A.; Steele, C.M.; Rango, A. Multispectral remote sensing from unmanned aircraft: Image processing workflows and applications for rangeland environments. *Remote Sens.* **2011**, *3*, 2529–2551. [CrossRef]
51. Hsu, K.L.; Gupta, H.V.; Sorooshian, S. Artificial neural network modeling of the rainfall-runoff process. *Water Resour. Res.* **1995**, *31*, 2517–2530. [CrossRef]
52. Reddick, W.E.; Glass, J.O.; Cook, E.N.; Elkin, T.D.; Deaton, R.J. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE Trans. Med. Imaging* **1997**, *16*, 911–918. [CrossRef] [PubMed]
53. Heermann, P.D.; Khazenie, N. Classification of multispectral remote sensing data using a back-propagation neural network. *IEEE Trans. Geosci. Remote Sens.* **1992**, *30*, 81–88. [CrossRef]
54. Lu, R. Multispectral imaging for predicting firmness and soluble solids content of apple fruit. *Postharvest Biol. Technol.* **2004**, *31*, 147–157. [CrossRef]
55. Wu, S.G.; Bao, F.S.; Xu, E.Y.; Wang, Y.-X.; Chang, Y.-F.; Xiang, Q.-L. A leaf recognition algorithm for plant classification using probabilistic neural network. In Proceedings of the 2007 IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, 15–18 December 2007; pp. 11–16.
56. Noh, H.; Zhang, Q.; Shin, B.; Han, S.; Feng, L. A neural network model of maize crop nitrogen stress assessment for a multi-spectral imaging sensor. *Biosyst. Eng.* **2006**, *94*, 477–485. [CrossRef]
57. Carpenter, G.A.; Gopal, S.; Macomber, S.; Martens, S.; Woodcock, C.E.; Franklin, J. A neural network method for efficient vegetation mapping. *Remote Sens. Environ.* **1999**, *70*, 326–338. [CrossRef]
58. Baranowski, P.; Jedryczka, M.; Mazurek, W.; Babula-Skowronska, D.; Siedliska, A.; Kaczmarek, J. Hyperspectral and thermal imaging of oilseed rape (brassica napus) response to fungal species of the genus alternaria. *PLoS ONE* **2015**, *10*, e0122913. [CrossRef] [PubMed]
59. Wang, X.; Zhang, M.; Zhu, J.; Geng, S. Spectral prediction of phytophthora infestans infection on tomatoes using artificial neural network (ANN). *Int. J. Remote Sens.* **2008**, *29*, 1693–1706. [CrossRef]
60. Liu, M.; Liu, X.; Li, M.; Fang, M.; Chi, W. Neural-network model for estimating leaf chlorophyll concentration in rice under stress from heavy metals using four spectral indices. *Biosyst. Eng.* **2010**, *106*, 223–233. [CrossRef]
61. King, B.; Shellie, K. Evaluation of neural network modeling to predict non-water-stressed leaf temperature in wine grape for calculation of crop water stress index. *Agric. Water Manag.* **2016**, *167*, 38–52. [CrossRef]
62. Ahumada-Orellana, L.E.; Ortega-Farías, S.; Searles, P.S.; Retamales, J.B. Yield and water productivity responses to irrigation cut-off strategies after fruit set using stem water potential thresholds in a super-high density olive orchard. *Front. Plant Sci.* **2017**, *8*, 1280. [CrossRef] [PubMed]
63. Moriana, A.; Fereres, E. Plant indicators for scheduling irrigation of young olive trees. *Irrig. Sci.* **2002**, *21*, 83–90.
64. Bellvert, J.; Zarco-Tejada, P.J.; Girona, J.; Fereres, E. Mapping crop water stress index in a 'Pinot-noir'vineyard: Comparing ground measurements with thermal remote sensing imagery from an unmanned aerial vehicle. *Precis. Agric.* **2014**, *15*, 361–376. [CrossRef]
65. Laliberte, A.S.; Rango, A. Texture and scale in object-based analysis of subdecimeter resolution unmanned aerial vehicle (UAV) imagery. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 761–770. [CrossRef]
66. Carlson, T.N.; Ripley, D.A. On the relation between ndvi, fractional vegetation cover, and leaf area index. *Remote Sens. Environ.* **1997**, *62*, 241–252. [CrossRef]
67. Zhang, J.-R.; Zhang, J.; Lok, T.-M.; Lyu, M.R. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training. *Appl. Math. Comp.* **2007**, *185*, 1026–1037. [CrossRef]
68. Santesteban, L.; Di Gennaro, S.; Herrero-Langreo, A.; Miranda, C.; Royo, J.; Matese, A. High-resolution uav-based thermal imaging to estimate the instantaneous and seasonal variability of plant water status within a vineyard. *Agric. Water Manag.* **2017**, *183*, 49–59. [CrossRef]
69. Bishop, C.M. *Neural Networks for Pattern Recognition*; Oxford University Press: Oxford, UK, 1995.
70. Ballesteros, R.; Ortega, J.F.; Moreno, M.Á. Foreto: New software for reference evapotranspiration forecasting. *J. Arid Environ.* **2016**, *124*, 128–141. [CrossRef]
71. Vogl, T.P.; Mangis, J.; Rigler, A.; Zink, W.; Alkon, D. Accelerating the convergence of the back-propagation method. *Biol. Cybern.* **1988**, *59*, 257–263. [CrossRef]

72. Willmott, C.J.; Robeson, S.M.; Matsuura, K. A refined index of model performance. *Int. J. Climatol.* **2012**, *32*, 2088–2094. [CrossRef]
73. Ballesteros, R.; Ortega, J.; Hernández, D.; Moreno, M. Applications of georeferenced high-resolution images obtained with unmanned aerial vehicles. Part I: Description of image acquisition and processing. *Precis. Agric.* **2014**, *15*, 579–592. [CrossRef]
74. Major, D.; Baret, F.; Guyot, G. A ratio vegetation index adjusted for soil brightness. *Int. J. Remote Sens.* **1990**, *11*, 727–740. [CrossRef]
75. Pu, R.-L.; Gong, P. *Hyperspectral Remote Sensing and Its Applications*; Higher Education: Beijing, China, 2000; Volume 8.
76. Gitelson, A.A.; Merzlyak, M.N. Remote sensing of chlorophyll concentration in higher plant leaves. *Adv. Space Res.* **1998**, *22*, 689–692. [CrossRef]
77. Chen, J.M. Evaluation of vegetation indices and a modified simple ratio for boreal applications. *Can. J. Remote Sens.* **1996**, *22*, 229–242. [CrossRef]
78. Haboudane, D.; Miller, J.R.; Tremblay, N.; Zarco-Tejada, P.J.; Dextraze, L. Integrated narrow-band vegetation indices for prediction of crop chlorophyll content for application to precision agriculture. *Remote Sens. Environ.* **2002**, *81*, 416–426. [CrossRef]
79. Bellvert, J.; Marsal, J.; Girona, J.; Zarco-Tejada, P.J. Seasonal evolution of crop water stress index in grapevine varieties determined with high-resolution remote sensing thermal imagery. *Irrig. Sci.* **2015**, *33*, 81–93. [CrossRef]
80. Trajkovic, S.; Todorovic, B.; Stankovic, M. Forecasting of reference evapotranspiration by artificial neural networks. *J. Irrig. Drain. Eng.* **2003**, *129*, 454–457. [CrossRef]
81. Wu, C.; Chau, K.; Fan, C. Prediction of rainfall time series using modular artificial neural networks coupled with data-preprocessing techniques. *J. Hydrol.* **2010**, *389*, 146–167. [CrossRef]
82. Torkashvand, A.M.; Ahmadi, A.; Nikravesh, N.L. Prediction of kiwifruit firmness using fruit mineral nutrient concentration by artificial neural network (ANN) and multiple linear regressions (MLR). *J. Integr. Agric.* **2017**, *16*, 1634–1644. [CrossRef]
83. Lin, M.-I.B.; Groves, W.A.; Freivalds, A.; Lee, E.G.; Harper, M. Comparison of artificial neural network (ANN) and partial least squares (PLS) regression models for predicting respiratory ventilation: An exploratory study. *Eur. J. Appl. Physiol.* **2012**, *112*, 1603–1611. [CrossRef] [PubMed]
84. Iñón, F.A.; Garrigues, S.; de la Guardia, M. Combination of mid-and near-infrared spectroscopy for the determination of the quality properties of beers. *Anal. Chim. Acta* **2006**, *571*, 167–174. [CrossRef] [PubMed]
85. Gonzalez Viejo, C.; Fuentes, S.; Torrico, D.; Howell, K.; Dunshea, F.R. Assessment of beer quality based on foamability and chemical composition using computer vision algorithms, near infrared spectroscopy and artificial neural networks modelling techniques. *J. Sci. Food Agric.* **2017**. [CrossRef] [PubMed]
86. Granger, C.W. Strategies for modelling nonlinear time-series relationships. *Econ. Rec.* **1993**, *69*, 233–238. [CrossRef]
87. Zhang, S.; Chau, K.-W. Dimension reduction using semi-supervised locally linear embedding for plant leaf classification. In Proceedings of the 5th International Conference on Intelligent Computing (ICIC 2009), Ulsan, Korea, 16–19 September 2009; pp. 948–955.
88. Papendick, R.; Camprell, G. *Theory and Measurement of Water Potential*; National Agricultural Library: Beltsville, MD, USA, 1981.
89. Zhang, J.; Chau, K.-W. Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization. *J. Univers. Compt. Sci.* **2009**, *15*, 840–858.
90. Taormina, R.; Chau, K.-W. Data-driven input variable selection for rainfall–runoff modeling using binary-coded particle swarm optimization and extreme learning machines. *J. Hydrol.* **2015**, *529*, 1617–1632. [CrossRef]
91. Chau, K.; Wu, C. A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *J. Hydroinformatics* **2010**, *12*, 458–473. [CrossRef]
92. Tu, J.V. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J. Clin. Epidemiol.* **1996**, *49*, 1225–1231. [CrossRef]
93. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]

94. Gamon, J.; Field, C.; Bilger, W.; Björkman, O.; Fredeen, A.; Peñuelas, J. Remote sensing of the xanthophyll cycle and chlorophyll fluorescence in sunflower leaves and canopies. *Oecologia* **1990**, *85*, 1–7. [CrossRef] [PubMed]
95. Gamon, J.; Penuelas, J.; Field, C. A narrow-waveband spectral index that tracks diurnal changes in photosynthetic efficiency. *Remote Sens. Environ.* **1992**, *41*, 35–44. [CrossRef]
96. Evain, S.; Flexas, J.; Moya, I. A new instrument for passive remote sensing: 2. Measurement of leaf and canopy reflectance changes at 531 nm and their relationship with photosynthesis and chlorophyll fluorescence. *Remote Sens. Environ.* **2004**, *91*, 175–185. [CrossRef]
97. Peñuelas, J.; Gamon, J.; Fredeen, A.; Merino, J.; Field, C. Reflectance indices associated with physiological changes in nitrogen-and water-limited sunflower leaves. *Remote Sens. Environ.* **1994**, *48*, 135–146. [CrossRef]
98. Wang, F.-M.; Huang, J.-F.; Tang, Y.-L.; Wang, X.-Z. New vegetation index and its application in estimating leaf area index of rice. *Rice Sci.* **2007**, *14*, 195–203. [CrossRef]
99. Ollinger, S. Sources of variability in canopy reflectance and the convergent properties of plants. *New Phytol.* **2011**, *189*, 375–394. [CrossRef] [PubMed]

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

Sensors Editorial Office
E-mail: sensors@mdpi.com
www.mdpi.com/journal/sensors

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34
Fax: +41 61 302 89 18

www.mdpi.com

ISBN 978-3-03897-092-7