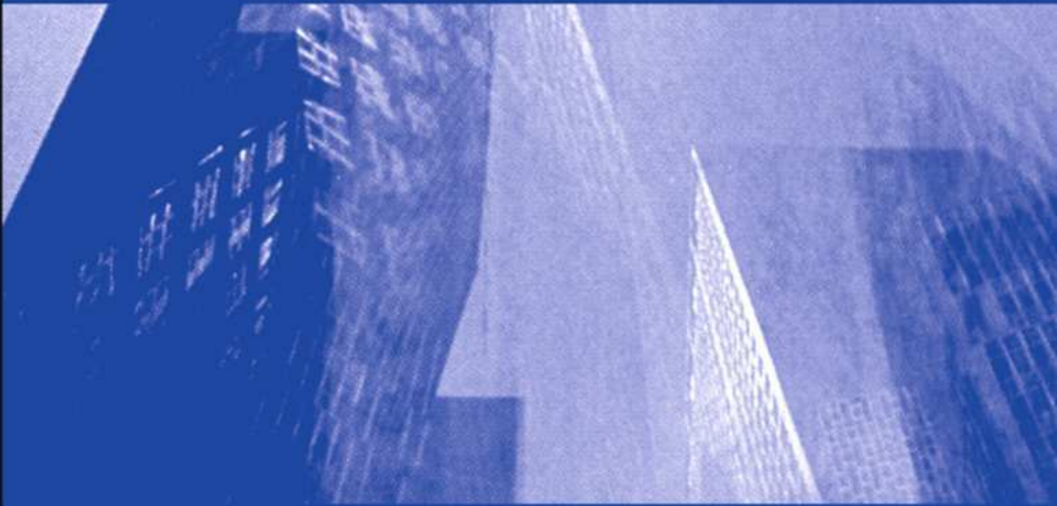
ADVANCED TOPICS IN SCIENCE AND TECHNOLOGY IN CHINA

Zhongxue Gan
Qing Tang

Visual Sensing and its Applications

Integration of Laser Sensors to Industrial Robots



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

Springer

ADVANCED TOPICS
IN SCIENCE AND TECHNOLOGY IN CHINA

ADVANCED TOPICS IN SCIENCE AND TECHNOLOGY IN CHINA

Zhejiang University is one of the leading universities in China. In Advanced Topics in Science and Technology in China, Zhejiang University Press and Springer jointly publish monographs by Chinese scholars and professors, as well as invited authors and editors from abroad who are outstanding experts and scholars in their fields. This series will be of interest to researchers, lecturers, and graduate students alike.

Advanced Topics in Science and Technology in China aims to present the latest and most cutting-edge theories, techniques, and methodologies in various research areas in China. It covers all disciplines in the fields of natural science and technology, including but not limited to, computer science, materials science, life sciences, engineering, environmental sciences, mathematics, and physics.

Zhongxue Gan
Qing Tang

Visual Sensing and its Applications

Integration of Laser Sensors to Industrial Robots

With 215 figures

ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

Springer

Authors

Dr. Zhongxue Gan
ENN Group
065001, Langfang, Hebei, China
E-mail: ganzhongxue@enn.cn

Dr. Qing Tang
InterSmart Robotic Systems Co., Ltd.
065001, Langfang, Hebei, China
E-mail: tangqing@enn.cn

ISSN 1995-6819 e-ISSN 1995-6827
Advanced Topics in Science and Technology in China

ISBN 978-7-308-08051-4
Zhejiang University Press, Hangzhou

ISBN 978-3-642-18286-0 ISBN 978-3-642-18287-7 (eBook)
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011920979

Printed on acid-free paper

Springer is a part of Springer Science+Business Media (www.springer.com)

Foreword

With the rapid development of information technology, the manufacturing sector has entered a new stage, from mass production to user-oriented and customized production, making the flexibility of manufacturing equipment become increasingly demanding. Implementation of flexible robotic manufacturing equipment is the core technology in industrial automation with high efficiency and low-cost. The application of a robot vision system will greatly enhance the degree of flexibility of the robot system for customized products.

Robot vision that integrates vision technology to robot technology adds the capability of a robot to visually perceive the environment and interact with it. To make the robot carry out a given task even in a structural environment, such as welding, palletizing, deburring, and grinding, usually a “teach” or “program” procedure is required manually. The manual teaching entails moving the robot into a number of successive points along the work piece. When the work piece has variations or location of the working changes the existing program that has been taught previously may not be able to work properly. This is mainly due to the fact that the robot lacks a human understanding of a task and the human eyes in identifying the work piece. There have been numerous effort and methods to make the robot to adapt the new tasks without reprogramming by adding sensory components and acquiring feedback from the sensors. Visual sensing is the most powerful mean to equip robots to gather and interpret the necessary information required for execution of new tasks through interaction and on-line learning. Combination of the robot and vision system propels the automation to a higher level of flexibility and reliability that can accomplish the complex tasks like online measurement and inspection, identification and localization, and visual servoing.

This book summarizes the theory of robot vision system and author’s engineering practices. The principles, algorithms, and implementations for robot vision have been reviewed, analyzed and discussed systematically. Specifically, it addresses multi-level calibrations of the robot vision system, including calibration of the visual system itself, TCP calibration, and calibration of the robot to improve the system precision. This book can be a reference source for researchers, engineers, and graduate students in the area of flexible automation, robotics, and machine vision. It can serve as a reference tool for field engineers in design of the robot vision system with technical guidance and practical application examples.

Tzyh Jong Tarn
Washington University, St. Louis
Tsinghua University, Beijing
January 2011

Preface

Industrial robots are designed for tasks such as pick and place, welding, grinding, parts assembly and painting, where repeated work is needed and the robot path is programmed previously. Consequently, if the working condition is changed and deviates from the programmed parameters, the robot may not be able to function properly. To ensure that the robot adapts to new tasks without reprogramming, sensing technology is integrated to the robot system to enhance the robot's capability to work in a dynamic environment. It makes the robot system easy-to-use for the end user and yet operative with a human. Vision sensing is a vital sensing technology where the robot mimics human vision to guide itself through the complex process.

This book focuses on the laser sensor based robotic vision system, due to its robustness and because it is relatively less sensitive to its working environment, and on the rapid growth in its applications. A laser structured light is the main concern in a discussion about vision sensing. Also addressed in this book is each component of the robotic vision system and an emphasis on how to increase the accuracy of the system by three levels of calibration. This includes calibration of the vision system (eye calibration), calibration of eye-to-hand configuration and calibration of robot kinematics (hand calibration). It starts from characteristics of the laser vision system, leading to a discussion on how to build a robotic vision system and its fundamental advantages and limitations in Chapter 2. In Chapter 3, calibration of the vision system is introduced to determine the intrinsic and extrinsic parameters of the vision system. In Chapter 4, calibration of the tool center point (TCP) of the robot vision system is discussed to determine the position of the vision system relative to the robot frame. In Chapter 5, calibration of robot kinematics for serial and parallel robots to increase robot accuracy is presented. In Chapter 6, image processing algorithms involved in the vision sensing technology are addressed. Finally, in Chapter 7, various applications of the laser sensor based robotic vision system are presented.

Furthermore, a general model of the TCP calibration problem and its solution are presented in Chapter 4. It shows that the general mathematical model of TCP calibration is a robot kinematics in addition to a geometrical constraint. The solution to the model is, in general, a nonlinear optimization process that minimizes the cost function defined in the model, which is expected to pave a way for unified calibration of all kinds of robotic manipulators.

Various laser sensor based robot vision systems and their applications are presented in Chapter 7. There are three kinds of applications: measurement and inspection, identification and localization, and visual servoing. In terms of measurement and inspection, the method for detecting the position and orientation of holes or channels in a 3D space using a robotic vision system is presented in Section 7.1. A robotic grinding system for a free form work piece is presented in Section 7.2 where the laser stripe sensor is used to measure and locate the work piece. A profile modeling based grinding approach is presented and discussed in Section 7.3. This approach is applied to blade tip profile refurbishing in the aerospace maintenance industry. A flexible robotic machining system that can compensate for parts shape variations is presented in Section 7.5, by using the measurement result of individual parts as feedback for the robot controller. A highly accurate relative measurement robot system is presented in Section 7.6, which is used to measure the material removal of a free-form work piece in the grinding process. Regarding identification and localization applications, in Section 7.4, a sand core handling/assembly system is presented, which handles and assembles sand cores to sand boxes to form sand molds for sand casting with an accuracy requirement of ±0.3 mm. In Section 7.8, two pick and place robot systems are presented where point laser sensors are used to identify edges of parts. One system is used for log loading and unloading. The other system is designed to automatically install the large solar panel in the field. For visual servoing applications, a general robotic seam tracking system which can tune robotic poses with 6 degrees of freedom is presented in Chapter 7.7. This includes the architecture of the system, the welding joint detection, the path generation algorithm, and computer-robot communication.

Most of the applications presented in the book (mainly in Chapter 7) are primarily based on R&D projects and engineering projects conducted by the authors and their colleagues, including Dr. Yunquan Sun, Mr. Shuihua Wu, Mr. Lizhe Qi, Dr. Hongliang Cui, Dr. Xinbo Huang, and Dr. Xiaoming Liu at InterSmart Robotic Systems Co., Ltd. and New Dimension Technology Corporation, subsidiary companies of the ENN group at Langfang, China. Some fundamental work and concepts can be traced back 10 years ago, when the authors worked in the ABB Cooperate Research Center at Windsor, Connecticut, USA. Research work in this book is partially supported by the National High-Tech R&D Program (863 program) of China, under grant 2007AA04Z243, and the International Science & Technology Cooperation Project of China, under grant 2008DFB70200. The authors appreciate all the support from government funding and the ENN group

for providing the R&D platform.

We wish to acknowledge our sincere appreciation to many individuals who contributed to the writing of the book. In particular, we wish to express our thanks to Mr. Shuihua Wu for providing materials for Sections 7.2, 7.4, 7.5, 7.6, 7.7, and for a helpful discussion of Chapter 4; to Dr. Xinbo Huang for providing materials for Section 7.3; to Mr. Lizhe Qi for providing materials for the writing of Chapter 5; to Dr. Hongliang Cui for providing materials and a helpful discussion of Chapter 6; to Dr. Xinbo Huang and Mr. Lizhe Qi who helped prepare the graphics and plots.

Finally we wish to thank our families for their patience, understanding and support.

Zhongxue Gan
Qing Tang
January 2011

Contents

1 Introduction

When an industrial robot is integrated with a vision sensor to form a robotic vision system, it becomes more flexible and intelligent. Industrial robots are originally designed for the tedious and repeated work, where the robot can be programmed in advance. It lacks of flexibility to handle the situation where the working conditions are changing. If the robot cannot be aware of its own internal working variables as well as the environment around it, it is just a simple manipulator. Combination of vision and robot technology propels robot automation to a new level of higher intelligence and reliability (Florczyk, 2005; Fayor, 2006; Xu, 2008; Kragic, 2009). It makes the robot system easy-to-use for the end user.

Vision sensors are well established in manufacturing industries, but they have been seen limited deployment in robotic applications. Vision systems have historically been expensive and complex. With the increase in microprocessor capability and the expanding commercial imaging market, they have become practical and economical. A significant amount of work has been done in the area of complementary metaloxide semiconductor (CMOS) sensor development. Research is being done in areas of vision sensing, from conventional arrays to sensors, which can monitor motion and implement 3D vision. A vision sensor can be used to extend the robot's control capabilities. 2D or 3D images of objects can be extracted from their environment with the vision system, then information or model can be reconstructed from these images to control the robot. The control that uses the outcomes, like images or a reconstructed 3D model of the vision system as the feedback signals, is known as vision-based control.

Force sensors are usable with an end effector to manipulate the work object. Force sensors have been used for tasks like complex assembly and manipulating objects that can be easily damaged by excessive force. Those sensors have six degrees of freedom (DOF) and can get force feedback for strength and orientation. The force feedback has been integrated into the control of the robot.

Z. Gan et al., *Visual Sensing and its Applications*

1.1 3D Acquisition Techniques

In general, for vision-based robot control, the purpose of the vision system is to acquire 3D information of the object of interest. There are many techniques that can be used to acquire 3D information of the objects. They are also referred to as range sensing technologies. The hierarchy of range sensing technologies is indicated in Fig. 1.1.

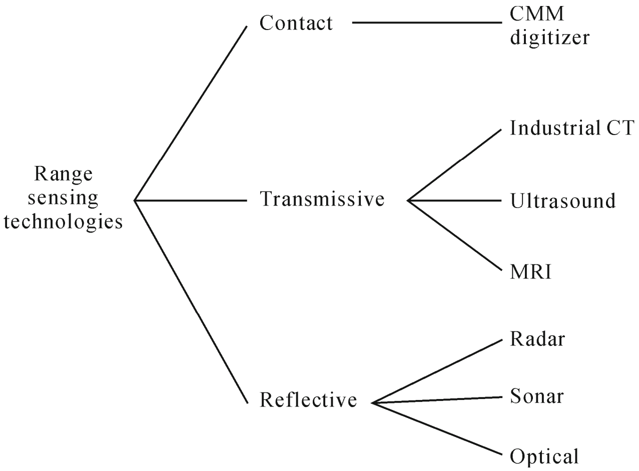


Fig. 1.1. Hierarchy of range sensing technologies

There are two different kinds of sensing technologies (Everett, 1995): contact and non-contact (transmissive and reflective). Contact sensors are typically touch probes that come in a wide range of accuracies as well as costs. Coordinate measuring machine (CMM) is extremely accurate and very expensive, being a standard tool for shape measurement in industry. The main drawback is the contact with the surface, which may be undesirable for fragile objects. Active non-contact methods usually operate by projecting energy waves onto the object of interest and by processing the transmitted or reflected signal. The energy waves can be X-rays for industrial computer tomography (CT) technology, sound energy for sonar sensors, microwave energy for radar. In terms of non-contact range sensing technologies, Blais (2004) gives an overview. Optical methods and sensor hierarchy are summarized in Fig. 1.2.

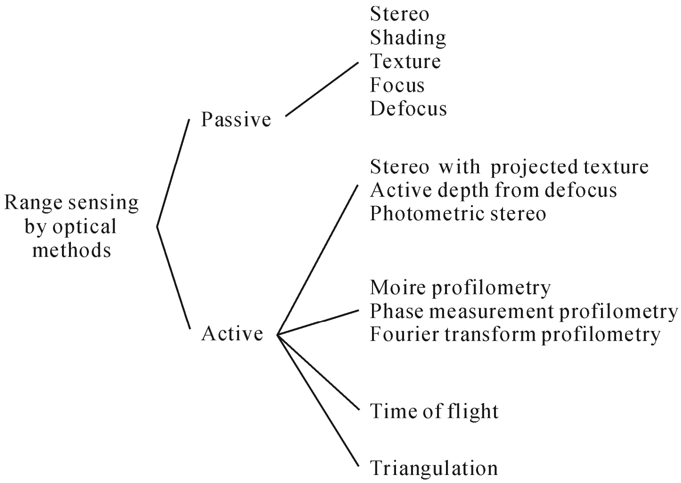

Range sensing by optical methods
Passive
Stereo
Shading
Texture
Focus
Defocus
Active
Stereo with projected texture
Active depth from defocus
Photometric stereo
Moire profilometry
Phase measurement profilometry
Fourier transform profilometry
Time of flight
Triangulation

Fig. 1.2. Optical range sensor hierarchy

An active optical method is our concern in applications. For this category, a light pattern is projected onto an object in a structured manner, and the shape of the object is determined by measuring the deformation of the reflected pattern. In contrast to passive and non-optical methods, many active optical range sensors can provide a range of dense and highly accurate data. They are less sensitive to the ambient light and environmental noise. They are good candidates for use in conjunction with industrial robots.

In the following, a brief introduction will be given to the commonly used 3D vision technologies that can be used together with the robot to form a robot vision system.

1.1.1 2D Vision

2D vision is a relatively well-developed vision technology and has been seen many successful industrial applications in the past three decades. Today, most of the installed machine vision systems are 2D vision like charge couple device (CCD) cameras and they are the most affordable vision products. 2D vision is used to identify and locate features or artifacts in video images. The main limitation of 2D vision is its lack of ability to determine the depth information of parts. 2D vision systems are only capable of determining the *X*, *Y* coordinates and planar rotation of parts. This is insufficient for some industrial applications.

To get depth information 2D, one-half-dimensional (1.5D) vision is introduced that typically employs scaling techniques to estimate distance. The system adds other dimensions to the 2D system by calculating the change in size of the feature part. The change in size is related to the change in distance due to the perspective projection. However, this technique cannot be used to determine rotations out of the part plane. These limitations and the inherent inaccuracy of depth estimation limit its applications.

The type of 2D vision sensors used with the robot system depends on the applications. One typical example is the pick and place application where the camera or cameras are mounted above the robot workcell to identify and locate the parts so that the robot arm can pick and sort the parts. For example, FlexPicker of ABB[1] can be used to pick up cookies on the conveyor based on the location information from the cameras mounted above the conveyor, as shown in Fig. 1.3.

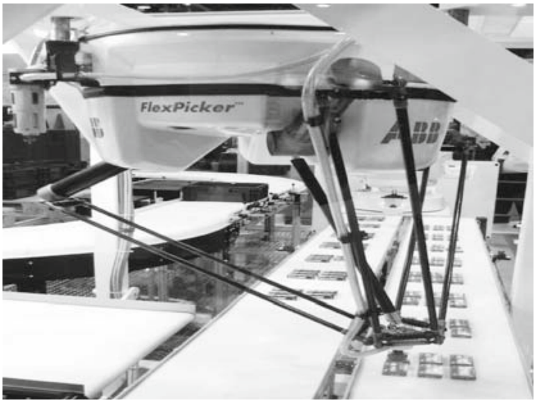


Fig. 1.3. The FlexPicker designed for industrial pick and pack of small lightweight objects, such as chocolates, with high speed. The cameras (not shown here) are integrated with the robot to identify and locate the object to be picked up

1.1.2 Stereo Vision

Another popular 3D vision technique is stereo vision (Yakimovsky, 1978). This is for calculating the depth of features on a given object relative to the sensor. The depth information from multiple features can then be used to create a model of the object or determine the 3D pose of the object. Stereo vision uses images from dual cameras aimed at the same object and finds common features in both images called correspondence. Based on the geometrical relationship between the two cameras and the location difference of each feature point in both images, the depth of each feature can be calculated and a depth map can be constructed from various feature points.

The stereo vision principles and algorithms have been around for decades. The challenge, however, is the successful identification and location of corresponding

[1] http://www.abb.com/product/us/9AAC910011.aspx

features in both images of dual cameras (Faugeras, 1993). If the object has no obvious features, like a uniform wall for example, the stereo vision will not work in this case. Another problem associated with stereo vision is that the depth calculation is very sensitive to the measurement error of the feature image. That means a small error in the position of the feature in one or both images results in large errors in the depth measurement. This sensitivity has severely limited the use of stereo vision algorithms in high accuracy industrial applications.

In order to determine the pose of an object, at least three feature points on the object have to be located to construct a coordinate frame. Due to the sensitivity of stereo vision to feature position errors, multiple stereo pairs are often used to minimize errors. Applications with stereo vision typically have a large standoff distance between cameras and the part, and the accuracy requirement is not very high.

Stereo vision is widely used in mobile robots for navigation and also has applications in industrial robots.

1.1.3 Time of Flight

Long-Range sensors with a measurement range exceeding 10 m are usually based on the time-of-flight (TOF) technology. These types of sensors have been used on airborne mapping systems (Baltsavias, 1999) and the reconstruction of buildings (Lange, 1999). The distance from the sensor to an object (z) is measured by sending a relatively short impulse of light on a reflective surface and measuring the time of travel (t), that is $z = ct / 2$, where c is the speed of light. If one wants a resolution of 1 mm, then a time delay of about 3.33 ps needs to be measured. Most commercial systems provide a resolution of about 10 mm to 100 mm. For short-range applications this is not satisfactory.

Other systems based on continuous wave (CW) modulation can be used to measure short distance by modulating the power of the laser beam. The modulated signal is projected onto a surface, and the reflected beam is analyzed with a circuit to measure the phase difference between the two waveforms and hence a time delay. The range z is given by $z = \pi c \Delta \phi /(4 f_{AM})$, where $\Delta \phi$ is the phase difference and f_{AM} is the modulation frequency. For example, with a frequency of 5 MHz and a phase resolution of 0.01°, the resolution in z is about 1 mm.

Because the returned signal cannot be associated with a specific original signal, the absolute distance information from a simple CW method cannot be calculated (Figueroa, 1992). The uncertainty is given by $za = c / (2 f_{\mathrm{AM}})$. In the example above, the uncertainty interval is about 30 m.

Based on the TOF principle, a commercially available laser measurement sensor LMS from SICK can be installed in the robot's gripper tool in order to position the gripper.

Recently 3D cameras based on the TOF principle become commercially available in the market[1]. They provide a depth map of the object or the scene of interest. They operate with modulated visible and near-infrared radiation, which is detected and demodulated simultaneously by a 2-D array of lock-in pixels employing the charge-coupled device principle. Each pixel individually measures the amplitude, offset and phase of the received radiation. Depending on the distance of the target, the captured image is delayed in phase compared to the originally emitted light wave. By measuring the phase delay in the array the depth map is determined (Lange, 2001; Oggier, 2003).

1.1.4 Laser Triangulation Sensor

This technology uses a laser triangulation sensor to determine the depth of the object of interest (Jarvis, 1983). The sensor typically consists of a CCD or CMOS camera and a low-power laser diode. The more sophisticated versions of the laser triangulation device use scanning lasers or laser stripe emitters that project a plane onto the surface of the object. The laser plane intersects the surface in a line and its deformation due to the uneven surface can be analyzed to determine the depth information of the surface. Laser triangulation technology has been successfully used for robotic applications, like seam tracking for welding, glue deposition, grinding, water jet cutting, part localization and de-burring of flexible and dimensionally unstable parts. Other applications include coordinate measurement and verification of dimensions.

1.2 Structure of Robot Visual Control System

An example of the robot vision sensing system is for a robot arm to acquire an object from a pallet. The exact location of the object is unknown. Therefore, the robot cannot be programmed with knowledge of the object position. In this case, the task environment is called unstructured. To achieve this task, a vision sensor like a camera or laser sensor is mounted on the robot arm to provide the visual sensing capability. The information acquired by the sensor is processed by a vision system in order to identify the object and the relationship between the spatial position of the object and the sensor position. Such a relationship is used to guide the robot to acquire the object from the pallet.

There are two types of robot control mode using a vision feedback loop. One is analogous to “look then action” and the other is analogous to “look and action”.

[1] http://www.mesa-imaging.ch/ http://www.baumeroptronic.com/

Open Loop Robot Control

In the “look then action” mode the response speed is not important. Once the vision sensor is to determine the pose of an object, the control sequence of the robot is generated based on the calculated visual information. Two actions are separated and no on-line interaction between the robot and the environment exists. The measurements of the object and the robot action are done only once. In order to make the robot action work properly, different levels of calibration are critical. To determine the pose of the object, the model of the object must be available and the sensors have to be calibrated with respect to the robot. The robot direct and inverse kinematic models have to be available and kinematics parameters need to be calibrated. The robot can then execute the task by performing planned movements which assume that the environment remains static.

Closed Loop Robot Control

In the “look and action” mode, the response speed of the control system is critical. It has the capability to deal with the realtime change of the relative part position to the robot. This control mode is referred to as “visual servoing”, indicating the realtime visual feedback control of the robot. The measurement of the object is repeated and the result is updated for each robot control cycle. It is not critical to have all the components of the robot visual system calibrated precisely. Fig. 1.4 illustrates the different types of robot control by using the visual feedback.

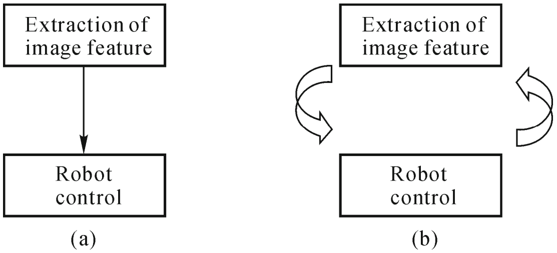


Fig. 1.4. Robot open loop and closed loop controls. (a) In the open loop control, the robot control sequence is generated after the feature of the image is extracted; (b) In the visual servoing control, on-line interaction between the robot and the environment is repeated and updated

Two basic approaches to visual servoing are introduced (Sanderson, 1980). They are position based visual servoing (PBVS) and image based visual servoing (IBVS). A tutorial and review of visual servoing can be found in (Hutchinson, 1996; Chaumette, 2006; Chaumtte, 2007).

In PBVS techniques, a 3D pose of a specific object is acquired and used to control the robot movement in 3D space. To get 3D information of an object, normally a vision sensor (i.e., camera) is mounted on the robot arm and the position and orientation of the object relative to the sensor is retrieved. In pick-and-place systems, for example, the sensor can be used to acquire the position and orientation (pose) of a part. This information is then transferred to the robot controller, which can pick and place the object. The structure of PBVS is shown in Fig. 1.5.

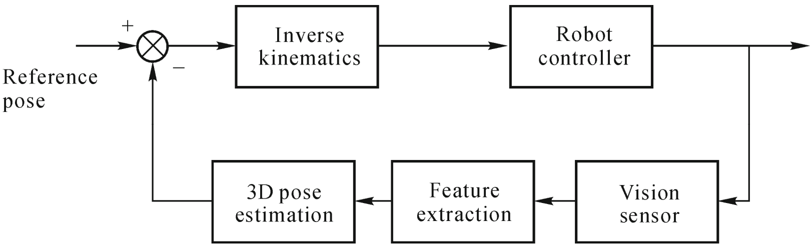


Fig. 1.5. Block diagram of PBVS. The pose of a target is measured and compared with a reference pose. The robot is moved to minimize the pose difference

Sensors used in PBVS include stereo cameras, laser structured light sensors and other range sensors (Blais, 2004) that can provide pose information of the object.

In IBVS, based on the conventional definition, 2D information of the image is directly used to control the robot movement to reduce the image distance error between a set of current and desired image features in the image plane. Extending the concept the feature may not come from images, it can come from any vision sensors. The structure of IBVS is shown in Fig. 1.6.

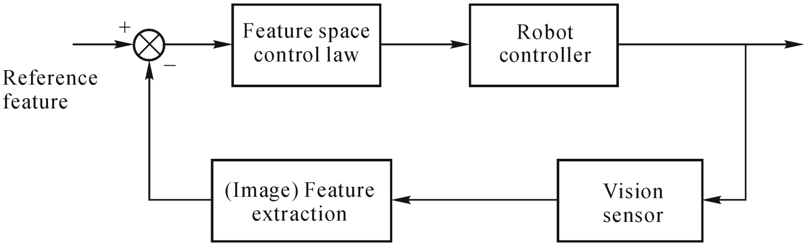


Fig. 1.6. Block diagram of IBVS. The target (image) is acquired and the (image) feature is extracted and compared with a reference feature. The robot is moved to minimize the difference

Integrating both servoing approaches, a third method known as hybrid or 2.5D visual servo control is introduced to decouple the rotation and translation component of the transformation matrix, to improve the stability and reliability of the algorithm (Malis, 1999; Corke, 2001). By using image data via PBVS techniques, the rotation of the object can be calculated. Then by using IBVS to generate translation information, rotation and translation information are effectively decoupled, ensuring more accurate robot positioning. To perform PBVS, IBVS, or 2.5D visual servoing, a number of different sensing technologies can be used.

1.2.1 Structured-Light Sensor Based Visual Control

Conventional visual servoing is based on a camera to acquire the information of an object target. This concept can be extended to include any sensor that can acquire either 3D information or feature of an object. In IBVS, the key point is to get depth information of the object. This can be implemented by using a number of 3D acquisition techniques: stereo vision, time-of-flight, single camera with multiple view and structured light sensor.

By using a single camera or dual cameras that are mounted on a robot arm, applications such as product sorting and pick-and-placement of parts can be implemented. When the object is lacking in features, like a uniform surface, both PBVS or IBVS by using the camera fail, since no feature is observed. In this case, a structured light based sensor is used to create features on the object surface (Pages, 2005). This can be done by using, for example, a laser stripe sensor. In operation, the laser stripe sensor projects a single light stripe over the part to be examined. By moving either the laser sensor or part, the structured light profiles can be digitized and used to create a 3D model of the part.

1.2.2 Selection of Industrial Robots

As we discussed previously, there are various methods and technologies to get 3D information and the pose of an object. There are also various industrial robots that vision systems can work with. The simplest one may be a 2D vision based actuator to sort parts. Industrial robots are generally classified based on the number of degrees of freedom in which they can operate. In three axes Cartesian robots, motion is limited to the *X*, *Y* and *Z* directions. Another popular industrial robot is called the selective compliance assembly robot arm (SCARA) robot. The robot has compliance only in specific directions (*X* and *Y* directions) and has high rigidity in another direction (*Z* direction). It has been designed mainly for automation in assembling objects. If any point and orientation in 3D space needs to be reached, a six degrees of freedom (6 DOF) robot is required that consists of six axis. Such robots are commonly used for applications such as welding, palletizing and complex part assembly. Most of the current industrial robots are of the six-axis type. Mounting six-axis robots on a track enables them to be moved to various positions on the factory floor. In some applications an external axis is added to the robot workcell to rotate the work object so that the entire part of the object can be reached by the robot arm. Because of this added flexibility, the systems are known as seven-axis robots. Fig. 1.7 shows three typical robots.

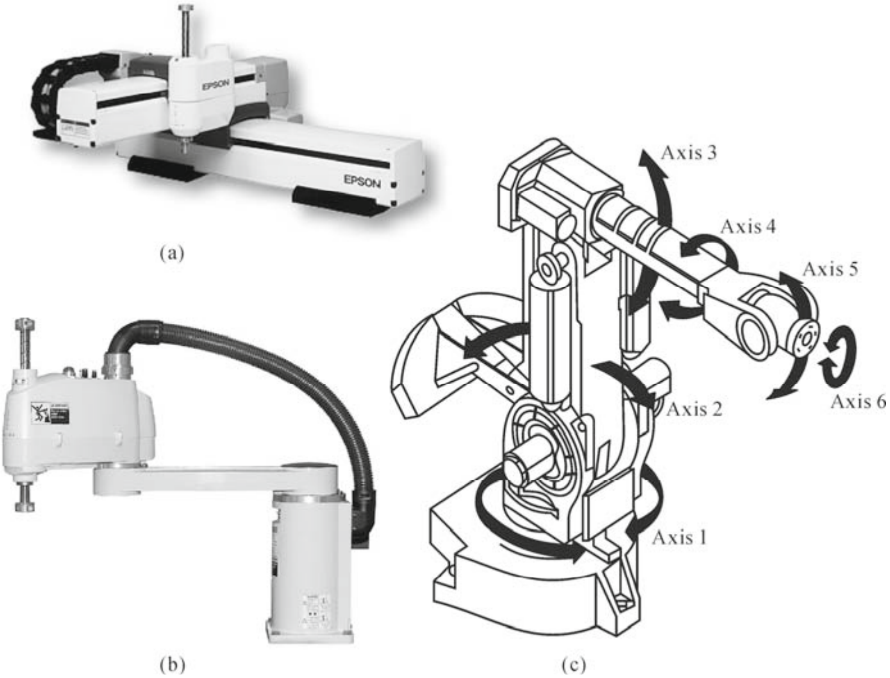


Fig. 1.7. Three types of industrial robot. (a) Cartesian coordinate robot; (b) SCARA robot; (c) Multiple joint robot

In terms of the relative position of the vision system, the robot visual system can be categorized as an eye-in-hand and an eye-to-hand system. In the eye-in-hand configuration the vision system (for example camera or laser scanner) is mounted on the robot arm as the robot end effector and moves along with the robot. The work object of interest is placed at a fixed position so the measurement is conducted by swinging the robot arm. The tool center point (TCP) for the vision system to be calibrated is called the moving TCP. In the eye-to-hand configuration the vision system is placed in a fixed position in the robot workcell and the work object to be measured is held by the robot arm. The measurement is done by moving the work object through the laser scanner. And the TCP to be calibrated for the vision system is called the fixed TCP (Fig. 1.8).

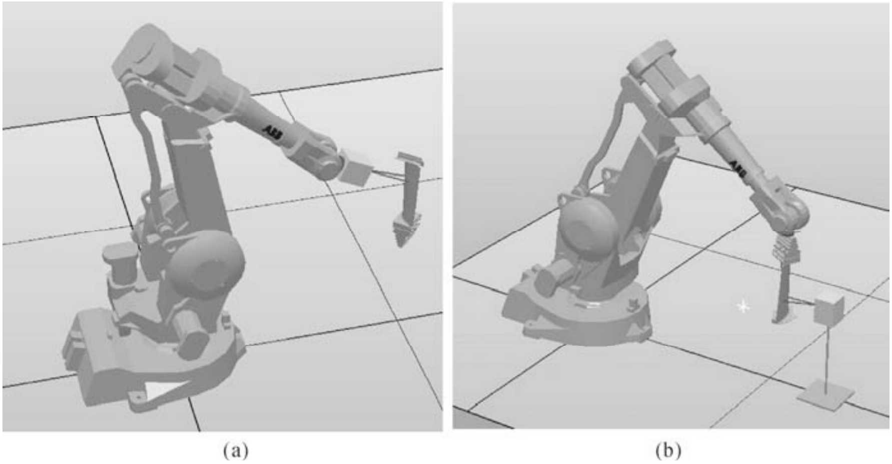


Fig. 1.8. Robot visual system with a robot manipulator and a 3D laser scanner. (a) Eye-in-hand configuration where the laser scanner is held by the robot arm and the object to be measured is placed at a fixed location; (b) Eye-to-hand configuration where the laser scanner is placed at a fixed location inside the workcell and the object to be measured is held by the robot arm

1.2.3 Applications of Robot Visual Systems

Many of the current applications of machine vision are inspection tasks that do not involve the use of an industrial robot. The system is installed on a high-speed production line to either accept or reject finished parts by some mechanical device that is communicating with the vision system. The combination of robot and vision system propels the automation to a higher level of reliability and flexibility. Robot vision falls into three categories listed below:

(1) Measurement and inspection;
(2) Identification and localization;
(3) Visual control and visual servoing.

Measurement and Inspection

The robot is used to position the vision system to perform measurement and inspection tasks including, for example, checking for gross surface defects, discovery of flaws, verification of the presence of components in assembly, measuring for dimensional accuracy and checking for the presence of holes and other features in a part. In this type of application the robot works as a CMM.

Identification and Localization

A robot vision system is used to recognize and classify an object rather than to inspect it. This process involves the determination of the part itself and its position and/or orientation. This is a process that is usually followed by a subsequent robot action to form robot servoing applications. This process includes part sorting and palletizing.

Visual Control and Visual Servoing

The robot is controlled based on the feedback of the vision system. This is called visual control. When the control is a closed loop and is usually in real time it is termed visual servoing. One general application of visual servoing is when the vision system is used to control the trajectory of the robot's end effector toward an object in the workspace. Applications include part positioning, retrieving and reorienting parts moving along a conveyor, assembly, bin picking and seam tracking in arc welding. In these applications, visual data are the main input data to the robot controller and a great deal of intelligence is required in the controller to apply the data for navigation and collision avoidance.

1.2.4 Calibration of Robot Visual Systems

There are two main tasks for robot visual systems: one is the identification or localization of the object by using the vision sensor and the other is the action of the robot. For the identification or localization process, the object 3D pose relative to the robot should be measured accurately. An efficient calibration routine needs to be performed to compensate for robot kinematics errors and vision system errors.

There are three types of calibrations for the robot visual system. In order for a robot to use the vision system to measure the 3D pose (X, Y, Z, raw, pitch, yaw) of an object relative to its own base, it is necessary to know the relative pose between the hand and the robot base, between the vision sensor and the hand, and between the object and the vision sensor. These three sets of poses can be obtained accurately with trio calibrations of the system:

(1) Vision sensor calibration (analogous to eye calibration);

(2) Vision to robot TCP calibration (analogous to eye-to-hand calibration);

(3) Robot kinematics calibration (analogous to hand calibration).

Fig. 1.9 indicates the relationship and scope between those three calibrations.

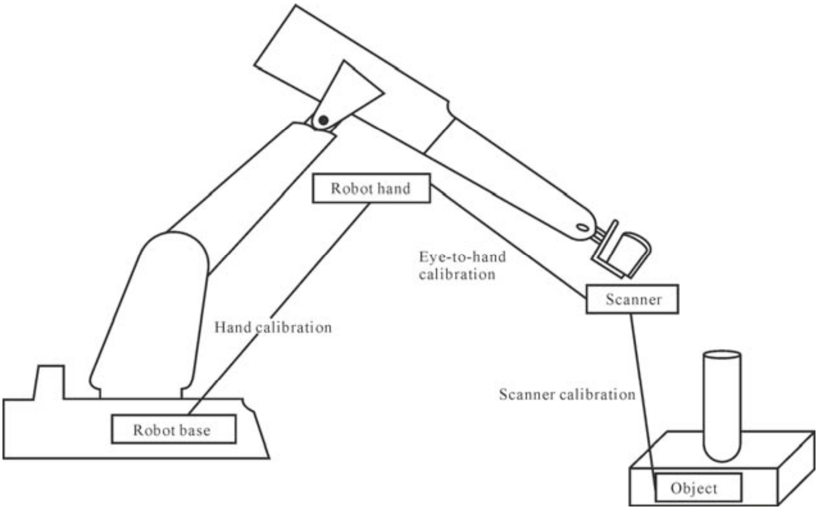


Fig. 1.9. To obtain the accurate pose of the object in the robot base frame, three calibrations are needed in the robot visual systems

Vision sensor calibration

The purpose of sensor calibration is to identify its intrinsic and extrinsic parameters. Extrinsic parameters contain information about the sensor position and orientation relative to a reference coordinate frame. The physical meaning of intrinsic parameters varies, depending on the mathematical model of the sensor and sensor type. In this book a laser stripe sensor is the main concern. Since the laser stripe sensor consists of a laser projector and a camera, camera modeling and calibration become an integrated part of laser stripe sensor calibration procedure.

Vision to Robot TCP Calibration

Vision to robot calibration, referred to as robot TCP calibration, is used to determine the position and orientation (i.e., pose) of the vision sensor with respect to the robot end point (called the robot mounting flange) or the robot base.

Robot Kinematics Calibration

Robot kinematics calibration is used to enhance robot positioning accuracy through software rather than by changing the mechanical structure, to compensate for mechanical variations and inconsistency of the robot in the manufacturing process.

1.2.5 Laser Sensor Based Commercial Robot Visual Systems

The laser stripe sensor or laser scanner technologies have been studied over the last few decades. Laser sensor products are available from numerous companies such as Perceptron, Hexagon, LDI and SICK. Nowadays, laser scanners can accurately capture millions of points within seconds. They are widely used as measurement instruments in various applications such as reverse engineering, rapid prototyping and quality control/inspection, where the sensors are usually mounted on a coordinate measurement machine, a measuring arm or other measurement device to extend the working space.

Besides being used in measurement systems, laser sensors can also be integrated into robotic systems (or some other automatic systems) to enhance the system performance. Visual instruments, like cameras, have been used in robotic systems to help locate work pieces. Compared with 2D cameras, laser sensors can not only provide 3D data but can also be less sensitive to environmental conditions such as lighting.

The integration of the laser sensor in a robotic system is generally case by case, depending on the requirements of the system. Some laser sensors can provide the measurement data to the high level software for secondary development through API, Ethernet communication, serial communications or other interfaces. Some laser sensors have also been developed for more specific purpose. Examples include AutoFit and AutoGuide systems from Perceptron[1], iRVision from FANUC[2], Meta seam tracking system from Meta Vision Systems[3], RobPal and Servo Robot seam tracking system from Servo Robot[4], RobotScan from InterSmart[5]. AutoFit systems can be used to detect and remedy fit and misalignment problems that arise in vehicle body assembly. AutoGuide, iRVision and RobPal can be used to visually locate work pieces through measurement of specific features or simple geometries like holes and corners. Meta and Servo Robot seam tracking systems can be used in the welding process for realtime compensation of position error caused by part loading, thermal distortion, etc. RobotScan is capable of identifying and locating the freeform work object without obvious geometrical features by using a global registration algorithm. Laser sensors have promising prospects in automation applications.

[1] http://www.perceptron.com/index.php/en/-industrial/gap-and-flush.html

[2] http://www.fanucrobotics.com/products/intelligent-solutions.aspx

[3] http://www.meta-mvs.com/robotic.htm

[4] http://www.servorobot.com/en/manufacturing-solutions/material-brhandling

[5] http://www.inter-smart.com/display.asp?ibws=10

1.3 Outline of Chapters

Block diagram of chapter framework is shown in Fig. 1.10.

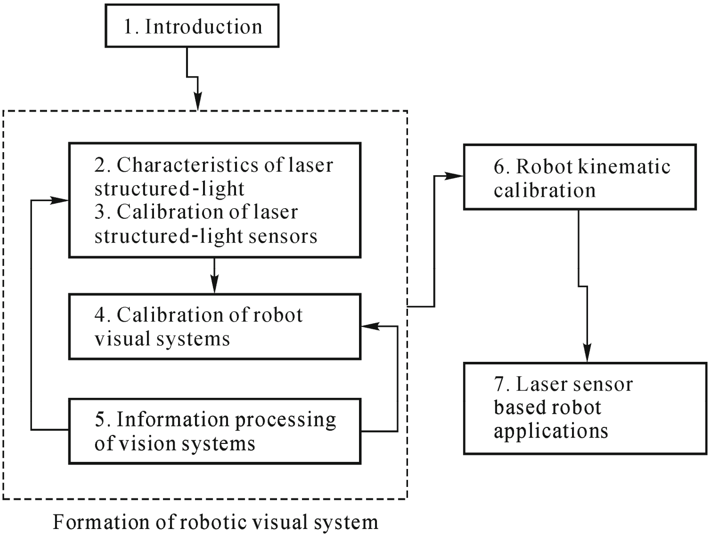

1. Introduction
2. Characteristics of laser structured-light
3. Calibration of laser structured-light sensors
4. Calibration of robot visual systems
5. Information processing of vision systems
6. Robot kinematic calibration
7. Laser sensor based robot applications
Formation of robotic visual system

Fig. 1.10. Block diagram of chapter framework

Laser structured-light sensors have been widely used in industrial robots for various applications like on-line programming, parts measurement and quality control, parts identification and localization, etc. In Chapter 2, some fundamental issues relating to such laser sensors are addressed. We first show how a laser structured light sensor is formed in Section 2.1. It is followed by a discussion on the limitation of accuracy and the effect of laser speckle and environmental factors on the measurement accuracy in Section 2.2. Finally, we list the commercially available laser structured light systems in Section 2.3.

In Chapter 3, we introduce laser stripe sensors calibration. In Section 3.1, two general models of the laser stripe sensor are presented and the calibration strategy for both models is discussed. In Section 3.2, camera modeling is introduced. It includes pinhole modeling and nonlinear modeling with consideration of lens distortion. In Section 3.3, algorithms and implementation of various camera calibration methods are presented and discussed. It includes conventional direct linear transform, Tai's RAC based algorithm and Zhang's flexible multiple view calibration algorithm. In Section 3.4, calibration of laser stripe sensors with various techniques, algorithms and implementation is presented. All kinds of calibration techniques are summarized in a table for comparison, in terms of the form of a calibration target, methods for extracting control points for camera calibration and laser plane calibration, and the algorithm and mathematical model used.

In Chapter 4, we first address a general model of the TCP calibration problem and its solution in Section 4.1. It shows that the general mathematical model of TCP calibration is a robot kinematics in addition to a geometry constraint. The solution to the model is in general a nonlinear optimization process that minimizes the cost function defined in the model. The algorithm can be simplified into linear equations by selecting a specific calibration target and/or having a constrained robot movement to decouple the components of unknown variables. Therefore, the TCP can be solved by using a linear least squares algorithm. The general model can be categorized into two types of TCP calibration approaches. One is for a point sensor-typed tool like a probe pin, a touch trigger probe and a point laser, where only a single point is measured each time. In this case the measurement points on the calibration target are constrained by the geometrical shape of the calibration target and the TCP calibration algorithm is formulated. The other is for an array-type measurement tool like a camera or a laser scanner where multiple points can be measured each time. In this case the calibration target is treated as the reference coordinate frame and the TCP calibration algorithm is formulated. Specific implementations of calibration algorithms derived from the general model with various geometrical shapes like point, line, sphere, plane and structured pattern as the calibration target are discussed in detail. After the discussion of the general TCP calibration algorithm and methodology we will focus on the calibration of laser sensor based tools, which we will be using frequently throughout the book, especially in Chapter 7. In Section 4.2, TCP calibration of a point laser is discussed. The orientation and position of the sensor are calibrated separately. Since a laser scanner consists of a camera and a laser plane calibration of the camera, TCP can be considered as part of a laser scanner TCP calibration process. In Section 4.3, TCP calibration of a camera is discussed with a linear and nonlinear algorithm. In Section 4.4, TCP calibration for a laser scanner with a calibration target of a sphere, plane and structured pattern is presented. In Section 4.5, TCP calibration of a mechanical tool like a spindle is presented by using direct measurement with the measurement tool that has been calibrated previously. In Tables 4.1 and 4.2 we summarize various TCP calibration methods for different types of tools by using various calibration targets as an implementation of the general calibration model.

In Chapter 5, for laser stripe sensors there are two types of image processing tasks. One is to find locations of the control points in the calibration target. This is usually for the calibration procedure. The other is to find the center position of the laser line, which is used for the reconstruction procedure. For the first task, accuracy of the algorithm is essential and for the second task, reliability, sensitivity to the ambient light, accuracy, as well as speed, need to be addressed. In this Chapter, we will also review image processing techniques involved in laser stripe sensors.

In Chapter 6, we first address the general model function of robots in Section 6.1, and then we review the D-H model in Section 6.2. Error budget analysis and error parameters solutions are presented in Sections 6.4 and 6.5, respectively. In section 6.5, detailed discussion of a TAU robot calibration including forward and inverse kinematic modeling with and without all error parameters, and by Jacobian matrix with all error parameters, is conducted. Simulation and experimental results are also presented.

In Chapter 7, various laser senor based robot visual systems and their applications are presented. There are three kinds of applications: measurement and inspection, identification and localization, and visual servoing.

In terms of measurement and inspection applications, in Section 7.1 the method for detecting the position and orientation of holes or channels in a 3D space using a robotic vision system is presented. It includes the approaches and algorithms for detecting the hole position, size and orientation by using a vision system mounted on the robot arms. The hole orientation is determined based on the alignment of the vision system and the hole axis. The position of the hole is the intersection between the hole axis and the surface region around the hole opening. Experimental results have indicated that the concept of cooling hole identification is feasible. It has been shown that the reproducible detection of the cooling channel position has ±0.15 mm accuracy and cooling channel orientation is within ±3° under current test conditions.

In Section 7.2, a robotic grinding system is presented. The freeform workpiece is first measured and located to close the kinematics chain of the robotic system, and then the robotic system error is taken into consideration and well compensated for. A registration algorithm is adopted to locate the work piece, and then two novel error compensation methods, which do not depend on the absolute accuracy of the robot, are proposed to enhance the accuracy of the robotic grinding system approaching the repeatability of the robot. Experimental results are also shown for the effectiveness of the methods.

In Section 7.3, a profile modeling based grinding approach is presented and discussed. This approach is applied to blade tip profile refurbishment in the aerospace maintenance industry. On-line profile calibration and fine-tune methods are adopted to generate an accurate processing path for different parts that are deformed after a few years service under severe conditions. Demo experiments are developed with a robot grinding work cell. A 3D laser scanner and LVDT in the robot workcell are used for measuring fine-tune and on-line quality control. Experimental results indicate that this profile modeling based grinding approach is competent for blade tip refurbishing tasks.

In Section 7.5, a flexible robotic machining system that can compensate for parts shape variations is presented. To compensate for the shape variation error, the measurement result of individual parts is used as the feedback of the robot controller. With the feedback, it becomes possible for the robotic system to generate the machining program in process for each individual work piece. This section will present a visual feedback based robotic solution for work pieces with geometrical shape variation. In Section 7.6, a highly accurate relative measurement robot system is presented which is used to measure the material removal of freeform work pieces (faucets) in the grinding process. The measured material removal data is useful for the adjustment of the grinding parameters for the faucet grinding. The material removal data is calculated based on two measurements of the same surface before and after grinding. Therefore, if the robot programs for these two measurements of the surface are the same, the robot system error involved in one measurement tends to cancel out the other, which makes the measurement essentially a relative measurement.

Regarding identification and localization applications, in Section 7.4 a sand core handling/assembly system is presented, which handles and assembles sand cores in sand boxes to form sand molds for sand casting with an accuracy requirement of ±0.3 mm. In this system, sand cores and sand boxes are loaded on two conveyors respectively, no strict constraints being enforced for the position of parts. The sand cores and boxes are transported to the working positions for a robot to pick-up the sand cores and then assemble them in the sand boxes. Because of the lack of constraints for part loading and the transportation error of the conveyor, a positioning error for the cores/boxes in the working position can easily reach ±50 mm. To compensate for the error, a laser sensor based 3D vision system is integrated in the robotic system to guarantee high accuracy material handling and assembly.

For visual sensing applications, in Section 7.7 a general robotic seam tracking system which can tune robotic poses with 6 DOF is presented and explained in detail, including the architecture of system, the welding joint detection, the path generation algorithm, computer-robot communication, etc. A pipe welding system with a seam tracking system is also introduced as an example of the commercial applications of seam tracking systems.

References

Baltsavias EP (1999) Airborne laser scanning: Basic relations and formulas. ISPRS Journal of Photogrammetry & Remote Sensing, 54:199-214.

Blais F (2004) Review of 20 years of range sensor development. Journal of Electronic Imaging, 13(1):231-240.

Chaumette F, Hutchinson S (2007) Visual servo control I: Basic approaches. IEEE Robotics and Automation Magazine, 13(4):82-90.

Chaumette F, Hutchinson S (2007) Visual servo control II: Advanced approaches. IEEE Robotics and Automation Magazine, 14(4):109-118.

Corke P, Hutchinson S (2001) A new partitioned approach to image-based visual servo control. IEEE Transactions on Robotics and Automation, 17:507-515.

Everett HR (1995) Sensors for Mobile Robots—theory and Application. A K Peters, Ltd., Wellesley, Massachusetts.

Faugeras O (1993) Three-Dimensional Computer Vision. MIT Press, Boston, Massachusetts.

Figueroa JF, Lamancusa JS (1992) A method for accurate detection of time of arrival: analysis and design of an ultrasonic ranging system. Journal of the Acoustical Society of America, 91(1):486-494.

Florczyk S (2005) Robot Vision: Video-based Indoor Exploration with Autonomous and Mobile Robots. Wiley-VCH Verlag.

Hutchinson S, Hager GD, Corke PI (1996) A tutorial on visual servo control. IEEE Transactions on Robotics and Automation, 12:651-670.

Jarvis RA (1983) A perspective on range finding technologies for computer vision. IEEE Transactions on Pattern Analysis and Intelligence, PAMI-1(2):122-139.

Kragic D, Vincze M (2009) Vision for Robots (Foundations and Trends in Robotics). Now Publishers Inc.

Lange R, Seitz P (2001) Solid-State time of flight range camera. IEEE Journal of Quantum Electronics, 37(3):390-397.

Lange R, Seitz P, Biber A, Schwarte R (1999). Time-of-Flight range imaging with a custom solid state image sensor. In: EOS/SPIE International Symposium on Industrial Lasers and Inspection, 3823:180-191.

Malis E, Chaumette F, Boudet S (1999) 2-1/2-D visual servoing. IEEE Transactions on Robotics and Automation, 15:238-250.

Oggier T, et al (2003) An all-solid state optical camera for 3D-real-time imaging with sub-centimeter depth-resolution (SwissRanger). In: Proc. SPIE 5249: 634-645.

Pages J (2005) Assisted visual servoing by means of structured light. Ph.D thesis, Universitat de Girona.

Sanderson A, Weiss L (1980) Image-Based visual servo control using relational graph error signals. In: Proceedings of the IEEE Conference on Robotics and Automation, 1074-1077.

Tayor G, Kleeman L (2006) Visual Perception andRobotic manipulation: 3D Object recognition, Tracking and Hand-Eye Coordination. Springer.

Xu D, Tan M, Li Y (2008) Visual Measurement and Control for Robots. National Defense Industry Press, Beijing (In Chinese)

Yakimovsky Y, Cunningham R (1978) A system for extracting three dimensional measurements from a stereo pair of TV cameras. Computer Graphics and Image Processing, 7:195-210.

2

Characteristics of Laser Structured-Light Sensors

Laser structured-light sensors have been widely used in industrial robots for various applications like on-line programming, parts measurement and quality control, parts identification and localization, etc. In this chapter we address some fundamental aspects of the laser structured-light sensor. We first present how a laser structured light sensor is formed. It is followed by a discussion on limitations of accuracy and the effect of laser speckle and environmental factors concerning measurement accuracy. Finally, we list the commercially available laser structured-light systems.

2.1 Formation of Laser Structured-Light Sensors

A basic structured-light vision system consists of one or multiple cameras and one projector that projects optical patterns. Unlike passive stereo, which uses two cameras, a structured-light system generates dense reconstructed points by locating image points on each light pattern in the image. Use of the structured light avoids the so-called correspondence problem occurring in the passive stereo vision. The structured-light sensor becomes an attractive method for many shape measurement tasks (Jarvis, 1983). The projector can project white light patterns and laser patterns. With the laser patterns the system is called a laser structured light sensor, which is the focus of the discussion in this chapter. For simplicity we also call it a laser sensor sometimes, without ambiguity. In terms of the laser projected patterns the system normally projects a laser spot, a single laser stripe, and multiple laser stripes as shown in Fig. 2.1. More patterns like a circle (Zhang, 2005), concentric multiple circles and a grid are also used. The projected laser pattern can be generated by laser projectors with cylindrical lenses or hologram lenses.

Z. Gan et al., *Visual Sensing and its Applications*

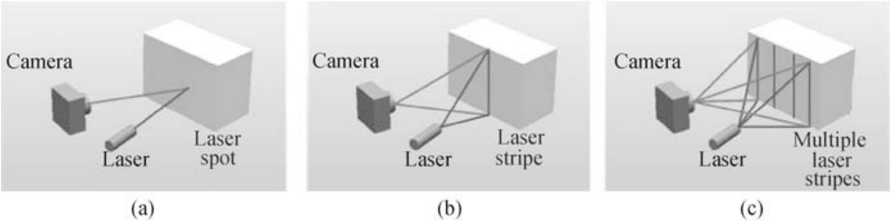


Fig. 2.1. Configuration of laser structured-light sensors with projected pattern of (a) laser point, (b) a single laser stripe, and (c) multiple laser stripes

With the configuration of a laser point sensor, the position of the laser spot which is the intersection point of the laser beam and the object under measurement can be easily identified and measured with high accuracy. However, only one point can be measured at a time. The major difficulty involved in the laser structured-light systems with multiple stripes or more complicated patterns is the ambiguity in identifying light stripes in regions where depth discontinuities occur. To compromise the ease of measurement and measurement capacity, a laser structured light system with a single stripe plane is widely used and investigated. It is usually called a laser stripe sensor. It has the advantages of optical and mechanical simplicity and cost. It is a natural extension of the laser single point sensor, allowing the projection of a laser stripe and the simultaneous detection of a complete profile of points in a single video frame. Deformation of the projected laser stripe will give the range information of the object. In order to measure the whole surface of the object, a laser stripe plane needs to swipe through the surface by using mechanical devices such as galvanometers or robot arms. The system with the scanning capability is called a laser scanner, and the laser stripe sensor is called the laser scanner head (in most literatures it is also called the laser scanner although there is no scanning capability). Recently, the introduction of a low cost CMOS camera with on chip processing unit makes the laser tripe sensor more attractive and cost effective.

Fig. 2.2 illustrates some common terms used in typical triangulation sensors.

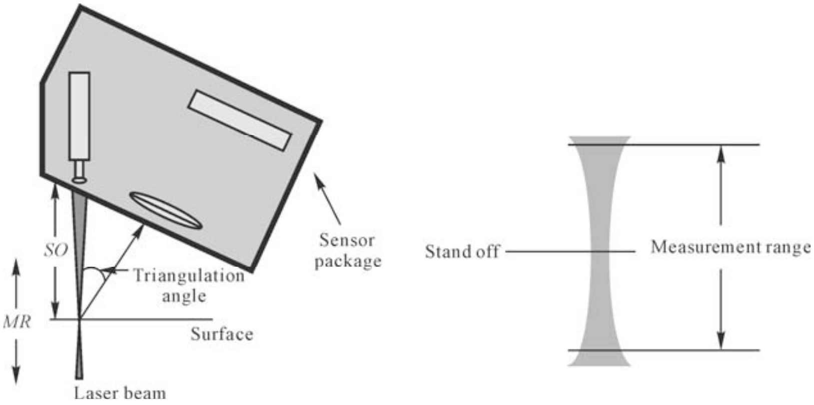


Fig. 2.2. Geometrical terms for the typical triangulation sensor

Standoff Distance

This is the distance from the sensor to the designed measurement position. Normally this is where the waist of the Gaussian beam is located and is where the characteristics of the sensor are optimized. At the standoff distance the spot size is the smallest, providing the highest resolution. Sensors with higher accuracy will have a shorter standoff distance, resulting in a small package size. The sensor package size becomes larger as the standoff increases. For a typical laser triangulation sensor that is integrated with the robot system, the standoff distance is from 50 mm to 300 mm. For a large standoff distance the vision system has a better clearance and a big field of view. However, the resolution and accuracy of the sensor is decreased.

Measurement Range

This is the range over which the sensor gives a valid distance output signal. The standoff distance is around the middle of the measurement range. It is determined by the combination of the sensor structure and the characteristics of the laser beam and imaging lens. To have a large measurement range, the laser beam needs to have a large depth of view and the Scheimpflug configuration is used, in which the detector plane is at a tilted angle to the imaging plane. Normally, the sensor with a large standoff distance has a large measurement range. The measurement range for a typical laser sensor is within 100 mm.

Triangulation Angle

This is the angle between the laser beam and the optical axis of the imaging lens. In general, as the angle increases, the measurement range of the sensor decreases and the resolution increases. In practice, the laser triangulation angle may be as low as 10° for a low-resolution sensor and up to about 45° for a high-resolution sensor. The resolution of triangulation sensors improves with a smaller measurement range.

2.1.1 Light Source

The light source used in the laser structured-light sensor is obviously a laser source although a conventional light source can be used for a general structured-light sensor. The advantage of the laser source is its brightness, coherence and compactness. Strong brightness makes the projected patterns immune to the interference of ambient light. The coherence of the light source means the beam is in focus when projected on the object under measurement. Most laser structured-light sensors use solid state laser diodes as the source, similar to the type used in the common laser pointer. The laser diode provides a compact, efficient, long-life light source for sensors. Laser diodes also emit light in a narrow band of frequencies or with one colour. This property can be used to filter out environmental noise when a narrow band pass optical filter is placed in front of the detector. The filter allows only light of the laser wavelength to reach the detector, and blocks other wavelengths, reducing position detection errors caused by stray light from ambient conditions. Laser diodes can be operated in two modes. One is continuous mode and the other is modulated or pulsed mode. Using a modulated laser can be useful in reducing ambient light by filtering the detector output at the modulation frequency.

The spot size projected on the object surface is determined by the beam width at the point of intersection. The beam width depends on the distribution of irradiance, which is described by Gaussian distribution as

$$I(r,z) = I_0 \exp\left(-\frac{2r^2}{w(z)^2} \right) \tag{2.1}$$

where I_0 is the irradiance of the beam along the central axis, r is the radial distance perpendicular to the central axis, $w(z)$ is the beam radius at distance z from the beam waist w_0 (Williams, 1993) as defined by

$$w(z) = w_0 \left[1 + \left(\frac{\lambda z}{\pi w_0^2} \right)^2 \right]^{1/2} \tag{2.2}$$

where beam waist can be determined by the Rayleigh criterion

$$w_0 \approx \frac{1.22\lambda f}{D} \tag{2.3}$$

where λ is the laser wavelength, D is the aperture diameter of the laser focusing lens, f is the focal length of the lens. The depth of field or depth of focus is defined by the distance D_f when the beam radius becomes $\sqrt{2}w_0$, that is

$$D_f = 2z_0 = \frac{2\pi w_0^2}{\lambda} =\propto \frac{\lambda f^2}{D^2} \tag{2.4}$$

Eq. (2.4) defines the depth of focus of a laser projector. It indicates that the depth of focus is proportional to square of the focal length, and inversely proportional to the square of the aperture diameter of the projection lens. For a laser stripe sensor that consists of a laser projector and an imaging lens, the depth of focus is the combination of the depth of focus of the laser projector defined in Eq. (2.4) and the depth of focus of the imaging lens.

The maximum number of resolvable volume elements (Vf) along each axis within the depth of focus range is given by Beraldin (2000)

$$V_f = \frac{D_f}{w_0} = \frac{2\pi w_0}{\lambda} \tag{2.5}$$

The beam profile and parameter definitions are shown in Fig. 2.3.

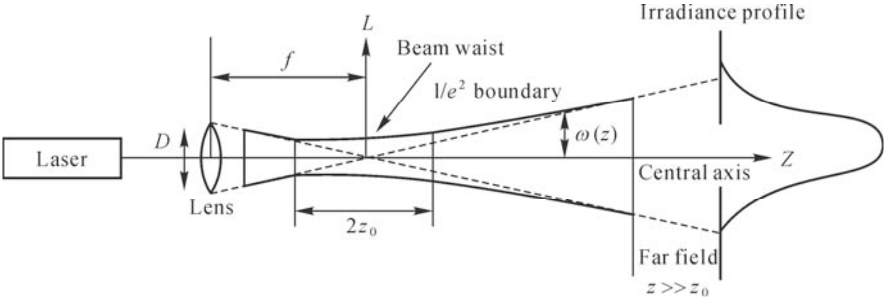


Fig. 2.3. Laser beam profile

The laser plane is created by using a cylindrical lens followed by a spherical lens as shown in Fig. 2.4. The narrow laser beam comes out from a diode laser source and passes through a cylindrical lens and a spherical lens. The beam gets extended to a width of L in Y direction and focused into a width W in X direction at distance S from the lens. The depth of focus is R within which the divergence of thickness of the laser plane is less than $\sqrt{2}W$. The profiles in both X and Y directions are the Gaussian distribution as illustrated in Fig. 2.4. For a commercial laser projector, the thickness (W) of the laser plane is as thin as 0.05 mm.

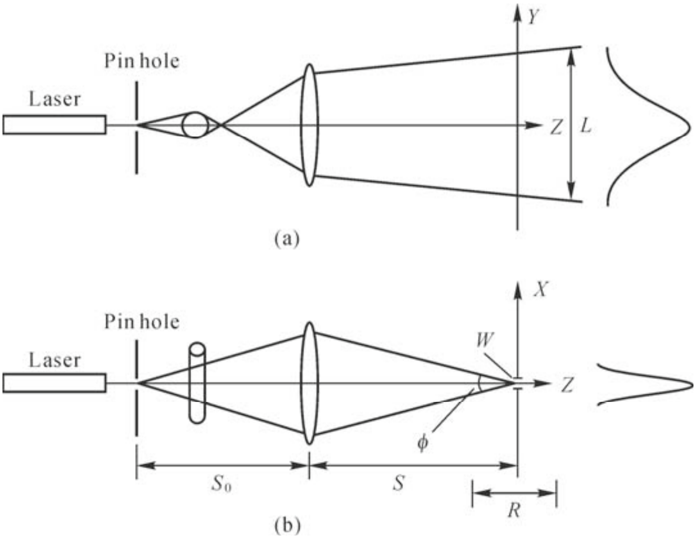


Fig. 2.4. Creation of laser plane. (a) View from *YX* plane (b) View from *XZ* plane. (modified from Fig. 13.2.2 of (Su, 1999), permission granted)

Due to the feature of the Gaussian distribution of the laser beam, the laser line that is expanded from a laser beam by using a conventional cylindrical lens will have the same distribution as shown in Fig. 2.3, which will degrade the performance of the laser stripe sensor since that requires a large dynamic range imaging detector to cover the whole range of the laser line. In order to make uniform distribution of the intensity along the laser line, some research has been conducted by using specially designed optics called a beam shaping device (Dickey, 2000). After the beam shaping device the Gaussian profile becomes a relatively flat top profile as illustrated in Fig. 2.5(a) compared with the Gaussian distribution as shown in Fig. 2.5(b).

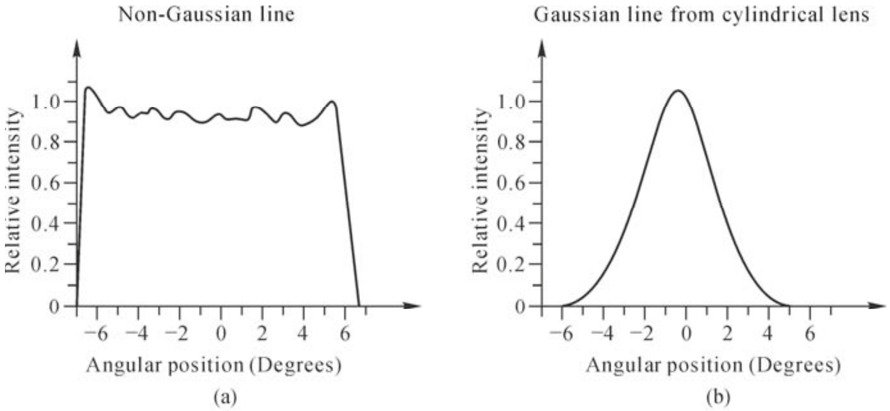


Fig. 2.5. Line intensity profile along line length. (a) Relatively uniform distribution line from beam shaping device; (b) Gaussian line from cylindrical lens

2.1.2 Detector Types

Various methods are available for detecting the position of incident light. These include methods using small discrete detector arrays or multi-element sensors such as CCD and CMOS sensors, and by using analog detectors to obtain continuous position data, such as PSD sensors. Those three types of detectors are fully solid state, and are integrated circuit chips of rugged construction and reliable performance, being suitable for hostile environments.

2.1.2.1 PSD Sensor

A position sensitive detector (PSD) provides continuous position data by making use of the surface resistance of the photodiode. It is essentially an analog device and offers advantages such as high position resolution and especially high-speed response, with data rates of up to 200 kHz or faster. It can be implemented with very fast light level control and has a very good dynamic range. The disadvantages of PSD include lack of ability to display an image or profile of the detector pattern. Also, PSD determines the center of all light that falls on the detector area. If more than one spot or other light falls on the detector, it will report the position of the center of all light, possibly giving an erroneous signal.

The principle of position detection by using a 2D PSD sensor is illustrated in Fig. 2.6. The detector chip has four outputs in both X and Y directions, i.e., X_1, X_2, Y_1 and Y_2. The amount of current from each output is proportional to the position of the imaged spot on the detector. If the spot is centered on the detector, equal currents are seen from opposite outputs. If the imaged spot moves off center, the two opposite outputs change. The spot position x and y can be calculated from the relative values of the outputs

$$
\begin{aligned}
x &= \frac{1}{2}\frac{I_{x_2} - I_{x_1}}{I_{x_2} + I_{x_1}} L_x \\
y &= \frac{1}{2}\frac{I_{y_2} - I_{y_1}}{I_{y_2} + I_{y_1}} L_y
\end{aligned}
\tag{2.6}
$$

where I_{x_1}, I_{x_2}, I_{y_1} and I_{y_2} are the output currents obtained from the electrodes, L_x and L_y are the dimensions of the detector active area.

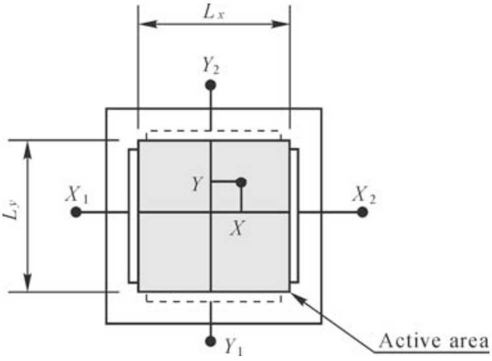


Fig. 2.6. Position detecting principle of PSD sensor

2.1.2.2 CCD Sensor

A charge-coupled device (CCD) is best described as a semiconductor chip sensitive to light. The light sensitive face is rectangular in shape and is subdivided into a grid of discrete rectangular areas called pixels. The incident of a photon on a pixel generates a small electrical charge which is stored for later read-out. The size of the charge increases cumulatively as more photons strike the surface: the brighter the illumination the greater the charge. The principle for detecting the spot position is illustrated in Fig.2.7.

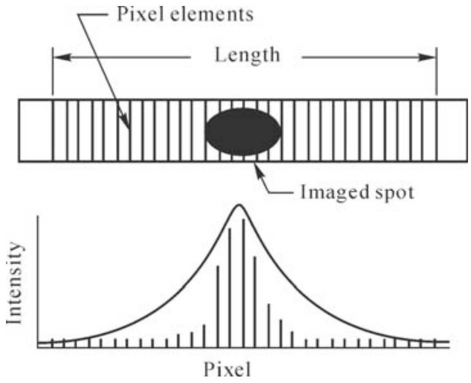


Fig. 2.7. Position detection of CCDs

Early CCDs used in the 1970s often had 64×64 elements. 256×256 or 512×512 element chips were typical in the 1980s and 1,024×1,024 or even 2,048×2,048 elements chips are common now.

CCDs require more post-processing than PSDs do, and the data rates are normally slower. CCDs have a number of unique properties that make them of interest for triangulation sensors. The ability to view the intensity distribution of the imaged spot allows the user to observe and understand the nature of the material or part being examined. The most important advantage of a CCD-based triangulation sensor is the ability to perform post-processing that filters out the noise and makes the system robust. The position of the spot is determined by performing a weighted centroid on the array to obtain sub-pixel accuracy.

2.1.2.3 CMOS Sensor

A CMOS image sensor captures light intensity via an array of photodetectors that are then coupled with an amplifier in order to obtain a high level of charge. Just like in CCDs, the data in each photodetector would then correspond to a pixel in the image output. The advantage of CMOS sensors over CCD is the cheap cost. This low cost of production would then easily translate to cheaper cameras and laser structured-light sensors. CMOS sensors also consume a considerably smaller amount of power compared to CCD sensors. That is why most cameras embedded in mobile phones and laptops have CMOS sensors in them. However, CMOS sensors are less sensitive to light compared to CCD sensors, resulting in lower quality images especially in poor light conditions. CMOS sensors are also more susceptible to noise, meaning that images captured by CMOS sensors are less clean or are grainy. When used in a laser structured-light sensor the CMOS sensor has a similar performance to the CCDs thanks to the advanced image processing algorithm. This makes the laser sensor cost effective and more attractive.

2.1.3 Triangulation Measurement Principle

The measurement principle of the laser structured-light sensor has been discussed (Su, 1999). It can be understood with the simple laser triangulation setup as shown in Fig. 2.8, where *P*(*X*, *Y*, *Z*) is a point in the world coordinate, and *P*(*x*, *y*) is its focused point in the image plane with analogy image coordinate (*x*, *y*). According to the geometrical optics and similar triangles, the coordinates of point *P*(*X*, *Y*, *Z*) can be calculated by relations

$$x = f\frac{X}{Z},\ b + X = Z\cot\theta \tag{2.7}$$

where f is the focal length, θ is the angle between the X axis and the laser light direction and b is the distance between the light source and the lens optical center. From the above equation we can get

$$X = \frac{bx}{f\cot\theta - x} \tag{2.8}$$

Similarly, in the Y direction we have

$$y = f\frac{Y}{Z},\ \frac{X}{Y} = \frac{x}{y} \tag{2.9}$$

yielding

$$Y = \frac{by}{f\cot\theta - x},\ Z = \frac{-bf}{f\cot\theta - x} \tag{2.10}$$

Assuming that the digital image coordinate (*u*, *v*) is related to the analogy image coordinate (*x*, *y*) by

$$x = s_x u,\ y = s_y v \tag{2.11}$$

where s_x and s_y are the pixel size in x and y directions, respectively, we have

$$\begin{aligned} X &= \frac{bs_x u}{f\cot\theta - s_x u} \\ Y &= \frac{bs_y v}{f\cot\theta - s_x u} \\ Z &= \frac{-bf}{f\cot\theta - s_x u} \end{aligned} \tag{2.12}$$

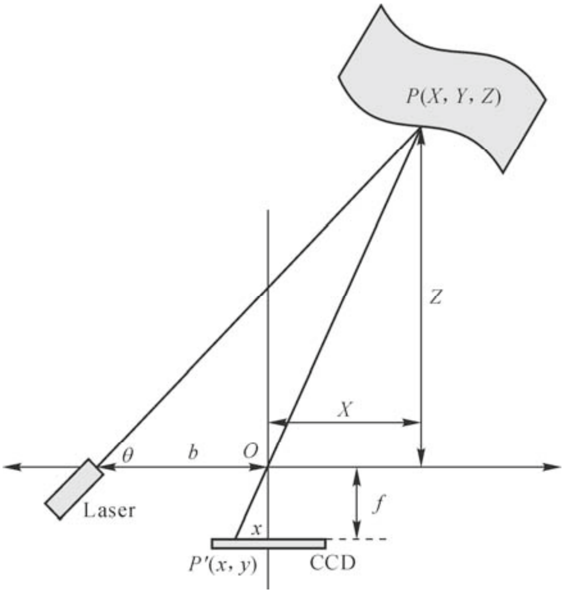


Fig. 2.8. Principle of triangulation measurement

Therefore, given the system parameters $\{b, f, \theta\}$, the 3D position $P(X, Y, Z)$ can be calculated from its image position $p(u, v)$. From Eq. (2.12), it shows that the range distance Z is inversely proportional to the image offset u and the relation between them is nonlinear, yielding a non-uniform measurement accuracy along the Z axis.

Variations of the triangulation setup of Fig. 2.8 with different coordinate system are shown in Fig. 2.9, in which the Z direction is aligned with the laser beam direction. In Fig. 2.9(a), the laser beam is in parallel with the optical axis of the imaging lens. This is the simplest triangulation setup and the measured distance Z can be calculated by using the similar triangles principle that is given by $Z = bf / x$. In Fig. 2.9(b), the optical axis of the imaging lens is aiming at the observation point to extend the measurement range due to the near optical axis configuration. Based on the geometrical relationship we have

$$Z = \frac{lx}{(f \sin\theta + x\cos\theta)} \tag{2.13}$$

where x is the image offset from an reference image point that is corresponding to the space reference point O.

When the measured distance Z changes, the image offset moves along the X direction of the detector. Since the detector plane is perpendicular to the optical axis of the detector, only one image position is in exact focus and the rest of them are out of focus, yielding the reduction in measurement accuracy due to the diffused spot. Both configurations in Figs. 2.9(a) and 2.9(b) limit the measurement range due to the defocus of the imaged laser spot or laser line.

In order to increase the measurement range, a Scheimpflug configuration is used where the detector plane has a tilted angle with respect to the imaging plane, as shown in Fig. 2.9(c).

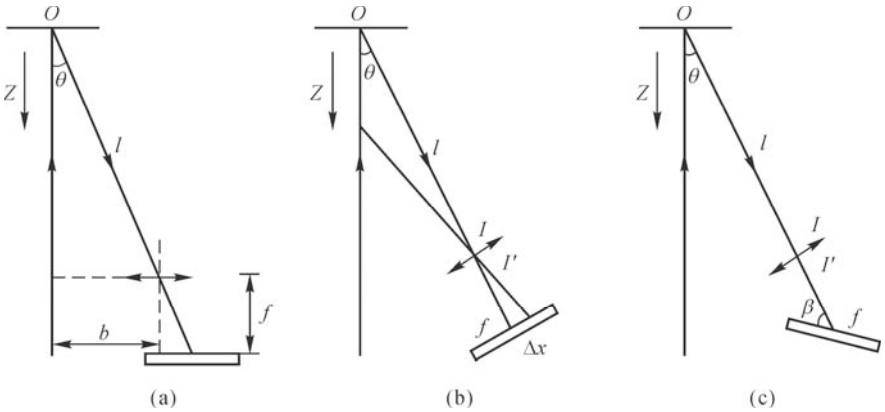


Fig. 2.9. Three typical setups of triangulation based measurement (modified from Fig. 13.3.1 of (Su, 1999), permission granted)

For the largest depth of view of the triangulation system, the Scheimpflug condition needs to be satisfied as

$$\tan\theta = k\tan\beta \tag{2.14}$$

where k is the magnification factor of the imaging lens. This condition ensures the best focus along the X direction of the detector when the object height changes. In this case, one obtains the measured distance

$$Z = \frac{(l-f)x\sin\beta}{(f\sin\theta + x\cos\theta\sin\beta)} \tag{2.15}$$

For a general arrangement between the laser plane and the camera position, the 3D reconstruction becomes complicated and may not be written analytically. The general solution will be based on the camera model and laser plane position relative to the camera, which will be addressed in the following sections.

2.2 Accuracy Analysis

In order to derive the effect of the system parameters on the measurement accuracy, we take derivatives of X, Y and Z to Eq. (2.12), yielding (Wu, 2006)

$$\left.\begin{aligned}
\delta X &= \frac{f\cot\theta s_x b}{(f\cot\theta - s_x b)^2}\delta u \\
\delta Y &= \frac{v s_y s_x b}{(f\cot\theta - s_x b)^2}\delta u + \frac{s_y b}{f\cot\theta - s_x u}\delta v \\
\delta Z &= \frac{f s_x b}{(f\cot\theta - s_x b)^2}\delta u
\end{aligned}\right\} \tag{2.16}$$

From Eq. (2.12) we also have

$$\left.\begin{aligned}
s_x u &= \frac{fX\cot\theta}{b+X} \\
s_y v &= fY\cot\theta
\end{aligned}\right\} \tag{2.17}$$

Substituting Eq. (2.17) into Eq. (2.16) yields

$$\left.\begin{aligned}
\delta X &= \frac{s_x(b+X)^2}{fb\cot\theta} \\
\delta Y &= \frac{Ys_x(b+X)^2}{fb\cot\theta}\delta u + \frac{s_y(b+X)}{f\cot\theta}\delta v \\
\delta Z &= \frac{s_x(b+X)^2}{fb\cot^2\theta}\delta u
\end{aligned}\right\} \tag{2.18}$$

In Eqs. (2.16) and (2.18) δX, δY and δZ indicate the resolution of the laser triangulation system in *X*, *Y* and *Z* directions, respectively. A smaller value means a higher resolution. Similarly, δu and δv indicate the resolution of the digital image coordinate. From Eqs. (2.16) and (2.18) we have the following conclusions:

(1) The resolution of the measurement system in *X* and *Z* direction (δX and δZ) is proportional to the resolution of the digital image coordinate in *X* direction (δu). The resolution of the measurement system in *Y* direction is proportional to the resolution of the digital image coordinate in *X* and *Y* direction (δu and δv).

(2) The resolution of the measurement system in space (δX, δY, δZ) is proportional to the focal length f of the camera. That means the longer the focal length, the higher the resolution, since the long focal length camera provides a narrow field of view.

(3) The resolution of the measurement system in space (δX, δY, δZ) is inversely proportional to the pixel size (s_x, s_y) of the image sensor (for example, CCD or CMOS). That means the smaller the pixel size, the higher the resolution, since the smaller pixel size indicates the finer detector array.

(4) The resolution of the measurement system in space (δX, δY, δZ) is proportional to the distance b between the laser source and the detector. That means the bigger the separation, the higher the resolution.

(5) The resolution of the measurement system in space (δX, δY, δZ) is proportional

to the angle θ between the laser beam and the X direction. That means the bigger the angle, the higher the resolution.

From Eq. (2.7), we have $Z = \dfrac{b+X}{\cot\theta}$. Substituting it into Eq. (2.18) yields

$$\left.\begin{aligned} \delta X &= \frac{s_x(b+X)Z}{fb}\delta u \\ \delta Y &= \frac{s_x(b+X)Z}{fb}\delta u + \frac{s_y Z}{f}\delta v \\ \delta Z &= \frac{s_x Z^2}{fb}\delta u \end{aligned}\right\} \tag{2.19}$$

It can be shown from the third term of Eq. (2.19) that the resolution of the measurement system is not uniform in the whole measurement range. The further from the image sensor (bigger Z), the lower is the resolution.

2.2.1 Effect of Laser Speckle Noise on the Measurement Accuracy

From Eq. (2.19) we know that the system measurement resolution or uncertainty is proportional to the uncertainty in the localization of the spot image on the detector. This uncertainty is caused by laser speckle noise. For the imaging system, speckle arises because the light wave amplitude at each point of the image is the summation of contributions from all the scattering points of the object (Baribea, 1991; Goodman, 1975). When the object is roughly on a scale comparable to the wavelength of the illumination source, the summation involves random phasors. For some portions of the image, these phasors cancel each other, leading to dark speckles, while for other parts of the image they reinforce each other, leading to bright speckles as shown in Fig. 2.10.

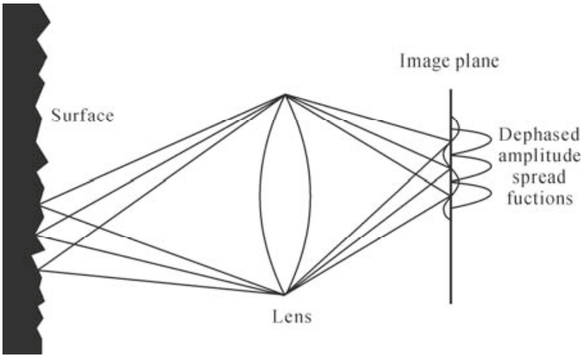


Fig. 2.10. Speckle noise arises from the interference of a series of diffraction patterns, each generated by a speckle element (Modified from (Goodman, 1984), permission granted)

Based on the laser speckle statistics theory, the uncertainty of the image coordinate is given by (Dorsch, 1995)

$$\delta x = \frac{1}{2\pi}\frac{\lambda}{\sin u} \tag{2.20}$$

where δx is the standard deviation of the laser spot position and sin u is the observation aperture. It can be written as $\sin u \approx a / z$ $(z >> a)$ where a is the radius of the camera lens and z is the distance from the lens to the observation surface. It has been indicated that the uncertainty of the spot position due to the laser speckle noise is a function of the wavelength λ and the observation aperture. For example, with practical parameters sin u = 0.2 and $\lambda = 670$ nm, the image resolution will be δx = 0.53 μm that yields the resolution of the measurement δZ = 6.7 μm for a typical measurement scenario with Z = 100 mm, b = 50 mm and f = 16 mm, based on Eq. (2.19).

Eq. (2.20) is for the coherent illumination light source. For the partially coherent illumination, we have the image spot location uncertainty

$$\delta x = C\frac{1}{2\pi}\frac{\overline{\lambda}}{\sin u} \tag{2.21}$$

where C is the speckle contrast and $\overline{\lambda}$ is the mean weighted wavelength.

The above equation indicates that the reduction in the location uncertainty can be achieved by reducing the coherence length of the light source.

In terms of the laser structured-light sensor, where the center line of the projected laser profile needs to be detected accurately, speckle noise makes the laser line noisy and salty, as shown in Fig. 2.11, for its Gaussian profile distribution affects the detection of the center line to sub-pixel accuracy regardless of algorithms.

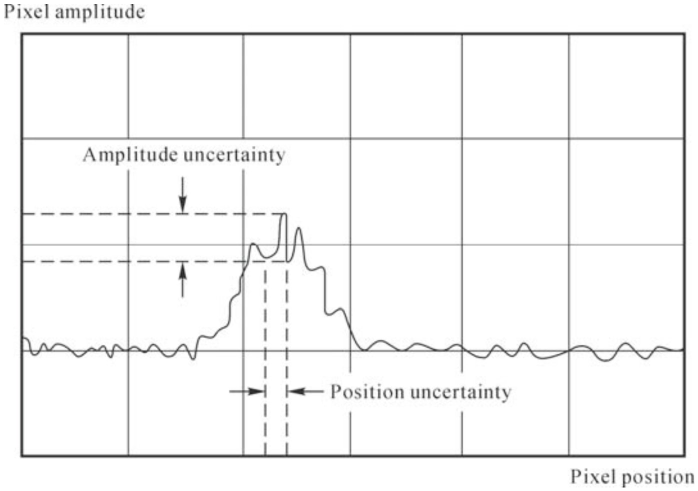


Fig. 2.11. Image of a laser line with speckle noise and uncertainty of the position measurement

Speckle noise can be reduced generally by integrating a single measurement over several intensity samples as the laser spot is moved over the surface being scanned. Another method to reduce speckle is to move the laser source along the laser plane (Wu, 2005) and take the average of all the images obtained from different source locations. The use of incoherent light, like light emitting diodes (LEDs), reduces speckle noise associated with lasers and consequently provides a better image. However, the depth of view is smaller and the focused spot size is bigger when compared to laser sensors, yielding lower resolution performance.

2.2.2 Effect of the Environmental Factors on the Measurement Accuracy

The environmental factors like incident angle, surface reflectivity, ambient lighting, surface discontinuity, will have an impact on the measurement accuracy as listed in Table 2.1

Table 2.1 Environmental factors affecting the measurement accuracy (modified from Table 1 of (MacKinnon et al., 2008), permission granted)

Error source	Effect
Range	Range uncertainty generally increases with range
Angle of incidence	Range uncertainty increases with increased angle of incidence (Prito, 2002)
Surface material	Translucent non-homogeneous materials increase range uncertainty (Hancock, 1998a)
Surface complexity	Surface discontinuities introduce range errors
Reflectivity	Range uncertainty increases with a decrease in reflectivity (El-Hakin, 1995)
Ambient lighting	Range uncertainty increases with an increase in ambient lighting (Hancock, 1998b)

A laser structured-light sensor operates by imaging the laser spot from the surface onto a position-sensing detector. Most surfaces to be measured are a combination of diffuse and specular in which they scatter light in many directions. A diffuse surface will scatter the light reflected from the surface in all directions such that light from one direction can be received by the detector, and the sensor will perform properly. When the surface is very specular like a mirror, the detector will not receive any light and the sensor will not perform properly. A change in reflectivity will affect the intensity of light reaching the detector. When the intensity of light getting into the detector is too high, the detector will be saturated. In this case the Gaussian profile will be clipped and the detected centroid position of the image profile will not be accurate. When the received intensity is too low, the signal-to-noise ratio (SNR) of the detected spot will decrease. In order to have the reflected intensity match the detector measurement range, the laser intensity is controlled automatically based on the feedback of the received intensity. When the incident angle to the surface is far from normal, the spot size will be enlarged and skewed. This will reduce the resolution of the sensor.

Fig. 2.12 shows the effect of the surface discontinuity on the measurement accuracy. Due to the block of the partial spot area, the detected centroid position is shifted from its peak location. This shift occurs if the object is smaller than the size of the laser beam or on edges. Assume that the center position is determined by calculating the profile gravity, the shift is given by (Blais, 2005)

$$\Delta x = \frac{\int_{x_1}^{x_2} I_0 \cdot x \cdot e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2} \alpha x}{\int_{x_1}^{x_2} I_0 \cdot e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma}\right)^2} \alpha x} \tag{2.22}$$

where x_1 and x_2 correspond to transitions and x_0 is the expected centroid of the laser spot .

From Eq. (2.16) we obtain

$$\Delta z = \Delta x / \cot \theta \tag{2.23}$$

If only 1/2 of 100 μm laser spot is imaged then a centroid shift is given by Δx =20 μm. This yields the error in Z direction Δz = 34 μm if $\theta = 60°$.

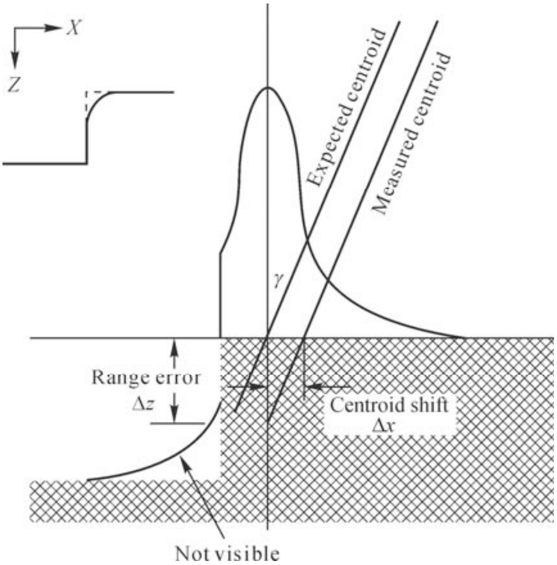


Fig. 2.12. A surface discontinuity results in a shift (Δx) in the position of the centroid in a triangulation laser range scanner. This results in a range error Δz. (modified from Fig. 7 of (Blais, 2005), permission granted)

Fig. 2.13 indicates the effect of the surface non-uniform reflectivity on the measurement uncertainty.

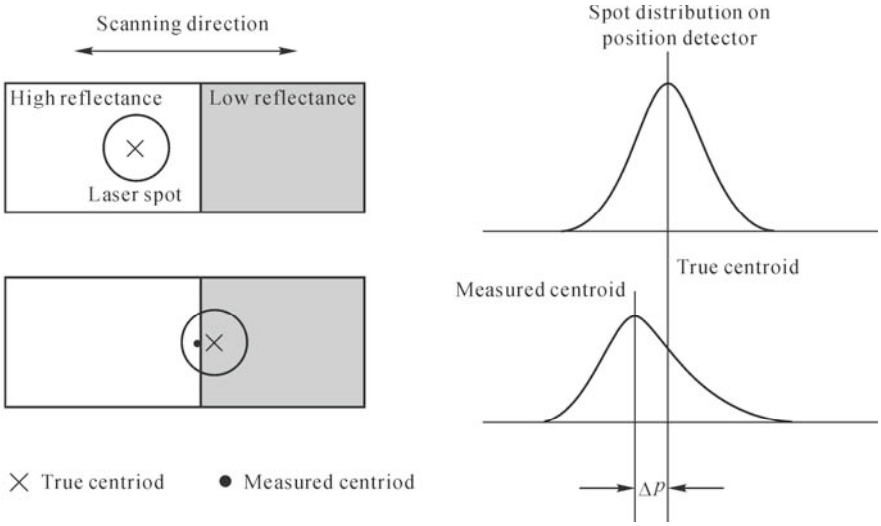


Fig. 2.13. Transitions between regions of different surface reflectivity can affect the accuracy of the range measurement (modified from Fig. 1 of (El-Hakan, 1995), permission granted)

2.3 Commercial Systems

Table 2.2 lists companies that can offer 3D laser range sensors, brief descriptions of scanning methods, measurement range and measurement accuracy. Most of the company names are from the reference (Blais, 2004) .

Table 2.1 Company list for laser range sensors (modified from (Blais, 2004), permission gruanted)

Company	Description	Range	Accuracy
Steintek (www.steintek.de)	Probe on multi-axis mechanical structure	10 – 60 cm	20 – 300 μm
ShapeGrabber (www.shapegrabber.com)	Plane of light on translation or rotation stage	40 – 650 mm	25 – 200 μm
Cyberware (www.cyberware.com)	Several products; body scanner, color, simple scanning unit		50 – 300 μm
3D scanners-Model Maker (www.3dscanners.com)	Hand-held laser stripe camera mounted on mechanical probe		
KonicaMinolta-Vivid Systems (www.konicaminolta-3d.com)	General purpose-galvanometer / mirror scanner	0.7 – 1.4 m	0.1 – 7 mm
Kreon (www.kreon3d.com)	Probe for CMM/CNC machine-laser stripe sensor	50 – 100 mm	10 – 25 μm
Virtek Vision International (www.virtekvision.com)	Multiple lines of industrial products	25 mm	100 μm

(to be continued)

(Table 2.2)

Company	Description	Range	Accuracy
Hamamatsu-Body Scanner (Usa.hamamatsu.com)	Full body scan (10 s)	0.5 m	
Metris (www.metris.be)	CMM/CNC laser stripe scanner probe	100 mm	5 – 12 µm
DLR 3D Modeller (www.robotic.dlr.de)	Multi-sensor device ; robotics applications		
Servo Robot Inc. (www.servorobot.com)	Robotic control; automatic welding; industrial inspection		
Arius3D (www.arius3d.com)	High accuracy and resolution color		
Laser Design Inc.-Surveyor (www.laserdesign.com)	Line of products, from portable to CMM mounted, reverse eng.		
Virtual 3D (www.virtual3dtech.com)	Several products		
Vitronic (www.vitronic.com)	Industrial inspection to body scan		
Polhemus-FastSCAN-3Draw (www.polhemus.com)	Hand-held slit scanner with magnetic trackers		
Nextec (www.nextec-wiz.com)	CMM-based optical probe		
3D Digital Corp. (www.3ddigitalcorp.com)	Laser strip with optional color texture, inspection		
Perceptron (www.perceptron.com)	CMM or portable arm hand-held, high accuracy inspection		
Scantech (www.scantech.net)	Dual view triangulation, laser stripe	40 – 400 mm	20 – 200 µm
Neptec (www.neptec.com)	Space, industrial, mining, autosynchronized laser scanning		
www.riegl.com	Laser range finder		
LMI Technologies (www.lmint.com)	Over 40 products: elect., inspection, manufacturing, forest		
Acuity Research-AR600 (www.acuityresearch.com)	Probe	Up to 0.5 m	0.10%
RobotScan (www.inter-smart.com)	Laser scanner head	120 – 300 mm	30 µm
Steintek-3D-SCAN	Triangulation based, laser stripe	100 – 600 mm	20 – 300 mm

References

Baribea R, Rioux M (1991) Influence of speckle on laser range finders. Applied Optics, 30:2873-2878.

Beraldin JA, Blais F, Rioux M, Cournoyer L, Laurin D, Maclean S (2000) Eye-Safe digital 3-D sensing for space applications. Optical Engineering, 39:196-211.

Blais F (2004) Review of 20 years of range sensor development. Journal of Electronic Imaging, 13(1):231-240.

Blais F, et al. (2005) High resolution imaging at 50 μm using a portable XYX-RGB color laser scanner. In: Proc. Int. Workshop on Recording, Modeling and Visualization of Cultural Heritage, NRC, Ottawa, Canada.

Curless BL (1997) New methods for surface reconstruction from range images. Ph.D Dissertation, Stanford University, CA.

Dickey FM, Holswade SC (Eds.) (2000) Laser Beam Shaping, Theory and technologies. Marcel Dekker.

Dorsch RG, Hausler G, Herrmann JM (1995) Laser triangulation: fundamental uncertainty in distance measurement. Applied Optics, 33(7):1306-1314.

El-Hakin S, Beraldin JA (1995) Configuration design for sensor integration. Proc. SPIE, 2598:274-285.

Goodman JW (1975) Some fundamental properties of speckle. Journal of Optical Society of America, 66:1145-1150.

Goodman JW (1984) Laser Speckle and Related Phenomena. Springer-Verlag, Chapter 1, 9-76.

Hancock J, et al. (1998) Active laser radar for high performance measurements. Proc. IEEE International Conference on Robotics and Automation, 2:1465-1470.

Hancock J, Hebert M, Thorpe C (1998) Laser intensity based obstacle detection. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, 3:1541-1546.

Jarvis RA (1983) A perspective on range finding techniques for computer vision. IEEE Transaction on Pattern Analysis and Machine Intelligence, 5(2):122-139.

MacKinnon D, Aitken V, Blais F (2008) Review of measurement quality metrics for range imaging. Journal of Electronic Imaging, 17(3):1-14.

Prieto F, Boulanger P, Lepage R, Redarce T (2002) Automated inspection system using range data. In: Proc. IEEE International Conference on Robotics and Automation, 3:2557-2562.

Su, XY, Li JT (1999) Information Optics. Science Press, Beijing (In Chinese).

Williams D (1993) Optical Methods in Engineering Metrology. Chapman & Hall, London, 11-16.

Wu QY (2006) A study for key techniques of 3D-sensor based on line-structure light. Chapter 2, Ph.D thesis, Sichuan University (In Chinese).

Wu QY, Su XY, Li Y (2005) An approach for reducing the influence of speckle on three-dimensional sensing through surface laser light scanning. Laser Journal, 26(3):63-64 (In Chinese).

Zhang G, He J, Li X (2005) 3D vision inspection for internal surface based on circle structured light. Sensors and Actuators A, 122:68-75.

3

Laser Stripe Sensor Calibration

The purpose of laser stripe sensor calibration is to identify its intrinsic parameters and extrinsic parameters. Extrinsic parameters contain information about the sensor position and orientation relative to a reference coordinate frame. The physical meaning of intrinsic parameters varies, depending on the mathematical model of the sensor. Since the laser stripe sensor consists of a laser projector and a camera, camera modeling and calibration become an integrated part of the sensor calibration procedure.

In Section 3.1 two general models of the laser stripe sensor are presented and the calibration strategy for both models is discussed. In Section 3.2 camera modeling is introduced. It includes pinhole modeling and nonlinear modeling with consideration of lens distortion. In Section 3.3 algorithms and implementations of various camera calibration methods are presented and discussed. It includes conventional direct linear transform, Tsai's RAC based algorithm, and Zhang's flexible multiple view calibration algorithm. In Section 3.4 calibration of laser stripe sensor with various techniques, algorithms and implementation is presented. All kinds of calibration techniques are summarized in a table for comparison in terms of the form of a calibration target, method for extracting control points for camera calibration and laser plane calibration, and the algorithm and mathematical model used.

3.1 Modeling of Laser Stripe Sensor and Calibration Strategy

Since the laser stripe sensor consists of a camera and a laser projector, there are two methods for modeling the laser stripe sensor. One method for modeling the sensor is just the combination of a camera model and the plane model. Another method is to use laser plane to image plane direct mapping as the model. Determination of the system parameters, like camera parameters and the relative position between the camera and the laser projector is called the calibration process. There are two types of approach based on the modeling method as shown in Fig. 3.1.

Z. Gan et al., *Visual Sensing and its Applications*

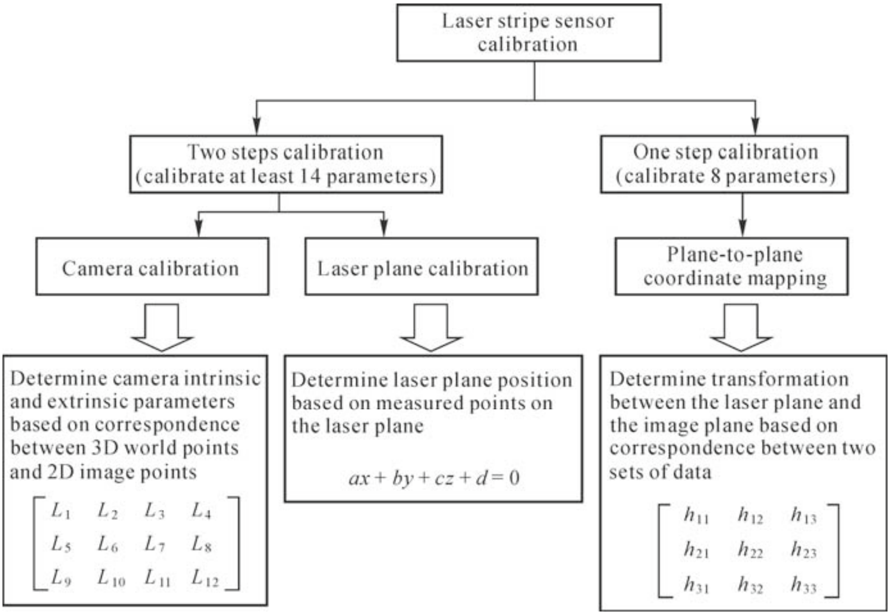

Laser stripe sensor calibration
Two steps calibration (calibrate at least 14 parameters)
One step calibration (calibrate 8 parameters)
Camera calibration
Laser plane calibration
Plane-to-plane coordinate mapping
Determine camera intrinsic and extrinsic parameters based on correspondence between 3D world points and 2D image points
$\begin{bmatrix} L_1 & L_2 & L_3 & L_4 \\ L_5 & L_6 & L_7 & L_8 \\ L_9 & L_{10} & L_{11} & L_{12} \end{bmatrix}$
Determine laser plane position based on measured points on the laser plane
$ax+by+cz+d=0$
Determine transformation between the laser plane and the image plane based on correspondence between two sets of data
$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$

Fig. 3.1. Two methods of calibrating laser stripe sensor

The first approach is to calibrate the camera parameters and laser plane position separately since the laser stripe sensor is composed of a camera and a laser plane. In this case, at least 14 parameters need to be determined. Among them there are 11 parameters for the camera if the lens distortion is ignored or 17 parameters if the lens distortions of the tangential and radical are considered. The laser plane is defined by three independent parameters. This is a physical model based calibration. All the calibration parameters have a physical meaning. The other method is to treat the inside of the laser stripe sensor as a black box and only care about the laser plane and image plane as input and output plane to the plane mapping function. This requires calibrating 8 parameters.

For both methods there are linear models and nonlinear models. Parameters in the linear model can be solved with a linear least squares algorithm and parameters in the nonlinear model are obtained with a nonlinear optimization algorithm.

3.2 Camera Modeling

Before camera calibration is presented we will present a review of camera modeling. The model is a mathematical relationship between a space point and its corresponding image coordinate. This is the mapping function from 3D space to

2D space. In the following two sections we will address two types of camera models: a distortion free pinhole model and a nonlinear model with consideration of lens distortion.

3.2.1 Pinhole Model of the Camera

Any point in space can be imaged into the image plane through a pinhole camera model. The pinhole camera model is a simplification of the imaging process through an optical system when the distance of an object to the lens is much bigger than the focal length. In this case, the image plane coincides with the focal plane of the optical system and the imaging system is simplified as the projective projection system.

The geometric model of a pinhole camera consists of an image plane I and eye point C on the focal plane F, as shown in Fig. 3.2(a). The fundamental property of the perspective is that every image point m is collinear with C and its corresponding world point M. The point C is called the optical center. The line Cc, perpendicular to image plane I and focal plane F, is called the optical axis. C is called the principle point. From the geometric viewpoint there is no difference in replacing the image plane by a virtual image plane located on the other side of the focal plane as shown in Fig. 3.2(b).

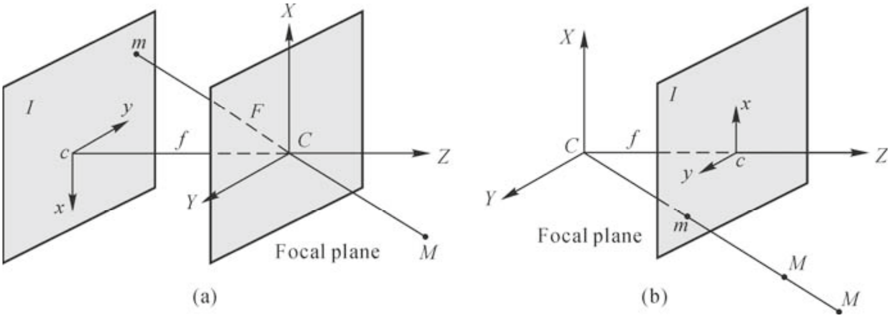


Fig. 3.2. Illustration of camera pinhole model. (a) Image plane is placed behind the focal plane reflecting the physical layout of the pinhole camera; (b) Pinhole camera model with a virtual image plane placed in front of focal plane

Let (C, X, Y, Z) be the camera coordinate system and (c, x, y) be the image coordinate system. It can be seen that

$$x = f\frac{X}{Z}; y = f\frac{Y}{Z} \tag{3.1}$$

where (x, y) is the coordinate of point m in the image coordinate system and (X, Y, Z) is the coordinate of point M in the camera coordinate system. In projective geometry, any point along the ray going through the optical center projects to the same image point. Eq. (3.1) can be rewritten as the following matrix format

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

where

$$\boldsymbol{P} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.3}$$

is called the camera perspective projection matrix and λ is the scale factor.

The digital image coordinate (u, v) with "pixel" is related to the analogy image coordinate (x, y) with unit "micro meter" as

$$\left.\begin{aligned} u - u_0 &= \alpha_x x \\ v - v_0 &= \alpha_y y \end{aligned}\right\} \tag{3.4}$$

where (u_0, v_0) is the digital image coordinate of the principle point and (α_x, α_y) are the scaling factors for the conversion from the coordinate system (x, y) to coordinate system (u, v).

Combination of Eqs. (3.1) and (3.4) results in

$$\left.\begin{aligned} u - u_0 &= \alpha_x f \frac{X}{Z} = k_u \frac{X}{Z} \\ v - v_0 &= \alpha_y f \frac{Y}{Z} = k_v \frac{Y}{Z} \end{aligned}\right\} \tag{3.5}$$

and its matrix format is

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix} \tag{3.6}$$

where $k_u = \alpha_x f$ and $k_v = \alpha_y f$ are the scaling factors along the X and Y axes of the image plane, respectively. Matrix $\boldsymbol{K}$ is called the intrinsic parameter of the camera, which is independent of the camera position

$$\boldsymbol{K} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

If the image plane axes u and v are not orthogonal, the intrinsic parameter of the camera is represented by

$$\boldsymbol{K} = \begin{bmatrix} k_u & k_u \cot\theta & u_0 \\ 0 & k_v / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

where θ is the skew between the axes as shown in Fig. 3.3.

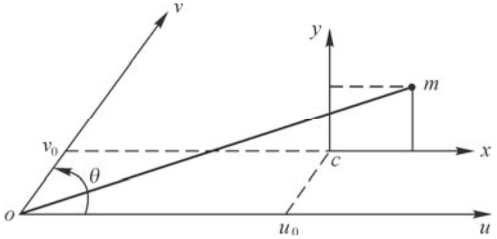


Fig. 3.3. Illustration of the relationship between two coordinate systems *xcy* and *uov*. Note that coordinate system *uov* may not be orthogonal if the angle between the two axes is not a right angle

Instead of being expressed in the camera coordinate system, a 3D point can be expressed in any arbitrary reference coordinate system called the world coordinate system. The position of the camera coordinate system relative to the world coordinate system can be expressed by the transformation

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \boldsymbol{t} \tag{3.9}$$

where (X_c, Y_c, Z_c) and (X_w, Y_w, Z_w) are the coordinates of a 3D point in the camera coordinate system and world coordinate system, respectively. $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation matrix and translation vector between two coordinate frames.

$$\boldsymbol{R} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}, \quad \boldsymbol{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \tag{3.10}$$

They are called extrinsic parameters of the camera. Combining Eqs. (3.6) and (3.9) gives the mapping of the coordinate in the world coordinate frame and in the image coordinate frame

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} [\boldsymbol{R} \quad \boldsymbol{t}] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.11}$$

This is the general model of the pinhole camera, describing the mapping from the world coordinate frame to the image coordinate frame.

3.2.2 Nonlinear Modeling with Lens Distortion

It is well known that actual lenses sustain a variety of aberrations and thus do not obey the ideal pinhole model, as described above. Lens distortions can be modeled as radial distortions and tangential distortions, as indicated in Fig. 3.4. Distortion dr is called radial distortion that causes an inward and outward displacement of a given image point from its ideal location. Distortion dt is called a tangential distortion that causes a decentering displacement of a given image point from its ideal location. The distortion free pinhole model has to be replaced by a model with the distortion and position errors being taken into account.

$$\left.\begin{aligned}\tilde{u} &= u + \delta_u(u,\ v),\\ \tilde{v} &= v+\delta_v(u,\ v)\end{aligned}\right\} \tag{3.12}$$

where (u, v) are the unobservable distortion free image coordinates, $(\tilde{u},\ \tilde{v})$ are the corresponding coordinates with distortion correction, δ_u and δ_v are nonlinear distortion in u and v direction, respectively. They are decided by the position of the image points in the image plane and given by

$$\left.\begin{aligned}&\delta_u(u,v) = k_1u'r^2 + k_2u'r^4 + k_3u'r^6 + p_1(2u'^2 + r^2) + p_2(2u'v')\\ &\delta_v(u,v) = k_2v'r^2 + k_2v'r^4 + k_3v'r^6 + p_1(2u'v') + p_2(2v'^2 + r^2)\\ &u' = u - u_0,\ v' = v - v_0,\ r^2 = u'^2 + v'^2\end{aligned}\right\} \tag{3.13}$$

where k_1, k_2, k_3, p_1, p_2 are distortion parameters.

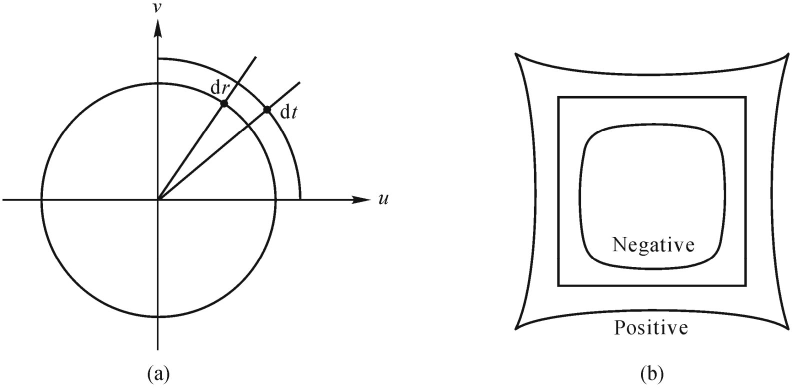


Fig. 3.4. (a) Effects of radial and tangential distortion on the location of an image point; (b) Effect of radial distortion on a rectangular

Parameters k_1, k_2, k_3 are coefficients of radial distortion, which is only dependent on the distance between an image point and lens center. The influence of these parameters on the distortion of a rectangular grid is shown in Fig. 3.4(b). The parameters p_1, p_2 represent coefficients of tangential or decentering distortions.

In most situations p_1, p_2 are usually negligibly small and k_2 and k_3 are also often neglected, leaving a simple radial model with only one parameter.

3.3 Calibration of Cameras

After the camera model was established in previous sections, the purpose of this section is to identify its parameters in the model by using the correspondence between the pre-designed space points (control points) and corresponding image coordinates. Calibration of the camera is to determine intrinsic parameters that define the optical characteristic of the camera, and extrinsic parameters that contain information on the position and orientation of the camera relative to a reference coordinate frame. In the following, three basic calibration methods will be presented. In each method the linear model is used first and solved by a linear least squares algorithm as an approximation solution, and then the model is modified to become the nonlinear function and solved with a nonlinear optimization algorithm, as shown in Fig. 3.5.

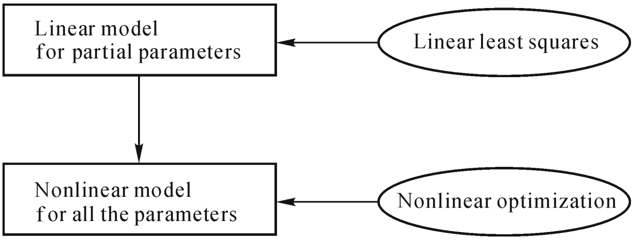


Fig. 3.5. Flow of camera calibration

3.3.1 Calibration with Direct Linear Transform Method

The direct linear transformation (DLT) is a linear relation between a 3D point and its image coordinate. It was introduced first by Abdel-aziz and Karara (1971) and further modified by Marzan and Karara (1975). It can be derived from the pinhole model. Eq. (3.11) can be rewritten in the following form

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \lfloor \boldsymbol{R} \quad \boldsymbol{t} \rfloor \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \\ L_5 & L_6 & L_7 & L_8 \\ L_9 & L_{10} & L_{11} & L_{12} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{3.14}$$

Eq. (3.14) is equivalent to the following form

$$\left.\begin{aligned} u &= \frac{L_1 X_w + L_2 Y_w + L_3 Z_w + L_4}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + L_{12}} \\ v &= \frac{L_5 X_w + L_6 Y_w + L_7 Z_w + L_8}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + L_{12}} \end{aligned}\right\} \tag{3.15}$$

L_{12} can be eliminated by dividing denominator and numerator of the right hand side of the equation by L_{12}, yielding a DLT standard form that relates the world coordinates $\{X_w, Y_w, Z_w\}$ to the image coordinates $\{u, v\}$ as

$$\left.\begin{aligned} u &= \frac{L_1 X_w + L_2 Y_w + L_3 Z_w + L_4}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + 1} \\ v &= \frac{L_5 X_w + L_6 Y_w + L_7 Z_w + L_8}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + 1} \end{aligned}\right\} \tag{3.16}$$

where $(L_1, L_2, \ldots, L_{11})$ is called a DLT parameter that describes the linear mapping from the 3D coordinates and 2D coordinates. They can be solved by using a set of 3D points (X_{wi}, Y_{wi}, Z_{wi}) and corresponding image coordinates (u_i, v_i). The advantage of DLT is that the mapping coefficients $(L_1, L_2, \ldots, L_{11})$ can be solved by using a linear least squares method.

Parameters $(L_1, L_2, \ldots, L_{11})$ are the combination of the physical parameters of the camera and the coordinate transform. The relationship between both sets of parameters is given by the following equations (Sabel, 1999)

$$\left.\begin{aligned} L_1 &= (u_0 m_{13} + k_u m_{11}) / L \\ L_2 &= (u_0 m_{23} + k_u m_{21}) / L \\ L_3 &= (u_0 m_{33} + k_u m_{31}) / L \\ L_4 &= -L_1 t_x - L_2 t_y - L_3 t_z \\ L_5 &= (v_0 m_{13} + k_v m_{12}) / L \\ L_6 &= (v_0 m_{23} + k_v m_{22}) / L \\ L_7 &= (v_0 m_{33} + k_v m_{32}) / L \\ L_8 &= -L_5 t_x - L_6 t_y - L_7 t_z \\ L_9 &= m_{13} / L \\ L_{10} &= m_{23} / L \\ L_{11} &= m_{33} / L \\ L &= -(m_{13} t_x + m_{23} t_y + m_{33} t_z) \end{aligned}\right\} \tag{3.17}$$

Extrinsic parameters and intrinsic parameters of the camera can be derived from L_i (Faugeras, 1993). The camera position in the world coordinate frame is given by

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} -L_4 \\ -L_8 \\ -1 \end{bmatrix} \tag{3.18}$$

The principle point and scale factors of the camera are

$$u_0 = \frac{\boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_3}{\boldsymbol{b}_3^{\mathrm{T}} \boldsymbol{b}_3}; v_0 = \frac{\boldsymbol{b}_2^{\mathrm{T}} \boldsymbol{b}_3}{\boldsymbol{b}_3^{T} \boldsymbol{b}_3}; k_u = \sqrt{\frac{\boldsymbol{b}_1^{\mathrm{T}} \boldsymbol{b}_1}{\boldsymbol{b}_3^{\mathrm{T}} \boldsymbol{b}_3} - u_0^2};\ k_v = \sqrt{\frac{\boldsymbol{b}_2^{\mathrm{T}} \boldsymbol{b}_2}{\boldsymbol{b}_3^{\mathrm{T}} \boldsymbol{b}_3} - v_0^2} \tag{3.19}$$

where

$$\boldsymbol{b}_1 = (L_1\ L_2\ L_3)^T;\ \boldsymbol{b}_2 = (L_5\ L_6\ L_7)^{\mathrm{T}}; \boldsymbol{b}_3 = (L_9\ L_{10}\ L_{11})^{\mathrm{T}} \tag{3.20}$$

The rotation matrix from the camera coordinate frame to the world coordinate frame is given by

$$\boldsymbol{R} = \begin{bmatrix} \dfrac{-\boldsymbol{b}_1}{\|\boldsymbol{b}_1\|} & \dfrac{-\boldsymbol{b}_2}{\|\boldsymbol{b}_2\|} & \dfrac{-\boldsymbol{b}_3}{\|\boldsymbol{b}_3\|} \end{bmatrix} \tag{3.21}$$

where

$$\boldsymbol{b}_3' = (\boldsymbol{b}_1 - u_0 \boldsymbol{b}_3) / k_u; \quad \boldsymbol{b}_2' = (\boldsymbol{b}_2 - v_0 \boldsymbol{b}_3) / k_v \tag{3.22}$$

3.3.1.1 Solution of DLT Parameters

Eq. (3.16) can be solved either by using a linear least squares method or nonlinear optimization method. Given a set of 3D points (X_{wi}, Y_{wi}, X_{wi}) in the world coordinate frame and their corresponding image coordinates (u_i, v_i), the DLT parameter can be obtained by solving the linear equations of the form

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{3.23}$$

$$\left.\begin{aligned} &\boldsymbol{A} = \begin{bmatrix} X_{w1} & Y_{w1} & Z_{w1} & 1 & 0 & 0 & 0 & 0 & -u_1 X_{w1} & -u_1 Y_{w1} & -u_1 Z_{w1} \\ 0 & 0 & 0 & 0 & X_{w1} & Y_{w1} & Z_{w1} & 1 & -v_1 X_{w1} & -v_1 Y_{w1} & -v_1 Z_{w1} \\ & & & \cdots & \cdots & \cdots & & & & & \\ & & & \cdots & \cdots & \cdots & & & & & \\ X_{wn} & Y_{wn} & Z_{wn} & 1 & 0 & 0 & 0 & 0 & -u_n X_{wn} & -u_n Y_{wn} & -u_n Z_{wn} \\ 0 & 0 & 0 & 0 & X_{wn} & Y_{wn} & Z_{wn} & 1 & -v_n X_{wn} & -v_n Y_{wn} & -v_n Z_{wn} \end{bmatrix} \\ &\boldsymbol{x} = [L_1\ L_2\ L_3\ L_4\ L_5\ L_6\ L_7\ L_8\ L_9\ L_{10}\ L_{11}]^{\mathrm{T}} \\ &\boldsymbol{b} = [u_1\ v_1 \ldots u_n\ v_n]^{\mathrm{T}} \end{aligned}\right\} \tag{3.24}$$

where A is a $2n \times 11$ matrix; b is a $2n \times 1$ vector; n is the number of measurement points. $\boldsymbol{x}$ can be solved by using linear least squares by minimizing the error function min $\|\boldsymbol{Ax}-\boldsymbol{b}\|^2$, yielding

$$\boldsymbol{x} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \tag{3.25}$$

In order to solve the over determined equations, the number of equations must be more than the number of variables (i.e., 11), requiring that at least 6 calibration data points are used (i.e., $n \geq 6$). To ensure that the equations are not linearly dependent, all the 3D calibration points cannot be distributed on the same plane. A typical calibration target is shown in Fig. 3.6. Coordinates of all the dots or markers are known in the reference coordinate frame.

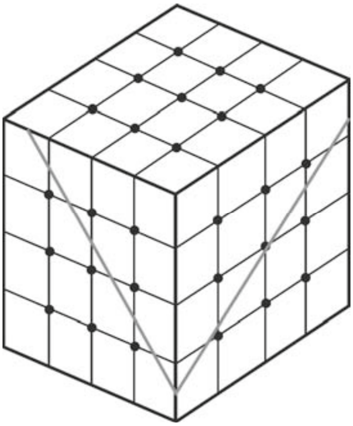


Fig. 3.6. DLT parameters are calculated using known 3D points and measured feature image points. 3D points are non-coplanar

3.3.1.2 Solution of DLT Parameters with Lens Distortion

With distortion correction the linear DLT model Eq. (3.16) becomes

$$\left.\begin{aligned} u + \delta_u(u,\ v) &= \frac{L_1 X_w + L_2 Y_w + L_3 Z_w + L_4}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + 1} \\ v + \delta_v(u,\ v) &= \frac{L_5 X_w + L_6 Y_w + L_7 Z_w + L_8}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + 1} \end{aligned}\right\} \tag{3.26}$$

This is a nonlinear equation and can be solved by using a nonlinear least squares method that minimizes the objective function

$$F(L_1,\ \ldots,\ L_{11},\ k_1,\ k_2,\ k_3,\ p_1,\ p_2) = \sum_{i=1}^{n} (F_{u,i}{}^2 + F_{v,i}{}^2) \tag{3.27}$$

$$\left.\begin{aligned} F_{u,i} &= \left(u_i + \delta_u(u_i,\ v_i) - \frac{L_1 X_{wi} + L_2 Y_{wi} + L_3 Z_{wi} + L_4}{L_9 X_{wi} + L_{10} Y_{wi} + L_{11} Z_{wi} + 1} \right) \\ F_{v,i} &= \left(v_i + \delta_v(u_i,\ v_i) - \frac{L_5 X_{wi} + L_6 Y_{wi} + L_7 Z_{wi} + L_8}{L_9 X_{wi} + L_{10} Y_{wi} + L_{11} Z_{wi} + 1} \right) \end{aligned}\right\} \tag{3.28}$$

by choosing parameters listed as variables in *F*, where *n* is the number of image points used for calibration. The problem is reduced to a standard nonlinear least squares operation. All the parameters are optimized iteratively by using a nonlinear least squares algorithm, i.e., the Levenberg-Marquard algorithm (Press, 1994), with the initial values obtained from Eq. (3.25) by using the linear least squares algorithm.

3.3.2 Calibration with Tsai's RAC Based Algorithm

In a DLT model all the calibration points need to be non-coplanar, which makes it difficult to obtain the calibration target. An alternative is to use the radical alignment constraint (RAC) algorithm proposed by Tsai (1987), in which a coplanar calibration target can be used. Tsai's camera model is based on the pinhole model of perspective projection, as indicated in Fig. 3.2, and the relationship between a 3D point (X_w, Y_w, Z_w) and its image coordinate (u, v) is expressed as

$$\left.\begin{aligned} u'(1 + kr^2) &= f_x \frac{r_1 X_w + r_2 Y_w + r_3 Z_w + t_x}{r_7 X_w + r_8 Y_w + r_9 Z_w + t_z} \\ v'(1 + kr^2) &= f_y \frac{r_4 X_w + r_5 Y_w + r_6 Z_w + t_y}{r_7 X_w + r_8 Y_w + r_9 Z_w + t_z} \end{aligned}\right\} \tag{3.29}$$

or

$$\frac{u'}{v'} = \mu^{-1} \frac{r_1 X_w + r_2 Y_w + r_3 Z_w + t_x}{r_4 X_w + r_5 Y_w + r_6 Z_w + t_y} \tag{3.30}$$

where

$$\left.\begin{aligned} r^2 &= \mu^2 u'^2 + v'^2 \\ \mu &= f_y / f_x \\ u' &= u - u_0 \\ v' &= v - v_0 \end{aligned}\right\} \tag{3.31}$$

f_x, f_y are the effective focal lengths of the pinhole camera in the *u* and *v* directions of the image plane, respectively; *k* is the 1st order radial lens distortion coefficient;

$(u_0,\ v_0)$ are coordinates of the center of radial lens distortion; $\boldsymbol{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$ and $\boldsymbol{t} = [t_x,\ t_y,\ t_z]^{\mathrm{T}}$ form the rotation matrix and translation vector of the transformation between the camera coordinate frame and the world coordinate frame, as indicated in Eq. (3.9). Eq. (3.30) is called an RAC equation.

The purpose of calibration is to determine the coefficients in Eq. (3.29) by using a set of calibration data points. Similar to the DLT method, calibration data for the model consists of a set of 3D points $(X_{wi},\ Y_{wi},\ Z_{wi})$ in the world coordinate frame (in mm for example) and corresponding 2D image coordinates $(u_i,\ v_i)$ (typically in pixels).

Tsai's RAC algorithm has two stages. In the first stage a linear least squares algorithm is used to get the initial values of some parameters. In the second stage all the parameters are obtained simultaneously by using a nonlinear optimization method that finds the best fit between measured image points and those predicted from the model. The parameters estimated from the first step are refined in this process. In the algorithm, only radial distortion is considered and other distortion like tangential distortion is ignored. The calibration target can be coplanar and non-planar that results in a different approach. An accurate planar target is easier to make and maintain than a 3D target, but it has the limitation that the target normally has to be turned away from the optical axis of the camera.

From Eq. (3.30) we have

$$\mu X_w v' r_1 + \mu Y_w v' r_2 + \mu Z_w v' r_3 + \mu v' t_x - X_w u' r_4 - Y_w u' r_5 - Z_w u' r_6 = u' t_y \tag{3.32}$$

Eq. (3.32) can be rewritten in the matrix form

$$[X_w v' \quad Y_w v' \quad Z_w v' \quad v' \quad -X_w u' \quad -Y_w u' \quad -Z_w u'] \begin{bmatrix} \mu r_1 / t_y \\ \mu r_2 / t_y \\ \mu r_3 / t_y \\ \mu t_x / t_y \\ r_4 / t_y \\ r_5 / t_y \\ r_6 / t_y \end{bmatrix} = u' \tag{3.33}$$

In Eq. (3.33) the vector $[X_w v' \quad Y_w v' \quad Z_w v' \quad v' \ {-X_w u'} \quad {-Y_w u'} \quad {-Z_w u'}]$ and u' are known parameters, and $[\mu r_1/t_y \quad \mu r_2/t_y \quad \mu r_3/t_y \quad \mu t_x/t_y \quad r_4/t_y \quad r_5/t_y \quad r_6/t_y]^{\mathrm{T}}$ need to be solved. There are slightly different approaches for coplanar and non-coplanar target points. Here we only discuss the case for coplanar calibration points, for simplicity. For the coplanar case μ cannot be determined, and it is assumed that μ=1, thus $f_x = f_y = f$. All the target points are placed in the world coordinate frame with $Z_{wi} = 0$.

(1) Computation of the rotation matrix $\boldsymbol{R}$ and translation parameters t_x and t_y.

Define a set of intermediate parameters

$$\{v_1,\ v_2,\ v_3,\ v_4,\ v_5\}=\{r_1/t_y,\ r_2/t_y,\ t_x/t_y,\ r_4/t_y,\ r_5/t_y\} \tag{3.34}$$

and recall $u_i'=u_i-u_0,\ v_i'=v_i-v_0$. Assuming that (u_0, v_0) is given and $Z_{wi}=0$, then the intermediate parameters can be solved from the linear equations

$$[X_{wi}v_i'\ \ Y_{wi}v_i'\ \ v_i'-X_{wi}u_i'-Y_{wi}u_i']\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{bmatrix}=u_i' \tag{3.35}$$

or

$$\boldsymbol{Av}=\boldsymbol{b} \tag{3.36}$$

By using the linear least squares method, we can obtain

$$\boldsymbol{v}=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b} \tag{3.37}$$

In order to establish an over-determined linear equation the number of the calibration points must be more than five ($N>5$).

$\boldsymbol{R}$ and t_x and t_y can be computed from $\{v_1, v_2, v_3, v_4, v_5\}$.

Based on the orthogonal property of matrix $\boldsymbol{R}$ we can obtain

$$t_y^2=\frac{s-[s^2-4(v_1v_5-v_4v_2)^2]^{\frac{1}{2}}}{2(v_1v_5-v_4v_2)^2} \tag{3.38}$$

where $s=v_1^2+v_2^2+v_4^2+v_5^2$. If $v_1v_5-v_4v_2\neq 0$, we have

$$t_y^2=\frac{1}{v_1^2+v_2^2+v_4^2+v_5^2} \tag{3.39}$$

After t_y is solved, $\{r_1, r_2, t_x, r_4, r_5\}$ can be obtained from $\{v_1, v_2, v_3, v_4, v_5\}$ as

$$[r_1,\ r_2,\ t_x,\ r_4,\ r_5]^{\mathrm{T}}=[v_1t_y,\ v_2t_y,\ v_3t_y,\ v_4t_y,\ v_5t_y]^{\mathrm{T}} \tag{3.40}$$

The sign of t_y will be determined by the factor that the camera coordinates X_c and Y_c have the same sign as the computer image coordinates u' and v' as defined in Eq. (3.31), respectively

$$\left.\begin{aligned} X_c &= r_1X_w+r_2Y_w+t_x \\ Y_c &= r_4X_w+r_5Y_w+t_y \end{aligned}\right\} \tag{3.41}$$

Pick the sign of t_y to be positive, calculate (X_c, Y_c) for an arbitrary world coordinate point (X_w, Y_w) by using Eqs. (3.40) and (3.41). If sign(u') = sign(X_c) and sign(v') = sign(Y_c) we retain the sign of t_y, otherwise reverse the sign of t_y and recalculate $\{r_1, r_2, t_x, r_4, r_5\}$.

The rest of the components of $\boldsymbol{R}$ can be determined by using the orthonormal condition of the rotation matrix. There are two sets of solutions because of the sign uncertainty of the square root, resulting in different values of f. The correct one corresponds to a positive f that will be determined in stage (3)

$$\left.\begin{aligned} &r_3 = (1-r_1^2-r_2^2)^{\frac{1}{2}} \\ &r_6 = \operatorname{sign}(r_1r_4+r_2r_5)(1-r_4^2-r_5^2)^{\frac{1}{2}} \\ &[r_7,\ r_8,\ r_9]^{\mathrm{T}} = [r_1,\ r_2,\ r_3]^{\mathrm{T}} \times [r_4,\ r_5,\ r_6]^{\mathrm{T}} \end{aligned}\right\} \tag{3.42}$$

or

$$\left.\begin{aligned} &r_3 = -(1-r_1^2-r_2^2)^{\frac{1}{2}} \\ &r_6 = -\operatorname{sign}(r_1r_4+r_2r_5)(1-r_4^2-r_5^2)^{\frac{1}{2}} \\ &[r_7,\ r_8,\ r_9]^{\mathrm{T}} = [r_1,\ r_2,\ r_3]^{\mathrm{T}} \times [r_4,\ r_5,\ r_6]^{\mathrm{T}} \end{aligned}\right\} \tag{3.43}$$

(2) Computation of t_z, k, f.

Given that $\boldsymbol{R}$, t_x, and t_y are known, $\{t_z, f\}$ is solved from the following linear equations that can be derived from Eq. (3.29) by letting $k = 0$ (ignore lens deformation for linear solution) as

$$\begin{bmatrix} -u_i'(r_1X_{wi}+r_2Y_{wi}+t_x) \end{bmatrix} \begin{bmatrix} t_z \\ f \end{bmatrix} = u_i'(r_7X_{wi}+r_8Y_{wi}) \tag{3.44}$$

With calibration points $N \geq 3$ the above equation is an over-determined system of linear equations that can be solved for unknowns $\{t_z, f\}$ as the initial estimate by using a linear least squares method. Note that the calibration plane must not be exactly parallel to the image plane, otherwise Eq. (3.44) becomes linearly dependent. The accurate solutions for parameters $\{t_z, k, f\}$ can be obtained from the nonlinear equation Eq. (3.29) by using a standard optimization algorithm. The initial values for $\{t_z, f\}$ are obtained from the linear solutions of Eq. (3.44) and the initial value for k is set to be zero. It can be solved by using a nonlinear least squares method that minimizes the objective function $c\{k, f, t_z\}$

$$c(k,\ f,\ t_z) = \min \left\| v_i'(1+kr_i^2) - f\,\frac{r_4X_{wi}+r_5Y_{wi}+r_6Z_{wi}+t_y}{r_7X_{wi}+r_8Y_{wi}+r_9Z_{wi}+t_z} \right\|^2 \tag{3.45}$$

where r is defined by Eq. (3.31). Usually, only one or two iterations are needed.

3.3.3 Calibration with Multiple View Algorithms

Zhang (2000) proposed a flexible calibration algorithm in which a camera observes a planar calibration pattern from different points of view. The camera and the calibration pattern can be freely moved and the relative camera positions among different poses are unknown. The calibration procedure gives a closed form solution, followed by a nonlinear refinement based on the maximum likelihood criterion.

3.3.3.1 Mapping between 3D Calibration Planar Points and 2D Image Points

Based on the pinhole camera model, the mapping between a 3D point in the world coordinate frame to the image coordinate frame is given as

$$s\tilde{\boldsymbol{m}} = A[\boldsymbol{R}\ \boldsymbol{t}]\tilde{\boldsymbol{M}} \tag{3.46}$$

where $\tilde{\boldsymbol{m}} = [u,\ v,\ 1]^{\mathrm{T}}$ is the augmented image coordinate vector and $\tilde{\boldsymbol{M}} = [X_w,\ Y_w,\ Z_w,\ 1]^{\mathrm{T}}$ is the augmented vector for a 3D point in the world coordinate frame; ***s*** is an arbitrary scale factor; ***R*** and ***t*** are the rotation and translation components of the transformation matrix from the world coordinate frame to the camera coordinate frame. ***A*** is the camera intrinsic parameters matrix when the skew of the two image axes is considered as described in Eq. (3.8). We recall this here with a slightly different notation.

$$A=\begin{bmatrix} \alpha & r & u_0 \\ \upsilon & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.47}$$

Assume that all the calibration points are placed on a plane with $Z_w = 0$ and $\boldsymbol{R}=[\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]$. We have

$$s\tilde{\boldsymbol{m}} = A[\boldsymbol{r}_1\quad \boldsymbol{r}_2\quad \boldsymbol{r}_3\quad \boldsymbol{t}]\begin{bmatrix} X_w \\ Y_w \\ 0 \\ 1 \end{bmatrix} = A[\boldsymbol{r}_1\quad \boldsymbol{r}_2\quad \boldsymbol{t}]\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{3.48}$$

Assume that $\boldsymbol{H} = \lambda A[\boldsymbol{r}_1\quad \boldsymbol{r}_2\quad \boldsymbol{t}]$, where λ is the scale factor and $\tilde{\boldsymbol{M}} = [X_w,\ Y_w,\ 1]^{\mathrm{T}}$, we obtain the mapping between the 2D image points and the 3D calibration planar points as

$$\tilde{\boldsymbol{m}} = \boldsymbol{H}\tilde{\boldsymbol{M}} \tag{3.49}$$

Given a set of calibration points and their corresponding image coordinates, the transformation matrix ***H*** can be obtained by solving the linear Eq. (3.49). After the linear least squares solution is obtained, a maximum likelihood estimation method is used to solve ***H*** accurately by minimizing the objective function

$$c = \min \left\| \boldsymbol{m}_i - \boldsymbol{m}_i \right\|^2 \tag{3.50}$$

where $\boldsymbol{m}_i = \dfrac{1}{\bar{\boldsymbol{h}}_3^{\mathrm{T}}\boldsymbol{M}_i}\begin{bmatrix} \bar{\boldsymbol{h}}_1^{\mathrm{T}}\boldsymbol{M}_i \\ \bar{\boldsymbol{h}}_2^{\mathrm{T}}\boldsymbol{M}_i \end{bmatrix}$ with $\bar{\boldsymbol{h}}_i$ is the *i*-th row of ***H***.

The initial value of the nonlinear estimation is obtained from the linear least squares result.

3.3.3.2 Closed Form Solution

Assuming that $\boldsymbol{H} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3]$ where $\boldsymbol{h}_i$ is the i-th column vector of $\boldsymbol{H}$ we have

$$[\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \boldsymbol{h}_3] = \lambda \boldsymbol{A}[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{t}] \tag{3.51}$$

Since $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are orthonormal, i.e., $\boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{r}_2 = 0$ and $\boldsymbol{r}_1^{\mathrm{T}}\boldsymbol{r}_1 = \boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{r}_2 = 1$, we have

$$\left.\begin{aligned} &\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{h}_2 = 0 \\ &\boldsymbol{h}_1^{\mathrm{T}}\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{h}_1 = \boldsymbol{h}_2^{\mathrm{T}}\boldsymbol{A}^{-\mathrm{T}}\boldsymbol{A}^{-1}\boldsymbol{h}_2 \end{aligned}\right\} \tag{3.52}$$

These are the constraints on the lens intrinsic parameter matrix $\boldsymbol{A}$. To solve for $\boldsymbol{A}$ let

$$\boldsymbol{B} = \boldsymbol{A}^{-\mathrm{T}}\boldsymbol{A}^{-1} = \begin{bmatrix} B_{11} & B_{13} & B_{13} \\ D_{12} & D_{22} & D_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{1}{\alpha^2} & -\dfrac{\gamma}{\alpha^2\beta} & \dfrac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\dfrac{\gamma}{\alpha^2\beta} & \dfrac{\gamma^2}{\alpha^2\beta^2} + \dfrac{1}{\beta^2} & -\dfrac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \dfrac{v_0}{\beta^2} \\ \dfrac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\dfrac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \dfrac{v_0}{\beta^2} & \dfrac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \dfrac{v_0^2}{\beta^2} + 1 \end{bmatrix} \tag{3.53}$$

Assuming that the i-th column vector of $\boldsymbol{H}$ be $\boldsymbol{h}_i = [h_{i1} \quad h_{i2} \quad h_{i3}]^{\mathrm{T}}$ we have

$$\boldsymbol{h}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{h}_j = \boldsymbol{v}_{ij}^{\mathrm{T}}\boldsymbol{b} \tag{3.54}$$

where

$$\boldsymbol{v}_{ij} = [h_{i1}h_{j1} \quad h_{i1}h_{j2} + h_{i2}h_{j1} \quad h_{i2}h_{j2} \quad h_{i3}h_{j1} + h_{i1}h_{j3} \quad h_{i3}h_{j2} + h_{i2}h_{j3} \quad h_{i3}h_{j3}]^{\mathrm{T}}$$

$$\boldsymbol{b} = [B_{11} \quad B_{12} \quad B_{22} \quad B_{13} \quad B_{23} \quad B_{33}]^{\mathrm{T}}$$

Therefore, Eq. (3.52) can be rewritten as

$$\begin{bmatrix} \boldsymbol{v}_{12}^{\mathrm{T}} \\ (\boldsymbol{v}_{11} - \boldsymbol{v}_{22})^{\mathrm{T}} \end{bmatrix} \boldsymbol{b} = 0 \tag{3.55}$$

In Eq. (3.55) v_{ij} are known parameters that derived from the homography matrix $\boldsymbol{H}$ and we need to solve for $\boldsymbol{b}$ with six unknowns. Multiple images of the calibration planar targets are required to form over-determined linear equations to solve for $\boldsymbol{b}$. If n images of the calibration planar targets are observed, by stacking n such equations as Eq. (3.55), we obtain

$$\boldsymbol{V}\boldsymbol{b} = 0 \tag{3.56}$$

where $\boldsymbol{V}$ is a $2n\times 6$ matrix. When $n\geq 3$ we can solve for $\boldsymbol{b}$ as the eigenvector of $\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}$ associated with the smallest eigenvalue.

After $\boldsymbol{b}$ is obtained, the camera intrinsic parameters can be obtained from Eq. (3.53) as

$$\left.\begin{aligned}
&v_0 = (B_{12}B_{13} - B_{11}B_{23}) / (B_{11}B_{22} - B_{12}^2)\\
&\lambda = B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})] / B_{11}\\
&\alpha = \sqrt{\lambda / B_{11}}\\
&\beta = \sqrt{\lambda / B_{11}(B_{11}B_{22} - B_{12}^2)}\\
&\gamma = -B_{12}\alpha^2\beta / \lambda\\
&u_0 = \gamma v_0 / \alpha - B_{13}\alpha^2 / \lambda
\end{aligned}\right\} \tag{3.57}$$

Once A is determined by Eq. (3.47), the extrinsic parameters of the camera can be derived from Eq. (3.51) as

$$\left.\begin{aligned}
&\boldsymbol{r}_1 = \lambda A^{-1}\boldsymbol{h}_1\\
&\boldsymbol{r}_2 = \lambda A^{-1}\boldsymbol{h}_2\\
&\boldsymbol{r}_3 = \boldsymbol{r}_1 \times \boldsymbol{r}_2\\
&\boldsymbol{t} = \lambda A^{-1}\boldsymbol{h}_3
\end{aligned}\right\} \tag{3.58}$$

where $\lambda = 1 / \left\|A^{-1}\boldsymbol{h}_1\right\| = 1 / \left\|A^{-1}\boldsymbol{h}_2\right\|$. Due to the measurement noise obtained, r_1, r_2, r_3 may not satisfy the orthonormal property of the rotation matrix. They can be orthonormalized through the singular value decomposition (SVD) procedure (Golub, 1996).

3.3.3.3 Nonlinear Solutions

Accurate nonlinear solutions to the parameters can be obtained by using the maximum-likelihood estimation method. Given n poses of the camera, there are m calibration points for each pose. The maximum-likelihood estimation can be implemented by minimizing the following objective function:

$$c(\boldsymbol{A},\ \boldsymbol{R}_i,\ \boldsymbol{t}_i) = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\|\boldsymbol{m}_{ij} - \boldsymbol{m}(\boldsymbol{A},\ \boldsymbol{R}_i,\ \boldsymbol{t}_i,\ \boldsymbol{M}_j)\right\| \tag{3.59}$$

where $\boldsymbol{m}(\boldsymbol{A},\ \boldsymbol{R}_i,\ \boldsymbol{t}_i,\ \boldsymbol{M}_j)$ is the projection of point M_j in image i, according to Eq. (3.49). This is a nonlinear optimization problem that can be solved with the Levenberg-Marquardt algorithm (Press et al., 1994). An initial guess at the solution $\{\boldsymbol{A}, \boldsymbol{R}_i, \boldsymbol{t}_i\}$ can be obtained with the method described above.

If the radical distortion is considered, we have

$$\left.\begin{aligned}
&\tilde{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]\\
&\tilde{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]
\end{aligned}\right\} \tag{3.60}$$

where k_1 and k_2 are the coefficients of the radical distortion. (x, y) are analogy image coordinates. All the parameters including radical distortion can be solved by minimizing the objective function

$$c(\boldsymbol{A}, \boldsymbol{R}_i, \boldsymbol{t}_i, k_1, k_2) = \min \sum_{i=1}^{n} \sum_{j=1}^{m} \left\| \boldsymbol{m}_{ij} - \boldsymbol{m}(\boldsymbol{A}, \boldsymbol{R}_i, \boldsymbol{t}_i, k_1, k_2 \boldsymbol{M}_j) \right\| \tag{3.61}$$

where $\boldsymbol{m}(\boldsymbol{A}, \boldsymbol{R}_i, \boldsymbol{t}_i, k_1, k_2, \boldsymbol{M}_j)$ is the projection point M_j in image i according to Eq. (3.49), followed by the distortion model in Eq. (3.60). The initial values of k_1 and k_2 can be set as zeros.

The example of the calibration planar target is shown in Fig. 3.7, with different points of view and different distances to the target plane.

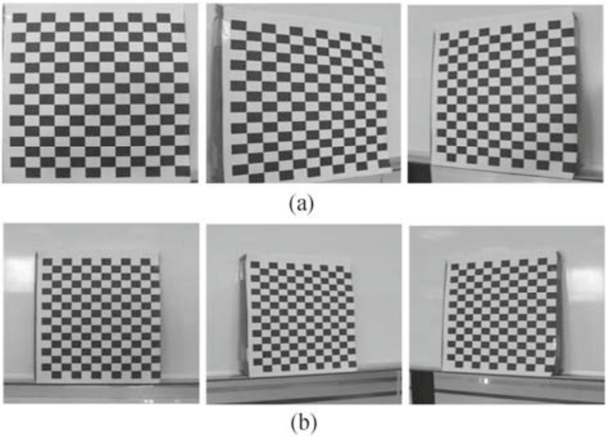


Fig. 3.7. Two sets of images taken at different distances to the calibration pattern. (a) Three images from the first set taken at a closer distance; (b) Three images from the second set taken at a larger distance

3.4 Calibration of Laser Stripe Sensor

There are two types of laser stripe sensor models, as mentioned in Section 3.1. One is based on the structure of the laser stripe sensor and the other is based on the input-output black box, regardless of the internal structure. We first derive their general mathematical forms and then review some calibration methods.

For the first one, the mathematical model of the laser stripe sensor consists of the model of the camera and the model of the laser plane. The model for the camera that has been discussed in Section 3.2 is the relationship between a 3D point and its image coordinate, as indicated by Eq. (3.11). The model for the laser plane is the equations for defining the plane either in the camera coordinate or in the world coordinate. For a single laser stripe sensor, the plane equation is given by

$$aX_w + bY_w + cZ_w + d = 0 \tag{3.62}$$

where (a, b, c, d) are the plane parameters that need to be determined through the calibration procedure.

Depending on the camera model, the model for a laser stripe sensor has a slightly different format. If the DLT model for the camera is used, a laser stripe sensor is described by the combination of Eqs. (3.26) and (3.62)

$$\left.\begin{aligned} u+\delta_u(u,v) &= \frac{L_1X_w+L_2Y_w+L_3Z_w+L_4}{L_9X_w+L_{10}Y_w+L_{11}Z_w+1} \\ v+\delta_v(u,v) &= \frac{L_5X_w+L_6Y_w+L_7Z_w+L_8}{L_9X_w+L_{10}Y_w+L_{11}Z_w+1} \\ aX_w+bY_w+cZ_w+d &= 0 \end{aligned}\right\} \tag{3.63}$$

Its matrix format is given by

where

$$\boldsymbol{Ax}=\boldsymbol{b}$$

$$\boldsymbol{A}=\begin{bmatrix} \tilde{u}L_9-L_1 & \tilde{u}L_{10}-L_2 & \tilde{u}L_{11}-L_3 \\ \tilde{v}L_9-L_5 & \tilde{v}L_{10}-L_6 & \tilde{v}L_{11}-L_7 \\ a & b & c \end{bmatrix} \tag{3.64}$$

$$\boldsymbol{x}=\begin{bmatrix} X_w \\ I_w \\ Z_w \end{bmatrix}, \boldsymbol{b}=\begin{bmatrix} L_4-\tilde{u} \\ L_8-\tilde{v} \\ -d \end{bmatrix}$$

and (u, v) is defined by Eqs. (3.12) and (3.13).

Similarly, if the RAC camera model is used, the laser stripe scanner equation is given by

$$\left.\begin{aligned} u'(1+kr^2) &= f_x\frac{r_1X_w+r_2Y_w+r_3Z_w+t_x}{r_7X_w+r_8Y_w+r_9Z_w+t_z} \\ v'(1+kr^2) &= f_y\frac{r_4X_w+r_5Y_w+r_6Z_w+t_y}{r_7X_w+r_8Y_w+r_9Z_w+t_z} \\ aX_w+bY_w+cZ_w+d &= 0 \end{aligned}\right\} \tag{3.65}$$

Its matrix format is given by

$$\left.\begin{aligned} &\boldsymbol{Ax}=\boldsymbol{b}\\ &\boldsymbol{A}=\begin{bmatrix} \tilde{u}r_7-f_xr_1 & \tilde{u}r_8-f_xr_2 & \tilde{u}r_{11}-f_xr_3\\ \tilde{v}r_7-f_yr_4 & \tilde{v}r_8-f_yr_5 & \tilde{v}r_9-f_yr_6\\ a & b & c \end{bmatrix}\\ &aX_w+\boldsymbol{x}=\begin{bmatrix} X_w\\ Y_w\\ Z_w \end{bmatrix},\boldsymbol{b}=\begin{bmatrix} f_xt_x-\tilde{u}t_z\\ f_yt_y-\tilde{v}t_z\\ -d \end{bmatrix}bY_w+cZ_w+d=0 \end{aligned}\right\} \tag{3.66}$$

where $\tilde{u}=u'(1+kr^2)$, $\tilde{v}=v'(1+kr^2)$ and parameters $\{u', v', k, r\}$ are defined by Eq. (3.31).

Given a measured image coordinate (u, v), the reconstructed 3D point (X_w, Y_w, Z_w) can be calculated as $\boldsymbol{x}=\boldsymbol{A}^{-1}\boldsymbol{b}$.

A special case for the model as described by Eqs. (3.63) and (3.65) is when the laser plane coincides with the XY plane of the world coordinate frame, ie., $Z_w=0$. In this case, calibration of the laser plane can be achieved by just calibrating the camera extrinsic parameters.

For the second type of model of the laser stripe sensor, the laser plane and image plane can be mapped with a general mapping function.

Assume that the laser plane is the XY plane of the reference coordinate system. Let (X_{wi}, Y_{wi}) be a coordinate in the laser plane and (u_i, v_i) the corresponding image coordinate. The mapping between those two planes is given by

$$\begin{bmatrix} X_w\\ Y_w \end{bmatrix}=\boldsymbol{M}\begin{bmatrix} u\\ v \end{bmatrix} \tag{3.67}$$

where $\boldsymbol{M}$ is a general mapping function.

Table 3.1 summarizes various calibration methods of laser stripe sensor in terms of the form of calibration target, method for extracting control points for camera calibration and laser plane calibration, and calibration algorithms.

Table 3.1 Summary of laser stripe sensor calibration method

Method	Form of calibration target	Method to extract control points for camera calibration	Method to extract control points for laser calibration	Calibration algorithm and reference
(1) Strained threads	Non-collinear multiple threads (lines)		Control points are the intersection points between the non-planar multiple thin threads strained in space and the laser plane. Position of the control points are measured by using a theodolite measurement system	Plane-to-Plane coordinate mapping (Dewar, 1988; James, 1988)

(to be continued)

(Table 3.1)

Method	Form of calibration target	Method to extract control points for camera calibration	Method to extract control points for laser calibration	Calibration algorithm and reference
(2) Zigzag face	Zigzag-like face		Control points are the intersection points between the ridge of zigzag face and the laser plane. The laser plane is perpendicular to the zigzag	Plane-Plane coordinate mapping (Duan et al., 2000)
(3) Neural networks	Moving gauge block or photo detector		Control points on the laser plane are directly measured by using a gauge block or photo detector	Plane-Plane coordinate mapping based on neural networks or least squares polynomial fitting (Trucco et al., 1994; Chang, 1995; Zhang and Wei, 2002; Fan, 2001)
(4) Line constraint	Multiple cross lines		Control points are not determined explicitly. Instead, all the control points satisfy line constraints that are used to solve homogenous 3×4 transform matrix	Plane-plane coordinate mapping (Chen and Kak, 1987)
(5) Plane constraint	Multiple planes		Control points are not determined explicitly. Instead, all the control points satisfy plane constraints that are used to solve homogenous 3×4 transform matrix	Plane-Plane coordinate mapping (DePiero, 1995; Reid, 1996)
(6) 3D Target with ray tracing	Two or three perpendicular planes	The control points are defined by the markers on the surfaces of the calibration target	A control point is created by the intersection of a ray with the calibration plane. The ray is the connection of a laser point and the projection center	Two steps calibration method: camera and laser plane are calibrated separately (McIvor, 2002)
(7) 3D target with invariance of cross ratio	Two or three perpendicular planes	The control points are defined by the corner position of blocks on the calibration planes	The control points on the laser plane are determined by using the invariance of the cross ratio	Two steps calibration method (Xu, 1995; Huynh, 1999; Zhou, 2005)

(to be continued)

(Table 3.1)

Method	Form of calibration target	Method to extract control points for camera calibration	Method to extract control points for laser calibration	Calibration algorithm and reference
(8) 2D plane with controlled movement	Plane on movable stage	The control points are defined by the markers on the planar target. Non-Coplanar control points are obtained by moving the planar target with the precision control	A control point is created by the intersection of a ray with the calibration plane. The ray is the connection of a point on the laser line and the camera projection center. Multiple non-collinear points are obtained by moving the planar target	Two steps calibration method (Tiddeman, 1998; Li et al., 2007; Li et al., 2008)
(9) 2D plane with free movement	Plane with unknown position	The control points are defined by the markers on the planar target. Non-Coplanar control points are obtained by moving the planar or camera freely	There are two methods to acquire laser control points. One is to use ray-plane intersection method as used in method (8). The other is to use the invariance of the cross ratio or double cross ratio	Two steps calibration method (Wei and Zhang, 2003; Zhou and Zhang, 2004)

The following will give a review of the individual calibration method.

3.4.1 Laser Stripe Plane Calibration with Two Known Planes

As shown in Fig. 3.8, a calibration target with two planes is used to calibrate the camera and laser stripe plane. The markers on the target are used to calibrate the camera parameters and two planes defined in the world coordinate frame can be used to calibrate the laser stripe plane relative to the world coordinate frame.

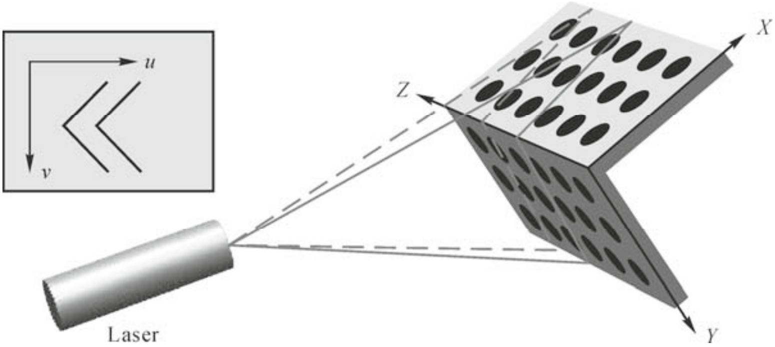


Fig. 3.8. Calibration target with two planes to calibrate the laser stripe plane

In a general case we assume multiple planes are used to calibrate the laser plane and they are defined in the world coordinate frame as

$$a_k X_w + b_k Y_w + c_k Z_w = 1 \tag{3.68}$$

where $\{a_k, b_k, c_k\}$ ($k = 1, \ldots, n$) are k-th plane parameters. They are known parameters. Combining the plane equation Eq. (3.68) and the camera DLT model Eq. (3.26), the laser line points coordinate (X_{wi}, Y_{wi}, Z_{wi}) ($i = 1, \ldots, n$) can be obtained from their image coordinates (u_i, v_i) ($i = 1, \ldots, n$) by using Eq. (3.64). Since, for each calibrate plane the reconstructed point set (X_{wi}, Y_{wi}, Z_{wi}) is collinear, at least two calibration planes (i.e., $k \geq 2$) are needed to create non-collinear points that determine the laser plane position by best fitting the reconstructed points.

3.4.2 Laser Stripe Plane Calibration Based on Invariance of Cross Ratios

Invariance of Cross Ratios

In order to obtain a set of calibration points, the cross ratios invariance principle is used (Huynh et al., 1999). In Fig. 3.9, there are four 3D points ***A***, ***B***, ***C*** and ***D*** lying on a line, their corresponding projective points are ***a***, ***b***, ***c***, and ***d***, respectively. ***O*** is the perspective projection center.

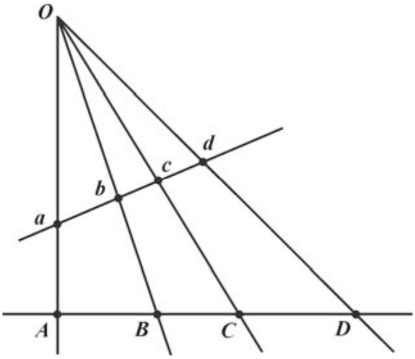


Fig. 3.9. Invariance of collinearity and cross ratios under perspective projection

Collinearity and cross ratio are known to be invariant under perspective projection (Semple and Kneebone, 1952). This invariant property can be used to determine uniquely the coordinate of a point that is on the same line with three other points if the cross ratio's of those four points and the coordinates of three other points are known. If four world points ***A***, ***B***, ***C***, and ***D*** are collinear in space, their corresponding perspective projection points or image points are also collinear. The cross ratios of the two sets of points are identical. Since ***a***, ***b***, ***c*** and ***d*** are lying on a line, any point on the line can be expressed by the parameter equation as

$$\boldsymbol{x} = \boldsymbol{a} + (\boldsymbol{b} - \boldsymbol{a})\boldsymbol{t} \tag{3.69}$$

where t is the parameter that defines points on the line. For example, point b corresponds to t_b=1 and point $\boldsymbol{a}$ results in t_a=0. The cross ratios $\{\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{c}, \boldsymbol{d}\}$of these points are defined as

$$r = \{\boldsymbol{a}, \boldsymbol{b}; \boldsymbol{c}, \boldsymbol{d}\} = \left.\frac{\overline{ac}}{\overline{bc}}\right/ \frac{\overline{ad}}{\overline{bd}} \tag{3.70}$$

where $\overline{ac}$ denotes the distance between point $\boldsymbol{a}$ and point $\boldsymbol{c}$; similar notations are pled to the other three distances. Using the parameters expression we have

$$r = \{a, b; c, d\} = \left.\left(\frac{t_a - t_c}{t_b - t_c}\right)\right/ \left(\frac{t_a - t_d}{t_b - t_d}\right) \tag{3.71}$$

Similarly, the cross ratios $\{\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{C}, \boldsymbol{D}\}$of world points $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ are given by

$$R = \{\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{C}, \boldsymbol{D}\} = \left.\frac{\overline{AC}}{\overline{BC}}\right/ \frac{\overline{AD}}{\overline{BD}} = r \tag{3.72}$$

Given that the cross ratio is r and that the coordinates of the three world points $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are known, the coordinate of the fourth world point $\boldsymbol{D}$ can be calculated as

$$t_D = \frac{r(t_B - t_C)t_A - t_B(t_A - t_C)}{r(t_B - t_C) - (t_A - t_C)} \tag{3.73}$$

This invariant property of the cross ratio will be used to determine the world points that are located on the laser stripe plane.

3.4.2.2 Invariance of Double Cross Ratio

In order to create more calibration points with limited known points, an invariant of the double cross ratio method was proposed (Wei et al., 2003). As shown in Fig. 3.10, there are three lines $A_1B_1C_1$, $A_2B_2C_2$, and $A_3B_3C_3$ on calibration plane π_c, the other line $D_1D_2D_3$ on the same plane intersects with the previous three lines at points D_1, D_2, and D_3, respectively. This line is created by intersecting the laser stripe plane and the calibration plane. And their images on plane π_i with respective to perspective projection center O are $a_1b_1c_1$, $a_2b_2c_2$, $a_3b_3c_3$, and $d_1d_2d_3$, respectively.

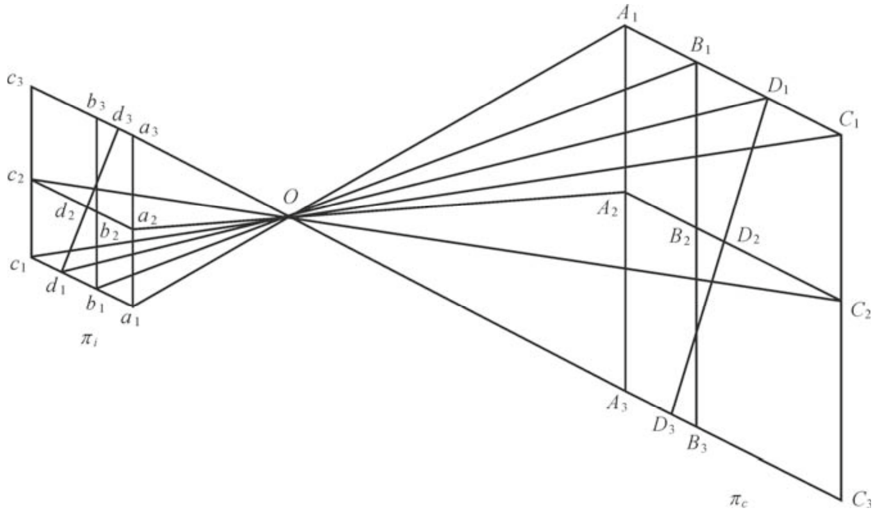


Fig. 3.10. Invariance of the cross ratio to determine the world points on the laser stripe plane

Based on the invariance of the cross ratio, we have

$$R(\boldsymbol{A}_i,\ \boldsymbol{B}_i;\ \boldsymbol{C}_i,\ \boldsymbol{D}_i) = r(\boldsymbol{a}_i,\ \boldsymbol{b}_i;\ \boldsymbol{c}_i,\ \boldsymbol{d}_i)\quad i = 1,\ 2,\ 3 \tag{3.74}$$

Once world points $\boldsymbol{A}_i$, $\boldsymbol{B}_i$, $\boldsymbol{C}_i(i = 1, 2, 3)$ and image points $\boldsymbol{a}_i$, $\boldsymbol{b}_i$, $\boldsymbol{c}_i$, $\boldsymbol{d}_i$ $(i = 1, 2, 3)$ are known, point $\boldsymbol{D}_i$ can be obtained from Eq. (3.74) . Choose an arbitrary point $\boldsymbol{D}_4$ on the line $D_1D_2D_3$. The perspective projection point is $\boldsymbol{d}_4$ for $\boldsymbol{D}_4$ with respect to center $\boldsymbol{O}$. Using invariance of the cross ratio again we have

$$R(\boldsymbol{D}_1, \boldsymbol{D}_2; \boldsymbol{D}_3, \boldsymbol{D}_4) = r(\boldsymbol{d}_1, \boldsymbol{d}_2; \boldsymbol{d}_3, \boldsymbol{d}_4) \tag{3.75}$$

Using the invariance of the cross ratio relation in Eq. (3.74) we can obtain points $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, and $\boldsymbol{D}_3$. Using the invariance of the cross ratio Eq. (3.75), $\boldsymbol{D}_4$ can be obtained given $\boldsymbol{d}_1$, $\boldsymbol{d}_2$, $\boldsymbol{d}_3$ and $\boldsymbol{d}_4$. Repeat the preceding action, an arbitrary number of points on line $D_1D_2D_3$ can be obtained.

3.4.2.3 Selection of Calibration Target

The calibration target, that utilizes the invariance of the cross ratio to determine the calibration points, is shown in Fig. 3.11.

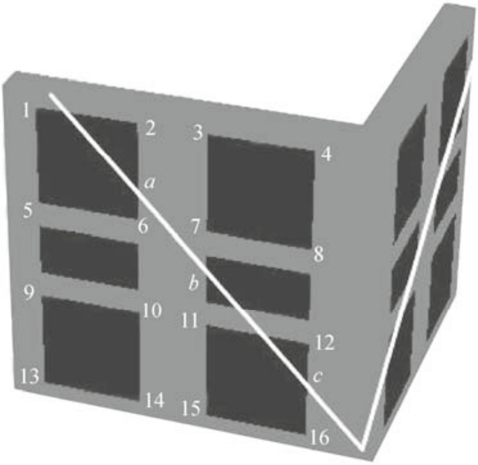


Fig. 3.11. Calibration target with two grid planes

The calibration target consists of two grid planes that are perpendicular to each other. On each plane are four black squares and two rectangles. Their edges are aligned vertically and horizontally in the world coordinate frame. Each corner of the black pattern can be used as the calibration point, since its space location in the reference coordinate frame is well defined and its corresponding image coordinate is easily identified with high accuracy. Therefore, the calibration target can provide in total 48 non-coplanar calibration points that can be observed from a single camera viawpoint. The two white lines on the two planes of the calibration target form the intersection line of the laser stripe plane and the target planes. Positions of the points on the intersection line can be determined based on the principle of the invariance of the cross ratio. For example, the position of point ***a***, (ie,) the intersection between the laser line and the black square, is determined by positions of points 2, 6, and 10. Similarly, the positions of point ***b*** and ***c*** are determined by the positions of points 3, 7, 11, and points 4, 12,16, respectively. All the other intersection points on the line can be determined in the same way. Those intersection points, including ***a***, ***b*** and ***c***, can be used to determine the laser line on the first calibration plane. More points on the line can be determined by using the invariance of the cross ratio again, that is to use the formally determined points like ***a***, ***b*** and ***c*** to determine other arbitrary points on the line. When enough points on the two lines (i.e., ten points for each line) are calculated, the laser stripe plane can be determined by using the best fit to create its plane equation.

Determination of Laser Stripe Plane

The plane equation Eq. (3.62) can be written in the form of three independent variables $\{a', b', c'\}$

$$\left.\begin{aligned} a'X_w + b'Y_w + c'Z_w = 1 \\ a' = -a/d,\ b' = -b/d,\ c' = -c/d \end{aligned}\right\} \tag{3.76}$$

Once the points of the laser stripe plane are calculated, the plane parameters in Eq. (3.76) can be determined by solving the linear equations in the form of

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$\boldsymbol{A} = \begin{bmatrix} X_{w1} & Y_{w1} & Z_{w1} \\ \vdots & \vdots & \vdots \\ X_{wm} & X_{wm} & X_{wm} \end{bmatrix}_{m\times 3},\ \boldsymbol{x} = \begin{bmatrix} a' \\ \upsilon \\ c' \end{bmatrix}_{3\times 1},\ \boldsymbol{b} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{m\times 1} \tag{3.77}$$

where $\{X_{wi}, Y_{wi}, Z_{wi}\}$ (i=1, 2, ..., m) are the measured points on the laser stripe plane. The linear least squares solution of Eq. (3.77), thus plane parameters in Eq. (3.62), are given by

$$\boldsymbol{x} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$$

$$\left.\begin{aligned} d = 1/\sqrt{a'^2 + b'^2 + c'^2} \\ a = -a'd \\ b = -b'd \\ c = -c'd \end{aligned}\right\} \tag{3.78}$$

This is equivalent to minimizing the error function $\min \|\boldsymbol{Ax} - \boldsymbol{b}\|^2$ between the measured points and the plane, that is minimizing the objective function

$$\left.\begin{aligned} e(a,\ b,\ c,\ d) = \min \sum_{i=1}^{m} d_i^2 \\ d_i = \frac{|aX_{wi} + bY_{wi} + cZ_{wi} + d|}{(a^2 + b^2 + c^2)^{1/2}} \end{aligned}\right\} \tag{3.79}$$

The solution $\{a, b, c, d\}$ can be obtained by using a nonlinear least squares method like the Levenberg-Marquardt algorithm (Press et al., 1994).

3.4.3 Laser Plane Calibration with a Planar Target

In Section 3.3.3, a method of camera calibration with a planar target is discussed, in which the target is observed from different points of view. Similarly, the laser stripe plane can be determined by using a planar target that is seen with multiple views (Zhou et al., 2004; Zhou and Zhang, 2005). In this case, the position of the laser stripe plane is defined in the camera coordinate frame and can be easily transferred to the world coordinate frame, if needed.

3.4.3.1 Determination of the Control Points Based on the Invariance of the Cross Ratio

As shown in Fig. 3.12, a planar target with grid pattern is used. All the square blocks are aligned vertically and horizontally. Positions of all the corner points are known in the local world coordinate frame. A grayscale line indicates the intersection between the laser stripe plane and the target plane. The intersection points between the line and black block can be determined, based on the invariance of the cross ratio as described in the previous section. If only one view of the calibration planar target is used, collinear calibration points are obtained that will not determine the laser stripe plane. We must construct the calibration control points by using multiple views of the same planar calibration target from a different orientation.

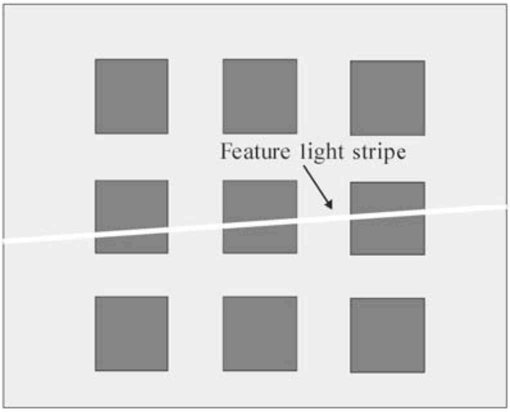


Fig. 3.12. Calibration target with one grid plane

As discussed in Section 3.2, the camera intrinsic parameters and extrinsic parameters can be calibrated with multiple views of such a planar pattern. All the views of the planar target are acquired by the same camera with different positions, thus each view corresponds to the different extrinsic parameters and the same intrinsic parameters. For each view, the feature points on the laser line are transferred into the camera coordinate frame. All the featured points calculated from different local coordinate frames form a laser stripe plane, thus determine the laser plane equation in the camera coordinate frame. Once the camera intrinsic parameters are determined, the transform between the local world coordinate frame, where the calibration planar target is defined, to the camera coordinate frame can be derived.

Assume that we have m views of the planar calibration target and n control points for each view. Let $\tilde{\boldsymbol{M}}^i_{wj} = (X^i_{wj}, Y^i_{wj}, Z^i_{wj}, 1)^{\mathrm{T}}$ and $\tilde{\boldsymbol{M}}^i_{cj} = (X^i_{cj}, Y^i_{cj}, Z^i_{cj}, 1)^{\mathrm{T}}$ be the homogenous coordinate of the j-th control point ($j = 1, \ldots, n$) in the i-th local world coordinate frame (i.e., i-th view of the target plane) ($i = 1, \ldots, m$) and the camera coordinate frame, respectively. They are related by

$$\tilde{\boldsymbol{M}}_{cj}^{i} = \boldsymbol{T}^{i}\tilde{\boldsymbol{M}}_{wj}^{i}$$
$$\boldsymbol{T}^{i} = \begin{pmatrix} \boldsymbol{R}^{i} & \boldsymbol{t}^{i} \\ 0 & 1 \end{pmatrix} \tag{3.80}$$

where $\boldsymbol{R}^i$ and $\boldsymbol{t}^i$ are the rotation matrix and translation vector from the i-th local world coordinate frame to the camera coordinate frame. Control points are on the laser line for each view. $\boldsymbol{T}^i$ is determined by Eq. (3.58) as discussed in Section 3.2.4. All the control points are the collections from different views

$$\tilde{\boldsymbol{M}}_{c}^{k}(k = 1, \ ..., \ n\times m) = \cup\tilde{\boldsymbol{M}}_{cj}^{i} \tag{3.81}$$

where $\tilde{\boldsymbol{M}}_{c}^{k} = (X_{ck}, \ Y_{ck}, \ Z_{ck}, \ 1)^{\mathrm{T}} \ (k = 1, \ ..., \ n\times m)$ can best fit a laser stripe plane in the camera coordinate frame. The procedure is the same as one discussed in the previous section.

A similar method, with three collinear points on the calibration planar target to determine the laser plane has been proposed (Han et al., 2009).

3.4.3.2 Determination of the Control Points based on Ray Tracing

As we have discussed earlier, the calibration procedure is to find control points on the laser stripe plane. Besides using the invariance of the cross ratio to find the control points on the laser plane we can use the ray tracing method. The control point is determined by using the intersection between the ray, which is the connection of the image point and the perspective center, and the calibration target plane.

Assume that the camera intrinsic and extrinsic parameters have been calibrated with a planar target, as described in Section 3.2.4 with Zhang's method (Zhang, 2000), and that there is a control point $\boldsymbol{M}_w = [X_w, Y_w, Z_w]^{\mathrm{T}}$ that is located on the laser line on the calibration plane. Its camera coordinate and image coordinate are $\boldsymbol{M}_c = [X_c, Y_c, Z_c]^{\mathrm{T}}$ and $\boldsymbol{m} = [u, v]^{\mathrm{T}}$, respectively. The plane equation of the calibration planar target that is originally defined in the local world coordinate frame can be converted by using Eq. (3.58) into the camera coordinate frame as

$$aX_c + bY_c + cZ_c + d = 0 \tag{3.82}$$

Image point $\boldsymbol{m} = [u, v]^{\mathrm{T}}$ and its camera coordinate $\boldsymbol{M}_c = [X_c, Y_c, Z_c]^{\mathrm{T}}$ are satisfied with the perspective projection relationship described in Eq. (3.6), that is

$$\left.\begin{aligned} u - u_0 &= k_u \frac{X_c}{Z_c} \\ v - v_0 &= k_v \frac{Y_c}{Z_c} \end{aligned}\right\} \tag{3.83}$$

where $\{u_0, v_0, k_u, k_v\}$ are the camera intrinsic parameters. Combining Eqs. (3.82) and (3.83), camera coordinate $\boldsymbol{M}_c = [X_c, Y_c, Z_c]^{\mathrm{T}}$ can be expressed by image coordinate $\boldsymbol{m} = [u, v]^{\mathrm{T}}$ as

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} k_u & 0 & -(u-u_0) \\ 0 & k_v & -(v-v_0) \\ a & b & c \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -d \end{bmatrix} \tag{3.84}$$

The determination of the laser stripe plane with the control points can be derived in the same way as described in the previous section.

3.4.4 Calibration of Dual Laser Stripe Sensor

In some applications, dual laser stripe sensors are used to improve the reliability of the laser scanning process. The calibration process needs to calibrate both laser stripe sensors in a common world coordinate frame. As shown in Fig. 3.13, dual laser stripe sensors consist of one laser projector and two cameras. The laser projector and each camera form a standard laser stripe sensor. The calibration procedure not only calibrates each system but also finds the relationship between the two systems. That can be done by using a common calibration planar target.

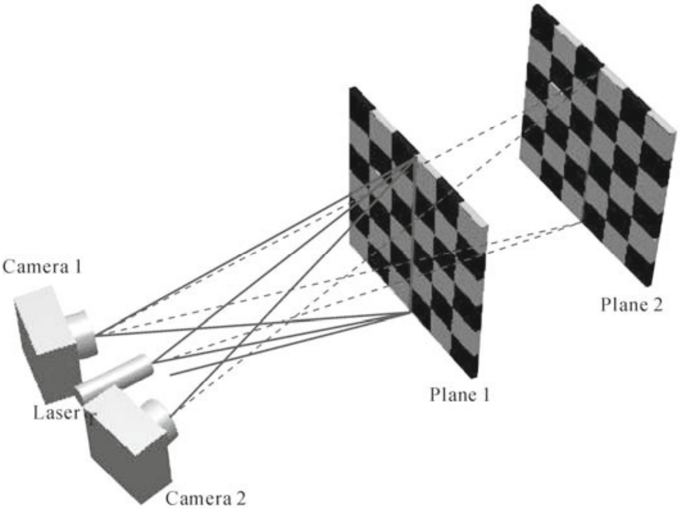


Fig. 3.13. Calibration setup for dual laser stripe sensor

3.4.5 Calibration of the Rotation Table

When a laser stripe sensor is used to scan the enclosed surface of an object, a rotation table is often required. Calibration of the rotation table is for determining the rotation axis relative to the world coordinate frame. There are basically two methods that can be used to determine the rotation axis. One is based on the camera measurement of the laser stripe sensor and the other is based on laser measurement. In the first method, a planar calibration target is placed on the rotation table as shown in Fig. 3.14. The camera of a laser stripe sensor is used to observe the calibration pattern and to determine the calibration pattern position relative to the camera coordinate frame.

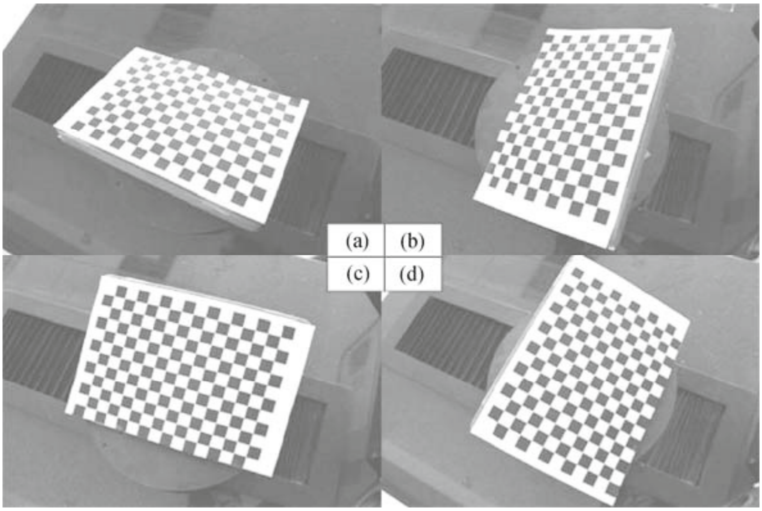


Fig. 3.14. Illustration of the rotation table calibration. (a) A planar calibration target is placed on the rotation table with a tilt angle; (b) – (d) The camera of the stripe sensor observes the target pattern for multiple rotation angles

The first step is to calibrate the camera intrinsic parameters. This can be done by Zhang's calibration method (Zhang, 2000) using a planar calibration target. The planar target is viewed multiple times when the rotation table is in different angle positions. Once the camera intrinsic parameters are known, the homograph matrix can be calculated by Eq. (3.49), and the transformation between the calibration target coordinate frame and the camera coordinate frame can be obtained by using Eq. (3.58). The relationship between the point in the local world coordinate frame $\boldsymbol{M}_w = [X_w, Y_w, 0]^{\mathrm{T}}$ and the camera coordinate frame $\boldsymbol{M}_c^i = [X_c^i, Y_c^i, Z_c^i]^{\mathrm{T}}$, $(i = 1, ..., n)$ is given by

$$\begin{bmatrix} X_c^i \\ Y_c^i \\ Z_c^i \end{bmatrix} = \boldsymbol{R}^i \begin{bmatrix} X_w \\ Y_w \\ 0 \end{bmatrix} + \boldsymbol{t}^i \tag{3.85}$$

where n is the number of rotation table positions; $\boldsymbol{R}^i$ and $\boldsymbol{t}^i$ are the rotation matrix and the translation vector of the transform from the world coordinate frame to the camera coordinate frame, when the rotation table is in the i-th position. Therefore, the rotation axis can be obtained by using the circle fitting of the points $\boldsymbol{M}_c^i = [X_c^i,\ Y_c^i,\ Z_c^i]^{\mathrm{T}}\ (i=1,\ \ldots,\ n)$. The plane normal of the circle is the orientation of the axis and the circle center will pass through the axis.

Similar methods are reported in (Wu, 2008).

Another method is to use a sphere as the calibration target and its position is measured by the laser stripe sensor. As shown in Fig. 3.15, a sphere is located on the rotation table. A sphere is scanned by moving the mechanical fixture that holds the laser stripe sensor, and the sphere center is calculated by the scanned point using the sphere fitting algorithm. Rotate the table onto the next angular position and repeat the scanning process and calculate the fitted sphere center. All the fitted sphere centers are used to fit a circle whose center axis gives the location of the axis of the rotation table.

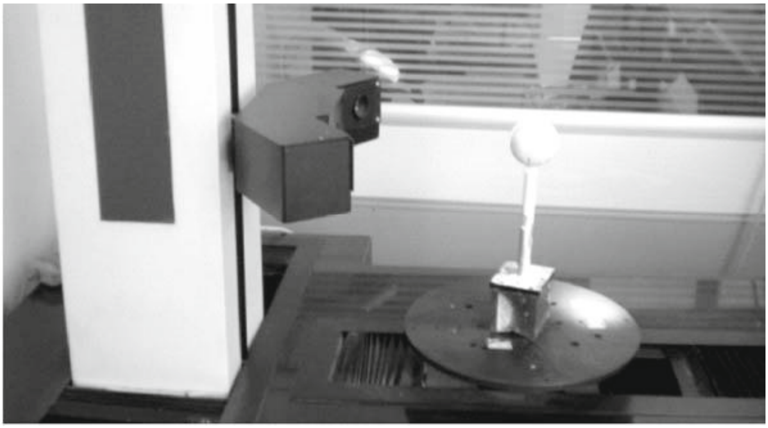


Fig. 3.15. Calibration of rotation table with a sphere

3.4.6 Calibration of the Laser Stripe Sensor with Robot Alignment

When the laser stripe sensor is used with the robot, the flexibility of the robot arm can be utilized. The pose of the laser stripe sensor can be easily controlled through the movement of the robot arm to create a certain geometrical relationship with the calibration target to decouple the calibration parameters. The manual alignment of the laser plane position through the robot movement can simplify the calibration procedure (Lin et al., 2007).

The calibration setup is illustrated in Fig. 3.16. The calibration target is a plate with a row of dots on it. The world coordinate system $\{W\}$ is defined on the calibration plate with the Y axis aligned with the direction of dots and the Z axis is perpendicular to the plate. The procedure is as follows:

(1) Place the calibration plate on the *XY* plane of the robot base coordinate system $\{B\}$. Allow the *Y* axis of the world coordinate frame $\{W\}$ to be aligned with the *Y* axis of the robot based frame $\{B\}$.

(2) Control the robot movement and make the laser beam align with the *Y* axis of the world coordinate frame $\{W\}$ by aligning the laser beam with the dots on the calibration plate, as shown in Fig. 3.16.

(3) Move the robot along the *Z* axis of the base frame $\{B\}$ and observe if the laser beam is still aligned with the dots on the calibration plate. If it is off alignment, rotate the robot tool around the *Y* axis of the robot base frame $\{B\}$ and translate along the *X* axis of $\{B\}$ to make it realigned. Repeat the procedure until the laser beam stays aligned with the dots while the robot is moving along the *Z* axis, indicating that the laser beam position in the world coordinate $\{W\}$ is perpendicular to the *XY* plane.

(4) Turn the laser beam off and index the robot along the *Z* axis and take images of the calibration dots for each height of the robot. Calculate the position of the dots in the image plane.

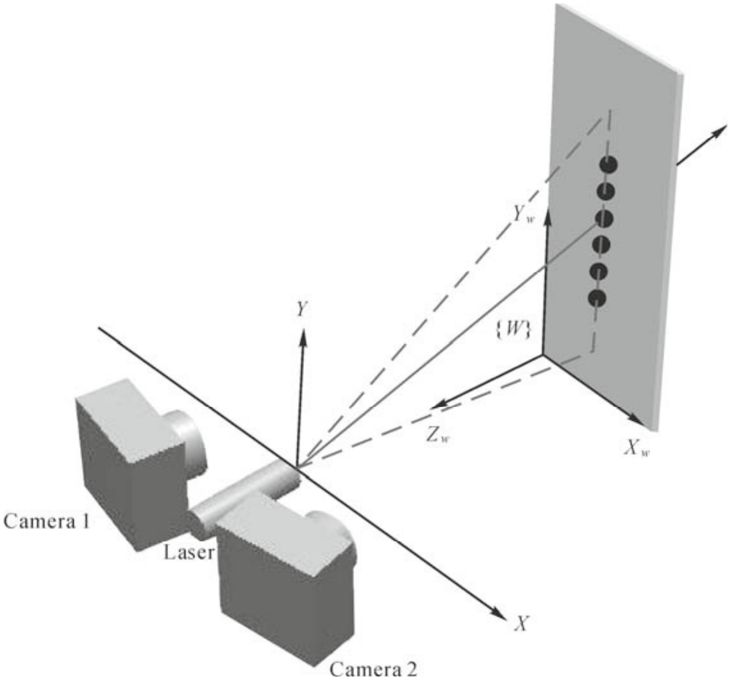


Fig. 3.16. Laser stripe calibration setup: laser stripe is positioned with the movement of the robot arm to align with the calibration dot

After step (3) is accomplished, in which the laser plane is aligned with the calibration plate, the laser plane equation is already determined by $X_w = 0$. The intrinsic parameters and extrinsic parameters of the camera can be calibrated with two sets of calibration data points by using Tsai's method (Tsai, 1987). The calibration control points in the world coordinate frame are defined by the positions of a group of dots. When the robot arm moves along the Z axis of the world coordinate frame, that is equivalent to moving the dots in the opposite direction, creating an array of dots on the YX plane of the coordinate frame.

3.4.7 Laser Scanner Calibration with Direct Coordinate Mapping

As we mentioned earlier, the second type of calibration method is to directly find the transform between the laser plane and the image plane based on two sets of correspondences between calibration points on the laser plane and their image points. Two types of mapping functions and corresponding calibration methods are discussed in the following sections.

3.4.7.1 Linear mapping

Mapping function $\boldsymbol{M}$ in Eq. (3.67) can be derived from the laser plane model and the camera perspective projection model. In fact, Eq. (3.49) describes the mapping between the image plane and a plane in the world coordinate system. If we assume this world plane is the laser plane, we already create the linear mapping relationship between the laser plane and the image plane. From Eq. (3.49) we have

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} \tag{3.86}$$

Therefore, the mapping function from the image plane to the laser plane is given by

$$\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{3.87}$$

That means the mapping from the image plane to the laser plane is linear, given that the lens distortion error is ignored. The mapping function $\boldsymbol{M}$ can be obtained by using the linear least squares method, as a derivation from the following.

For a set of corresponding points (X_{wi}, Y_{wi}) and (u_i, v_i), Eq. (3.87) can be rewritten as

$$\left.\begin{aligned} m_{11}u_i + m_{12}v_i + m_{13} - m_{31}X_{wi}u_i - m_{32}X_{wi}v_i &= m_{33}X_{wi} \\ m_{21}u_i + m_{22}v_i + m_{23} - m_{31}Y_{wi}u_i - m_{32}Y_{wi}v_i &= m_{33}Y_{wi} \end{aligned}\right\} \tag{3.88}$$

Without loss of generality, let $m_{33} = 1$ and $(m_{11}, m_{12}, \ldots, m_{13}, m_{32})$ can be obtained by solving the linear equations with the form $\boldsymbol{Ax} = \boldsymbol{b}$

$$\left.\begin{array}{l}
\boldsymbol{A}=\begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -X_{w1}u_1 & -X_{w1}v_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -Y_{w1}u_1 & -Y_{w1}v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_n & v_n & 1 & 0 & 0 & 0 & -X_{wn}u_n & -X_{wn}v_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -Y_{wn}u_n & -Y_{wn}v_n \end{bmatrix}_{2n\times 8} \\
\boldsymbol{x}=\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{21} & m_{21} & m_{21} & m_{31} & m_{32} \end{bmatrix}^{\mathrm{T}} \\
\boldsymbol{b}=[X_1 \quad Y_1 \quad \cdots \quad X_n \quad Y_n]^{\mathrm{T}}{}_{1\times 2n}
\end{array}\right\} \tag{3.89}$$

The least squares solution is therefore given by $\boldsymbol{x}=(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$.

3.4.7.2 Polynomial Mapping Function

The polynomial mapping function between the point (X_w, Y_w) on the laser plane and its image coordinate (u, v) is given by

$$\left.\begin{array}{l}
X_w(u,v)=\sum\limits_{j=0}^{n}\sum\limits_{i=0}^{n-j} c_{ij}u^i v^j \\
Y_w(u,v)=\sum\limits_{j=0}^{n}\sum\limits_{i=0}^{n-j} d_{ij}u^i v^j
\end{array}\right\} \tag{3.90}$$

where c_{ij}, d_{ij} are the coefficients of the polynomial mapping function. The related error functions E_x and E_y of $X_w(u, v)$ and $Y_w(u, v)$, respectively, are obtained by

$$\left.\begin{array}{l}
E_x=\sum\limits_{k=0}^{n}(X_{wk}-X)^2 \\
E_y=\sum\limits_{k=0}^{n}(Y_{wk}-Y)^2
\end{array}\right\} \tag{3.91}$$

The coefficients c_{ij}, d_{ij} can be determined from the minimum error

$$\left.\begin{array}{l}
\dfrac{\partial E_x}{\partial c_{ij}}=0 \\
\dfrac{\partial E_y}{\partial d_{ij}}=0
\end{array}\right\} \tag{3.92}$$

Once the coefficients c_{ij}, d_{ij} are obtained, the space coordinate can be calculated from its image coordinate, based on Eq. (3.90).

3.4.7.3 Measurement of the Calibration Points on the Laser Plane

Fig. 3.17(a) shows the experimental setup for the laser plane calibration. A laser

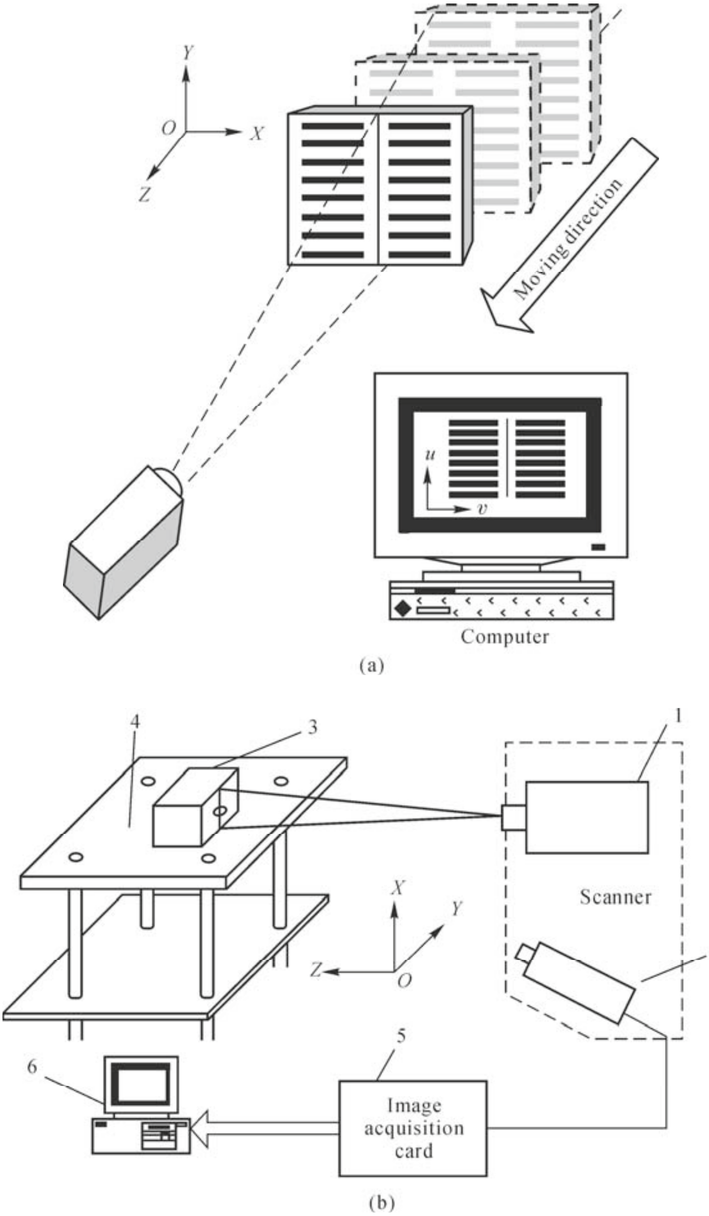


Fig. 3.17. Calibration setup to obtain and measure the calibration points on the laser plane by using (a) a template (Fan, 2001) with permission of Fan; (b) a photo-electrical aiming device (Zhang and Wei, 2002 permission granted)

stripe is projected onto the standard template and the CCD cameras detect the line image. The template is made using a laser writer with 1μm accuracy. Since the distance between each horizontal line on the standard template is known, the intersection point between each horizontal line and vertical laser line stripe indicates (X_w, Y_w) coordinates of the calibration points. The template is successively moved along the X axis direction to create non-collinear points. The CCD camera grabs the image of the calibration points (u, v).

Fig. 3.17(b) shows the experimental setup to measure the position of the laser plane. A photo electrical aiming device is used to sample the laser plane. When the detector intersects with the laser plane, the detected signal is maximized and the position of the detector is recorded. Move the detector at least three positions along the laser plane to determine the position of the laser plane.

3.4.8 Calibration of Laser Stripe Sensor with Scheimpflug Configuration

When 3D acquisition systems of small depth-of-view are applied to measure an object of large size and complicated profile, one has to edit multiple scanning paths and register the scanning data to obtain the entire 3D surface information. The multiple paths editing and scanning are time-consuming, and the data registering will reduce the measurement accuracy. Therefore, it is necessary to improve the depth-of-view to increase the working efficiency and the measurement accuracy. For a quality imaging lens, the resolution and the intensity are good at full aperture, but the depth-of-view is poor. Although the depth-of-view can be improved by decreasing the aperture, the intensity and resolution are becoming poor. This effect is serious for short-range measurement, because the defocusing problem becomes serious as the object to be measured is brought closer to the camera to obtain high resolution. Therefore, a decreasing aperture is not an effective way to obtain both large depth-of-view and high resolution. The Scheimpflug condition (Altenhofen, 1952) is well known and has been used in photography for a long time. This condition provides a considerable improvement in the depth-of-view without a loss of intensity (the lens aperture can be kept at maximum) (Bickel et al., 1985). In order to complete 3D reconstruction from the 2D image points, one has to build a mathematical model which describes the mapping between 3D coordinates and corresponding 2D image coordinates, and determine a set of unknown parameters of this model. This procedure is called camera calibration. All those camera calibration techniques described previously can only work for the camera model with the CCD plane being parallel to the lens plane, and will not work when the CCD plane is tilted. In this section, we extend the depth-of-view of a portable 3D laser scanner from less than 40 to 100 mm, according to the Scheimpflug condition. Based on the tilted camera model, the new object-image equations are derived theoretically. The traditional two-step camera calibration method is modified by introducing the angle factor for the determination of extrinsic and intrinsic camera parameters. Also, a novel segmental calibration approach, i.e., dividing the whole work range into two segments and calibrating, respectively, with corresponding system parameters, is presented to effectively improve the measurement accuracy of the large depth-of-view 3D laser scanner (Li et al., 2008; Li et al., 2006; Chen et al., 2006). In the process of 3D reconstruction, different calibration parameters are used to transform the 2D coordinates into 3D coordinates according to the different positions of the image in the CCD plane and a measurement accuracy of 60 μm is obtained experimentally. Finally, the experiment of scanning a blade by a portable 3D laser scanner in an industrial robot IRB 4400 is also employed to demonstrate the effectiveness and high measurement accuracy of our scanning system.

3.4.8.1 The Extension of Depth-of-View based on the Scheimpflug Condition

The Scheimpflug condition has to be enforced in the angular-displacement system, which requires that the image plane, the object plane and the lens plane intersect along a single line. A detailed derivation of the Scheimpflug condition is presented in (Prasad and Jensen, 1995). Fig. 3.18 shows how to construct Scheimpflug geometry in the laser scanning system. From Fig. 3.18 we have

$$\tan\theta = l'/d,\quad \tan\beta = d/l,\ l' = fl/(l-f) \tag{3.93}$$

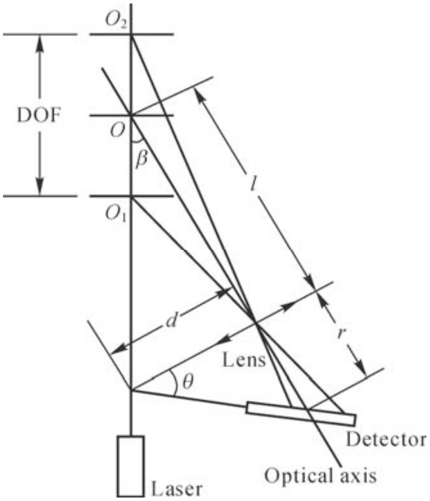


Fig. 3.18. Scheimpflug structure of a scanning system

where l and l' denote the object and image distance of the imaging system, respectively. f denotes the focal distance. d denotes the distance between the projection and detection axis at the level of the lens. θ denotes the angle between the projection axis, and the optical axis and β denotes the angle between the lens plane and the image plane. From Eq. (3.93), the following relationship can be obtained

$$\theta = \tan^{-1}\left(\frac{l'}{a}\right) = 90 - \tan^{-1}\left(\frac{\beta}{\kappa}\right) \tag{3.94}$$

where k is the one-axis magnification of an imaging system whose value is l'/l. The idea of this geometry is that any point along the projection axis is in focus on the image plane. This property can provide a considerable improvement in the depth-of-view without compromising the intensity of light. Note that the depth-of-view which can be improved by the Scheimpflug condition here is not the depth-of-view of the imaging system of the whole field-of-view but only the depth-of-view of the laser scanner of the scanning field-of-view. The longitudinal magnification can be defined as the ratio of the displacement of the object image on the position detector to the change in depth along the projection axis. Usually, the longitudinal magnification is considered as the resolution of the scanning system, which can be written as

$$M_r \approx \frac{l\,d}{l^2 \cos\theta} \tag{3.95}$$

It can be seen from Eq. (3.95) that the resolution of the scanning system is proportional to d. However, the field-of-view of the scanning system will decrease with the increase in d due to the shadow effects. On the other hand, the resolution of the scanning system on the detection axis will decrease with the increasing square of l. Therefore, the Scheimpflug condition is a necessary but not sufficient condition. A compromise has to be made between l and d to obtain both large depth-of-view and high resolution.

3.4.8.2 Camera and Scanner Calibration

Referring to the ideal pin-hole model, the tilted camera and projection model is presented in Fig. 3.19.

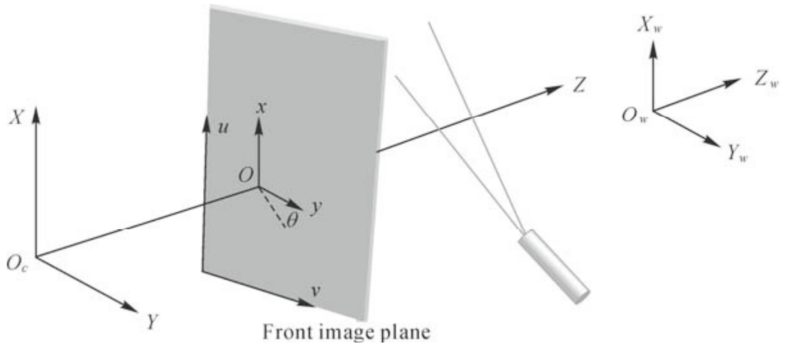


Fig. 3.19. Perspective tilted camera and projector model

In Fig. 3.19, (X_w, Y_w, Z_w) denotes the world coordinate system. (X, Y, Z) denotes the camera coordinate system, whose origin is located at the optical center point O_c and whose Z axis coincides with the optical axis. (x, y) is the analog coordinate centered at O (intersection of the optical axis and the front image plane) and is measured in millimeter. (u, v) denotes the computed image coordinates system centered at O_l and is measured in pixels. (u_0, v_0) is the computed image coordinates of the center O in the image plane. θ denotes the angle between the image plane and the lens plane, which can be calculated by Eq. (3.93). f denotes the effective focal length of the camera. Because the factor θ is introduced into the camera model, the new object-image equations and the mathematical model mapping the 2D and 3D points have to be derived.

3.4.8.3 Calibration of Camera

The parameters of the camera model are derived in two stages. In the first stage, a distortion-free model is used, the DLT procedure is devised to find the nominal values of the unknown parameters for the second stage. In the second stage, a nonlinear model is used, these nominal parameters are then used as initial conditions for the nonlinear least squares algorithm to find the globally optimal set of parameters of the camera model.

Remember that the transform equation from the camera coordinates (X, Y, Z) to the world coordinates (X_w, Y_w, Z_w) is expressed by Eq. (3.9). In the tilted "pin-hole model", the relationship between the image height and the object height can be rewritten as

$$\frac{x}{X}=\frac{f-y\sin\theta}{Z},\quad \frac{y\cos\theta}{Y}=\frac{f-y\sin\theta}{Z} \tag{3.96}$$

then

$$x=\frac{f\cdot X\cos\theta}{Z\cos\theta+Y\sin\theta},\ y=\frac{f\cdot Y}{Z\cos\theta+Y\sin\theta} \tag{3.97}$$

where θ denotes the angle between the image plane and the optical axis of the camera. The physical size of a pixel is dx and dy respectively, then the transform equation from the image coordinates to the computed image coordinates is

$$u=x/\alpha x+u_0,\ v=y/\alpha y+v_0 \tag{3.98}$$

Substituting Eq. (3.97) into Eq. (3.98), the computed image coordinates are given by

$$\begin{aligned} u&=\frac{f_x\cdot X\cos\theta}{Z\cos\theta+Y\sin\theta}+u_0\\ v&=\frac{f_y\cdot Y}{Z\cos\theta+Y\sin\theta}+v_0 \end{aligned} \tag{3.99}$$

where f_x and f_y are defined as follows

$$f_x = f / \mathrm{d}x,\ f_y = f / \mathrm{d}y \tag{3.100}$$

By substituting Eq. (3.9) into Eq. (3.99), the relationship between the computed image coordinates (u, v) and the world coordinates (X_w, Y_w, Z_w) can be derived as follows

$$\begin{aligned} u &= x_0 + f_x \frac{[m_{11}(X_w - t_x) + m_{21}(Y_w - t_y) + m_{31}(Z_w - t_z)]\cos\theta}{[m_{13}(X_w - t_x) + m_{23}(Y_w - t_y) + m_{33}(Z_w - t_z)]\cos\theta + [m_{12}(X_w - t_x) + m_{22}(Y_w - t_y) + m_{32}(Z_w - t_z)]\sin\theta} \\ v &= y_0 + f_y \frac{m_{12}(X_w - t_x) + m_{22}(Y_w - t_y) + m_{32}(Z_w - t_z)}{[m_{13}(X_w - t_x) + m_{23}(Y_w - t_y) + m_{33}(Z_w - t_z)]\cos\theta + [m_{12}(X_w - t_x) + m_{22}(Y_w - t_y) + m_{32}(Z_w - t_z)]\sin\theta} \end{aligned} \tag{3.101}$$

The above equations can be described as

$$\begin{aligned} u &= \frac{L_1 X_w + L_2 Y_w + L_3 Z_w + L_4}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + L_{12}} \\ v &= \frac{L_5 X_w + L_6 Y_w + L_7 Z_w + L_8}{L_9 X_w + L_{10} Y_w + L_{11} Z_w + L_{12}} \end{aligned} \tag{3.102}$$

where

$$\begin{cases} L_1 = [u_0(m_{13}\sin\theta + m_{12}\cos\theta) + k_u m_{11}\sin\theta] / L \\ L_2 = [u_0(m_{23}\sin\theta + m_{22}\cos\theta) + k_u m_{21}\sin\theta] / L \\ L_3 = [u_0(m_{33}\sin\theta + m_{32}\cos\theta) + k_u m_{31}\sin\theta] / L \\ L_4 = -L_1 t_x - L_2 t_y - L_3 t_z \\ L_5 = [v_0(m_{13}\sin\theta + m_{12}\cos\theta) + k_v m_{12}] / L \\ L_6 = [v_0(m_{23}\sin\theta + m_{22}\cos\theta) + k_v m_{22}] / L \\ L_7 = [v_0(m_{33}\sin\theta + m_{32}\cos\theta) + k_v m_{32}] / L \\ L_8 = -L_5 t_x - L_6 t_y - L_7 t_z \\ L_9 = (m_{13}\sin\theta + m_{12}\cos\theta) / L \\ L_{10} = (m_{23}\sin\theta + m_{22}\cos\theta) / L \\ L_{11} = (m_{33}\sin\theta + m_{32}\cos\theta) / L \end{cases}$$

$$\begin{aligned} L = -[&(m_{13}\sin\theta + m_{12}\cos\theta)t_x + (m_{23}\sin\theta + m_{22}\cos\theta)t_y \\ &+ (m_{33}\sin\theta + m_{32}\cos\theta)t_z] \end{aligned} \tag{3.103}$$

For the convenience of computation, L_{12} is usually chosen as 1. If the noncoplanar world coordinates (X_w, Y_w, Z_w) and their corresponding image coordinates (u, v) of n ($n \geq 6$) calibration points are known, an over-determined system of linear equations can be established and 11 unknown components L_i can be easily determined by using a linear least squares technique. The extrinsic parameters and the intrinsic parameters can be derived from L_i and they have the same format as the ones described for the standard DLT model, as indicated in Eqs. (3.18) to (3.22).

When the lens distortion is considered, a nonlinear optimization algorithm will be used. The procedure and equations are the same as for the standard camera configuration in which the image plane and the lens plane are in parallel. Therefore Eq. (3.28) can be directly applied here.

3.4.8.4 Experimental Results and Discussion

The large depth-of-view portable 3D scanner consists of a 1280 × 1024-pixel CCD camera, which is used to capture images via a 16 mm lens with a band-pass spectral filter and a 5 m laser diode of wavelength 650 nm that is used to generate the laser beam. The angle between the projection axis and the optical axis is 25°, and the angle between the CCD plane and the lens plane is 10°, which is calculated by Eq. (3.95). The choice of these parameters is a compromise: the longer the focal distance, the longer the working distance needed to obtain the proper image size, and thereby the larger the extended range of depth-of-view that can be obtained. However, the resolution of the scanning system decreases with an increase in l. Thus the resolution will be sacrificed though the extended range of depth-of-view is increased for a lens with long focal distance. On the other hand, the larger β, the larger d, and thereby the higher resolution that can be obtained. However, the field-of-view will decrease with the increase in d. We always minimize β to obtain the maximum field-of-view provided the resolution is satisfied. Therefore, in the experiment, the focal distance is chosen to be 16 mm and the angle between the projection axis and the optical axis is chosen to be 25° to ensure that the image size and the working distance are adaptable, so as to obtain a good compromise between resolution and depth-of-view. The schematic structure of an experimental calibration system is shown in Fig. 3.20. It can be seen that a calibration object with 14 calibration holes is fixed on an encoded motor-driven positioning platform with a positioning accuracy of 1 μm. The distance between the two nearest calibration holes is 4 mm, and the diameter of each hole is 2 mm. The portable 3D scanner is positioned in the front of the calibration object and the distance between them is approximately 100 mm. The laser beam projected by the laser diode is vertical to the calibration plane so that the moving direction of the platform is parallel to the laser plane. The CCD camera and the motor-driven positioning system are controlled by a computer. The world coordinate system is centered at the first hole of the calibration object which is positioned at the zero scale of the platform, and its Z axis is opposite to the moving direction of the platform. Therefore, the world coordinates of the center of all holes are known.

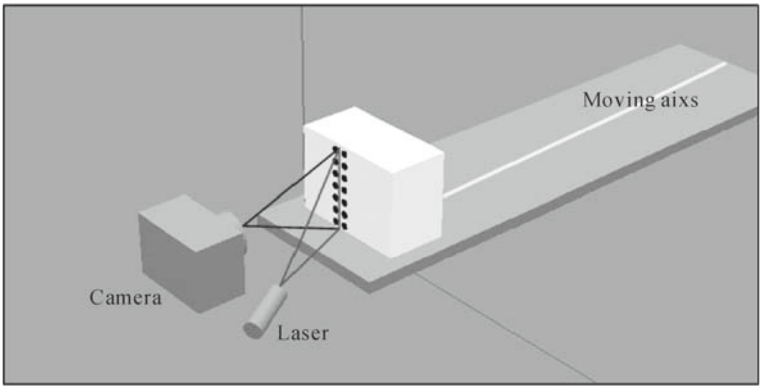


Fig. 3.20. Schematic structure of the experimental system for laser scanner calibration. The laser scanner consists of a CCD camera and a laser. The laser scanner calibration is to determine the position of the camera as well as the laser plane, relative to a reference coordinate frame that is defined on the calibration target

Comparison between Depth-of-View of Scanner Systems With and Without CCD Being Tilted

Figs. 3.21 (a) – 3.21(f) show the images of a calibration object at different positions along the Z axis of the world coordinate system for $\theta = 0°$ and $\theta = 9°$, respectively. Note that $\theta = 9°$ is computed by the Scheimpflug condition. As can be seen from Figs. 3.21(a) and 3.21(c), the images of the calibration holes are out of focus in $Z_w = -35$ mm and $Z_w = -75$ mm, which indicates that the depth-of-view of the camera is less than 40 mm. From Fig. 3.21 (d) – 3.21(f), it can be seen that the images of calibration holes in $Z_w = -3$ mm and $Z_w = -103$ mm are still in focus, which indicates that the depth-of-view of the camera is extended to 100 mm successfully.

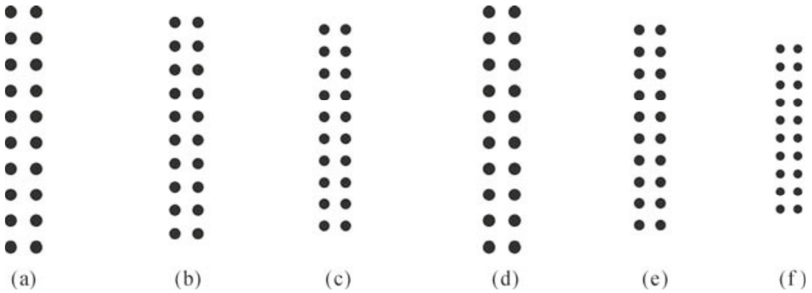


Fig. 3.21. Images of the calibration object in (a) $Z_w = -35$ mm; (b) $Z_w = -53$ mm; and (c) $Z_w = -75$ mm when the CCD plane is parallel to the lens plane. Images of the calibration objects in (d) $Z_w = -3$ mm; (e) $Z_w = -53$ mm; and (f) $Z_w = -103$ mm when the angle between the images and lens plane is 9°

The Calibration Results

In total, five images of a calibration object in different positions are captured, and a threshold and center-of-gravity algorithm is employed to process each image and determine the gravity of each hole in the image. In total, 70 calibration points, their world coordinates (X_{wn}, Y_{wn}, Z_{wn}) and corresponding image coordinates (u_n, v_n) are obtained in the experiments for the following data processing. Using Eq. (3.102), we can get the 11 nominal coefficients (L_1, L_2, …, L_{11}), which are used as the initial conditions for the nonlinear least-squares algorithm to obtain the global optimal parameters of the camera. The calibration results are shown in Table 3.2. Using the calibrated parameters, 70 test points with known 3D world coordinates are transformed into the 2D computed image coordinates. The 2D calibration accuracy is determined by measuring the discrepancy between the real 2D points and the computed ones. Meanwhile, for each image point, the calibrated parameters are used to compute their corresponding 3D coordinates. The 3D calibration accuracy is determined by measuring the discrepancy between the real and estimated 3D positions. Because the calibration accuracy on the Z axis is dominant, the estimated 3D coordinates X and Y are assumed to be the same as the real ones. The discrepancy between the real and estimated coordinates on the Z axis is considered as the 3D calibration accuracy. The 2D and 3D calibration accuracy calculated by using the above criteria is listed in Table 3.3. Meanwhile, the image points in the laser line can be estimated by using a sub-pixel operator such as a directional gravity operator, as described in Chapter 5, and 640 points per image are obtained with high accuracy. Substituting the obtained image coordinates (u_n, v_n) into Eq. (3.12), the corresponding distortion-corrected coordinates $(\tilde{u}_n, \tilde{v}_n)$ can be calculated. By using the real coordinate Z_{wn} and Eq. (3.28) we can get all the corresponding world coordinates (X_{wn}, Y_{wn}, Z_{wn}) of points on the laser plane. The parameters of the laser stripe plane [a, b, c, d] can be calculated by using Eq. (3.79), the values of which are [a, b, c, d] = [0.9999, 0.0015, 0.0052, 1.9573]. The distance from the points to the laser stripe plane dn can be considered as the measurement accuracy of X_w and Y_w because of the use of real coordinates Z_{wn}. By using the obtained camera and projector parameters, the image coordinates (u_n, v_n) in the laser line can be transformed into their corresponding world coordinates (X_{wn}, Y_{wn}, Z_{wn}). The 3D measurement accuracy of Z_w is determined by measuring the discrepancy between the real and estimated coordinates on the Z axis. The standard deviation of X_w and Y_w (σ_{xy}) and the standard deviation of $Z_w(\sigma_z)$ in different computed image coordinates are shown in Fig. 3.11(a). It can be seen from Fig. 3.22(a) that the 3D measurement accuracy on the Z-axis is 0.18 mm, which can be improved by a new calibration method.

Table 3.2 Result of camera calibration

Parameters	Results	Parameters	Results
m_{11}	–0.9620	t_z	118.4971
m_{12}	–0.0007	f_x	3301.0
m_{13}	–0.2730	f_y	3316.9
m_{21}	0.00008	x_0 (pixel)	622.9
m_{22}	–0.9996	y_0 (pixel)	398.7
m_{23}	–0.0269	k_1	–7.0012 e^{-8}
m_{31}	0.2729	k_2	0
m_{32}	0.0260	k_3	0
m_{33}	–0.9617	p_1	–4.6917 e^{-6}
t_x	–59.3403	P_2	4.4056 e^{-6}
t_y	7.3496		

Table 3.3 2D and 3D calibration accuracy

	Mean	Std Dev	Max
2D calibration accuracy (pixel)	0.223	0.368	0.869
3D calibration accuracy (mm)	0.013	0.024	0.051

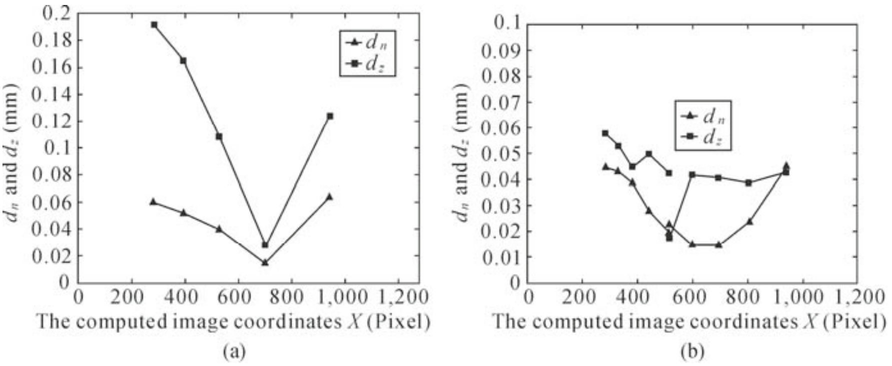


Fig. 3.22. Estimation of measurement error for the laser scanner in X and Y direction (d_n) and in Z direction (d_z) versus computed coordinates X by using (a) standard calibration method; and (b) segmental calibration method. Results for the first and second segments are shown in (b)

Segmental Calibration for Improving the Measurement Accuracy

With the significant extension of depth-of-view in the system, the measurement accuracy decreases due to some nonlinear effects. This problem can be overcome by a segmental calibration method to divide the whole depth-of-view into n segments and then to calibrate. Thereby the camera calibration accuracy and the fitting accuracy of the laser plane are increased due to the shortening calibration range. However, it does not mean that the more the divided segments, the higher the measurement accuracy, because the measurement accuracy is still limited by many factors, such as the resolution of CCD, the quality of lens and the laser beam projected by the projector. In the experiment, two segments are used to obtain a good compromise between accuracy and simplicity. The first range is from $Z_w = -3$ mm to $Z_w = -53$ mm, and the second range is from $Z_w = -53$ mm to $Z_w = -103$ mm. The calibration results and the camera calibration accuracy are listed in Tables 3.4 and 3.5, respectively. Comparing 3.5 with Table 3.3 , it can be observed that the segmental calibration method provides higher calibration accuracy. Using the segmental calibration parameters, the standard deviation of X_w and Y_w ($\sigma_{x,y}$) and the standard deviation of Z_w (σ_z) in different computed image coordinates are shown in Fig. 3.22(b). It can also be seen that, after segmental calibration, the measurement accuracy is improved significantly, which can attain 60 μm.

Table 3.4 Results of segmental calibration of camera and projector

Parameters	First range	Second range
m_{11}	–0.9460	–0.9293
m_{12}	–0.0004	–0.0012
m_{13}	–0.3240	–0.3692
m_{21}	–0.0045	–0.0132
m_{22}	–0.9999	–0.9999
m_{23}	–0.0150	0.0085
m_{31}	0.3240	0.3692
m_{32}	0.0129	–0.0127
m_{33}	–0.9459	–0.9293
t_x	–58.5841	–57.9167
t_y	7.1392	7.3896
t_z	117.0172	113.8807
f_x	3307.0	3301.0
f_y	3300.8	3316.9
x_0 (pixel)	633.6	622.9
y_0 (pixel)	400.8	398.7
k_1	–1.00 e^{-8}	–1.00 e^{-7}
k_2	0	0
k_3	0	0
p_1	–1.34 e^{-5}	–2.05 e^{-5}
p_2	3.08 e^{-6}	4.59 e^{-6}
a	–0.9999	–0.9999
b	–0.0015	–0.0024
c	–0.0067	–0.0003
d	–2.1281	–2.1140

Table 3.5 2D and 3D calibration accuracy for segmental calibration

		Mean	Std Dev.	Max
First range	2D calibration accuracy (pixel)	0.124	0.268	0.571
	3D calibration accuracy (mm)	0.009	0.015	0.029
Second range	2D calibration accuracy (pixel)	0.176	0.284	0.657
	3D calibration accuracy (mm)	0.011	0.019	0.035

3D Reconstruction after Segmental Calibration

After completing the above calibration and the robot hand-eye calibration, the large depth-of-view portable 3D laser scanner can be used by an industrial robot as a measurement tool to acquire the 3D surface information of the target from multiple angles and directions, the photograph of which is shown in Fig. 3.23.

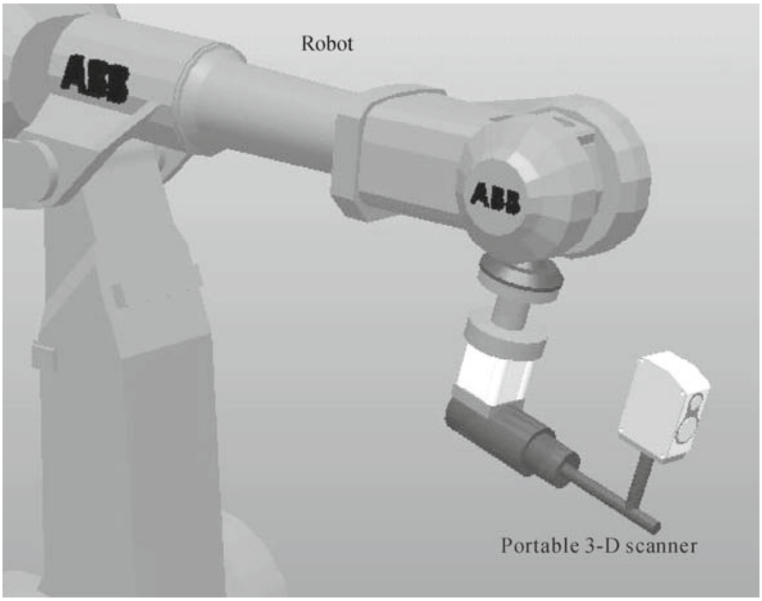


Fig. 3.23. A large-depth-of view portable laser scanner is used by an industrial robot as a measurement tool

For demonstrating the effectiveness and high measurement accuracy of the scanning system, the large depth-of-view laser scanner is held by an industrial robot ABB-4400 to scan a blade. In the process of 3D reconstruction, the 2D image coordinates can be transformed into the 3D world coordinates by using different calibration parameters decided by the image position of the object in the CCD plane. The preliminary scanning result is presented in Fig. 3.24(b), accompanied by a photograph of the actual blade, as shown in Fig. 3.24(a). Note that the measurement accuracy includes the calibration accuracy of scanner calibration, the calibration accuracy of robot hand-eye calibration and the robot positioning accuracy. Therefore, in order to further improve the measurement accuracy, a more accurate robot hand-eye calibration method and ways to decrease the effect of robot positioning error have to be studied. It is also worth mentioning that if the conventional laser scanner with a 30 mm of depth-of-view is used, one has to edit 12 scanning paths to scan the whole blade and spend 30 min to complete the whole process. But if we use the new laser scanner with 100 mm of depth-of-view, only three scanning paths and 10 min are needed to finish the scanning process. More over, the measurement accuracy of the latter system is higher than that of the former.

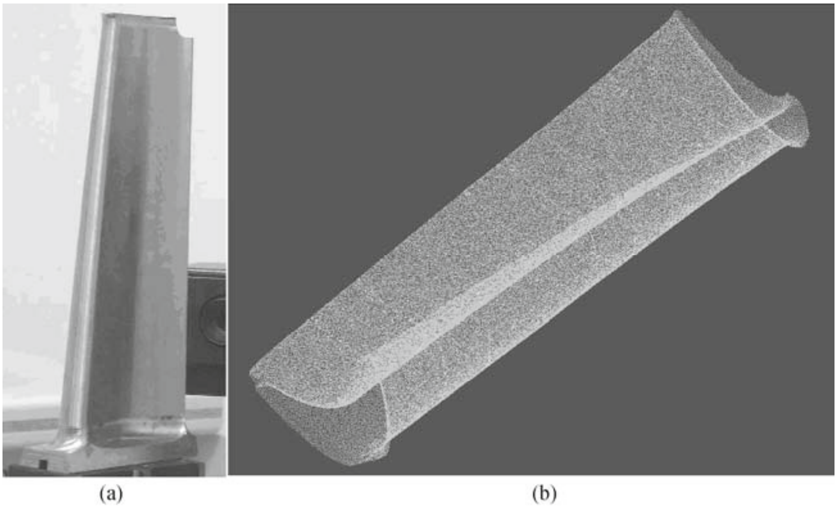


Fig. 3.24. 3D measurement result of a blade. (a) A photograph of a blade; (b) 3D points of cloud

Conclusion

A portable and compact 3D laser scanner with large depth-of-view and high measurement accuracy is presented. The depth-of-view of the laser scanner is extended from less than 40 to 100 mm according to the Scheimpflug condition. Based on the tilted camera model, the traditional two-step camera calibration technique is modified. Meanwhile, a method of segmental calibration that divides the whole depth-of-view into two segments and then calibration is used to obviously improve the measurement accuracy of this laser scanner from 180 to 60 μm. The preliminary experimental results show that the proposed calibration approach for the tilted camera system is stable, accurate and can be expected to have some practical applications in robot vision and 3D acquisition, etc.

3.5 Conclusion and Remarks

In this chapter we presented and discussed various calibration algorithms for thelaser stripe sensor. They are grouped into two categories: 1) calibration of camera and laser plane, respectively; 2) calibration of the mapping function between image plane and laser plane.

In terms of calibration algorithms for the camera, a linear model is used first to get an approximation solution or partial solution of the parameters and then a nonlinear model is used to obtain all the parameters accurately. The linear solution is used as the initial value of the nonlinear solution. For the linear model a linear least squares algorithm is used. And for the nonlinear model a standard nonlinear optimization method, like the Levenberg-Marquardt algorithm, is applied. If the lens distortion of the camera is ignored, the linear solution will be accurate enough for robotic applications.

In terms of implementation of calibration, the main task is to find correspondence between the world coordinates and image coordinates. Image coordinates are easy to find and world coordinates or control points need to identified through various calibration targets. Control points can be implemented with stationary or moving calibration targets like dots, lines, planar patterns and 3D patterns. There is always a trade-off between the calibration accuracy and the expense of the calibration setup. Which calibration method is selected really depends on the accuracy requirements and factory floor working conditions. For fast calibration in the field a flexible planar pattern can be used. For high accuracy requirements a 3D calibration pattern or a planar pattern with accuracy controlled movement is preferred.

Since the laser strip sensor is intended to be used in conjunction with an industrial robot, in this book selection of the calibration method needs to match the accuracy of the robot. When a laser strip sensor is attached to the robot arm, calibration of the sensor parameters can be done simultaneously with the TCP calibration.

References

Abdel-Aziz YI, Karara HM (1971) Direct linear transform from comparator coordinates into object-space coordinates. In: Proc. ASP/UI Symp. on Close-Range Photogrammetry, American Society of Photogrammetry, Falls Church, pp. 1-18.

Altenhofen RE (1952) Manual of photogrammetry. American Society of Photogrammetry, Washington DC.

Anchini R, Beraldin JA, Liguori C (2007). Sub-Pixel location of discrete target images in close range camera calibration: a novel approach. In: Proc. SPIE, 6491: 10.

Bickel G, Hausel G, Maul M (1985) Triangulation with expended range of depth. Optical Engineering, 24:955-957.

Blais F, Rioux M, Beraldin JA (1988). Practical considerations for a design of a high precision 3-D laser scanner system. In: Proc. SPIE, 959: 225-246.

Chang M, Tai WC (1995) 360-deg profile non-contact measurement using a neural network. Optical Engineering, 34(12):3572-3576.

Chen CH, Kak AC (1987) Modeling and Calibration of a structured light scanner for 3D robot vision: In: Proc. IEEE Conf. Robotics Automation, 2:807-815.

Chen L, Tang Q, Li JF, Zhu JH (2006) Large depth-of-view portable 3D scanner for robot vision. Opto-Electronics Engineering, 33(10) :72-78 (In Chinese)

DePiero FW, Trivedi MM (1995) 3-D computer vision using structured light: Design, calibration and implementation issues. Advanced Computers, 43: 243-278.

Dewar R (1988) Self-Generated targets for spatial calibration of structured light optical sectioning sensors with respect to an external coordinate system. In: Proc. Robots and Vision'88 Conf, 5:13-21.

Duan F, Liu F, Ye S (2000) A new accurate method for the calibration of line structured light sensor, Chinese Journal of Scientific Instruments, 21(1): 108-110 (In Chinese).

Fan KC, Tsai TH (2001) Optimal shape error analysis of the matching image for a free-form surface. Robotics and Computer Integrated Manufacturing, 17:215-222.

Faugeras O (1993) Three-Dimensional Computer Vision. MIT Press, London.

Golub G, Loan C (1996) Matrix Computations. The John Hopkins University Press, Baltimore, USA.

Han JD, Lu NG, Dong ML, Lou XP (2009) Fast method to calibrate structure parameters of line structured light vision sensor. Optics and Precision Engineering, 17(5):954-963 (In Chinese).

Huynh DQ, Owens RA, Hartmann PE (1999) Calibration of a structured light stripe system: a novel approach. International Journal of Computer Vision, 33(1):73-86.

James KW (1988) Non-Contact machine vision metrology within a CAD coordinate system. In: Proc. Autofact'88 Conf., 12:9-17.

Li JF, Guo YK, Zhu JH, Lin XD, Duan KL, Tang Q (2007) Large depth-of-view portable three-dimensional laser scanner and its segmental calibration for robot vision. Optics and Lasers Engineering, 45:1077-1087.

Li JF, Tang Q, Zhu JH, Xu M, Chen L (2006) Calibration method for the eye-in-hand relationship of the robot vision based on large depth-of-field portable 3-D scanner. Journal of Optoelectronics Laser, 17 (10):1238-1244 (In Chinese)

Li JF, Zhu JH, Guo YK, Lin XD, Duan KL, Wang Y, Tang Q(2008) Calibration of a portable laser 3D scanner used by a robot and its use in measurement. Optical Engineering, 47 (1):1-8.

Lin N, Ma Z, Hu Y, Zhao J (2007) New method for calibration of line structure-light vision sensor based on robot system. Transducer and Microsystem Technologies 26(9):101-103 (In Chinese).

Marzan GT, Karara HM (1975) A computer program for direct linear transformations solution of collinearity condition, and some applications of it. In: Proceedings of the Symposium on Close-Range Photogrammetric Systems. American Society of Photogrammetry, Falls Church.: 420-426.

McIvor AM (2002) Nonlinear calibration of a laser stripe profiler. Optical Engineering, 41(1):205-212.

Prasad AK, Jensen K (1995) Scheimpflug stereo camera for particle image velocimetry in liquid flows. Applied Optics, 34:7092-7099

Press WH, Teukolsk SA, Vetterling WT (1994) Numerical Recipes in C. Cambridge University Press, New York.

Reid ID (1996) Projective calibration of a laser-stripe range finder. Image Vision Computing, 14(9):659-666.

Sabel JC (1999) calibration and 3d reconstruction for multicamera marker based motion measurement. Chapter 3 theory , 26-60, Delft University Press, The Netherlands.

Semple JG, Kneebone GT (1952) Algebraic Projective Geometry. Oxford University Press, London.

Tiddeman B, Duffy N, Rabey G, Lokier J (1998) Laser-Video scanner calibration without the use of a frame store. IEE Proc. Vision Image Signal Process, 145(4):244-248.

Trucco E, Fisher RB, Fitzgibbon AW (1994) Direct calibration and data consistency in 3-D laser scanning. In: BMVC94: Proc. 5th British Mach. Vis. Conf.: 489-498.

Tsai RY (1987) A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-shelf TV camera and lenses. IEEE Journal of Automation, 394:323-334.

Wei Z, Zhang G, Xu Y (2003) Calibration approach for structured-light-stripe vision sensor based on the invariance of double cross ratio. Optical Engineering, 42(10):2956-2966.

Wu QY, Li JZ, Su XY, Hui B (2008) An approach for calibrating rotor position of three dimensional measurement system for line structured light. Chinese Journal of Lasers, 35(8):1224-1227 (In Chinese).

Xu G, Liu L, Zeng J, Shi D (1995) A new method of calibration in 3D vision system based on structure-light. Chinese Journal of Computer, 18(6):450 (In Chinese).

Zhang G, Wei Z (2002) A novel calibration approach to structured light 3D vision inspection. Optics & Laser Technology, 34:373-380.

Zhang ZY (2000) A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11):1330-1334.

Zhou FQ, Zhang G (2005) Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations. Image and Vision Computing, 23:59-67.

Zhou FQ, Zhang G, Jiang J (2004) Field calibration method for line structured light vision sensor. Chinese Journal of Mechanical Engineering, 40(6):169-173 (In Chinese).

4

Calibration of a Robot Visual System

Industry is now seeing a dramatic increase in robot simulation and off-line programming (Motta, 2004; Mitsi, 2005; Swider, 2007). In order to use off-line programming effectively, the simulated workcell has to be identical to the real workcell. This requires a more efficient and accurate robot workcell calibration. By making use of calibration, the simulated robot workcell will clone the real workcell in a simulation model, so that the off-line generated robot program from a simulated workcell will be accurate enough and can be directly downloaded to a real robot controller to drive the real robot with maximum accuracy and without further modification. In general, workcell calibration includes calibrating all the components inside the workcell, including robot calibration, robot tool calibration and work object calibration. Robot calibration is used to enhance robot positioning accuracy through software rather than by changing the mechanical structure, to compensate for mechanical variations and inconsistency of the robot in the manufacturing process. Robot tool calibration, referred to as robot TCP calibration, is used to determine the position and orientation (i.e. pose) of the tool with respective to the robot end point (that is called the robot mounting flange) or the robot base. This is the main topic of this chapter. Work object calibration is used to identify the position and orientation of the workpiece inside a robot workcell. This is normally measured by the robot with a calibrated measurement tool.

In this chapter we first address a general model of the TCP calibration problem and its solution in Section 4.1. It shows that the general mathematical model of TCP calibration is a robot kinematics in addition to a geometry constraint. The solution to the model is in general a nonlinear optimization process that minimizes the cost function defined in the model. The algorithm can be simplified into linear equations by selecting a specific calibration target and/or by having a constrained robot movement to decouple the components of unknown variables. Therefore, TCP can be solved by using a linear least squares algorithm. The general model can be categorized into two types of TCP calibration approach. One is for a point sensor-typed tool like a probe pin, a touch trigger probe or a point laser, where only a single point is measured each time. In this case, the measurement points on

Z. Gan et al., *Visual Sensing and its Applications*

the calibration target are constrained by the geometrical shape of the calibration target and the TCP calibration algorithm is formulated as in Eq. (4.10). The other approach is for an array-type measurement tool like a camera or a laser scanner where multiple points can be measured each time. In this case, the calibration target is treated as the reference coordinate frame and the TCP calibration algorithm is formulated as in Eq. (4.11). Specific implementations of calibration algorithms derived from the general model with various geometrical shapes like point, line, sphere, plane and structured pattern as the calibration target are discussed in detail. After the discussion of the general TCP calibration algorithm and methodology we will focus on calibration of laser sensor based tools, which we will be using frequently throughout the book, especially in Chapter 7. In Section 4.2, TCP calibration of a point laser is discussed. The orientation and position of the sensor are calibrated separately. Since a laser scanner consists of a camera and a laser plane, calibration of the camera TCP can be considered as part of a laser scanner TCP calibration process. In Section 4.3, TCP calibration of a camera is discussed with linear and nonlinear algorithms. In Section 4.4, TCP calibration for a laser scanner with a calibration target of a sphere, a plane and a structured pattern is presented. In Section 4.5, TCP calibration of a mechanical tool like a spindle is presented by using direct measurement with the measurement tool that has been calibrated previously. In Section 4.6, process relative workpiece calibration is presented. In Tables 4.1 and 4.2, we summarize various TCP calibration methods for different types of tools by using various calibration targets as an implementation of the general calibration model.

4.1 General Solution of Robot Tool Calibration

A general robot tool calibration setup with a calibration target is shown in Fig. 4.1. Let $\{B\}$ be the robot base coordinate frame; $\{M\}$ is the robot mounting flange coordinate frame called $Tool_0$ frame, and its pose relative to the robot base frame is denoted by $\boldsymbol{T}_0$, that is given by the robot controller; $\{T\}$ is the tool coordinate frame. There are two scenarios. One is when the tool is held by the robot arm as indicated in Fig. 4.1(a) in which $\boldsymbol{T}_s$ is the tool pose relative to the $Tool_0$ frame and is called the moving tool center point or moving TCP. The other scenario is when the tool is fixed in the robot workcell as indicated in Fig. 4.1(b), in which $\boldsymbol{T}_s$ is the tool pose relative to the based frame and is called the fixed TCP. This is what we need to calibrate. $\{G\}$ is the calibration target coordinate frame, which is often defined as the world coordinate frame in the literature. It is made of a geometrical shape or structured pattern that is used as the geometry constraint to enclose the robot kinematics chain. $\boldsymbol{T}_g$ is its pose relative to the robot base frame for the moving TCP case or to the $Tool_0$ frame for the fixed TCP case; transformation from $\{G\}$ frame to $\{T\}$ frame is $\boldsymbol{T}_v$. For simplicity we only discuss the scenario of the moving TCP and derive the general calibration model. For the fixed TCP scenario relevant equations follow without derivation.

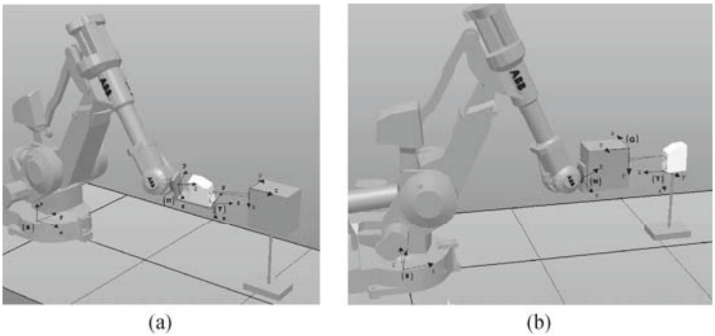


Fig. 4.1. A robot tool calibration setup with a calibration target for (a) a moving TCP case and (b) a fixed TCP case

If $\boldsymbol{T}_g$ and $\boldsymbol{T}_v$ are known, the calibration problem becomes trivial since $\boldsymbol{T}_s$ can be obtained directly with the robotic kinematics relationship as follows

$$\boldsymbol{T}_g = \boldsymbol{T}_0\boldsymbol{T}_s\boldsymbol{T}_v \Rightarrow \boldsymbol{T}_s = \boldsymbol{T}_0^{-1}\boldsymbol{T}_g\boldsymbol{T}_v^{-1} \tag{4.1}$$

But in an actual system, $\boldsymbol{T}_g$ is normally unknown. To obtain $\boldsymbol{T}_g$, extra constraints are required. Therefore, a general model of the calibration problem can be formulated as follows, that consists of a kinematic relationship and other extra constraints (usually geometric constraints introduced by the calibration target)

$$\left.\begin{aligned} \boldsymbol{T}_{gi} &= \boldsymbol{T}_{0i}\boldsymbol{T}_s\boldsymbol{T}_{vi} \\ f(\boldsymbol{T}_{gi}) &= 0 \end{aligned}\right\} \tag{4.2}$$

where subscript i indicates multiple measurements with different robotic poses, function f represents the extra constraints.

Although $\boldsymbol{T}_{gi}$ is normally an unknown matrix, the relationship between multiple measurements is given through the geometrical constraint or geometrical pattern of a calibration target, as indicated by the general form $f(\boldsymbol{T}_{gi}) = 0$. For example, if the calibration target is a sphere, the surface points on the target satisfy the sphere equation. Eq. (4.2) is the criterion function to calibrate the pose of TCP in which the tool is held by the robot arm. If the tool is stationary and fixed at the workcell and the calibration target is held by the robot arm, the robot kinematics become

$$\left.\begin{aligned} \boldsymbol{T}_{0i}^{-1}\boldsymbol{T}_{gi} &= \boldsymbol{T}_s\boldsymbol{T}_{vi} \\ f(\boldsymbol{T}_{gi)}) &= 0 \end{aligned}\right\} \tag{4.3}$$

In this case, $\boldsymbol{T}_g$ is the target pose relative to the *Tool*$_0$ coordinate frame and $\boldsymbol{T}_s$ is the tool frame relative to the robot base frame.

In Eq. (4.2), $\boldsymbol{T}_{0i}$ is the given matrix and it depends on the robot pose; $\boldsymbol{T}_s$ is a fixed unknown matrix that defines the relationship between the tool and the robot tool mounting flange; $\boldsymbol{T}_{vi}$ is the transformation from the calibration target and the tool frame. This transformation can often be determined based on the sensing of the tool on the calibration target. Considering $\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ 0 & 1 \end{pmatrix}$, Eq. (4.2) can be written as the following component equations for rotation matrix and translation vector

$$\left.\begin{aligned} \boldsymbol{R}_{gi} &= \boldsymbol{R}_{0i}\boldsymbol{R}_{s}\boldsymbol{R}_{vi} \\ \boldsymbol{t}_{gi} &= \boldsymbol{R}_{0i}\boldsymbol{R}_{s}\boldsymbol{t}_{vi} + \boldsymbol{R}_{0i}\boldsymbol{t}_{s} + \boldsymbol{t}_{0i} \end{aligned}\right\} \tag{4.4}$$

For a fixed TCP case, Eq. (4.3) becomes

$$\left.\begin{aligned} \boldsymbol{R}_{0i}\boldsymbol{R}_{gi} &= \boldsymbol{R}_{s}\boldsymbol{R}_{vi} \\ \boldsymbol{R}_{0i}\boldsymbol{t}_{gi} + \boldsymbol{t}_{0i} &= \boldsymbol{R}_{s}\boldsymbol{t}_{vi} + \boldsymbol{t}_{s} \end{aligned}\right\} \tag{4.5}$$

If we only calibrate the tool position and ignore the orientation, as we do for a point sensor like a probe pin and a touch trigger probe, letting $\boldsymbol{R}_s = \boldsymbol{I}$ (i.e., 3×3 identity matrix), yields the calibration criteria for the position of the moving TCP as

$$\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\boldsymbol{t}_{vi} + \boldsymbol{R}_{0i}\boldsymbol{t}_{s} + \boldsymbol{t}_{0i} \tag{4.6}$$

and for the position of the fixed TCP case as

$$\boldsymbol{R}_{0i}\boldsymbol{t}_{gi} + \boldsymbol{t}_{0i} = \boldsymbol{t}_{vi} + \boldsymbol{t}_{s} \tag{4.7}$$

In general, there are two methods for calibrating the tool pose.

(1) Use an external measurement device like CMM. An external measurement device can be used to directly measure the position and orientation of the tool, as well as the tool mounting flange frame $Tool_0$,with respect to the coordinate frame of the measurement device. Therefore, the tool frame relative to $Tool_0$ can be derived from the coordinate transformation. The orientation measurement of the tool is normally based on the geometrical model and multiple position measurements of the tool. But it is not practical to use an external measurement device on the work floor because it is normally very expensive and time consuming.

(2) Use the robot as a measurement tool. In Eqs. (4.2) and (4.3), since normally $\boldsymbol{T}_{gi}$ could not be determined previously, $\boldsymbol{T}_s$ has to be solved simultaneously with $\boldsymbol{T}_{gi}$. In general this can be done with multiple measurements of the calibration target with different robot poses. $\boldsymbol{T}_s$ and $\boldsymbol{T}_{gi}$ are solved based on a constraint optimization procedure that minimizes the object function. In summary, for the moving TCP case the general calibration model is given by

$$\left.\begin{aligned} &\min\left\|f(\boldsymbol{T}_{gi})\right\| \\ &s.t.: \boldsymbol{T}_{gi} = \boldsymbol{T}_{0i}\boldsymbol{T}_{s}\boldsymbol{T}_{vi} \end{aligned}\right\} \tag{4.8}$$

Similarly, the general model for the fixed TCP case is given by

$$\left.\begin{aligned} &\min\left\|f(\boldsymbol{T}_{gi})\right\| \\ &s.t.: \boldsymbol{T}_{0i}\boldsymbol{T}_{gi} = \boldsymbol{T}_{s}\boldsymbol{T}_{vi} \end{aligned}\right\} \tag{4.9}$$

There are two types of variation to Eq. (4.8) or Eq. (4.9). One is for a point sensor-typed tool like a probe pin, a touch trigger probe or a point laser, where only a single point is measured each time. Another is for an array-type measurement tool like a camera or a laser scanner where multiple points can be measured each time.

TCP Calibration Formula for a Point Sensor

For the first case, in which the position of TCP is concerned, we can use kinematics of the robot in the form of Eq. (4.6). In this equation ($\boldsymbol{R}_{oi}$, $\boldsymbol{t}_{oi}$) is the robot *Tool*$_0$ pose for the *i*-th measurement and is given by the robot controller. $\boldsymbol{t}_{vi}$ is the corresponding coordinate of the measurement point on the calibration target with respect to the tool coordinate frame, and its value is given by the sensor. $\boldsymbol{t}_{gi}$ is the position of the corresponding measurement point in the calibration target frame. For example, when a touch sensor like a touch trigger probe contacts the calibration target during the measurement, we have $\boldsymbol{t}_{vi} = 0$. For simplification and being easy to implement, standard geometrical shapes are normally used as the calibration target. In this case Eq. (4.8) will be used and simplified as

$$\left.\begin{aligned} &\min\left\|f(\boldsymbol{t}_{gi})\right\| \\ &\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\boldsymbol{t}_{vi} + \boldsymbol{R}_{0i}\boldsymbol{t}_{s} + \boldsymbol{t}_{0i} \end{aligned}\right\} \tag{4.10}$$

where $f(\boldsymbol{t}_{gi})=0$ indicates a geometry constraint, which is a linear or nonlinear function of $\boldsymbol{t}_{gi}$. When a point constraint is used, i.e., $\boldsymbol{t}_{gi} = \boldsymbol{t}_{g(i+1)}$, Eq. (4.10) is simplified as the linear equation. When the constraint is a sphere, i.e., $(\boldsymbol{t}_{gi}-\boldsymbol{t}_{o})^{\mathrm{T}}(\boldsymbol{t}_{gi}-\boldsymbol{t}_{o}) = r^2$, a nonlinear least squares algorithm will be used to solve $\boldsymbol{t}_s$ as well as $\boldsymbol{t}_0$, the center of the sphere.

TCP Calibration Formula for an Array-Type Sensor

For an array type sensor like a camera or a laser scanner we need to calibrate the position and orientation. Eq. (4.2) will be used as the starting point. In this equation $\boldsymbol{T}_{0i}$ is the robot *Tool*$_0$ pose for the *i*-th measurement and is given by the robot controller. $\boldsymbol{T}_{vi}$ is the corresponding coordinate frame of the calibration target with respect to the tool coordinate frame, and its value is given by the sensor. $\boldsymbol{T}_{gi}$ is the coordinate frame of the calibration target for the *i*-th measurement. Since the sensor senses multiple points, and each time it can measure the position and orientation of the calibration target frame, there are some requirements for the calibration target such that the single measurement can determine the transformation between the sensor frame and the calibration target frame. During the multiple measurements, the calibration target is normally fixed and therefore the constraint function $f(\boldsymbol{T}_{gi})=0$ becomes $\boldsymbol{T}_{gi}=\boldsymbol{T}_{g(i+1)}=\boldsymbol{T}_{g}$. Eq. (4.6) can be written as

$$\boldsymbol{T}_{0(i+1)}^{-1}\boldsymbol{T}_{0i}\boldsymbol{T}_{s} = \boldsymbol{T}_{s}\boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}^{-1} \tag{4.11}$$

This problem can be modeled as a homogenous transformation equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$ where $\boldsymbol{A} = \boldsymbol{T}_{0(i+1)}^{-1}\boldsymbol{T}_{0i}$, $\boldsymbol{B} = \boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}^{-1}$, and $\boldsymbol{X} = \boldsymbol{Ts}$. The equation can be solved by an optimization procedure that minimizes the objective function $\boldsymbol{AX} - \boldsymbol{XB}$.

The following tables summarize various TCP calibration methods as the special case and implementation of Eqs. (4.10) and (4.11). A detailed derivation of the algorithm and description of each method will be the main topic of this chapter and will be discussed in the following sections.

Table 4.1 Position calibration of tool center point

Type of tool to be calibrated	Form of calibration target	Criteria equation and algorithm used to solve TCP	Feature
Probe pin	A single point	Geometry constraint $f(\boldsymbol{t}_{gi}) = 0$ becomes a fixed point constraint $\boldsymbol{t}_{gi} = \boldsymbol{t}_{g(i+1)}$, and Eq. (4.10) is simplified as $\boldsymbol{R}_{0i}\boldsymbol{t}_s + t_{0i} = \boldsymbol{R}_{0(i+1)}\ \boldsymbol{t}_s + \boldsymbol{t}_{0(i+1)}$. Linear least squares algorithm is used to solve $\boldsymbol{t}_s$	Alignment to ensure $\boldsymbol{t}_{gi} = \boldsymbol{t}_{g(i+1)}$, is based on visual measurement
Tool tip (weld gun)	A line (beam) detector	Geometry constraint $f(\boldsymbol{t}_{gi}) = 0$ becomes a line equation $\boldsymbol{t}_{gi} = \boldsymbol{t}_0 + \boldsymbol{n}t$, by letting $\boldsymbol{t}_{vi} = 0$, Eq. (4.10) becomes $\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\ \boldsymbol{t}_s + \boldsymbol{t}_{0i}$. Nonlinear least squares algorithm is used to solve $\boldsymbol{t}_s$ and the line detector position $(\boldsymbol{t}_0, n)$.(Stenberg, 2002; Thorne, 1995)	Alignment to ensure $\boldsymbol{t}_{vi} = 0$ is based on the search algorithm and is done automatically. Edge detection and search algorithm are used to align a feature point on the tool tip with the calibration beam with various robot orientations
Point measurement sensors: Touch trigger probe, LVDT sensor, Point laser; Proximity sensor	Sphere	Geometry constraint $f(\boldsymbol{t}_{gi}) = 0$ becomes a sphere equation: $(\boldsymbol{t}_{gi} - \boldsymbol{t}_0)^{\mathrm{T}}(\boldsymbol{t}_{gi} - \boldsymbol{t}_0) = r^2$. Eq. (4.10) will be used here, $\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\ \boldsymbol{t}_{vi} + \boldsymbol{R}_{0i}\ \boldsymbol{t}_s + \boldsymbol{t}_{0i}$. Nonlinear least squares algorithm is used to solve $\boldsymbol{t}_s$ and the sphere center $\boldsymbol{t}_0$ (Zhu et al., 2004)	t_{vi} is determined by the sensor reading for the distance between the sensor and the sphere surface
Touch trigger probe	Plate	Geometry constraint $f(\boldsymbol{t}_{gi}) = 0$ becomes two planes (plate) equation $\boldsymbol{n}(\boldsymbol{t}_{gi} - \boldsymbol{t}_{g(i+1)}) = d$, and Eq. (4.10) is simplified as $\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\ \boldsymbol{t}_s + \boldsymbol{t}_{0i}$. Linear algebra is used. $\{X, Y, Z\}$ components of $\boldsymbol{R}_{0i}$ is aligned respectively with the plane normal $\boldsymbol{n}$ to decouple and solve the components $\{\boldsymbol{t}_x, \boldsymbol{t}_y, \boldsymbol{t}_z\}$ of $\boldsymbol{t}_s$ separately (Tang and Gan, 2008)	Decouple $\{\boldsymbol{t}_x, \boldsymbol{t}_y, \boldsymbol{t}_z\}$ with special arrangement of $Tool_0$

Table 4.2 Pose calibration of tool center point

Type of tool to be calibrated	Form of calibration target	Criteria equation and algorithm used to solve TCP	Feature
Camera	Pattern with precision grid or dots	The camera is held by the robot arm (moving TCP calibration). Geometry constraint $f(T_{gi}) = 0$ becomes $T_{gi} = T_{g(i+1)} = T_g$, indicating the calibration pattern is fixed for each measurement. Based on Eq. (4.11), $\boldsymbol{T}_{0(i+1)}{}^{-1}\boldsymbol{T}_{0i}\boldsymbol{T}_s = \boldsymbol{T}_s\boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}{}^{-1}$,rotation matrix and translation vector are solved separately. Quaternion algebra and both linear least squares (Tsai and Lens, 1989; Shiu and Ahmad, 1989) and nonlinear optimization (Motai and Kosaka, 2008) are used to solve the equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$ for $\boldsymbol{T}_s$ $\boldsymbol{T}_s$ and $\boldsymbol{T}_g$ can be solved simultaneously by using the equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$ with linear and nonlinear least square (Zhuang, 1994; Dornaik, 1998; Park, 1994)	$\boldsymbol{T}_{vi}$ is treated as the camera extrinsic parameter and can be determined by using camera calibration algorithms. Basic approach: the camera is looking at the calibration target from different view point. For each camera position the camera pose relative to the target is determined. Multiple camera poses and measurements are used to obtain the target pose in the robot base frame and therefore camera TCP
Laser scanner	Calibration fixture with four non–parallel edges	The laser scanner is held by the robot arm (moving TCP calibration) Geometry constraint $f(\boldsymbol{T}_{gi}) = 0$ becomes $\boldsymbol{T}_{gi} = \boldsymbol{T}_{g(i+1)} = \boldsymbol{T}_g$, indicating the calibration fixture is fixed for each measurement with different robot pose. Based on Eq. (4.11), $\boldsymbol{T}_{0(i+1)}{}^{-1}\boldsymbol{T}_{0i}\boldsymbol{T}_s = \boldsymbol{T}_s\boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}{}^{-1}$. As long as $\boldsymbol{T}_{vi}$ can be determined, the equations can be solved by using the similar method to that for camera TCP calibration as described above. Both need to solve the equation of the form $\boldsymbol{AX} = \boldsymbol{XB}$ for $\boldsymbol{T}_s$ (Huissoon, 2000; Greer and Kim, 2000)	$\boldsymbol{T}_{vi}$ is the calibration fixture position in the scanner coordinate frame, determined by using a single measurement of the laser scanner. This requires a fixture with at least four topographically defined features like non-parallel edges. The laser scanner is shooting at the calibration fixture from different view points

(to be continued)

(Table 4.2)

Type of tool to be calibrated	Form of calibration target	Criteria equation and algorithm used to solve TCP	Feature
Laser scanner	Sphere	The laser scanner is fixed at the robot workcell (fixed TCP calibration). Geometry constraint $f(\boldsymbol{T}_{gi}) = 0$ becomes $\boldsymbol{t}_{gi} = \boldsymbol{t}_{g(i+1)} = \boldsymbol{t}_g$, indicating the sphere center is fixed. According to Eq. (4.5), $\boldsymbol{R}_{0i}\,\boldsymbol{t}_{gi} + \boldsymbol{t}_{0i} = \boldsymbol{R}_s\boldsymbol{t}_{vi} + \boldsymbol{t}_s$, $\boldsymbol{t}_v$ is the sphere center position in the scanner frame and can be determined by using the scanner measurement. The rotation matrix and translation vector are calibrated separately. $\boldsymbol{R}_s$ is determined with further constraint for the robot movement: $\boldsymbol{R}_{0i} = \boldsymbol{R}_{0(i+1)}$, $\boldsymbol{t}_s$ is calculated with linear least squares (Tang and Gan, 2007; Li, 2008b)	The approach to calibration is similar to that of cameras. The scanner is scanning the sphere from different points of view. The sphere position relative to the scanner frame is determined for each robot pose. Multiple robot poses and measurements are used to determine the sphere position relative to the robot $Tool_0$ frame as well as the scanner TCP
Laser scanner	Plane	The laser scanner is held by the robot arm (moving TCP calibration). The laser plane in the camera coordinate frame is given by $ax_c + by_c + cz_c = 1$. Geometric constraint $f(T_{gi}) = 0$ becomes the plane equation $nt_{gi} = d$. Connection between the camera frame and the robot base frame is given by Eq. (4.4): $\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\boldsymbol{R}_s\boldsymbol{t}_{vi} + \boldsymbol{R}_{0i}\boldsymbol{t}_s + \boldsymbol{t}_{0i}$, where $(\boldsymbol{R}_s, t_s)$ is the camera TCP that has been calibrated previously. $\boldsymbol{t}_{vi} = (x_c, y_c, z_c)_i$ is the laser point in the camera coordinate frame. Laser plane parameters (a, b, c) are determined together with the calibration plane parameters $(\boldsymbol{n}, d)$ by solving those equations (Xu et al., 2005)	Determine the laser plane relative to the camera coordinate frame. The camera pose relative to $Tool_0$ frame is assumed to be calibrated previously. During the measurement the camera is rotating along its origin, that is $\boldsymbol{T}_{0(i+1)} = \boldsymbol{T}_0(i)\boldsymbol{T}_c\boldsymbol{R}\boldsymbol{T}_c^{-1}$. This decouples the calibration parameters
A machine tool (i.e., Spindle) modeled as a cylinder	Touch trigger probe, laser displacement sensor, proximity sensor	When the positions of a point sensor like a touch trigger probe, a laser displacement sensor, or proximity sensor have been calibrated using the method as discussed above, they can be used together with the robot as a measurement tool. Thus the position and orientation of the machine tool can be directly measured (Tang et al., 2003)	Position of the tool is measured directly and the orientation is determined by using a group of measurement data to fit the model

4.1.1 Calibration Target with Geometry Constraint: Point

A simple case is a point constraint and $f(\boldsymbol{t}_{gi})=0$ becomes linear constraint

$$\boldsymbol{t}_{gi} = \boldsymbol{t}_{g(i+1)} = \boldsymbol{t}_g \ (i = 1,\ 2,\ \ldots,\ N) \tag{4.12}$$

where $\boldsymbol{t}_g$ is a fixed unknown position.

Substituting Eq. (4.12) into Eq. (4.10) and letting $\boldsymbol{t}_{vi}=0$ give linear equations

$$\boldsymbol{R}_{0i}\boldsymbol{t}_s + \boldsymbol{t}_{0i} = \boldsymbol{R}_{0(i+1)}\boldsymbol{t}_s + \boldsymbol{t}_{0(i+1)} \tag{4.13}$$

where $\boldsymbol{t}_g$ can be solved together with the TCP position $\boldsymbol{t}_s$. In order to solve $\boldsymbol{t}_s$ and $\boldsymbol{t}_g$ from Eq. (4.13) at least two points have to be used. When more points are used Eq. (4.13) becomes overdetermined linear equation of the form $\boldsymbol{Ax} = \boldsymbol{b}$ and can be solved with a linear least squares algorithm. $\boldsymbol{t}_s$ (where $\boldsymbol{x} = \boldsymbol{t}_s$) can be written as the closed form as

$$\boldsymbol{t}_s = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\boldsymbol{b} \tag{4.14}$$

where
$$A = \begin{pmatrix} \boldsymbol{R}_{01} - \boldsymbol{R}_{02} \\ \vdots \\ \boldsymbol{R}_{0i} - \boldsymbol{R}_{0(i+1)} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{t}_{02} - \boldsymbol{t}_{01} \\ \vdots \\ \boldsymbol{t}_{0(i+1)} - \boldsymbol{t}_{0i} \end{pmatrix} \tag{4.15}$$

where $(i+1)$ is the number of measurements with different tool orientation. It is equivalent to the solution that minimizes $\|A\boldsymbol{t}_s - \boldsymbol{b}\|^2$, that is

$$G(\boldsymbol{t}_s) = \sum_{i=1}^{N} \left\| (\boldsymbol{R}_{0i} - \boldsymbol{R}_{0(i+1)})\boldsymbol{t}_s - \boldsymbol{t}_{0i} + \boldsymbol{t}_{0(i+1)} \right\|^2 \tag{4.16}$$

For the linear least squares problem $\boldsymbol{Ax} = \boldsymbol{b}$, assuming A and $\boldsymbol{b}$ are erroneous due to measurement and robot positioning errors, solution $\boldsymbol{x}$ of the linear equation with errors complies to

$$(\boldsymbol{A} + \boldsymbol{E})\boldsymbol{x} = \boldsymbol{b} + \delta\boldsymbol{b} \tag{4.17}$$

where $\boldsymbol{E}$ and $\delta\boldsymbol{b}$ are the introduced errors in $\boldsymbol{A}$ and $\boldsymbol{b}$ respectively.

Then the following bounds will be used to evaluate the reliability of the solution (Charles and Lawson, 1974)

$$\frac{\|\boldsymbol{x}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le \frac{\kappa(A)}{1-\|\boldsymbol{E}\|\|A^{+}\|}\left((1+\kappa(A)\frac{\|\boldsymbol{r}\|}{\|A\|\|\boldsymbol{x}\|})\frac{\|\boldsymbol{E}\|}{\|A\|}+\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}\frac{\|\boldsymbol{b}\|}{\|A\|\|\boldsymbol{x}\|}\right) \tag{4.18}$$

where $\kappa(A) = \|A\|\|A^{+}\|$, A^{+} is the pseudo inverse of matrix A, and $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$.

It is shown from the equation above that $\kappa(A)$, which is also known as the condition number of the matrix $\boldsymbol{A}$, is an amplification factor of the errors in $\boldsymbol{A}$ and $\boldsymbol{b}$. To reduce the effect of the errors on the calibration result (that is to reduce $\|\boldsymbol{x} - \boldsymbol{x}\| / \|\boldsymbol{x}\|$), a proper selection of robot poses ($\boldsymbol{R}_{0i}$) is required so that no large condition number of $\boldsymbol{A}$ is involved.

In the implementation, a probe pin point is used as the calibration target. It is placed at the location inside the robot workcell that can be reached from the robot tool that will be calibrated as shown in Fig. 4.2. Then manually jog the robot TCP to align with the pin point with different tool orientation. The visual inspection or sensor inspection is used to ensure that the TCP coincides with the fixed pin point. When $N = 3$, meaning 4 such points are used, the calibration procedure is then so called 4-point TCP calibration. This algorithm has been built into the robot control system and supported by many off-line programming systems and is provided as the tool-calibration procedure by robot manufacturers.

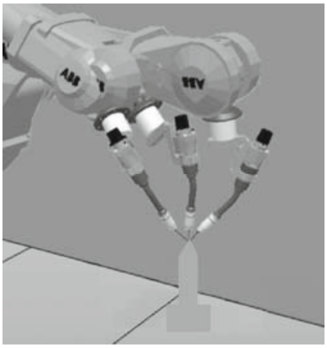


Fig. 4.2. Tool tip calibration with a single point constraint

4.1.2 Calibration Target with Geometry Constraint: Line

When a line is used as the calibration target, $f(\boldsymbol{t}_{gi}) = 0$ becomes linear constraint

$$\boldsymbol{t}_{gi} = \boldsymbol{t}_{g0} + \boldsymbol{n}k_i, \tag{4.19}$$

where $\boldsymbol{t}_{g0}$ is a point on th e line and it is a fixed unknown position. $\boldsymbol{n}$ is an orientation of the line, and k_i is a scalar parameter indicating the distance between two line points $\boldsymbol{t}_{g0}$ and $\boldsymbol{t}_{gi}$. When the TCP of the tool is contacting the line we have $\boldsymbol{t}_{vi} = 0$, yielding

$$\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\boldsymbol{t}_s + \boldsymbol{t}_{0i} \tag{4.20}$$

Combining Eqs. (4.19) and (4.20) results in

$$\boldsymbol{t}_{g0} + \boldsymbol{n}k_i = \boldsymbol{R}_{0i}\boldsymbol{t}_s + \boldsymbol{t}_{0i} \tag{4.21}$$

In general $\boldsymbol{t}_s$ can be solved by using a linear least squares algorithm that minimizes the objective function with respect to $\{\boldsymbol{t}_s, \boldsymbol{n}, \boldsymbol{t}_{g0}, k_1, k_2, ..., k_N\}$ when $i=N$

$$G(\boldsymbol{t}_s, \boldsymbol{n}, \boldsymbol{t}_{g0}, k_1, k_2, ..., k_N) = \sum_{i=1}^{N} \left\| \boldsymbol{t}_{g0} + \boldsymbol{n}k_i - \boldsymbol{R}_{0i}\boldsymbol{t}_s - \boldsymbol{t}_{0i} \right\|^2 \tag{4.22}$$

where $\boldsymbol{R}_{0i}$ and $\boldsymbol{t}_{0i}$ are known parameters and are obtained from the robot controller.

The direction of the line can be calibrated separately. When the direction of the line becomes a known parameter, Eq. (4.21) becomes the linear equation in the form

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{R}_{01} - \boldsymbol{R}_{02} & -\boldsymbol{n} & \boldsymbol{n} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \boldsymbol{R}_{02} - \boldsymbol{R}_{03} & 0 & -\boldsymbol{n} & \boldsymbol{n} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & \\ \boldsymbol{R}_{0i} - \mathbf{R}_{0(i+1)} & 0 & 0 & 0 & 0 & \cdots & 0 & -\boldsymbol{n} & \boldsymbol{n} \end{pmatrix}_{3i\times(i+3)} \tag{4.23}$$

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{t}_s \\ \kappa_1 \\ \kappa_2 \\ \vdots \\ k_i \end{pmatrix}_{(i+3)\times 1}, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{t}_{02} - \boldsymbol{t}_{01} \\ \boldsymbol{t}_{03} - \boldsymbol{t}_{02} \\ \vdots \\ \boldsymbol{t}_{0i} - \boldsymbol{t}_{0(i-1)} \\ \boldsymbol{t}_{0(i+1)} - \boldsymbol{t}_{0i} \end{pmatrix}_{3i}$$

where $(i+1)$ is the number of measurements with different tool orientation. $\boldsymbol{x}$ (thus $\boldsymbol{t}_s$) can be solved by using a linear least squares algorithm as the closed form $\boldsymbol{x} = (\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})^{-1}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{b}$ that is equivalent to minimization of $\|\boldsymbol{Ax} - \boldsymbol{b}\|^2$.

In the implementation (Stenberg, 2002; Thorne, 1995) as shown in Fig.4.3(a), a line detector, such as a reading fork of light-beam type, is placed in the robot workcell. Neither the position nor the direction of the beam needs to be known. It should be possible to detect the interruption of the beam. Such a detection is an indication that the surface of the tip is tangential to the detector line. There is a feature point on the tip of the tool that can align with the line by using the search procedure. Fig.4.3(b) illustrates the movement of the robot tool relative to the line detector. The edge points *A*, *B*, and *C* are detected to determine the feature point *D*. The orientation of the line can be determined by aligning the feature point of the tip to the line twice with the same robot orientation. The orientation of the line is given by $\boldsymbol{n} = (\boldsymbol{t}_{02} - \boldsymbol{t}_{01}) / \|(\boldsymbol{t}_{02} - \boldsymbol{t}_{01})\|$ where $\boldsymbol{t}_{02}$ and $\boldsymbol{t}_{01}$ are the $Tool_0$ positions for two measurements.

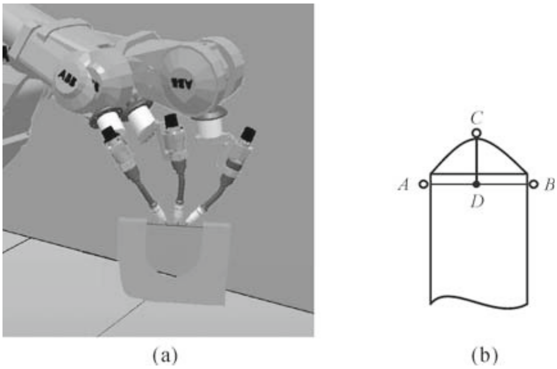


Fig. 4.3. (a) A light-beam type line detector is placed at the workcell. The robot moves around such that the tip of the tool can break the line and finally align with the line with different orientation. (b) The feature point *D* on the tip is aligned with the line using the search procedure to find the edge points *A*, *B* and *C* of the tip

4.1.3 Calibration Target with Geometry Constraint: Sphere

In this case a sphere with known radius r is used as the calibration target. The constraint function $f(\boldsymbol{t}_{gi}) = 0$ becomes

$$(\boldsymbol{t}_{gi} - \boldsymbol{t}_c)^{\mathrm{T}} (\boldsymbol{t}_{gi} - \boldsymbol{t}_c) = r^2 \tag{4.23}$$

where $\boldsymbol{t}_{gi} = (x_i, y_i, z_i)^{\mathrm{T}}$ ($i = 1, 2, \ldots, N$) are the measurement points on the sphere; $\boldsymbol{t}_c = (x_c, y_c, z_c)^{\mathrm{T}}$ is the unknown sphere position; r is the known sphere radius. In Eq. (4.10), let $\boldsymbol{t}_{vi} = 0$, indicating the contact of the TCP on the calibration sphere and Eq. (4.10) is simplified as

$$\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i} \boldsymbol{t}_s + \boldsymbol{t}_{0i} \tag{4.24}$$

The TCP value $\boldsymbol{t}_s$ can be solved together with the sphere position $\boldsymbol{t}_c$ by solving Eqs. (4.24) and (4.25). These are nonlinear equations and can be solved by using a nonlinear least squares algorithm by minimizing the following objective function with respect to $\boldsymbol{t}_s$ and $\boldsymbol{t}_c$

$$G(\boldsymbol{t}_s, \boldsymbol{t}_c) = \sum_i \left\| (\boldsymbol{R}_{0i} \boldsymbol{t}_s + \boldsymbol{t}_{0i} - \boldsymbol{t}_c)^{\mathrm{T}} (\boldsymbol{R}_{0i} \boldsymbol{t}_s + \boldsymbol{t}_{0i} - \boldsymbol{t}_c) - r^2 \right\|^2 \tag{4.25}$$

Eq. (4.26) can be solved by the modified Gauss-Newton method and implemented with the Levenberg-Marquardt algorithm (Marquardt, 1963). The result can be written as

$$\delta\gamma = \left(\left(\frac{\partial G}{\partial \gamma} \right)^{\mathrm{T}} \left(\frac{\partial G}{\partial \gamma} \right) + \lambda \boldsymbol{I} \right)^{-1} \left(\frac{\partial G}{\partial \gamma} \right)^{\mathrm{T}} \boldsymbol{e} \tag{4.26}$$

where γ represents the vector of error parameters $\boldsymbol{t}_s$ and $\boldsymbol{t}_c$, $\boldsymbol{e}$ is the error vector that is the difference between the measured value and calculated value with error parameters set to zero. The factor $\lambda > 0$ is used to determine the tendency of the algorithm towards either the steepest descent method (for a large value) or Gauss-Newton method (for a small value). Since $\lambda \neq 0$, numerical stability of the algorithm is maintained because any potential ill-determinacy in $\dfrac{(\partial G / \partial \gamma)^{\mathrm{T}}}{(\partial G / \partial \gamma)}$ is eliminated by the addition of $\lambda \boldsymbol{I}$.

As an example of implementation of the algorithm described above, a touch-trigger probe is mounted onto the robot arm so the robot can operate as a coordinate measuring machine. The probe is calibrated with the sphere as the calibration target. The sphere is placed in the robot cell. Program or jog the robot to a position where the tool can touch the surface of the sphere from various robot poses. Record the robot *Tool*$_0$ position ($\boldsymbol{R}_{0i}$, $\boldsymbol{t}_{0i}$) for each pose and the TCP can be solved based on Eqs. (4.24) and (4.25). The calibration setup is illustrated in Fig. 4.4. Note that the actual TCP of the trigger probe is the center of the small touch sphere and it will not contact the calibration sphere (the big one). The contact point on the touch sphere varies for each robot pose. The effective radius of the constraint sphere in Eq. (4.24) should be the sum of two radii for the calibration sphere and touch sphere.

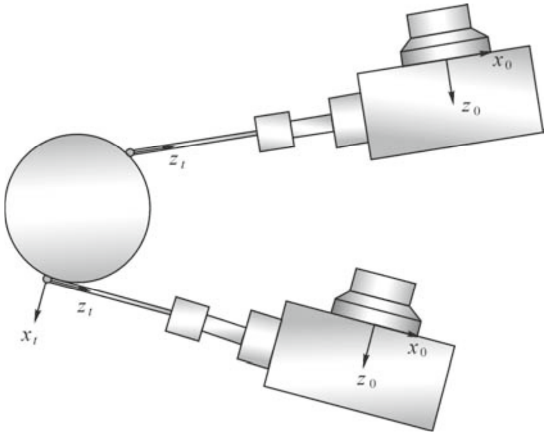


Fig. 4.4. A touch trigger probe TCP calibration illustration with the calibration target of a sphere. The sphere is fixed at the workcell and its location is unknown. The TCP is moved by the robot to touch the surface of the sphere with different orientation

4.1.4 Calibration Target with Geometry Constraint: Plane

By using a point or a sphere as the geometry constraint to form an optimization scenario, we can calibrate the TCP value as demonstrated previously. All the variables of TCP are solved simultaneously in the sense of linear or nonlinear least squares. An alternative approach is to separate those variables to be solved. One variable is solved at a time for each measurement. For the first approach, single value decomposition (SVD) analysis is usually used to make sure all the variables are observable in the measurement. The accuracy of the calibration for an individual variable is dependent on how observable this variable is in the measurement. For the later approach, since all the variables are decoupled, the calibration accuracy is easy to control in the measurement.

If a plate with a known thickness is used as the calibration target, three components of the TCP value can be calibrated separately. In this case the geometry constraint $f(\boldsymbol{t}_{gi}) = 0$ in the general calibration model Eq. (4.10) becomes

$$\boldsymbol{n}(\boldsymbol{t}_{gi} - \boldsymbol{t}_{gj}) = d \quad (i \neq j) \tag{4.27}$$

where $\boldsymbol{t}_{gi}$ and $\boldsymbol{t}_{gj}$ are the measurement points on the both sides of the plate, respectively; d is the known plate thickness. Let $\boldsymbol{t}_{vi} = 0$, indicating the contact of the TCP on the calibration plate, and Eq. (4.10) is simplified as $\boldsymbol{t}_{gi} = \boldsymbol{R}_{0i}\,\boldsymbol{t}_s + \boldsymbol{t}_{0i}$.

In an implementation, place the plate in the robot workcell that the robot can reach easily. The accurate position and orientation of the plate are not known. Program or jog the robot that is equipped with a touch-trigger probe to be calibrated, to measure five points on the plate surface to determine its orientation on the surface plane. It has been proven that the orientation measurement is independent of the TCP value once the robot keeps the same pose during the measurement of surface points. After the plate normal is determined, align the axis *x*, *y* or *z* of the robot *Tool*$_0$ frame with the plate surface orientation, respectively, to measure the surface offset plane. Align the axis –*x*, –*y*, and –*z* of the robot *Tool*$_0$ frame with the plate surface orientation, respectively, to measure the surface plane offset again. Then TCP components t_x, t_y, and t_z, of the TCP value $\boldsymbol{t}_s$ can be calculated, respectively, based on the above two measurements. The following will give the algorithm.

4.1.4.1 Measurement of Surface Plane Normal

First we will prove the normal measurement of the surface plane is independent of the robot TCP once the robot keeps the same orientation during the surface point measurements.

Assume that the TCP of the robot is $\boldsymbol{t}_s$. The measured point with the tool in the robot base is then determined by Eq. (4.25).

To measure a plane at least three measurement points are required. For simplicity of analysis we use three points. The plane determined by more than three points is based on least squares. But the principle for three points still holds. The plane normal can be obtained by the cross product of two vectors on the plane that are formed by the differences between three measured points

$$\boldsymbol{n} = (\boldsymbol{y}_1 - \boldsymbol{y}_2) \times (\boldsymbol{y}_1 - \boldsymbol{y}_3) \tag{4.28}$$

where $\boldsymbol{n}$ is the plane normal and $\boldsymbol{y}_1$, $\boldsymbol{y}_2$, and $\boldsymbol{y}_3$ are the measured positions on the plane. During the measurement of three points the robot keeps the same orientation as illustrated in Fig. 4.5, meaning that the rotation matrix $\boldsymbol{R}_0$ of *Tool*$_0$ is constant. From Eq. (4.25) we can derive Eq. (4.29) into

$$\boldsymbol{n} = (\boldsymbol{t}_{01} - \boldsymbol{t}_{02}) \times (\boldsymbol{t}_{01} - \boldsymbol{t}_{03}) \tag{4.29}$$

where $\boldsymbol{t}_{01}$, $\boldsymbol{t}_{02}$, and $\boldsymbol{t}_{03}$ are the translation vectors of *Tool*$_0$ corresponding to the measurement of three points. It can be seen that the plane normal calculated with Eq. (4.30) is independent of the TCP value. Then the plane offset is obtained by

$$\begin{aligned} \boldsymbol{d} &= \boldsymbol{n}(\boldsymbol{R}_0 \boldsymbol{t}_s + \boldsymbol{t}_{\boldsymbol{0}}) \\ &= n_x(r_{11}t_x + r_{12}t_y + r_{13}t_{13} + p_x) + n_y(r_{21}t_x + r_{22}t_y + r_{23}t_{13} + p_y) \\ &\quad + n_z(r_{31}t_x + r_{32}t_y + r_{33}t_{13} + p_z) \end{aligned} \tag{4.30}$$

Where (n_x, n_y, n_z) are the three components of the plane normal $\boldsymbol{n}$, $(r_{11}, r_{12}, \ldots, r_{33})$ are the components of the rotation matrix $\boldsymbol{R}_0$ and $(p_x, p_y, p_z)(p_x, p_y, p_z)$ forms the translation vector $\boldsymbol{t}_0$.

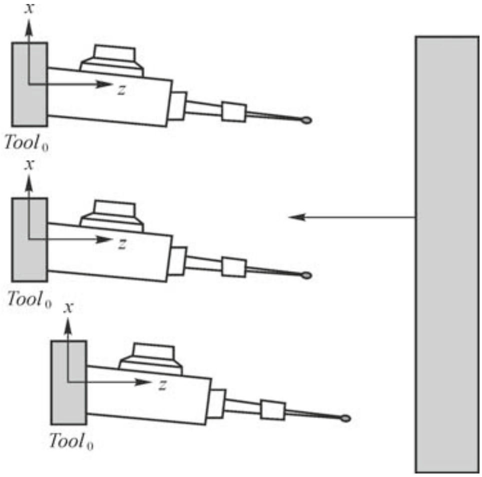


Fig. 4.5. Measurement of the plane normal, proving that the plane normal measured is independent of the robot TCP value as long as the robot pose remains unchanged during the measurement

4.1.4.2 Calibration of $\boldsymbol{t}_x$, $\boldsymbol{t}_y$, and $\boldsymbol{t}_z$

In the following we give the mathematical derivation of how to calibrate $\boldsymbol{t}_x$. Calibration of $\boldsymbol{t}_y$ and $\boldsymbol{t}_z$ is similar to that of $\boldsymbol{t}_x$. In the first step, the surface point on one side of the plate is measured. During the measurement, align the *X* axis of the robot $Tool_0$ with the plane normal by letting

$$(r_{11},\ r_{21},\ r_{31}) = (n_x,\ n_y,\ n_z) \tag{4.31}$$

By substituting Eq. (4.32) into Eq. (4.31), t_y and t_z will be eliminated because of the orthogonal of rotation matrix $\boldsymbol{R}_0$. Thus, Eq. (4.31) can be reduced as

$$d_1 = 3t_x + \boldsymbol{nt}_{01} \tag{4.32}$$

In the second step, the surface point on the other side of the plate is measured. During the measurement align the *X* axis of the robot $Tool_0$ with the plane normal as follows

$$(r_{11},\ r_{21},\ r_{31}) = -(n_x,\ n_y,\ n_z) \tag{4.33}$$

Eq. (4.31) is then reduced to

$$d_2 = -3t_x + \boldsymbol{nt}_{02} \tag{4.34}$$

Combining Eqs. (4.33) and (4.35) results in

$$t_x = \frac{(d_1 - d_2) - \boldsymbol{n}(\boldsymbol{t}_{01} - \boldsymbol{t}_{02})}{6} \tag{4.35}$$

where (d_1-d_2) is the thickness of the plate which is a given parameter. $\boldsymbol{t}_{01}$ and $\boldsymbol{t}_{02}$ are the translation vectors of the robot $Tool_0$ frame used for the point measurements on the two sides of the plate. $\boldsymbol{n}$ is the plane normal. They are all known parameters. Therefore, t_x can be calculated independently of the other two components t_y and t_z.

For the calibration of t_y, it is similar to that of t_x. During the two measurements, letting $(r_{12}, r_{22}, r_{32}) = (n_x, n_y, n_z)$ and $(r_{12}, r_{22}, r_{32}) = -(n_x, n_y, n_z)$, respectively, results in

$$t_y = \frac{(d_1 - d_2) - \boldsymbol{n}(\boldsymbol{t}_{01} - \boldsymbol{t}_{02})}{6} \tag{4.36}$$

For the calibration of t_x, letting $(r_{13}, r_{23}, r_{33}) = (n_x, n_y, n_z)$ and $(r_{13}, r_{23}, r_{33}) = -(n_x, n_y, n_z)$ for the two measurements, respectively, gives

$$t_z = \frac{(d_1 - d_2) - \boldsymbol{n}(\boldsymbol{t}_{01} - \boldsymbol{t}_{02})}{6} \tag{4.37}$$

Fig. 4.6 illustrates how to implement the calibration procedure for components t_x and t_z.

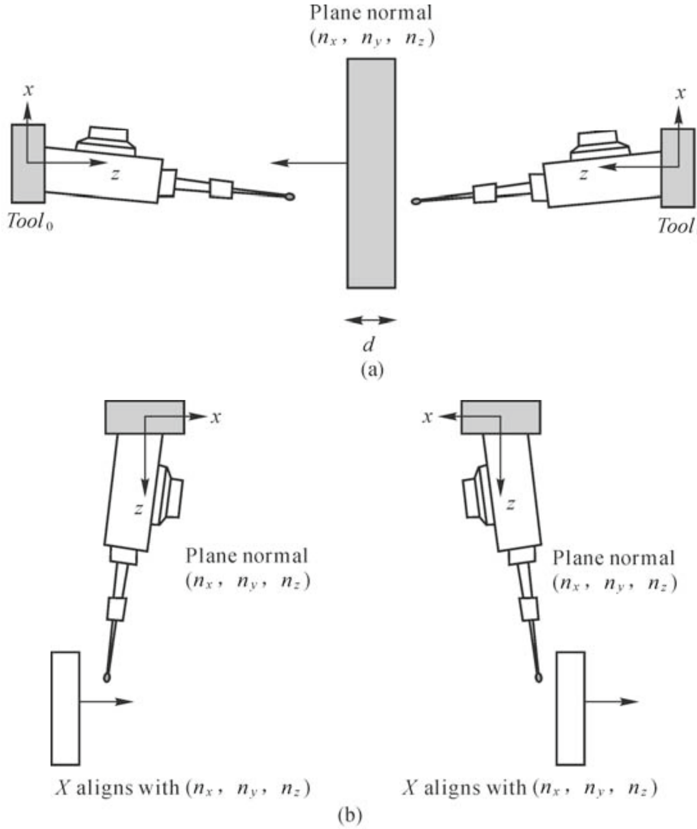


Fig. 4.6. (a) Illustration for the calibration of T_z. Two points on the two sides of the plate are measured with the z axis of robot $Tool_0$ frame aligned with the plane normal in opposite directions. (b) Illustration for the calibration of T_x. Two points on the two sides of the plate are measured with the x axis of robot $Tool_0$ frame aligned with the plane normal in opposite directions

4.2 TCP Calibration for a Point Laser

When an industrial robot with a point laser (laser displacement sensor) is used as a measurement tool, it is necessary to calibrate the position and orientation of the laser sensor relative to the robot-mounting flange $Tool_0$ frame. Fig. 4.7 shows a robot measurement system with a laser displacement sensor and the setup of TCP calibration (Zhu et al., 2004).

In the conventional TCP calibration procedure, as described in Section 4.1.1, the tool tip is moved to a fixed point in space using a robot reach pendant with jog move command, but with different orientations. This fixed point is called the target point. The robot positions are recorded after a visual inspection has verified that the TCP coincides with the target point. With this data available, the position of the TCP relative to $Tool_0$ can be determined. However, this procedure requires visual inspection to determine if the TCP coincides with the target point. The accuracy of calibration varies with operators. For non-contact sensors, such as laser displacement sensors, it is difficult to visually determine if the reference point on the beam corresponds to the target point. This problem can be overcome through the use of an automatic calibration algorithm and simple geometry with known dimensions.

4.2.1 Algorithms

The laser displacement sensor to be used in the calibration consists of a laser pointer and an imaging system. A CCD array in the imaging system detects the position of a laser spot. It operates based on a triangulation principle. As an example, an optoNCDT 1800 laser sensor from Micro Epsilon is used. However, the approach used here can be applied to other similar types of non-contact sensors. The displacement reading, L, from the laser sensor indicates the displacement of the laser beam point to a reference position. When this sensor is mounted onto a robot or other mechanical device with 3D position coordinates, we can measure the 3D position of the point that the laser beam is shooting at (x, y, z).

Assume that there is a virtual reference point p in the laser beam that corresponds to a laser sensor reading of zero. When the laser sensor is mounted on the robot-mounting flange, the 3D position of the virtual reference point p is (x_0, y_0, z_0) in $Tool_0$. Also assume that the laser beam orientation is (n_x, n_y, n_z) relative to $Tool_0$. Then, for any point p on the laser beam that corresponds to laser sensor reading L, the 3D coordinate of the point in $Tool_0$ is

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{n}L \tag{4.38}$$

where $\boldsymbol{x} = (x, y, z)^T$, $\boldsymbol{x}_0 = (x_0, y_0, z_0)^T$, and $\boldsymbol{n} = (n_x, n_y, n_z)^T$.

The task of the laser TCP calibration is to determine (x_0, y_0, z_0) and (n_x, n_y, n_z) such that the 1D reading (L) of the laser sensor can be converted into a 3D position based on Eq. (4.39). The orientation of the laser beam (n_x, n_y, n_z) is determined using the alignment method and the laser position (x_0, y_0, z_0) is determined based on a nonlinear optimization algorithm. When the laser displacement sensor is used as the robot measurement tool, we can calculate the position of the robot end effecter based on the sensor reading using the equation.

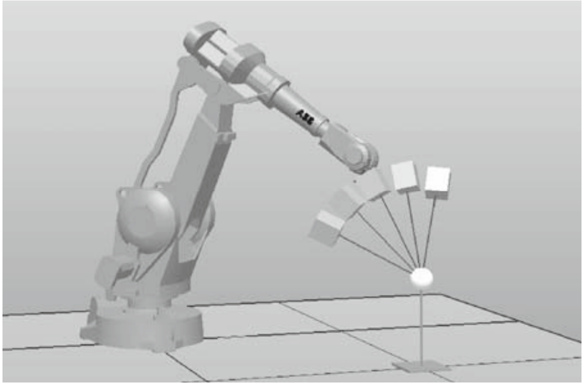


Fig. 4.7. A robot measurement system with a laser displacement sensor mounted on the robot arm. A sphere in the workcell is the calibration target and is used to calibrate the TCP of the laser sensor

4.2.2 Calibration of Laser Beam Orientation (n_x, n_y, n_z)

The value of the laser beam orientation (n_x, n_y, n_z) is determined by using the following procedure, as shown in Fig.4.8. Project the laser beam on a screen so that the laser spot can be viewed from a stationary camera. Record the current image position of the laser spot as (x_r, y_r) and the robot $Tool_0$ pose as the transform $\boldsymbol{T}_0$, where $\boldsymbol{T}_0$ is a 4×4 homogenous matrix representing the 3D position and orientation of a coordinate system. Move the robot along its Z axis of the $Tool_0$ coordinate system (use relative transnational movement with respect to $Tool_0$). The laser spot position will move away from its original position. Then translate (no rotation) the robot along its x and y axes, respectively, of $Tool_0$ so that the laser spot coincides with its original position (x_r, y_r). The alignment will be conducted with the vision feedback control of the robot. The image offset of the current laser spot position to the original one will be fed back to the robot controller and it will be used to command the next robot move until the offset reaches a preset value (i.e. 30 μm which is the robot repeatability.) Record the current robot $Tool_0$ pose as $\boldsymbol{T}_1$. The orientation of the laser beam in the robot base coordinate system is simply the connection of $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$. $\boldsymbol{v}_b = (p_0 - p_1)/\quad p_0 - p_1)$ where $\boldsymbol{v}_b$ is the orientation vector of the laser beam under the robot base coordinate system; p_0 and p_1 are the translation portions of the matrix $\boldsymbol{T}_0$ and $\boldsymbol{T}_1$, respectively. If more than two points are used, a 3D line-fitting algorithm will be used to determine the orientation vector. The orientation in $Tool_0$ is therefore converted from the robot base coordinate system by $\boldsymbol{v}_t = R \cdot \boldsymbol{v}_b$, where $\boldsymbol{v}_t$ is the laser beam orientation in $tool_0$ with the form of (n_x, n_y, n_z); $\boldsymbol{R}$ is the rotation matrix of the matrix $\boldsymbol{T}_0$.

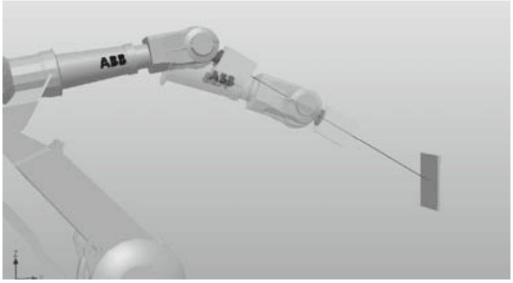


Fig. 4.8. Calibration setup of laser beam orientation. The calibration is conducted by aligning the laser beam with a fixed pin hole on the plate twice. The robot only translates during the alignment

4.2.3 Calibration of Laser Sensor Position (x_0, y_0, z_0)

From the general model Eq. (4.2) and its variation Eq. (4.4) we have the kinematic relation as

$$\boldsymbol{t}_{g\iota} = \boldsymbol{R}_{\upsilon i}\boldsymbol{R}_s\boldsymbol{t}_{vi} + \boldsymbol{R}_{\upsilon i}\boldsymbol{t}_s + \boldsymbol{t}_{\upsilon i} = \boldsymbol{R}_{\upsilon i}(\boldsymbol{R}_s\boldsymbol{t}_{vi} + \boldsymbol{t}_s) + \boldsymbol{t}_{\upsilon i} = \boldsymbol{R}_{\upsilon i}(\boldsymbol{n}L_i + \boldsymbol{x}_\upsilon) + \boldsymbol{t}_{0i} \qquad (4.39)$$

with $\boldsymbol{t}_s = \boldsymbol{x}_0$, $\boldsymbol{t}_{vi} = (0,\ 0,\ L_i)^{\mathrm{T}}$ and $\boldsymbol{R}_s = (\boldsymbol{o},\ \boldsymbol{a},\ \boldsymbol{n})$, where $\boldsymbol{a}$, $\boldsymbol{o}$ and $\boldsymbol{n}$ are the vector component s of the rotation matrix in X, Y and Z directions, respectively. $\boldsymbol{n} = (n_x,\ n_y,\ n_z)$ is the laser beam orientation that has been calibrated previously; L_i is the laser sensor reading; ($\boldsymbol{R}_0$, $\boldsymbol{t}_0$) represents the pose (rotation and translation) of the robot mounting flange frame ($Tool_0$) in the robot base coordinate frame. They can be obtained from the robot controller. $\boldsymbol{x}_0 = (x_0, y_0, z_0)^{\mathrm{T}}$ is the laser sensor position that needs to be determined. When the laser sensor in conjunction with the robot is used to measure certain geometry, such as a sphere with known diameter r, all the measurement points have to satisfy the sphere equation. The constraint in the general model Eq. (4.2) becomes

$$(\boldsymbol{t}_{gi} - \boldsymbol{x}_c)^{\mathrm{T}}(\boldsymbol{t}_{gi} - \boldsymbol{x}_c) = r^2 \qquad (4.40)$$

where $\boldsymbol{x}_c = (x_c, y_c, z_c)^{\mathrm{T}}$ is the center position of the sphere, which is an unknown parameter. $\boldsymbol{t}_{gi}$ represents the position of the laser beam on the sphere surface.

The TCP value $\boldsymbol{x}_0$ can be solved, together with the sphere position $\boldsymbol{x}_c$, by solving Eqs. (4.40) and (4.41). These are nonlinear equations and can be solved by using a nonlinear least squares algorithm by minimizing the following objective function with respect to $\boldsymbol{x}_0$ and $\boldsymbol{x}_c$

$$G(\boldsymbol{x}_0, \boldsymbol{x}_c) = \sum_i \left\| (\boldsymbol{R}_{0i}(\boldsymbol{n}L_i + \boldsymbol{x}_0) + \boldsymbol{t}_{0i} - \boldsymbol{x}_c)^{\mathrm{T}}(\boldsymbol{R}_{0i}(\boldsymbol{n}L_i + \boldsymbol{x}_0) + \boldsymbol{t}_{0i} - \boldsymbol{x}_c) - r^2 \right\|^2 \qquad (4.41)$$

The well-known nonlinear least squares algorithm—the Levenberg-Marquardt algorithm (Marquardt, 1963) is applied to minimize the error function defined in Eq. (4.42) by selecting optimized variables (x_0, y_0, z_0, x_c, y_c, z_c), and initial values for (x_0, y_0, z_0) and (x_c, y_c, z_c) are required. The initial value of the laser sensor position (x_0, y_0, z_0) can be obtained with the conventional robot four points TCP calibration procedure. The initial value of the sphere position (x_c, y_c, z_c) can be estimated by measuring the sphere position with the laser sensor that has not been mounted on the robot. At this point the laser sensor has not been calibrated yet, but the initial TCP value can be used.

4.2.4 Experimental Results

An ABB robot (RIB4400) is used in a workcell. The laser displacement sensor (optoNCDT 1800) is from Micro-Epsilon. Its resolution is 1 µm and the range of measurement is ±5 mm. The sensor's working distance is about 25 mm, and is mounted on the robot.

In the measurement of the laser beam orientation, a total of 13 robot movement positions are used. The robot $Tool_0$ positions after alignment are recorded and the laser orientation is calculated based on 13-point data fitting. The calculated orientation vector (n_x, n_y, n_z) is (0.003141, 0.8647, –0.50228) and the fitting error is $\sigma = 0.12$ mm.

For the calibration of the laser sensor TCP position (x_0, y_0, z_0), a sphere with diameter of 14.28 mm is placed at the position where the robot can easily reach it, as shown in Fig. 4.7. The sphere position (x_c, y_c, z_c) is unknown. It can be determined along with the laser sensor TCP. There are 3 test cases corresponding to different sphere positions to check the algorithm repeatability and stability. For each test case, 17 points are measured on the sphere with different robot orientations. The calculated results are listed in the following table.

Table 4.1 Experimental results for laser sensor TCP calibration (mm)

Test case	x_0	y_0	z_0	x_c	y_c	z_c	Std. Dev.	Max. Dev.
1	1.95	213.35	75.95	372.22	1361.30	685.66	0.07	0.12
2	1.89	213.28	75.90	372.80	1360.20	684.45	0.07	0.13
3	1.88	213.40	75.86	370.10	1358.80	683.89	0.07	0.12
Mean	1.91	213.34	75.90					
Std. Dev.	0.04	0.06	0.05					

In Table 4.3, (x_0, y_0, z_0) is the laser TCP position and (x_c, y_c, z_c) is the sphere position. Columns 8 and 9 indicate the standard deviation and maximum deviation of the nonlinear least squares-fitting algorithm. Rows 5 and 6 show the mean value and standard deviation for the calculated TCPs based on three tests. From the experimental results it has been shown that the results are quite robust for repeatable tests and the TCP calibration error (as shown in columns 8 and 9) is much smaller than the conventional visual inspection approach (standard deviation is around 0.20 mm). However, when compared with the conventional method, the nonlinear optimization algorithm requires correct initial values. To make the laser reading stable, multiple points around the sphere need to be acquired and the laser beam needs to be normal to the sphere surface. This may require robot offline programming.

4.3 TCP Calibration for Cameras

The purpose of this section is to calibrate the position and orientation of the camera relative to the robot mounting flange $Tool_0$ frame for the configuration, the so called eye-in-hand configuration where the camera is held by the robot arm or relative to the robot base frame for the configuration, or the so-called eye-to-hand configuration where the camera is placed at a fixed location. Since the laser scanner is composed of a camera and a laser plane, the calibration of the camera can be considered as the first step in the calibration for the laser scanner, if two steps calibration approaches are used. For both configurations, a calibration target with a known pattern such as a grid pattern is placed in the robot workcell so that the camera can view the calibration pattern through the motion of the robot arm. The position of the calibration target is unknown in general and that makes the calibration task challenging. Move the camera around and take images of the calibration target from different viewpoints. For each viewpoint, calculate the transformation between the camera frame and the calibration target frame based on the image taken by the camera. Then the camera pose relative to the robot $Tool_0$ frame can be calculated by solving linear or nonlinear equations. Fig. 4.9 illustrates the basic calibration setup for both configurations. The classic method for calculating the camera pose is to use the quaternion algebra and linear least squares method (Tsai and Lens, 1989; Shiu and Ahmad, 1989) where the calculation of the rotation matrix and translation vector is separate. The other way is to use the nonlinear optimization method to solve the camera pose (Motai and Kosaka, 2008). An alternative to the above methods is to solve the camera pose and calibration target pose simultaneously by using linear and nonlinear methods (Zhuang et. al., 1994; Dornaik and Horaud, 1998).

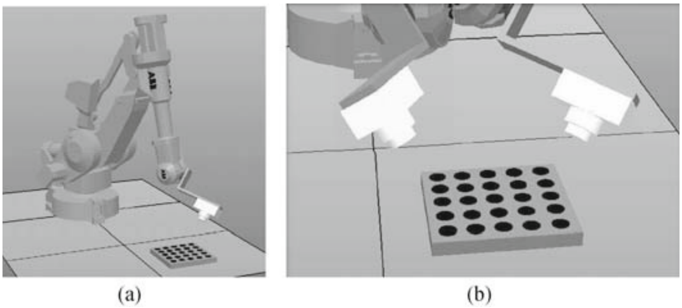


Fig. 4.9. (a) Robot workcell with the camera calibration setup. A calibration target with a precise planar dot pattern is located in the workcell so that the camera that is held by the robot arm can review the pattern to determine the pose of the target with respect to the camera coordinate frame; (b) In the calibration process the camera moves around and takes the image of the calibration target from at least two points of view

4.3.1 Camera Pose Calibration with Linear Equations

In an eye-in-hand setup, as shown in Fig. 4.9, the general calibration model is described by Eq. (4.11). Letting $\boldsymbol{A}_i = \boldsymbol{T}_{0(i+1)}{}^{-1}\boldsymbol{T}_{0i}$, $\boldsymbol{B}_i = \boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}{}^{-1}$, and $\boldsymbol{X} = \boldsymbol{T}_s$, Eq. (4.11) becomes the following equation (Tsai and Lens, 1989)

$$\boldsymbol{A}_i\boldsymbol{X} = \boldsymbol{X}\boldsymbol{B}_i \tag{4.42}$$

Matrices $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{X}$ can be written as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{R}_a & \boldsymbol{t}_a \\ 0 & 1 \end{pmatrix}, \boldsymbol{B} = \begin{pmatrix} \boldsymbol{R}_b & \boldsymbol{t}_b \\ 0 & 1 \end{pmatrix}, \ \boldsymbol{X} = \begin{pmatrix} \boldsymbol{R}_x & \boldsymbol{t}_x \\ 0 & 1 \end{pmatrix} \tag{4.43}$$

Substituting Eq. (4.44) into Eq. (4.11) yields

$$\left.\begin{aligned} \boldsymbol{R}_{ai}\boldsymbol{R}_x &= \boldsymbol{R}_x\boldsymbol{R}_{bi} \\ (\boldsymbol{R}_{ai} - \boldsymbol{I})\boldsymbol{t}_x &= \boldsymbol{R}_x\boldsymbol{t}_{bi} - \boldsymbol{t}_{ai} \end{aligned}\right\} \tag{4.44}$$

Therefore, finding the solution to Eq. (4.11) becomes finding the solution to Eq. (4.45). It is well known that any rotation transformation can be modeled as a rotation by an angle θ around an axis that passes through the origin. The direction of the rotation axis is denoted by $(n_x, n_y, n_z)^{\mathrm{T}}$. The rotation matrix can be written as

$$R = \begin{pmatrix} n_x^2 + (1 - n_x^2)\cos\theta & n_x n_y(1 - \cos\theta) - n_z \sin\theta & n_x n_z(1 - \cos\theta) + n_y \sin\theta \\ n_x n_y(1 - \cos\theta) + n_z \sin\theta & n_y^2 + (1 - n_y^2)\cos\theta & n_y n_z(1 - \cos\theta) - n_x \sin\theta \\ n_x n_z(1 - \cos\theta) + n_y \sin\theta & n_y n_z(1 - \cos\theta) + n_x \sin\theta & n_z^2 + (1 - n_z^2)\cos\theta \end{pmatrix} \tag{4.45}$$

One of the eigenvectors and eigenvalues of $\boldsymbol{R}$ should be the rotation axis and 1, that is

$$\boldsymbol{R}\boldsymbol{p}_r = \boldsymbol{p}_r \tag{4.46}$$

Define $\boldsymbol{p}_r$ as

$$\boldsymbol{p}_r = 2\sin\left(\frac{\theta}{2}\right)(n_x,\ n_y,\ n_z)^{\mathrm{T}} \tag{4.47}$$

Therefore $\boldsymbol{R}$ is expressed by $\boldsymbol{p}_r$ as

$$\boldsymbol{R} = (1-\frac{|\boldsymbol{p}_r|^2}{2})\boldsymbol{I} + \frac{1}{2}(\boldsymbol{p}_r\boldsymbol{p}_r^{\mathrm{T}} + \sqrt{4-|\boldsymbol{p}_r|^2}\,\boldsymbol{\Omega}(\boldsymbol{v})) \tag{4.48}$$

where $\boldsymbol{I}$ is unit matrix and $\boldsymbol{\Omega}(\boldsymbol{v})$ is a skew-symmetric matrix generated by a 3D vector $\boldsymbol{v} = (v_x, v_y, v_z)$ such that

$$\boldsymbol{\Omega}(\boldsymbol{v}) = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix} \tag{4.49}$$

Define

$$\boldsymbol{P}_r = \frac{1}{2\cos(\frac{\theta}{2})}\boldsymbol{p}_r = \frac{1}{\sqrt{4-|\boldsymbol{p}_r|^2}} \tag{4.50}$$

$\boldsymbol{P}_{Rr}$ can be solved through the equations

$$\boldsymbol{\Omega}(\boldsymbol{p}_{Rb,i} + \boldsymbol{p}_{Ra,i})\boldsymbol{p}_{\kappa r} = \boldsymbol{p}_{Rb,i} - \boldsymbol{p}_{Ra,i} \tag{4.51}$$

where $\boldsymbol{P}_{Ra}$ and $\boldsymbol{P}_{Rb}$ are the eigenvectors of the rotation matrices $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$, respectively. Since $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ are known parameters, so $\boldsymbol{P}_{Ra}$ and $\boldsymbol{P}_{Rb}$ can be calculated accordingly. $\boldsymbol{P}_{Rr}$ can be solved through linear equations Eq. (4.52) by using linear least squares algorithms. To get a unique solution to Eq. (4.52), at least two sets of $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ need to be used that requires placing the camera in at least three poses. That is $i \geq 2$. When i = 2, $\boldsymbol{P}_{Rr}$ is solved from the following linear equations

$$\begin{pmatrix} \boldsymbol{\Omega}(\boldsymbol{p}_{Rb,1} + \boldsymbol{p}_{Ra,1}) \\ \boldsymbol{\Omega}(\boldsymbol{p}_{Rb,2} + \boldsymbol{p}_{Ra,2}) \end{pmatrix}\boldsymbol{p}'_{\kappa x} = \begin{pmatrix} \boldsymbol{p}_{Rb,1} - \boldsymbol{p}_{Ra,1} \\ \boldsymbol{p}_{Rb,2} - \boldsymbol{p}_{Ra,2} \end{pmatrix} \tag{4.52}$$

by using linear least squares we have

$$\boldsymbol{p}'_{Rx} = (\boldsymbol{C}^{\mathrm{T}}\boldsymbol{C})^{-1}\boldsymbol{C}^{\mathrm{T}}\boldsymbol{d} \tag{4.53}$$

where

$$\boldsymbol{C} = \begin{pmatrix} \boldsymbol{\Omega}(\boldsymbol{p}_{Rb,1} + \boldsymbol{p}_{Ra,1}) \\ \boldsymbol{\Omega}(\boldsymbol{p}_{Rb,2} + \boldsymbol{p}_{Ra,2}) \end{pmatrix}; \quad \boldsymbol{d} = \begin{pmatrix} \boldsymbol{p}_{Rb,1} - \boldsymbol{p}_{Ra,1} \\ \boldsymbol{p}_{Rb,2} - \boldsymbol{p}_{Ra,2} \end{pmatrix} \tag{4.54}$$

Eq. (4.54) is equivalent to the solution that minimizes $\left\|\boldsymbol{C}\boldsymbol{p}'_{Rx} - \boldsymbol{d}\right\|^2$.

By tusing Eq. (4.51), $\boldsymbol{P}_{Rx}$ is obtained from $\boldsymbol{p}_{Rr}$

$$p_{Rx} = \frac{2\boldsymbol{p}_{Rx}}{\sqrt{1+\left|\boldsymbol{p}_{Rx}\right|^2}} \tag{4.55}$$

Therefore $\boldsymbol{R}_x$ can be solved based on Eq. (4.49).

After the rotation matrix of the camera pose is determined, the translation vector $\boldsymbol{t}_x$ can be solved directly from the second part of Eq. (4.45) by using a linear least squares algorithm. $\boldsymbol{t}_x$ can be written in the closed form as

$$\boldsymbol{t}_x = (\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E})^{-1}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{f} \tag{4.56}$$

where

$$\boldsymbol{E} = \begin{pmatrix} \boldsymbol{R}_{a1} - \boldsymbol{I} \\ \boldsymbol{R}_{a2} - \boldsymbol{I} \end{pmatrix}, \quad \boldsymbol{f} = \begin{pmatrix} \boldsymbol{R}_x\boldsymbol{t}_{b1} - \boldsymbol{t}_{a1} \\ \boldsymbol{R}_x\boldsymbol{t}_{b2} - \boldsymbol{t}_{a2} \end{pmatrix} \tag{4.57}$$

That is equivalent to the solution that minimizes $\left\|\boldsymbol{E}\boldsymbol{t}_x - \boldsymbol{f}\right\|^2$.

4.3.2 Camera Pose Calibration with Nonlinear Optimizations

Eq. (4.45) can be solved with a nonlinear optimization algorithm (Motai and Kosaka, 2008). Let $\{\alpha, \beta, \gamma\}$ be the yaw-pitch-roll angles associated with the rotation matrix $\boldsymbol{R}$. Then $\boldsymbol{R}$ can be written as

$$\boldsymbol{R}_x = \begin{pmatrix} \cos\gamma\cos\beta & \cos\gamma\sin\beta\sin\alpha - \sin\gamma\cos\alpha & \cos\gamma\sin\beta\cos\alpha + \sin\gamma\sin\alpha \\ \sin\gamma\cos\beta & \sin\gamma\sin\beta\sin\alpha + \cos\gamma\cos\alpha & \sin\gamma\sin\beta\cos\gamma - \cos\gamma\sin\alpha \\ -\sin\beta & \cos\beta\sin\alpha & \cos\beta\cos\alpha \end{pmatrix} \tag{4.58}$$

$\{\alpha, \beta, \gamma\}$ can be solved by using a nonlinear optimization algorithm, the so- called Broyden-Flecheler-Goldfarb-Shanno optimization method (Chong and Zak, 1996) that minimizes the following cost function

$$f(\alpha,\ \beta,\ \gamma) = \sum_i \left\|\boldsymbol{R}_{ai}\boldsymbol{R}_x - \boldsymbol{R}_x\boldsymbol{R}_{bi}\right\|^2 \tag{4.59}$$

where matrices $\boldsymbol{R}_{ai}$ and $\boldsymbol{R}_{bi}$ are known coefficients that are determined by the robot poses and camera positions relative to the calibration target, and $\boldsymbol{R}_x$ is the function of three angles. The initial value for the nonlinear optimization algorithm can be estimated based on the position and orientation of the camera relative to the robot mounting flange. Although the initial value may not be close to the true value, the algorithm still gives stable convergence.

Once $\boldsymbol{R}_x$ is solved it is straightforward to solve the translation vector $\boldsymbol{t}_x$ by using Eq. (4.57).

Instead of solving homogenous transformation equations of the form $\boldsymbol{AX} = \boldsymbol{XB}$, with linear and nonlinear approaches as described in previous sections (Tsai and

Lens, 1989; Shiu and Ahmad, 1989; Motai and Kosaka, 2008), the camera pose relative to the robot mounting flange can be solved together with the calibration target position by solving transformation equations of the form $\boldsymbol{AX} = \boldsymbol{YB}$ proposed by Zhuang et al. (1994). The equation is solved based on quaternion algebra and a linear least squares algorithm. At least four camera poses are required to uniquely solve variables. This form of equation was further discussed and solved to give the closed form solution and by the nonlinear constraint minimization method (Dornaik and Horaud, 1998). Furthermore, the camera pose can even be calibrated together with the camera intrinsic parameters and robot parameters by using a large scale nonlinear optimization procedure (Zhuang et al., 1995).

4.4 TCP Calibration for 3D Laser Scanner

Fig. 1.5 shows a robot scanning system consisting of a robot manipulator and a portable laser 3D scanner. It is necessary to perform the robot TCP calibration procedure to determine the relation between the laser scanner coordinate frame and the robot coordinate frame.

The laser scanner system consists of a CCD or CMOS camera and a laser plane. Normally, the coordinate frame of the laser scanner coincides with the coordinate frame of the camera. There are two approaches for calibrating the position of the laser scanner relative to the robot frame (i.e., the robot base frame for the eye-to-hand configuration or the robot $Tool_0$ frame for the eye-in-hand configuration). One approach is to treat the laser scanner as a coordinate frame. And the other is to calibrate the camera coordinate frame relative to the robot frame first and then find the position of the laser plane relative to the camera coordinate frame. In the following sections we will give two examples to illustrate the procedures.

4.4.1 TCP Calibration with a Sphere

For the eye-to-hand configuration, the laser scanner is placed at a fixed workcell location and a sphere as the calibration target is held by the robot arm. The calibration to find the coordinate frame of the laser scanner relative to the robot based frame is referred to as a fixed TCP calibration. The following description is applied to the fixed TCP case that the laser scanner is fixed at the robot work cell (Li, 2007; 2008a; 2008b). The method and algorithm can be easily extended to the moving TCP case that the scanner is held by the robot arm.

4.4.1.1 Calibration Algorithms

There are two steps for calibrating the rotation and translation components of the transform matrix between the laser scanner coordinate frame and the robot based coordinate frame, respectively.

***Step 1**. Calibration of Rotation Matrix* $\boldsymbol{R}_s$

$\boldsymbol{R}_s$ is the rotation matrix of the transformation between the laser scanner and the robot base frame. The algorithm derivation can start from robot general kinematic Eq. (4.5)

$$\boldsymbol{R}_{0i}\boldsymbol{t}_{gi} + \boldsymbol{t}_{0i} = \boldsymbol{R}_s\boldsymbol{t}_{vi} + \boldsymbol{t}_s \tag{4.60}$$

where ($\boldsymbol{R}_0$, $\boldsymbol{t}_0$) are the rotation and translation components of the robot $Tool_0$ frame; Index i is the number of measurements; ($\boldsymbol{R}_s$, $\boldsymbol{t}_s$) are the rotation and translation components of the laser scanner frame relative to the robot base frame, which are parameters to be calibrated. $\boldsymbol{t}_g$ is the sphere center position in the $Tool_0$ frame and $\boldsymbol{t}_v$ is the sphere center position in the scanner frame.

When the robot end effecter moves with the same orientation ($\boldsymbol{R}_{0i} = \boldsymbol{R}_{0j}$ when $i \neq j$), the positions of the same target point $\boldsymbol{t}_g$ are measured in the scanner frame. With the geometric constraint $\boldsymbol{t}_{gi} = \boldsymbol{t}_{gi}$ ($i \neq j$), we have

$$\boldsymbol{t}_{0j} - \boldsymbol{t}_{0i} = \boldsymbol{R}_s(\boldsymbol{t}_{vi} - \boldsymbol{t}_{vj}) \tag{4.61}$$

Therefore, with the measurements of the sphere center being more than 3 non-collinear robot positions, the orientation of the scanner frame ($\boldsymbol{R}_s$) can be calculated by minimizing the following object functions

$$G(\boldsymbol{R}_s) = \sum_{i \neq j} \left\| \boldsymbol{R}_s(\boldsymbol{t}_{vi} - \boldsymbol{t}_{vj}) - \boldsymbol{t}_{0j} + \boldsymbol{t}_{0i} \right\|^2 \tag{4.62}$$

If two measurements are used, $\boldsymbol{R}_s$ can be solved as

$$\boldsymbol{R}_s = \frac{(\boldsymbol{t}_{02} - \boldsymbol{t}_{01})}{(\boldsymbol{t}_{v1} - \boldsymbol{t}_{v2})} \tag{4.63}$$

The center position of the sphere ($\boldsymbol{t}_v$) in the laser scanner frame is obtained by the following procedure. The circle parameters including center $(x_c, y_c, z_c)^{\mathrm{T}}$, radius ($r$) and the normal of the circle plane ($\boldsymbol{n}$) are obtained by circle fitting of the reconstructed circle points. And the sphere center $\boldsymbol{t}_v=(t_x, t_y, t_z)^{\mathrm{T}}$ is calculated based on the geometrical relationship as shown in Fig. 4.10

$$\left.\begin{aligned} \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} &= \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \boldsymbol{n}d \\ d &= (R^2 - r^2)^{\frac{1}{2}} \end{aligned}\right\} \tag{4.64}$$

where R is the known sphere radius.

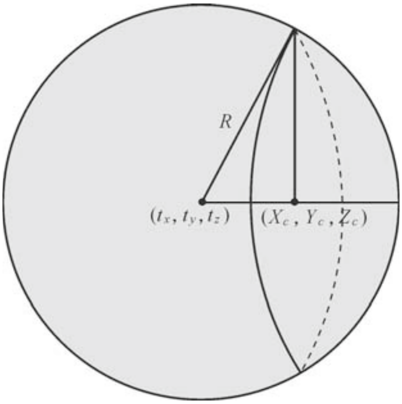


Fig. 4.10. Calculation of the sphere center from the reconstructed points

Fig. 4.11 shows the setup to calibrate the laser linear scanner TCP with a sphere of known radius.

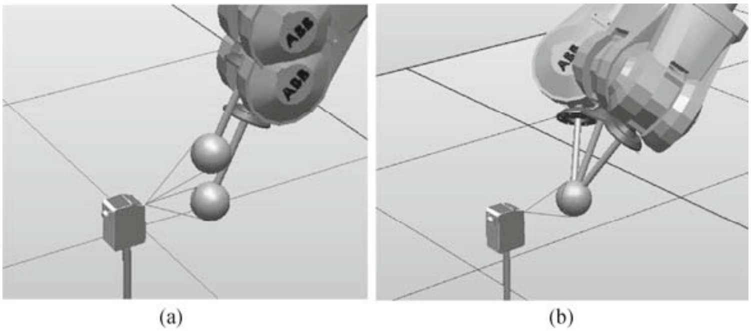


Fig. 4.11. Calibration of the laser scanner. (a) The sphere is scanned at least two times with the robot orientation unchanged to get the direction of the fitted sphere center in order to calibrate the orientation of the scanner; (b) The sphere is scanned at least three times to get the fitted sphere centers. Alignment of sphere centers is used to calibrate the position of the scanner

Step 2*. Calibration of Translation Vector* $\boldsymbol{t}_s$

In the position calibration, the sphere is scanned at more than three different robot poses, and the geometric constraint of the sphere center $\boldsymbol{t}_{gi} = \boldsymbol{t}_{gi}$ $(i \neq j)$ is used again to solve the position vector that minimizes the object functions of

$$G(\boldsymbol{t}_s) = \sum_{i \neq j} \left\| \boldsymbol{t}_{gi} - \boldsymbol{t}_{gj} \right\|^2 \tag{4.65}$$

where $\boldsymbol{t}_{gi}$ is the fitted sphere center obtained with the i-th robot pose. And the fitted sphere center is expressed as

$$\boldsymbol{t}_{gi} = \boldsymbol{X}_{ci} + \boldsymbol{R}_{0i}^{-1} \cdot \boldsymbol{t}_s \tag{4.66}$$

where X_{ci} is the fitted sphere center shifted by $R_{0i}^{-1} \cdot t_s$ for the i-th scanning. It is obtained from a set of reconstructed points on the sphere surface

$$X_{ik} = R_{0i}^{-1}(R_s t_{vik} - t_{0ik}) \tag{4.67}$$

where t_{vik} and t_{0ik} are the reconstructed points on the k-th line for the ith scanning and corresponding $Tool_0$ positions. To solve t_s, combining Eqs.(4.66) and (4.67) yields the object function of

$$G(t_s) = \sum_{i \neq j} \left\| X_{ci} - X_{cj} + (R_{0i}^{-1} - R_{0j}^{-1}) t_s \right\|^2 \tag{4.68}$$

Then t_s can be solved with a least squares algorithm.

4.4.1.2 Experimental Verification

In the experimental setup, an industrial robot (ABB IRB4400) is used in the work cell, and a pre-calibrated portable laser 3D scanner is fixed at a position which the robot can easily reach. A sphere with diameter 15.38 mm is mounted on the robot end effector (Tang and Gan, 2007). The rotation matrix R_s of the laser scanner is determined by the following procedure:

(1) Jog the robot to the position and make the laser beam project onto the sphere to obtain reconstructed points of the circle that is the intersection of the laser plane with the sphere. Do circle fitting to get the center of the circle, and derive the center of the sphere. All the measurement is based on the laser scanner coordinate frame.

(2) Translate the robot along its y axis of $Tool_0$, while keeping the robot pose unchanged, to another position and make the laser beam project onto the sphere to get the sphere center position.

(3) Repeat step (2) twice and obtain 4 measurements of the sphere center.

By putting these data into Eq.(4.64), R_s can be solved and given as follows

$$R_s = \begin{pmatrix} 0.0456 & -0.9978 & 0.0471 \\ 0.0415 & 0.0490 & 0.9979 \\ -0.9981 & -0.0436 & 0.0436 \end{pmatrix}.$$

The translation vector t_s is determined by the following procedure:

(1) Jog the robot to the position and make the laser beam project onto the sphere. With the robot orientation unchanged move the robot along its Y axis of $Tool_0$ to scan the sphere to obtain a reconstructed points cloud of the sphere. Use Eq. (4.68) to reconstruct the sphere points. Do sphere fitting to get the center position of the reconstructed sphere X_c.

(2) Change the orientation of the robot, and repeat step (1) to get another center position of the reconstructed sphere.

(3) Repeat steps (1) and (2) at least three times to obtain a few sets of X_c and R_0. Entering these data into Eq. (4.69), t_s is estimated to be t_s = (356.45, 1543.27, 862.56).

With the laser scanner position obtained, the robot scanning system is employed to scan a sphere to check the accuracy of this calibration approach. The scanning result is shown in Fig.4.12. It can be seen that the multiple scans are merged seamlessly into the shape of the sphere, indicating that the calibration parameters of the laser scanner are accurate. The calibration accuracy of the position of the laser scanner will affect the merging effect, i.e., the multiple scans would not be merged without a gap if the parameter is not accurate. It does not affect the shape for a single scan. The shape or scale of the sphere for a single scan is affected by the accuracy of the pose of the laser scanner. With these scanning data available, a spherical fitting is performed and 0.1 mm fitting error is obtained. This fitting error can be considered as the measurement accuracy, composed of the measurement accuracy of the portable laser 3D scanner, the accuracy of the robot and the accuracy of robot TCP calibration.

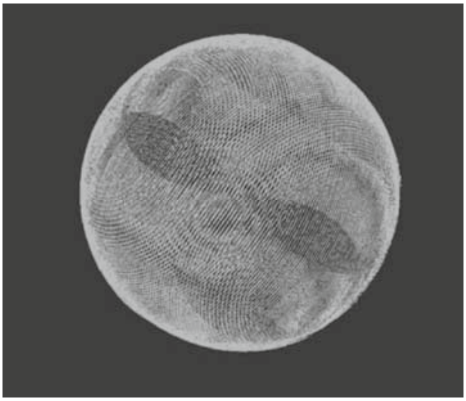


Fig. 4.12. Reconstructed sphere with multiple scans

4.4.2 TCP Calibration with a Plane

The other approach for calibrating the TCP position of the laser scanner is to calibrate the TCP position of the camera first and to determine the position of the laser plane relative to the camera coordinate frame. The TCP calibration of the camera has been discussed in the previous sections. Here we focus on the calibration of the laser plane in the camera frame (Xu et al., 2005). This can be done with a plane as the calibration object. The robot movement for each scan during the calibration operation is constrained to keep the camera origin unchanged, in order to simplify and decouple the calibration parameters as illustrated in Fig. 4.13.

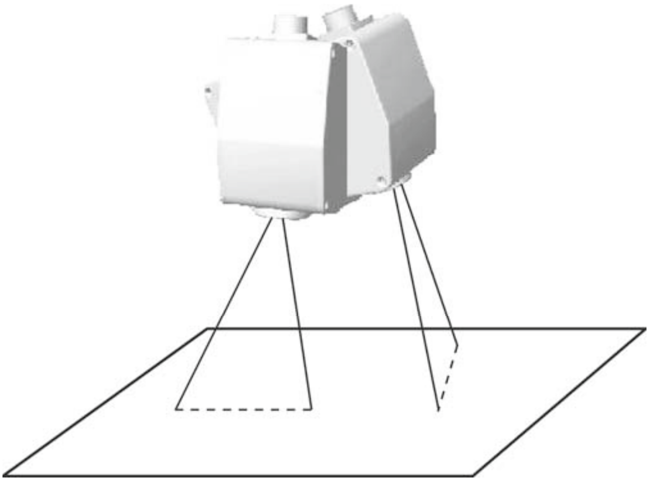


Fig. 4.13. A plane is placed at the workcell as the calibration target. The robot rotates with the constraint that the origin of the camera is kept unchanged to decouple calibration parameters

Assume that the equation of the laser plane in the camera frame is given by

$$ax_c + by_c + cz_c + 1 = 0 \tag{4.69}$$

where (a, b, c) are the plane parameters that need to be determined in the calibration procedure.

A point P on the laser plane can be expressed by Eq. (4.71) in the camera coordinate system

$$\left.\begin{aligned} x_c &= x_{cf}t \\ y_c &= y_{cf}t \\ z_c &= t \end{aligned}\right\} \tag{4.70}$$

where (x_{cf}, y_{cf}, 1) is the point on the normalized focal plane in the camera coordinate system. Combining Eqs. (4.70) and (4.71) results in the coordinate of the point P in the camera frame and the robot base frame

$$\left.\begin{aligned} x_c &= -x_{cf} / (ax_{cf} + by_{cf} + c) \\ y_c &= -y_{cf} / (ax_{cf} + by_{cf} + c) \\ z_c &= -1 / (ax_{cf} + by_{cf} + c) \end{aligned}\right\} \tag{4.71}$$

$$\boldsymbol{x}_0 = \boldsymbol{T}_0 \boldsymbol{T}_c \boldsymbol{x}_c \tag{4.72}$$

where $\boldsymbol{T}_0$ is the robot $Tool_0$ frame; $\boldsymbol{T}_c$ is the camera frame pose relative to the robot $Tool_0$ frame that is assumed to be calibrated previously and $\boldsymbol{x}_c$=(x_c, y_c, z_c) is the coordinate of the point in the camera frame. Since both transformations $\boldsymbol{T}_0$ and $\boldsymbol{T}_c$ are given, we can combine them by letting

$$\boldsymbol{T}_0\boldsymbol{T}_c = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{n} & \boldsymbol{o} & \boldsymbol{a} & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.73}$$

Substituting $\boldsymbol{T}_0\boldsymbol{T}_c$ in Eq. (4.73) with Eq. (4.74) results in

$$\left.\begin{aligned} x_b &= n_x x_c + o_x y_c + a_x z_c + p_x \\ y_b &= n_y x_c + o_y y_c + a_y z_c + p_y \\ x_b &= n_z x_c + o_z y_c + a_z z_c + p_z \end{aligned}\right\} \tag{4.74}$$

Assume that the laser beam is shooting on a plane with the equation

$$Ax_b + By_b + Cz_b + 1 = 0 \tag{4.75}$$

where (A, B, C) are the parameters of the plane in the robot base frame that are unknown. Combining Eqs. (4.75) and (4.76) we have

$$\begin{aligned} &A(n_x x_c + o_x y_c + a_x z_c) + B(n_y x_c + o_y y_c + a_y z_c) \\ &+C(n_z x_c + o_z y_c + a_z z_c) + Ap_x + Bp_y + Cp_z + 1 = 0 \end{aligned} \tag{4.76}$$

Let $D = Ap_x+Bp_y+Cp_z+1$. If point (p_x, p_y, p_z) is not on the plane, then $D\neq 0$. This can be implemented by keeping the distance of the camera origin from the calibration plane. By substituting (x_c, y_c, z_c) in Eq. (4.77) with Eq. (4.72) and dividing Eq. (4.77) by D we obtain

$$\begin{aligned} &A_1(n_x x_{cf} + o_x y_{cf} + a_x) + B_1(n_y x_{cf} + o_y y_{cf} + a_y) \\ &+C_1(n_z x_{cf} + o_z y_{cf} + a_z) - (ax_{cf} + by_{cf} + c) = 0 \end{aligned} \tag{4.77}$$

where $A_1 = A/D$, $B_1 = B/D$, $C_1 = C/D$. Since the laser plane cannot be perpendicular to the optical axis of the camera, in order to be able to observe the laser line by the camera, $c \neq 0$ in Eq. (4.70). Dividing Eq. (4.78) by c gives

$$\begin{aligned} &A_2(n_x x_{cf} + o_x y_{cf} + a_x) + B_2(n_y x_{cf} + o_y y_{cf} + a_y) \\ &+C_2(n_z x_{cf} + o_z y_{cf} + a_z) - (a_1 x_{cf} + b_1 y_{cf}) = 1 \end{aligned} \tag{4.78}$$

where $A_2=A_1/c$, $B_2=B_1/c$, $C_2=C_1/c$, $a_1=a/c$, $b_1=b/c$. They are variables to be determined. Vectors {$\boldsymbol{n}$, $\boldsymbol{0}$, $\boldsymbol{a}$} are determined by the robot $Tool_0$ pose. (x_{cf}, y_{cf}) is the point on the normalized focal plane in the camera coordinate system. That is the intersection between the plane $z = 1$ and the ray connecting the camera coordinate origin and the image point on the laser line. When the laser beam is projected on the calibration plane its image is a straight line. Pick up two points on the straight line to create two linear equations from Eq. (4.79). However, equations from more than two points on the line are not independent. In order to form at least 5 linear independent equations to solve variables {A_2, B_2, C_2, a_1, b_1}, change the camera pose to get a different image line while keeping the camera origin position (p_x, p_y, p_z) unchanged. Constant (p_x, p_y, p_z) makes D and therefore

$\{A_2, B_2, C_2, a_1, b_1\}$ unchanged for different camera poses. This can be implemented by changing the pose of the robot $Tool_0$ as

$$\boldsymbol{T}_{0(i+1)} = \boldsymbol{T}_{0(i)} \boldsymbol{T}_c \boldsymbol{R} \boldsymbol{T}_c^{-1} \tag{4.79}$$

where $\boldsymbol{T}_{0(i+1)}$ is the $(i+1)$-th pose of the robot Tool0 and $\boldsymbol{T}_{0(i)}$ is the i-th pose of the robot $Tool_0$. $\boldsymbol{R}$ is the rotation matrix of the camera relative to the robot base frame. Multiple camera poses will result in over-determined linear equations. Variables $\{A_2, B_2, C_2, a_1, b_1\}$ can be solved from the following linear equations by using least squares algorithms

$$\boldsymbol{EX} = \boldsymbol{F} \tag{4.80}$$

where $\boldsymbol{X} = \{A_2, B_2, C_2, a_1, b_1\}^{\mathrm{T}}$, $\boldsymbol{E}$ is the matrix with dimension $n \times 5$ formed by the coefficients and $\boldsymbol{F}$ is the unity matrix with dimension $n \times 1$. $\boldsymbol{X}$ can be solved as

$$\boldsymbol{X} = \boldsymbol{(E^{\mathrm{T}}E)^{-1}E^{\mathrm{T}}F} \tag{4.81}$$

The distance between two points on the laser line is given by

$$d = \sqrt{(x_{k1} - x_{k2})^2 + (y_{k1} - y_{k2})^2 + (z_{k1} - z_{k2})^2} = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{4.82}$$

where $\{d_x, d_y, d_x\}$ are components of d.

Inserting Eqs. (4.72) and (4.75) into Eq. (4.83) we have

$$\begin{aligned} d_x &= n_x(x_{c1} - x_{c2}) + o_x(y_{c1} - y_{c2}) + a_x(z_{c1} - z_{c2}) \\ &= \frac{1}{c}\left[n_x \left(\frac{x_{cf2}}{a_1 x_{cf2} + b_1 y_{cf2} + 1} - \frac{x_{cf1}}{a_1 x_{cf1} + b_1 y_{cf1} + 1} \right) \right. \\ &\quad + o_x \left(\frac{y_{cf2}}{a_1 x_{cf2} + b_1 y_{cf2} + 1} - \frac{y_{cf1}}{a_1 x_{cf1} + b_1 y_{cf1} + 1} \right) \\ &\quad \left. + a_x \left(\frac{1}{a_1 x_{cf2} + b_1 y_{cf2} + 1} - \frac{1}{a_1 x_{cf1} + b_1 y_{cf1} + 1} \right) \right] = \frac{1}{c} d_{x1} \end{aligned} \tag{4.83}$$

Similarly, we have $d_y = d_{y1}/c$ and $d_z = d_{z1}/c$. Therefore

$$d = \frac{1}{c}\sqrt{d_{x1}^2 + d_{y1}^2 + d_{z1}^2} = \frac{1}{c} d_1 \tag{4.84}$$

Therefore, one of the laser plane parameters is obtained as $c = d_1/d$, where d_1 is the calculated distance between two points on the laser line based on the values $\{a_1, b_1\}$. d is the physical distance of the same points measured by the ruler. And a and b can be solved as $a = a_1 c$, $b = b_1 c$.

4.4.3 TCP Calibration with a Structured Pattern

If a structured pattern with at least four topographically defined features is used as the calibration target, the TCP calibration of a laser scanner can be done mathematically in a similar way to that for a camera (described in Section 4.3). Reviewing the calibration model for array type sensors represented by Eq. (4.11), we have $\boldsymbol{T}_{0(i+1)}^{-1}\boldsymbol{T}_{0i}\boldsymbol{T}_{s} = \boldsymbol{T}_{s}\boldsymbol{T}_{v(i+1)}\boldsymbol{T}_{vi}^{-1}$ where $\boldsymbol{T}_{0i}$ and $\boldsymbol{T}_{0(i+1)}$ are the $Tool_0$ pose relative to the robot base frame. Those are known transforms and are given by the robot controller. $\boldsymbol{T}_{vi}$ is the corresponding coordinate frame of the calibration target with respect to the tool coordinate frame, and its value is given by the sensor. As long as $\boldsymbol{T}_{vi}$ can be determined, the equations can be solved by using the similar method to that for camera TCP calibration. Both methods need to solve the equation in the form $\boldsymbol{AX} = \boldsymbol{XB}$ for $\boldsymbol{T}_s$ (Huissoon, 2000; Greer and Kim, 2000). In this section we focus on a discussion of how to find $\boldsymbol{T}_{vi}$.

As shown in Fig. 4.14, a calibration target with three topographic edges and a virtual edge is used. Those edges are not in parallel. An edge can be represented by a line equation in the calibration target coordinate frame

$$\boldsymbol{x}_i = \boldsymbol{x}_{0i} + \boldsymbol{n}_i \mathrm{k}_i \tag{4.85}$$

where $\boldsymbol{x}_{0i}$ is a point on the line, $\boldsymbol{n}_i$ is a directional vector of the line and k_i is a scalar indicating the distance between the points $\boldsymbol{x}_{0i}$ and $\boldsymbol{x}_i$. $\boldsymbol{x}_{0i}$ and $\boldsymbol{n}_i$ are the known parameters. The transform of a point in the laser scanner frame to the calibration target frame is given by

$$\boldsymbol{R}\ \boldsymbol{x}_{vi} + \boldsymbol{t} = \boldsymbol{x}_i \tag{4.86}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation matrix and translation vector of the transformation $\boldsymbol{T}_g^v$ from the laser scanner frame to the calibration target frame; $\boldsymbol{t}$ is the coordinate of the point in the laser scanner frame. Insertion of Eq. (4.86) into Eq. (4.87) results in

$$\boldsymbol{Rx}_{vi} + \boldsymbol{t} = \boldsymbol{x}_{0i} + \boldsymbol{n}_i \mathrm{k}_i \tag{4.87}$$

In Eq. (4.88), $\boldsymbol{x}_{vi}$, $\boldsymbol{x}_{0i}$ and $\boldsymbol{n}_i$ are known parameters and $\boldsymbol{R}$, $\boldsymbol{t}$ and k_i are unknown variables. Considering $\boldsymbol{R}$ includes three independent variables $\{\alpha, \beta, \gamma\}$ and $\boldsymbol{t}$ includes three components $\{t_x, t_y, t_z\}$, when i=1 there are 3+3+1=7 independent variables to be solved with only three scalar equations. When i=4 there are 10 independent variables $\{\alpha, \beta, \lambda, t_x, t_y, t_z, k_1, k_2, k_3, k_4\}$ but there are 12 scalar equations. These are overdetermined equations. They can be solved with a nonlinear optimization algorithm by minimizing the following objective function with respect to $\{\alpha, \beta, \lambda, t_x, t_y, t_z, k_1, k_2, k_3, k_4\}$

$$f(\alpha,\ \beta,\ \gamma,\ t_x,\ t_y,\ t_z,\ k_1,\ k_2,\ k_3,\ k_4) = \sum_{i=1}^{4} \left\| \boldsymbol{Rx}_{vi} + \boldsymbol{t} - \boldsymbol{x}_{0i} + \boldsymbol{n}_i k_i \right\|^2 \tag{4.88}$$

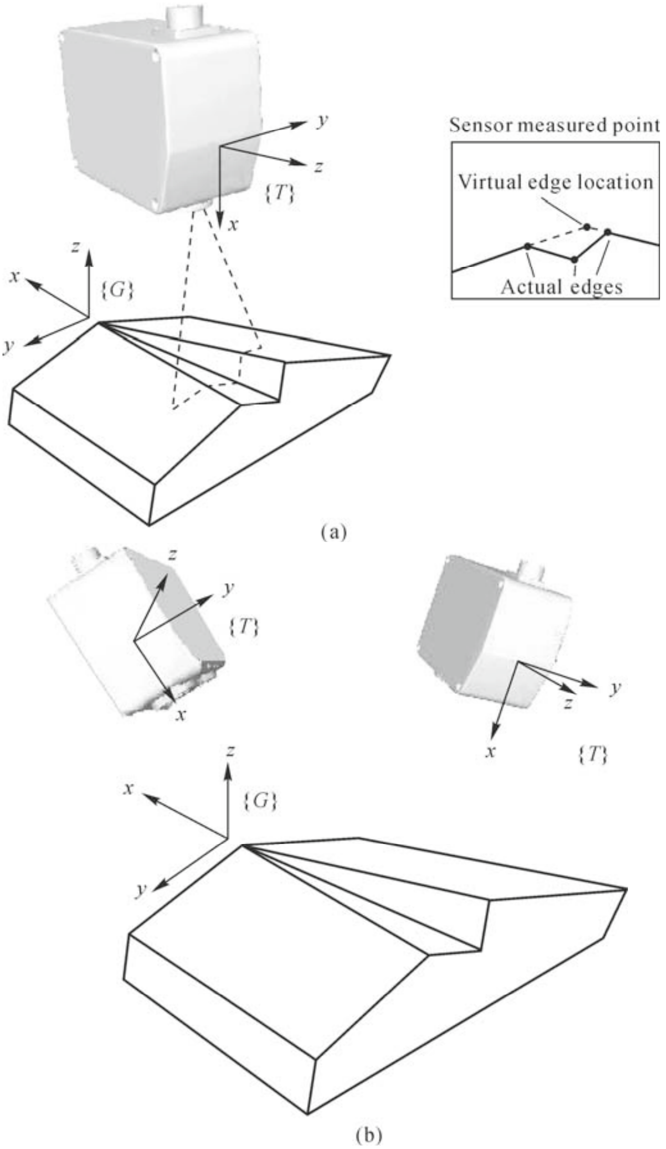


Fig. 4.14. Calibration setup for a laser scanner. (a) A calibration target with three topographic unparallel edges and a virtual edge is located in the robot workcell so that the laser scanner held by the robot arm can view and measure the pattern on the target. A single measurement of the laser scanner can obtain the pose of the calibration fixture relative to the laser scanner frame; (b) Two or more measurements with different orientation are used to calibrate the TCP of the laser scanner

Please refer to Appendix A.3 for the discussion of a nonlinear optimization algorithm. Assume that

$$\boldsymbol{R}=\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix},\ \boldsymbol{t}=(t_x,\ t_y,\ t_z)^{\mathrm{T}},\ \boldsymbol{x}_{vi}=(x_{vi},\ y_{vi},\ 0)^{\mathrm{T}},$$

$$\mathbf{x}_{0i}=(x_{0i},\ y_{0i},\ z_{0i})^{\mathrm{T}},\ \boldsymbol{n}_i=(n_{xi},\ n_{yi},\ n_{zi})^{\mathrm{T}}$$

The third component of $\boldsymbol{x}_{vi}$ is set to zero since we assume that the laser plane is on the *XY* plane in the laser scanner coordinate system. This assumption will eliminate 3 variables to be solved in Eq. (4.88). They can be solved by using the unit vector relation. Therefore, Eq. (4.88) can be written as

$$\begin{aligned} r_{11}x_{vi}+r_{12}y_{vi}+t_x &= x_{0i}+n_{xi}k_i \\ r_{21}x_{vi}+r_{22}y_{vi}+t_y &= y_{0i}+n_{yi}k_i \\ r_{31}x_{vi}+r_{32}y_{vi}+t_z &= z_{0i}+n_{zi}k_i \end{aligned} \tag{4.89}$$

When $i=4$, Eq. (4.90) becomes a linear equation as

$$\boldsymbol{Ax}=\boldsymbol{b} \tag{4.90}$$

where $\boldsymbol{x}$ is a 13×1 vector. $\boldsymbol{A}$ is a known measured 12×13 matrix of the orientation vector components for each edge and the sensed edge location, and $\boldsymbol{b}$ is a known 12×1 vector of parameters defining the points on the edges.

$$\boldsymbol{x}=(r_{11},\ r_{12},\ t_x,\ r_{21},\ r_{22},\ t_y,\ r_{31},\ r_{32},\ t_z,\ k_1,\ k_2,\ k_3,\ k_4)^{\mathrm{T}}$$

$$\boldsymbol{A}=\begin{pmatrix} \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{A}_4 \end{pmatrix}$$

$$\boldsymbol{A}_1=\begin{pmatrix} x_{v1} & y_{v1} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -n_{x1} & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{v1} & y_{v1} & 1 & 0 & 0 & 0 & -n_{y1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{v1} & y_{v1} & 1 & -n_{z1} & 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{A}_4=\begin{pmatrix} x_{v4} & y_{v4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -n_{x4} \\ 0 & 0 & 0 & x_{v4} & y_{v4} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -n_{y4} \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{v4} & y_{v4} & 1 & 0 & 0 & 0 & -n_{z4} \end{pmatrix}$$

$$\boldsymbol{b}=(x_{01}\quad y_{01}\quad z_{01}\quad x_{02}\quad y_{02}\quad z_{02}\quad x_{03}\quad y_{03}\quad z_{03}\quad x_{04}\quad y_{04}\quad z_{04})^{\mathrm{T}}$$

Eq. (4.91) is an underdetermined equation with 13 variables and 12 equations. By adding a 13th row of 0 to the $\boldsymbol{A}$ matrix, and a 13th 0 element to the vector $\boldsymbol{b}$, the problem may be solved using the singular value decomposition (SVD) technique, which will result in a solution in the form

$$\boldsymbol{x}_l=\boldsymbol{e}_l+f\boldsymbol{v}_l,\ 1\le l\le 13 \tag{4.91}$$

where each x_l is one of variables to be solved. f is a constant and $\boldsymbol{v}$ is a column of 13 values (corresponding to the zero row appended to $\boldsymbol{A}$) returned by the SVD algorithm, together with the minimum length solution vector ***e***. In order to determine the constant *f*, unit vector equations and the orthogonal requirement are used. That is

$$\left.\begin{array}{l} r_{11}^2 + r_{21}^2 + r_{31}^2 = 1 \\ r_{12}^2 + r_{22}^2 + r_{32}^2 = 1 \\ r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} = 0 \end{array}\right\} \tag{4.92}$$

All the variables included in ***R*** and ***t*** can be obtained from Eq. (4.88), yielding a solution for $\boldsymbol{T}_g^{\,v}$. Therefore $\boldsymbol{T}_{vi}$ can be obtained that is the inverse of $\boldsymbol{T}_g^{\,v}$.

As long as $\boldsymbol{T}_{vi}$ is obtained from a single measurement, the TCP of the laser scanner can be obtained by solving Eq. (4.11) with the algorithm described in Section 4.3 for camera TCP calibration.

4.5 TCP Calibration with Direct Measurement

This is a straightforward and reliable method to calibrate machine tools with direct measurement if a measurement tool has been calibrated and its coordinate system has been associated with the robot coordinate system. In Sections 4.1 and 4.2 we discussed the TCP calibration methods for point type sensors. After those sensors are determined they can be used to measure and calibrate the positions of work objects and tools used in the workcell. As an example, without losing generality, we will demonstrate the calibration of a set of tools like milling tools, cutting tools, that are held and rotated by a spindle (Tang et al., 2003). Since all the tools are symmetric and centric we can use a standard cylinder to model. As shown in Fig. 4.15, a standard cylinder is mounted on the spindle. It is used to calibrate the position and orientation of a reference position on the cylinder to determine the position and orientation of the spindle. With the spindle orientation and position calibrated, the tools only need to be calibrated with respect to their length from the spindle face. The measurement probe may include a CMM probe, touch probe, LVDT, or point laser sensor. Its position $\boldsymbol{t}_g$ is previously calibrated using a TCP calibration approach as discussed in Section 4.1.3. Once the position of the probe is known it can be used as the measurement tool in conjunction with the robot pose. The coordinate of the measured point on the $Tool_0$ frame is determined by the robot pose and the reading of the probe. If a 1D displacement sensor, like a laser displacement sensor or LCDT is used as the probe, the coordinate of the measured point $\boldsymbol{x}_t$ is given by

$$\boldsymbol{x}_t = \boldsymbol{R}_0^{-1}(\boldsymbol{n}L + \boldsymbol{t}_g - \boldsymbol{t}_0) \tag{4.93}$$

where $\{\boldsymbol{R}_0, \boldsymbol{t}_0\}$ are the $Tool_0$ rotation and translation components relative to the robot base frame, L is the sensor reading indicating the distance between the measured point and the reference point, ***n*** is the orientation of the laser beam direction or probe move direction relative to the robot base frame. If a contact sensor like a touch trigger is used, the measured point $\boldsymbol{x}_t$ is given by $\boldsymbol{x}_t = \boldsymbol{R}_0^{-1}(\boldsymbol{t}_g - \boldsymbol{t}_0)$.

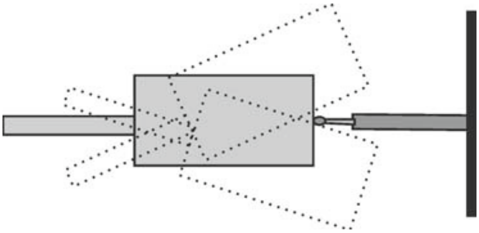


Fig. 4.15. Calibration of machine tool. A probe is fixed in the workcell and its position and orientation were calibrated previously. The cylinder is mounted on the spindle. Position of the cylinder is measured by programming the robot such that the cylinder surface is measured or touched by the probe

4.5.1 Calibration of Spindle

At the first position of the spindle (i.e., at 0 degree of the spindle encoder reading) the normal of the top surface of the cylinder is measured with the probe. The measurement is conducted by programming the robot to touch the probe with the cylinder top surface. The measurement of 5 or more points on the surface will determine the plane position (n_{x1}, n_{y1}, n_{x1}, d_1) relative to the $Tool_0$ coordinate frame. Rotate the spindle to 90°, 180°, 270° positions, respectively, to get the cylinder top surface positions (n_{x1}, n_{y1}, n_{x1}, d_1) (i = 1, 2, 3) corresponding to various spindle angles. For each spindle position the normal of the cylinder top surface will be slightly different each time, due to a mounting error or imperfection of the cylinder. However, the true spindle orientation will be determined by synthesis of all normal vectors

$$\left.\begin{aligned} n_x &= (n_{x0} + n_{x1} + n_{x2} + n_{x3})/4 \\ n_y &= (n_{y0} + n_{y1} + n_{y2} + n_{y3})/4 \\ n_z &= (n_{z0} + n_{z1} + n_{z2} + n_{z3})/4 \end{aligned}\right\} \qquad (4.94)$$

For each spindle position the axis (nl_x, nl_y, nl_z) of the cylinder will be measured with the probe. This axis may not be the same as the normal of the cylinder top surface due to imperfection of the cylinder. The axis of the cylinder is determined by measuring a number points on the side of the cylinder and a nonlinear least squares algorithm is used to fit the cylinder model.

A geometric cylinder can be described by its axis and radius. Assume the axis of the cylinder is represented by a straight line that has orientation (nl_x, nl_y, nl_z) and passes through a spatial point (x_0, y_0, z_0).

The cylinder equation is given by

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2$$

$$-\frac{[(x_i - x_0)nl_x + (y_i - y_0)nl_y + (z_i - z_0)nl_z]^2}{nl_x^2 + nl_y^2 + nl_z^2} = R^2 \tag{4.95}$$

where R is the radius of the cylinder and (x_i, y_i, z_i) are the measured positions on the cylinder side surface. The calibration of the cylinder position is to find the cylinder parameters $(nl_x, nl_y, nl_x, x_0, y_0, z_0)$ from the measurement points (x_i, y_i, z_i) based on Eq. (4.96). To solve this equation at least 6 measurement points are required. The equation can be solved for more measurement points by using a nonlinear least squares algorithm. During the calibration, two sections are measured along the cylinder top and bottom position.

The center position (T_{xi}, T_{yi}, T_{zi}) $(i = 0, 1, 2, 3)$ for various spindle positions of the top surface of the cylinder is calculated. This is the intersection between the measured top surface and cylinder axis. The average center position for all spindle positions will be used as the spindle position.

$$\left.\begin{aligned} T_x &= (T_{x0} + T_{x1} + T_{x2} + T_{x3})/4 \\ T_y &= (T_{x0} + T_{x1} + T_{x2} + T_{x3})/4 \\ T_z &= (T_{z0} + T_{z1} + T_{z2} + T_{z3})/4 \end{aligned}\right\} \tag{4.96}$$

4.5.2 Calibration of Tools with Different Length

Once the cylinder center and orientation is calibrated, it is possible to calibrate all of the other tools. We assume that all the tool orientations are the same as the spindle rotating axis. In order to get the TCP of all tools, the length of the tool needs to be measured and compared with the length of the cylinder. If the cylinder length is *Cyl_len* and the tool length is *Tool_len*, then the TCP of the tool will be calculated by

$$\left.\begin{aligned} Tl_x &= T_x + (Tool\_Len - Cyl\_len)n_x \\ Tl_y &= T_y + (Tool\_Len - Cyl\_len)n_y \\ Tl_z &= T_z + (Tool\_Len - Cyl\_len)n_z \end{aligned}\right\} \tag{4.97}$$

where (T_x, T_y, T_z) is the TCP of the cylinder and (n_x, n_y, n_z) is the spindle orientation.

Advantageously, the calibration technique described above does not rely on accounting for mechanical tolerances and can determine the rotation axis regardless of mounting error.

4.6 Relative Robot Workcell Calibration

A variety of attempts to develop a better robot calibration system have been made to improve robot accuracy. Current techniques, however, are typically tedious, time consuming and expensive. This is because most of the prior calibration methodologies are based on absolute calibration.

"Absolute calibration" refers to the method by which an external coordinate measurement system is utilized to measure the absolute position. Since the external system measures the coordinates of a point in the workspace, the absolute method can validate any path accuracy. However, absolute position measurement has many drawbacks, including the fact that it is time consuming, expensive and sometimes fails to meet accuracy requirements. One example is to use an optical coordinate measurement system (OCMS) to calibrate the robotic workcell, which is a very expensive and time-consuming way of calibrating the robot.

In contrast to absolute calibration, some development has been made in the area of "relative calibration" (Gan, 2004; Sun, 2009). Relative calibration is a method in which a standard reference target is used as the precision reference for the correction of robot kinematic error. This "standard reference" provides high-precision relative geometric quantities such as length, circularity and linearity. A standard reference could simply be a bar, a cube, a cylinder or a ball. During the calibration, the robot is driven to make the TCP follow the geometry of the selected standard reference. This standard reference therefore provides a constraint on the TCP process. Due to kinematic error, this constraint would be violated if the nominal kinematic model were used to calculate the Cartesian coordinates from the same joint angles. Minimization of the constraint violation (constraint error) will give the values of error parameters.

4.6.1 Robot Workcell Calibration

In a fixed TCP-based robot workcell, the forward kinematic chain includes the robot (robot based coordinate), the gripper (work-object coordinate) and the workpiece (object coordinate); the backward kinematic chain includes the tooling system (tool coordinate). In an ideal case, the errors in real or virtual contact points between the tooling and the object are zeros along the working path.

All of the errors from the two kinematic chains can be divided into two parts: forward chain error and backward chain error. Forward chain error includes robot error, gripper-setup error and object-installation error. Backward chain error includes tool-table error and tooling fixture error.

The role of calibration is to eliminate or correct all of these errors in order to create highly accurate paths for robot operation.

In a conventional absolute calibration environment, the goal is to calibrate all the components related to a global absolute reference, in order to eliminate all of these errors separately.

Absolute workcell calibration includes robot TCP calibration, tooling calibration and work-object coordinate calibration, where each is performed individually. Each calibration process will measure all the Cartesian coordinates to determine the error between the nominal and true value.

Unlike conventional absolute calibration methods, the relative calibration methods treat all of the errors as relative error between the tooling and the working object compared to a relative reference. Measuring this relative error and finding a way to correct this error is a major advantage of the relative calibration method. As long as the relative error is eliminated compared to the relative reference, the workcell is calibrated and the perfect path will be generated.

There are four steps for completing the relative calibration:

(1) The TCP calibration. This consists of performing a TCP calibration using the robot as a measurement tool. The calibration is accomplished by mounting a calibration target within the workcell and in a position that the robot can reach from various orientations. The calibration target can be a sphere, cylinder, cube or any other definable geometric shape as described in previous sections. The robot is programmed to touch the calibration target surface from various angles with a CMM touch probe or a laser sensor. All contact positions are recorded. The TCP is calculated from the measurements using a nonlinear least squares optimization algorithm.

(2) Set up a relative reference between the robot and a sample-working object. The relative reference is established by having the robot hold a finished sample of the working object (workpiece) while a series of measurements is performed to compensate for the error between the perfect CAD model and the finished sample to obtain a relative reference. This compensation process will make the standard reference in a cost-effective way.

(3) Workpiece Calibration. The robot will hold a raw or unfinished workpiece and the measurement of the raw workpiece will generate a relative error map compared with the relative reference set up in the second step.

(4) An error compensation matrix to calibrate the work-object coordinate, called a virtual work-object coordinate, will be calculated based on the relative error map obtained in the third step. An iterative nonlinear optimization algorithm is employed to obtain this error compensation matrix.

These four steps complete the workcell calibration offline. The subsequent online calibration requires only two steps from those four steps. The first is taking relative measurements of a workpiece utilizing the robot and calibration station. The second step is calculating a new, updated error compensation matrix for the virtual work-object coordinate. These two steps can be performed in real-time and in process.

4.6.1.1 TCP Calibration

For robotic belt grinding, the TCP is fixed and considered to be at the center of the grinding wheel. Now that the LVDT trigger or a laser displacement sensor has been calibrated, the TCP calibration begins by moving the trigger to one edge of the grinding wheel in the proximity of the grinding contact area. The LVDT is displaced by a small amount so that contact is maintained. The robot is commanded to move along the contact area to the opposite edge of the grinding wheel. The LVDT measures the relative displacement of the wheel edge. Any axis orientation error can be detected and calibrated to update the TCP as indicated in Fig. 4.16.

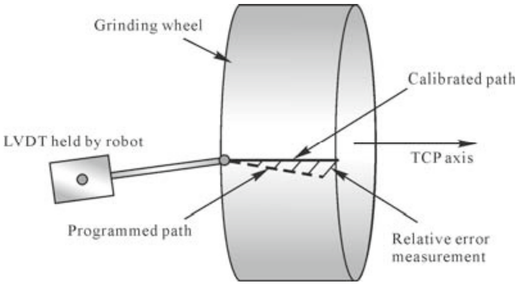


Fig. 4.16. Relative TCP calibration (modified from (Sun, 2009), permission granted)

4.6.1.2 Zero Reference Path Generation

In an ideal environment, the CAD model of the working object is a perfect relative reference. However, in the real case, when the robot path goes through all target points along the workpiece surface, the relative error will not be zero.

$$\varDelta_{\text{Ref}}(i) = \varDelta_{\text{Forward}}(i) + \varDelta_{\text{Backward}}(i) + \varepsilon(i) \tag{4.98}$$

That means the relative error between the robot tooling and the work object $\varDelta_{\text{Ref}}(i)$ is the sum of the robot forward chain error $\varDelta_{\text{Forward}}(i)$, the backward chain error $\varDelta_{\text{Backward}}(i)$ and the random error $\varepsilon(i)$.

Furthermore, the forward chain error $\varDelta_{\text{Forward}}(i)$ is the sum of the manipulator error $\varDelta_{\text{Robot}}(i)$, gripper installation error $\varDelta_{\text{Gripper}}(i)$ and workpiece installation error $\varDelta_{\text{Wheel}}(i)$.

$$\varDelta_{\text{Forward}}(i) = \varDelta_{\text{Robot}}(i) + \varDelta_{\text{Gripper}}(i) + \varDelta_{\text{Wheel}}(i) \tag{4.99}$$

The backward chain error can be described as the tooling system base installation error $\varDelta_{\text{Table}}(i)$ and the error of the tool installation $\varDelta_{\text{Tool}}(i)$

$$\varDelta_{\text{Backward}}(i) = \varDelta_{\text{Table}}(i) + \varDelta_{\text{Tool}}(i) \tag{4.100}$$

We are essentially substituting Eqs.(4.101) and (4.100) into Eq. (4.99) and regrouping the errors into two groups. One is systematic error, which is a natural error from the system, mainly from the robot. It also is a nonlinear error. The other group is installation error. The installation error is a linear error and can be added into one resulting error. In this way, the relative error can be rewritten as the sum of the nonlinear error mainly from the robot manipulator $\varDelta_{\text{Robot}}(i)$ and installation error from all of the remaining components $\varDelta_{\text{UserFrame}}(i)$

$$\varDelta_{\text{Ref}}(i) = \varDelta_{\text{Robot}}(i) + \varDelta_{\text{Userframe}}(i) + \varepsilon(i) \tag{4.101}$$

This error can be transferred into the working object coordinate, called virtual error userframe matrix, since all of these errors can be added linearly. It can be seen that the proposed in-process workcell calibration can be divided into two straightforward steps: relative reference set up and the installation error calibration. In this sense, the relative reference will be

$$f_{\mathrm{Ref}} = f(x,\ y,\ z) + \Delta_{\mathrm{Robot}}(i) \tag{4.102}$$

where $f(x, y, z)$ is the ideal model of the workpiece, usually the CAD model.

The procedure of zero path generation is shown in Fig. 4.17. This begins by obtaining the geometric model of the workpiece, typically in the form of a CAD file. A perfect workpiece can be produced by comparing the CAD data to positions measured on a sample workpiece with a CMM. Once obtained, this perfect workpiece is fixed to the robot mounting plate. The CAD model is then used to generate a calibration path along the profile of the workpiece. This is used to program the path of the robot. For calibration purposes, an LVDT or a laser displacement sensor should be mounted within the work-space of the robot. The robot arm is moved until the workpiece makes contact with the LVDT at the first point of the workpiece calibration path. The LVDT is given a small off-set so that manipulator and installation errors will not cause the workpiece to lose contact with the LVDT. The robot program of the calibration path is executed and the LVDT measures the relative displacement along the path with respect to the first point of contact. When the relative displacement measurements are compared with the original calibration path, the relative measurement error is determined. This error is added to the calibration path to obtain the zero reference path. When this path is used to program the robot, the relative error measurement is zero.

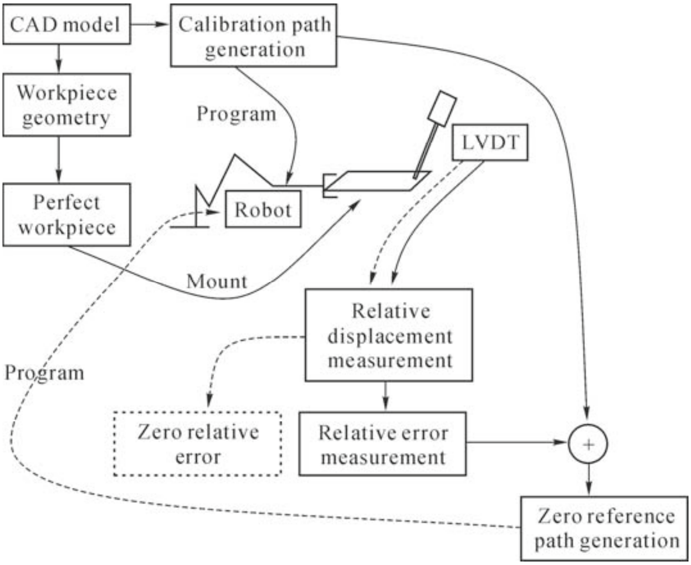

CAD model
Calibration path generation
Workpiece geometry
Program
LVDT
Perfect workpiece
Robot
Mount
Relative displacement measurement
Program
Zero relative error
Relative error measurement
+
Zero reference path generation

Fig. 4.17. Zero reference path generation (from (Sun, 2009), permission granted)

4.6.1.3 Workpiece Calibration

The third and last benchmark in the pre-process stage is to calibrate the workpiece (refer to Fig. 4.18). A typical workpiece is mounted on the robot; the zero reference path is programmed and executed. Again, the LVDT or a laser sensor measures the relative displacement along the path. Any relative error measurement corresponds directly to the geometric errors of the typical workpiece. For workpiece calibration, three sections of the workpiece should be measured with the LVDT, each with its own corresponding zero reference path. The three measured sections, along with their corresponding zero reference and calibration paths, can be used to produce a rigid body of the actual workpiece. The workpiece is calibrated by simply computing the displacement matrix that moves the actual workpiece to best line up with the CAD image. There are many ways this can be done (e.g., nonlinear optimization or singular-value-decomposition). The relative calibration processes described above are performed on an ABB IRB4400_45 Robot Arm interacting with a turbine blade. Note that in this case three zero reference paths were required for the workpiece calibration of the 3D workpiece in space, as shown in Fig. 4.19.

The measurement is designed to decouple the error matrix into two parts as the displacement error $(\Delta x, \Delta y, \Delta z)$ and rotation $(\Delta\theta, \Delta\alpha, \Delta\beta)$ from the mounting plate. The Δx , Δy , $\Delta\theta$ can be obtained by measuring the closed 2D sectional workpiece profiles. In order to get Δz , $\Delta\alpha$, $\Delta\beta$, multiple sections of 2D closed profile of the measurements are needed and the displacement of the workpiece along the *Z* axis needs to be measured too.

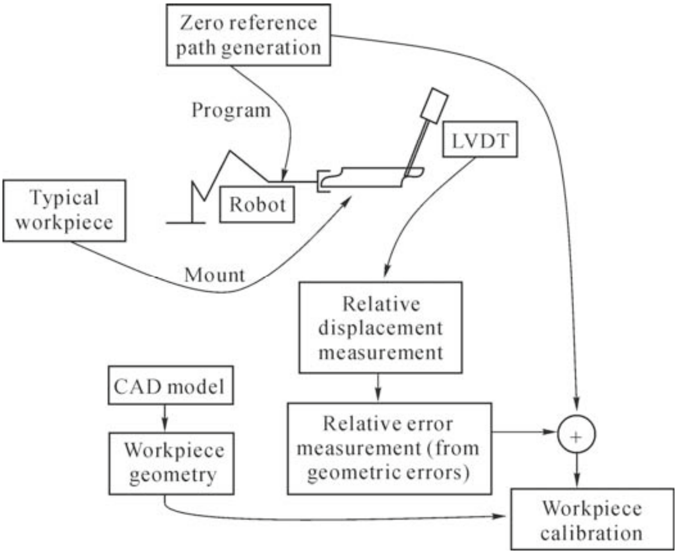

Zero reference path generation
Program
LVDT
Typical workpiece
Robot
Mount
Relative displacement measurement
CAD model
Workpiece geometry
Relative error measurement (from geometric errors)
+
Workpiece calibration

Fig. 4.18 Workpiece calibration flow (from (Sun, 2009), permission granted)

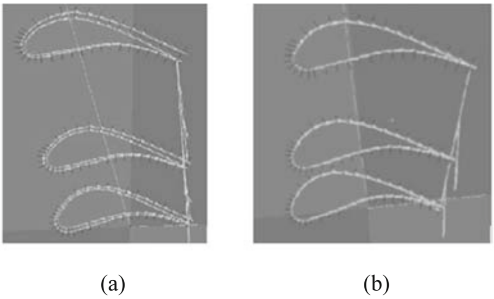


Fig. 4.19. Alignment of three reference paths during workpiece calibration. (a) Before calibration; (b) After calibration (from (Sun, 2009), permission granted)

4.6.2 Robot Error Compensation with Relative Measurement

In some industrial applications, such as inspection of holes on a part, the accuracy requirements (around 0.1 mm) exceed what the robot can provide. This limits the applications of the robot in this area. Inaccuracy of the robot is mainly due to the robot joint mechanical tolerance, arm length deformation, gearbox backlash and so on. When the robot has large joint angle changes, normally a few millimeters error will be observed. This error affects the accuracy of the work object measurement as well as the TCP calibration.

However, although the overall accuracy of the robot is relatively low, it has much higher repeatability. When the robot moves in a small area and with certain joint configurations, it behaves with high repeatability and high accuracy. The goal of calibration is to increase robot accuracy so as to approach this high repeatability when working in a small area. It has been found that when the robot has the translation movement only in a small range, it can maintain the high accuracy (close to its repeatability in a small area). For the translation movement almost all the joints will change to make a linear motion. However, the changes of the joints are relatively small.

Since robot accuracy can be maintained if the robot has only small joint angle changes, in applications such as the inspection of holes in a part, as an example we can divide hundreds holes into zones. For each zone, the robot can reach all the holes for inspection and machining with small joint angle variations. The calibration is conducted with a master or dummy workpiece for all different zones. The following is a hole locating and inspection procedure as an example to illustrate the principle of error compensation with the relative measurement.

(1) Making a master workpiece. A master workpiece is selected that is similar to the workpiece that will be inspected. The master piece can be one in which all the hole locations are known. The location of all the holes on the master workpiece is determined precisely. They are considered as the theoretical values.

(2) Determining the work zone. Group all the holes on the working piece into individual zones. In each zone, all the hole positions can be inspected with small robot joint angle changes as shown in Fig. 4.20(a).

For a specific working zone on the part, the location of all the holes are measured with very high precision tools to make the tolerance within 50 microns. The results are formed as the theoretical data set $\boldsymbol{M}_1$ and are listed as follows

$$\boldsymbol{M}_1 = \begin{bmatrix} -98.6000 & -102.5325 & -75.4425 & -73.9980 & -49.0867 \\ 89.8775 & 142.8750 & 83.6950 & 143.1480 & 73.8055 \\ -26.8625 & -21.8225 & -48.4525 & -50.4000 & -66.6400 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix} \quad (4.103)$$

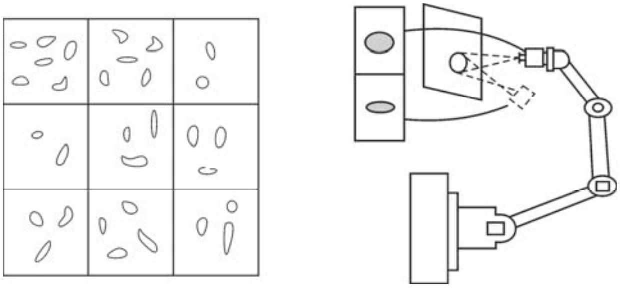


Fig. 4.20. (a) All the holes on the working piece grouped into individual zones; (b) Measurement of holes using robot vision system

(3) Measurement of the master piece. Use the vision system (or other measurement tool) to measure the positions of all the holes on the master piece (Fig. 4.20(b)). First use the vision system to measure some geometrical feature and create the work object coordinate system based on the measured geometry. All the measurements will be based on this work object coordinate system. The measured results with the vision system and robot are formed as the calibration data set $\boldsymbol{M}_2$ and are listed as follows

$$\boldsymbol{M}_2 = \begin{bmatrix} -98.7050 & -102.8150 & -75.6650 & -74.5875 & -49.9175 \\ 90.6100 & 143.5625 & 84.4650 & 143.8400 & 74.8075 \\ -26.6250 & -21.6100 & -48.2325 & -49.9900 & -66.0850 \\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000 \end{bmatrix} \quad (4.104)$$

(4) The transformation matrix is formed based on the measured data set and theoretical data (calibration matrix) (Fig. 4.21). A transform matrix is formed based on the measured hole positions and their theoretical values for each individual zone. This matrix will be used to compensate for the measurement error due to the robot tolerance. It is determined by the following relations

$$\boldsymbol{T}_{12} = \boldsymbol{T}_2 \cdot \text{pinv}(\boldsymbol{T}_1) \quad (4.105)$$

where pinv is the pseudo inverse of the matrix. That is

$$\boldsymbol{T}_{12}=\begin{bmatrix}0.9466 & -0.0047 & -0.0452 & -6.1049\\ 0.0262 & 0.9988 & 0.0261 & 4.1184\\ 0.0234 & 0.0014 & 1.0192 & 2.8641\\ 0 & 0.0000 & -0.0000 & 1.0000\end{bmatrix} \tag{4.106}$$

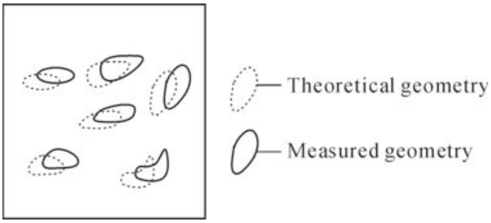


Fig. 4.21. Theoretical and measured positions of the holes in one specific zone of the master workpiece. Then the calibration matrix for this specific zone can be determined

(5) Measurement of the workpiece under inspection. Place the workpiece under inspection at the same location as the master work piece. Run the same robot program to create the work object coordinate system. Measure all the cooling holes on the part. During the measurement try to keep the same robot joint configuration as for the masterpiece for each individual zone. The measured results are

$$\boldsymbol{M}_{3}=\begin{bmatrix}-98.71 & -102.82 & -75.62 & -74.48 & -49.90\\ 90.51 & 143.40 & 84.50 & 145.74 & 74.70\\ -26.62 & -21.51 & -48.24 & -49.91 & -66.10\\ 1.0 & 1.0 & 1.0 & 1.0 & 1.0\end{bmatrix} \tag{4.107}$$

(6) Correct the measurement results with calibration matrix. Due to robot error the measured hole positions may not be accurate. Multiplying the measured results by the calibration matrix performs the correction. The corrected position will be

$$\boldsymbol{M}_{3}=\text{pinv}(\boldsymbol{T}_{12})\cdot\boldsymbol{M}_{3}$$

$$=\begin{bmatrix}-98.6599 & -102.4988 & -75.3354 & -73.9111 & -49.0798\\ 89.7829 & 142.7665 & 83.7830 & 145.0459 & 73.6967\\ -26.7929 & -21.7655 & -48.5317 & -50.2888 & -66.6428\\ 1.0000 & 1.0000 & 1.0000 & 1.0000 & 1.0000\end{bmatrix} \tag{4.108}$$

This section has shown a calibration method for improving the accuracy of a robotic measuring system with joint configuration dependent performance. With such an increased measuring accuracy, the robotic measuring system can be used in many industrial applications requiring a robotic measuring system of high accuracy. One such application is the automated measurement of the orientations and positions of many holes located on the surface of a part.

4.7 Summary

In this chapter, a general calibration model is proposed. It consists of a kinematic relation and a geometric constraint. The solution to the model is in general a nonlinear optimization process that minimizes the cost function defined in the model. The algorithm can be simplified into linear equations and solved by linear least squares algorithm by selecting a specific calibration target or having a constrained robot movement to decouple the components of unknown variables. Various calibration methods are reviewed and categorized, and show a good compliance to the general mode. It is believed that the general model can be used as a reference for the deduction of new calibration methods, which adopt different kinds of accessorial equipment/mechanism to provide a variety of constraints and for various measurement tools.

References

Charles L, Lawson RJH (1974) Solving Least Squares Problems: Automatic Computation. Prentice-Hall, New Jersey.

Chong EK, Zak SH (1996) An Introduction to Optimization. Wiley Press, New York, 147-165.

Dornaik F, Horaud R (1998) Simultaneous robot-world and hand-eye calibration. IEEE Transactions on Robotics and Automation, 14(4):617-622.

Gan ZX, Sun YQ, Tang Q (2004) In process relative robot workcell calibration. US Patent 6,812,665 B2.

Greer DR and Kim M (2000) Calibration and compensation of robot based gauging system. US Patent 6,078,846.

Hakan B, Mohamed SK (1997) A three-step camera calibration method. IEEE Transactions on Instrumentation and Measurement, 46(5):1165-1172.

Huissoon JP (2000) Method and device for robot tool frame calibration. US Patent 6,004,308.

Li JF, Tang Q, Wang P, Zhu JH, Guo YK (2008a) A new multiple robot laser 3D scanning system. Journal of Image and Graphics 13(4)741-748 (In Chinese).

Li JF, Zhu JH, Guo YK, Lin XD, Duan K, Wang Y, and Tang Q (2008b) Calibration of a portable laser 3D scanner used by a robot and its use in measurement. Optical Engineering, 47(1):1-8.

Li JF, Zhu JH, Tang Q, Chen L, Xu M, Guo YK (2007) Scanning method of robot 3D scanner system and its industrial application. Opto-Electronic Engineering 34(2):15-21 (In Chinese).

Mao J, Zhu J (2004) Study on high accurate calibration of industrial robot vision positioning system. Robot, 26(2):139-144 (In Chinese).

Marquardt D (1963). An algorithm for least squares estimation of nonlinear parameters. Journal of the Society for Industrial and Applied Mathematics, 11(2):431 - 441.

Mitsi S, Bouzakis KD, Mansour G, sagris D, Maliarls G (2005) Off-Line programming of an industrial robot for manufacturing. The International Journal of Advanced Manufacturing Technology, 26:262-267.

Motai Y, Kosaka A (2008) Hand-Eye calibration applied to viewpoint selection for robotic vision. IEEE Transactions on Industrial Electronics, 55(10):3731-3741.

Motta J, Sonsa C, Afonso F (2004), An off-line robot programming system including workcell and robot calibration. ABCM Symposium Series in Mechatronics, 1:134-143.

Park FC, Martin BJ (1994) Robot sensor calibration: solving $AX = XB$ on the Euclidean group. IEEE Transactions on Robotics and Automation, 10(5):717-721.

Shiu YC, Ahmad S (1989) Calibration of wrist-mounted robotic sensors by solving homogenous transform equations of the form $AX = XB$. IEEE Transactions on Robotics and Automation, 5(1):16-29.

Stenberg B (2002) Method for cell alignment and identification and calibration of robot tool. US patent 6,356,808 B1.

Sun YQ, Gilblin D, Kazerounian K (2009) Accurate robotic belt grinding of workpieces with complex geometries using relative calibration techniques. Robotics and Computer-Integrated Manufacturing, 25:204-210.

Świder J, Foitk, Wszolek G, Mastrowski D (2007) The off-line programming and simulation software for the Mitsubishi Movemaster RV-M1 robot. Journal of Achievements in Materials and Manufacturing Engineering, 20(1-2):499-502.

Tang Q, Gan ZX, Wu SH, Sun YQ (2007) Calibration method for the laser scanner position relative to the robot coordinate frame. Chinese Patent ZL2005100534601 (In Chinese).

Tang Q, Gan ZX, Wu SH, Ning SJ, Sun YQ (2008) Robot tool center point calibration based on the measurement of a plate. Chinese Patent ZL 200510087041X (In Chinese).

Tang Q, Brantmark H, Gan ZX, Brogardh T (2003) Robot machining tool position and orientation calibration. US Patent 6,941,192 B2.

Thorne HF (1995) Tool center point calibration apparatus and method. US patent 5,457,367.

Tian MQ (2006) A survey of calibration in a vision-robot system. Chinese Journal of Industrial Instrumentation and Automation, 2:14-17 (In Chinese).

Tsai RY (1987) A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-shelf TV camera and lenses. IEEE Journal of Automation, 3(4):323-334.

Tsai RY, Lenz RK (1989) A new technique for fully autonomous and efficient 3D robot hand/eye calibration. IEEE Transactions on Robotics and Automation, 5(3):345-358.

Xu D, Wang LK, Tan M (2005) A calibration method of structured light stripe for hand-eye system based on robot movement. Chinese Journal of Scientific Instrument, 26(11):1101-1113 (In Chinese).

Zhang ZY (2000) A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(11): 1330-1334.

Zhu ZQ, Tang Q, Li J, Gan Z (2004) Calibration of laser displacement sensor used by industrial robots. Optical Engineering, 43(1):12-13.

Zhu ZY, et al. (2004) High-Speed calibration method for the relationship of eye-in-hand of robot vision. Chinese Journal of Optical Technique, 30(2): 150-152 (In Chinese).

Zhuang HQ, Roth ZS (1996) Camera-Aided Robot Calibration. CRC Press, Florida.

Zhuang HQ, Roth ZS, Sudhakar R (1994) Simultaneous robot/world and tool/flange calibration by solving homogenous transformation equations of the form $AX = YB$. IEEE Transactions on Robotics and Automation, 10(4):549-554.

Zhuang HQ, Wang K, Roth ZS (1995) Simultaneous calibration of a robot and a hand-mounted camera. IEEE Transactions on Robotics and Automation, 11(5):649-660.

5

Image Processing of Laser Structured-Light Based Vision System

For laser stripe sensors there are two types of image processing tasks. One is to find locations of the control points in the calibration target. This is usually for the calibration procedure. The other is to find the center position of the laser line, which is used for the reconstruction procedure. For the first task, accuracy of the algorithm is essential. And for the second task, reliability, insensitivity to the ambient light, accuracy as well as speed need to be addressed. In applications associated with reconstructed range images, the registration algorithm is a very powerful tool used in workpiece calibration, localization, and shape inspection, especially for free form surface. In this chapter we will review 2D image processing techniques as well as range image registration algorithms associated with laser stripe sensors and robot vision systems. These techniques are widely used in visual sensing applications as described in Chapter 7.

5.1 Control Point Extraction from Pattern Images

An accurate location of control points in the image plane is a very critical step in the camera calibration process. There are two methods. The first one is based on determining the corners of a set of squares (Pedersini, 1997), which are regularly located on a plane, as shown in Fig. 5.1. The second approach calculates the center positions of a group of circles (Heikkila and Silver, 1996), which are regularly distributed on a plane, as shown in Fig. 5.2.

Z. Gan et al., *Visual Sensing and its Applications*

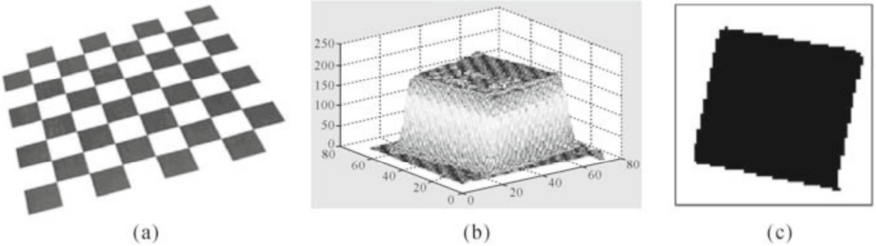


Fig. 5.1. Calibration plate with square patterns. (a) Image of calibration pattern; (b) Intensity level of one square; (c) Binary image of the square

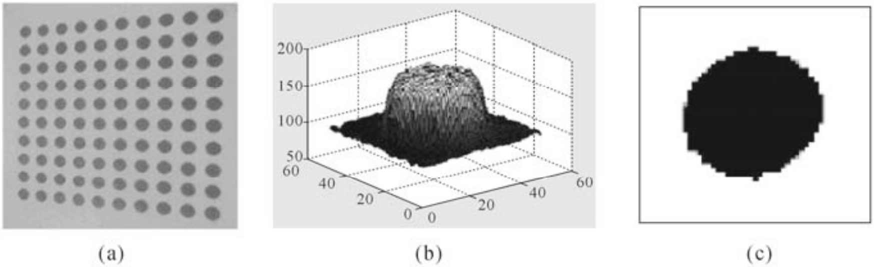


Fig. 5.2. Calibration plane with circle patterns. (a) Image of calibration pattern; (b) Intensity level of one circle; (c) Binary image of the circle

5.1.1 Feature Extraction from Squared Control Points

The positions of square corners are computed by the following procedure:

(1) Acquire a gray scale image.

(2) Extract the edge of the image.

(3) Fit straight lines to find edge points. The intersection between two cross straight lines yields the corner point.

In the above procedure, the key process for feature extraction from squared control points is the edge extraction algorithm that determines the measurement accuracy of the control point.

The extraction of edges from a gray scale image is a critical step in many image processing techniques. A variety of approaches are available which determine the magnitude of contrast changes and their orientation. Extensive literature exists, describing the available operators and the post-processing methods. A trade-off exists between efficiency and quality of the edge detection. Fast and simple edge detection can be performed by filters, such as the popular Sobel operator (Gonzalez, 1992) which conducts the convolution of a small kernel (3×3 pixels) over the image. Alternatively, more computationally intensive contour detection techniques are available, such as the Deriche (1987) or Canny (1986) method. These detectors require that a set of parameters be varied to detect the desired scale and curvature of edges in the image. It is necessary to compare the simple Sobel detector and the complex Deriche-type detectors before selecting the edge detection scheme of preference.

The following gives a step by step procedure to illustrate how to compute the corner positions of the squared pattern.

5.1.1.1 Edge Extraction by Using the Gradient Operator

This algorithm is used for extraction of the edges from the gray scale image. There are mainly two steps for edge extraction using this algorithm.

***Step1**:Smoothing the Initial Gray Scale Image Using a Gaussian Filter*

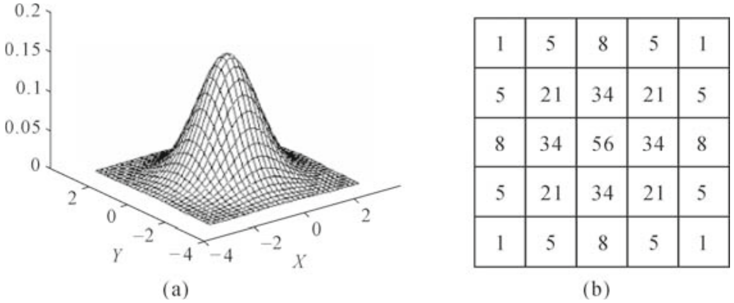
In order to reduce the noise, a Gaussian filter is used to convolute the input image as

$$g(x, y) = f(x, y) \otimes h(x, y) \tag{5.1}$$

where $g(x, y)$ is the output image, $f(x, y)$ is the input image, $\otimes$ is the convolution operator, $h(x, y)$ is the Gaussian filter that is defined as

$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp(-\frac{x^2 + y^2}{2\sigma^2}) \tag{5.2}$$

where σ is the standard deviation of the distribution. The distribution of a Gaussian filter is shown in Fig. 5.3(a) and its corresponding discrete form is shown in Fig. 5.3(b).



1	5	8	5	1
5	21	34	21	5
8	34	56	34	8
5	21	34	21	5
1	5	8	5	1

Fig. 5.3. (a) 2D Gaussian filter with $\sigma = 1$ and (b) its discrete approximation with window size of 5×5

The idea of Gaussian smoothing is to use this 2D distribution as a "point-spread" function to smooth the image to remove detail and noise. This is achieved by convolution. Since the image is stored as a collection of discrete pixels, we need to produce a discrete approximation to the Gaussian function before we can perform the convolution. In theory, the Gaussian distribution is non-zero everywhere, which would require an infinitely large convolution kernel, but in practice it is effectively zero at more than about three standard deviations from the mean, and so we can truncate the kernel at this point. Fig. 5.3(b) shows a suitable integer-valued convolution kernel that approximates a Gaussian with $\sigma = 1$.

***Step2**: Compute the Gradient of the Smoothed Gray Scale Image to Get the Edges*

The edges of an input image are obtained by the first derivative of the image. The definition of the gradient of function $f(x, y)$ at (x, y) is a vector described by

$$\nabla_f = [G_x,\ G_y]^{\mathrm{T}} = [\frac{\partial f}{\partial x},\ \frac{\partial f}{\partial y}]^{\mathrm{T}} \tag{5.3}$$

and the gradient magnitude is given by

$$\left|\nabla_f\right| = \lfloor G_x^{\ 2} + G_y^{\ 2} \rfloor^{\frac{1}{2}} = \left[\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2\right]^{\frac{1}{2}} \tag{5.4}$$

It can be simplified as

$$\left|\nabla_f\right| \approx \left|G_x\right| + \left|G_y\right| \tag{5.5}$$

For the digital image process, it uses the template to compute the gradient of a gray scale image. The common template of the gradient is shown in Fig. 5.4, which is also called a Sobel operator.

The convolution of an image with the Sobel kernel can be approximated as

$$\left|\nabla_f\right| \approx \left|(p_1 + 2p_8 + p_9) - (p_1 + 2p_2 + p_3)\right| + \left|(p_3 + 2p_6 + p_9) - (p_1 + 2p_4 + p_7)\right| \tag{5.6}$$

using the kernel as shown in Fig. 5.4(c).

−1	−2	−1
0	0	0
1	2	1

(a)

−1	0	1
−2	0	2
−1	0	1

(b)

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

(c)

Fig. 5.4. Sobel operators in (a) x direction G_x and in (b) y direction G_y and (c) Pseudo convolution kernel

5.1.1.2 Line Intersection to Calculate the Corner Positions

All the points detected along the square side from the above procedure can be fitted into a straight line in order to remove the measurement noise and outliers. The straight line equation is given by

$$a_i x + b_i y = 1 \tag{5.7}$$

where $\{a_i, b_i\}$ are the fitting parameters for a straight line L_i. The line parameters are determined by using the linear least-squares method.

The corner position is then determined by the intersection of the two cross lines L_m and L_n, which are nearly perpendicular to each other. That is given by

$$x = \frac{b_n - b_m}{a_m b_n - a_n b_m}; \; y = \frac{a_n - a_m}{a_n b_m - a_m b_n} \tag{5.8}$$

This method can provide the sub-pixel precision position of the corner points.

Fig. 5.5 illustrates the process for identifying and locating the corner points. A calibration plate with square patterns is used as an input image, as shown in Fig. 5.5(a). The image is smoothed with a Gaussian filter and then convoluted with a Sobel gradient operator to extract the edges points of squares, as shown in Fig. 5.5(b). All the points along the edge line are used to fit the straight line, as shown in Fig. 5.5(c). Intersections of vertical and horizontal lines are corner points, as shown in Fig. 5.5(d), which are used as the control points for camera calibration.

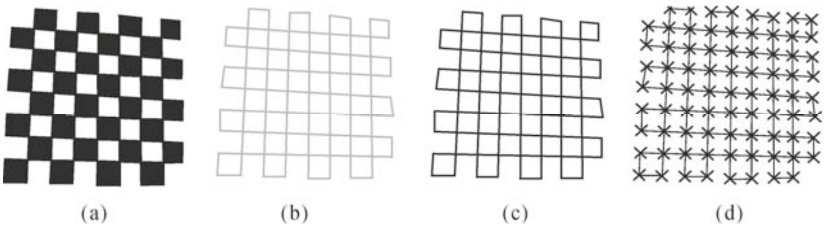


Fig. 5.5. Process to identify and locate the corner points of a square pattern. (a) Original image of the square pattern; (b) Edges detected by using a gradient operator; (c) Fitted lines to the edge points; (d) Corner points determined by intersecting two sets of edge lines

5.1.2 Feature Extraction from Circle Control Points

Targets of circular features are used commonly, due to the fact that the projective projection of a circle is always a circle or an ellipse. A circular feature is easy to locate with high accuracy. In practice, a median filter is first used to remove scattering noise and then the gray scale image is binarized to create the binary image. It is followed by a labeling operation to identify each circular disk in the binary image. The location of each disk is calculated by using either a center of gravity algorithm or a circle fitting algorithm. In the binarization process, an adaptive thresholding is sometimes applied, since the illumination on the object surface is seldom uniform and a fixed thresholding would not work properly. For a non-uniform illumination, the center of gravity method will cause the feature location to become biased. This can be overcome by first detecting the feature boundaries with sub-pixel precision, and then fitting a geometric model (ellipse or circle) to the measurement data.

5.1.2.1 Median Filtering

A 3×3 median filter is applied to the input image in the first step, to reduce noise from the original image. Median filtering is similar to an averaging filter, in which each output pixel is set to an average of the pixel values in the neighborhood of the corresponding input pixel. However, with median filtering, the value of an output pixel is determined by the median of the neighborhood pixels, rather than the mean. The median is much less sensitive than the mean to extreme values called outliers. Median filtering is therefore better able to remove these separate noises or outliers without reducing the sharpness of the image. Fig. 5.6 illustrates the effort of a median filter on the calibration target with the dot pattern.

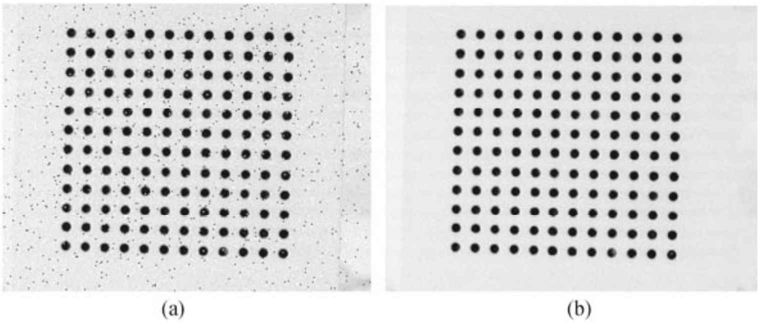


Fig. 5.6. Effort of median filtering. (a) Original image; (b) Processed image with median filtering with a 3×3 operator

5.1.2.2 Adaptive Thresholding

In computer vision applications, image segmentation is an important process algorithm. For understanding a high-level image in practical applications involving visual inspection, it is very useful to be able to separate the regions of the image corresponding to objects of interest from the regions of the image that correspond to background, under the conditions of various contrast. A thresholding image is used to segment an image by setting all pixels, whose intensity values are above a threshold to a foreground value and all the remaining pixels to a background value.

Whereas the conventional thresholding operator uses a global threshold for all pixels, it works well only for images with a well-shaped bimodal histogram. Adaptive thresholding changes the threshold dynamically over the image. This more sophisticated version of a thresholding algorithm can overcome segmentation problems for a non-uniform illumination image.

The adaptive thresholding algorithm takes gray scale images as input and outputs binary images. In order to convert a gray scale image to a binary image, the threshold is set at a certain value. Pixels with a gray level above the threshold are set to one or white (255) and all other pixels are set to zero or black. Threshold selection is usually made based on the information contained in the gray level histogram of a given image. The objective of this approach is to find the optimal location at the bottom of the valley of the histogram which separates the two groups or segments in a better effort.

Based on locally adaptive image thresholding, the following algorithm is used for thresholding of the calibration pattern image:

(1) Convolute the image with a mean operator with a window size of 7×7.

(2) Subtract the original image from the convoluted image.

(3) Threshold the different image with a constant threshold C (set C =10 as default).

(4) Invert the threshold image.

Fig. 5.7 shows the original image with non-uniform illumination and the result of the adaptive thresholding algorithm.

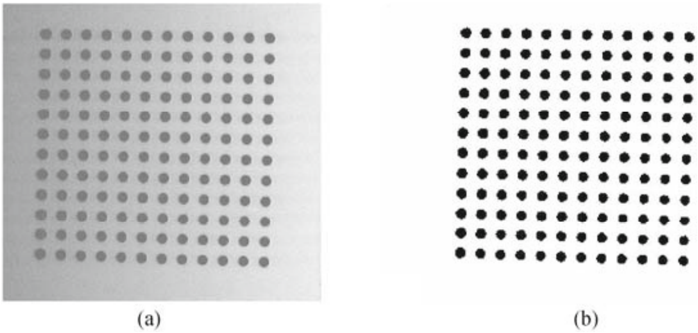


Fig. 5.7 (a) Calibration target under non-uniform illumination; (b) Resulting image of adaptive thresholding

5.1.2.3 Binary Image Labeling

The purpose of image labeling is to find the connected components of the image. All pixels in a connected component share similar pixel intensity values. Once all groups have been determined, each pixel is labeled with a gray level or a color (color labeling) according to the component it was assigned to. Extracting and labeling of various connected components in an image are central to many automated image analysis applications.

Information about each connected component is listed in the table, containing the number of the regions, its areas (in pixels) and bounding rectangles. The regions will be filtered by presetting filter regions parameters, including their area, width, height, Euler number, etc. Only regions that fall into the range of the filter are kept for further processing.

A better and wiser approach is to use the histogram of the region area as part of the criteria of filtering. Since there are multiple circle points in a calibration pattern, they comprise a peak in the histogram and so the other parts of the histogram can be removed as being false circular parts or noises.

The geometric property of a/an circle/ellipse is a good candidate for filtering as well. The ellipse area is calculated by area = πab, where a and b equal the major and minor radius of the projected ellipse, respectively, which can be approximated by half of the width and height of the bounding rectangle. Using the above criteria, spurious components, which are not corresponding to circular features, will be removed from the list. Fig. 5.8 shows the result of image labeling for the calibration target in Fig. 5.7.

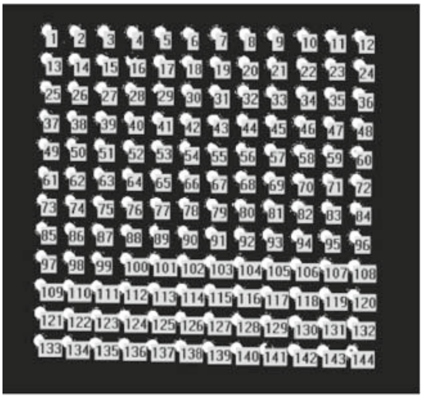


Fig. 5.8. Image labeling result for the calibration patterns as shown in Fig. 5.7

5.1.2.4 Calculation of the Circle Position

The circle position is usually represented by the centroid of the circular disk. The centroid of every circular disk is computed by means of the gravity center method and the ellipse fitting algorithm for an accurate result.

Gravity Center Method

The gravity center of a circular disk is calculated by the following equation

$$x_0 = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} I(x,y)x}{\sum_{x=1}^{m}\sum_{y=1}^{n} I(x,y)}, \qquad y_0 = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n} I(x,y)y}{\sum_{x=1}^{m}\sum_{y=1}^{n} I(x,y)} \tag{5.9}$$

where (x_0, y_0) is the gravity center of a circular disk with the intensity distribution $I(x, y)$ and the rectangular range $[m, n]$. The rough position needs to be determined by using an image labeling algorithm, as described previously. The image used in the gravity center calculation can be either a gray scale or a binary image. Noise of the original image may affect the accuracy of the centroid position, so preprocessing of the image, such as Gaussian filtering and median filtering, is usually applied to the original image before the gravity center calculation.

Fig. 5.9 shows the result using this method to compute the centroids.

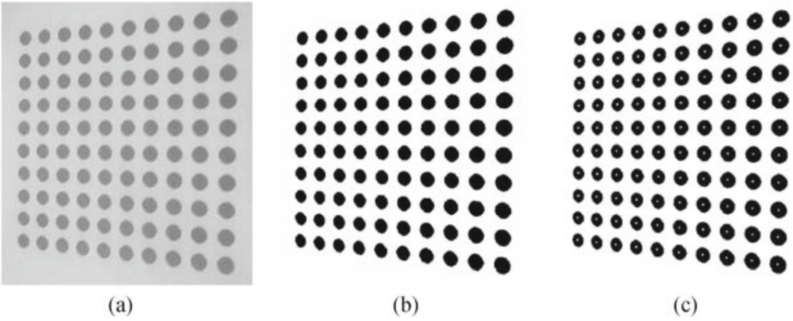


Fig. 5.9. Determination of center position of the circular disk by using gravity center method. (a) Initial gray image; (b) Binary image; (c) Centroid of the circular disk

Curve Fitting Method

When the circle or ellipse is big enough, curve fitting method is more accurate than the gravity center method. The first step is to find the edge of the circle or ellipse by using an edge detection algorithm or simply by thresholding method. Then the curve fitting algorithm is applied to the edge points to compute the center of the circle or ellipse.

The ellipse curve is expressed as

$$f(\boldsymbol{a},\ \boldsymbol{x}) = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{5.10}$$

where $\boldsymbol{a} = [\,a \quad b \quad c \quad d \quad e \quad f\,]$, $\boldsymbol{x} = [x^2 \quad xy \quad y^2 \quad x \quad y \quad 1]$, and then the center of the ellipse will be computed as

$$X_c = \frac{2cd - be}{b^2 - 4ac}, \qquad Y_c = \frac{2ae - bd}{b^2 - 4ac}, \qquad b^2 - 4ac \neq 0 \tag{5.11}$$

Fig. 5.10 shows the center position of the dot array by using a curve fitting algorithm.

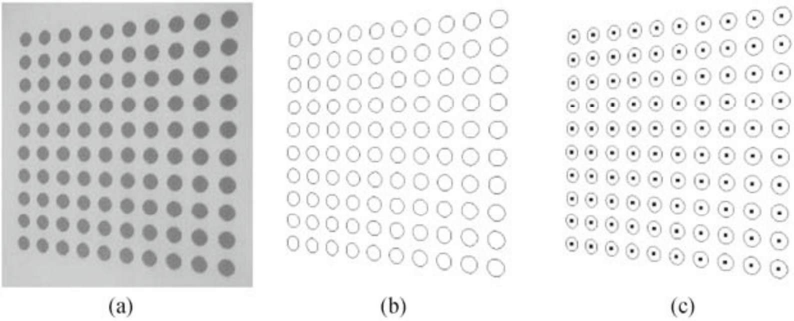


Fig. 5.10. Determination of center position of the circular disk by using curve fitting method. (a) Initial gray image; (b) Edge detection result; (c) Calculated center of each circle by using curve fitting method

5.2 Laser Stripe Sub-Pixel Positioning

Extraction of the center of a laser stripe is the main task of a structured light measurement system. Reliability, insensitivity to the ambient light, accuracy and speed are necessary for the laser stripe positioning process. In general, there are two types of methods for detecting the center position. One is based on geometry center detection and the other is based on energy center calculation (Sui, 2003). Calculation of the geometry center of the laser stripe is simple and fast but it is sensitive to noise and illumination. It includes, for example, the middle point method, the thinning and pruning algorithm (Wu, 2006). The method for finding the energy center is more robust and not very sensitive to the illumination change. It includes a gray scale gravity algorithm (Wu, 2007), curve fitting method (Faugeras, 1993), etc.

In the middle point method, the two edges of the laser stripe are detected first by using thresholding or gradient methods and then the center of the stripe is obtained. That is the middle point of the two edge points. Although this method is simple and fast, it is sensitive to noise.

The other method is simply to find the position that has the maximum intensity value along the cross laser stripe profile. This position is considered as the center of the laser stripe. Obviously, this method is also sensitive to noise although it may be simple and fast.

Since the intensity of the laser profile is usually a Gaussian distribution, its parameters can be determined by using the curve fitting algorithm (Faugeras, 1993). The position of the laser stripe is the center of the Gaussian distribution. This method is robust and insensitive to noise. However, it requires more calculation time.

5.2.1 Thinning and Pruning Algorithm

The position of the laser stripe can be detected by thinning and pruning algorithms. Thinning is a morphological operation that is used to remove selected foreground pixels from binary images, somewhat like erosion or opening. It is used for skeletonization. It is commonly used to tidy up the output of edge detectors by reducing all lines to single pixel thickness. Thinning is normally only applied to binary images and produces another binary image as output.

The procedure is as follows:

(1) Acquisition of a gray scale image of the laser line (Fig. 5.11(a)).

(2) Reduction of noise, as indicated in Fig. 5.11(b) by using a 3×3 median filter.

(3) Binarization of the gray scale image (Fig. 5.11(c)).

(4) Thinning of the binary image (Fig. 5.11(d)).

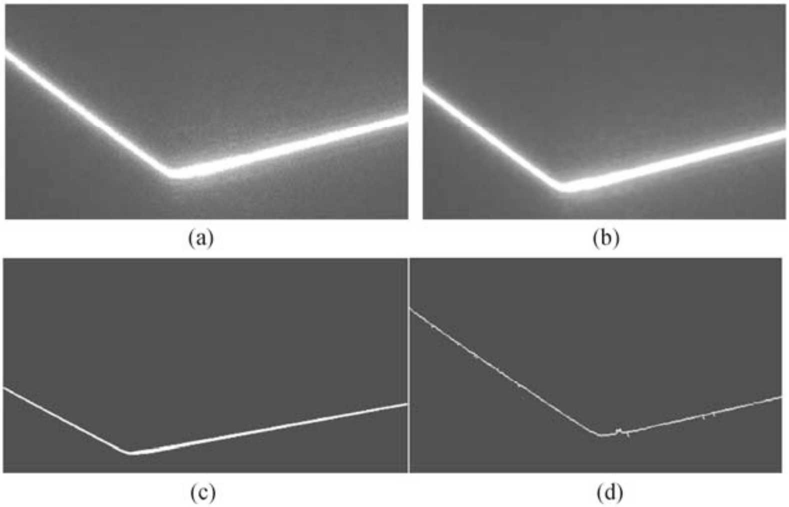


Fig. 5.11. Thinning process of a gray scale image. (a) Initial image; (b) Filtered image with a 3×3 median filter; (c) Binary image; (d) Image after thinning process

The process for thinning the image is as follows:

(1) Aligning one element (p_5) of the moving window template shown in Fig. 5.12 with an individual pixel on the binary image to obtain the value of fifteen elements p_1 to p_{15}

(2) Comparing p_1 to p_9 with delete templates, as shown in Fig. 5.13. If it matches any of the eight delete templates, p_5 is deleted, otherwise p_5 is reserved.

(3) Comparing p_1 to p_{15} with reservation templates shown in Fig. 5.14. If it matches one of the six reservation templates, p_5 is reserved, otherwise p_5 is deleted

(4) Repeating (1)~(3) for all the pixels in the binary image completes the thinning operation.

p_1	p_2	p_3	p_{13}
p_4	p_5	p_6	p_{14}
p_7	p_8	p_9	p_{15}
p_{10}	p_{11}	p_{12}	×

Fig. 5.12. 4×4 template

0	0	0
×	1	×
1	1	1

(a)

0	×	1
0	1	1
0	×	1

(b)

1	1	1
×	1	×
0	0	0

(c)

1	×	0
1	1	0
1	×	0

(d)

×	0	0
1	1	0
×	1	×

(e)

0	0	×
0	1	1
×	1	×

(f)

×	1	×
0	1	1
0	0	×

(g)

×	1	×
1	1	0
×	0	0

(h)

Fig. 5.13. Delete templates

×	1	×	0
0	1	1	0
×	1	×	0
×	×	×	×

(a)

×	×	0	0
0	1	1	0
×	×	1	×
×	×	×	×

(b)

×	×	1	×
0	1	1	0
×	×	0	0
×	×	×	×

(c)

×	0	×	×
1	1	1	×
×	1	×	×
0	0	0	×

(d)

×	0	×	×
×	1	×	×
1	1	0	×
×	0	0	×

(e)

×	0	×	×
×	1	×	×
0	1	1	×
0	0	×	×

(f)

Fig. 5.14. Reservation templates

Pruning Operation

After thinning, the image may have some burrs, as shown in Fig. 5.15. In order to obtain the clean and smooth laser line, the pruning operation is applied.

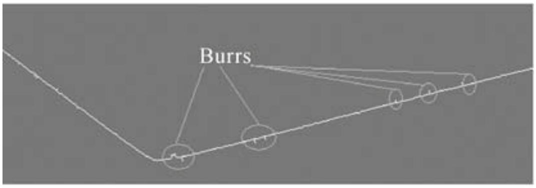


Fig. 5.15. Burrs on the thinning image

Fig. 5.16 shows a 3×3 moving window that applies to an binary image and its value $p_1 - p_9$ is evaluated.

p_1	p_2	p_3
p_4	p_5	p_6
p_7	p_8	p_9

Fig. 5.16. 3×3 template

Fig. 5.17 shows the pruning templates. By comparing the value of the moving window with these pruning templates, the image can be pruned:

(1) If the value of the moving window matches the template shown in Fig. 5.17(a), the pixel under investigation is an isolated element and will be deleted.

(2) If the value of the moving window matches one of templates shown in Fig. 5.17(b), which means the pixel is an endpoint and will be deleted.

(3) If the value of the moving window matches the templates shown in Fig. 5.17 (c), which means the pixel is the laser strip and will be kept.

(4) If the value of the moving window matches the templates shown in Fig. 5.17 (d), which means the pixel is a branch point and will be deleted.

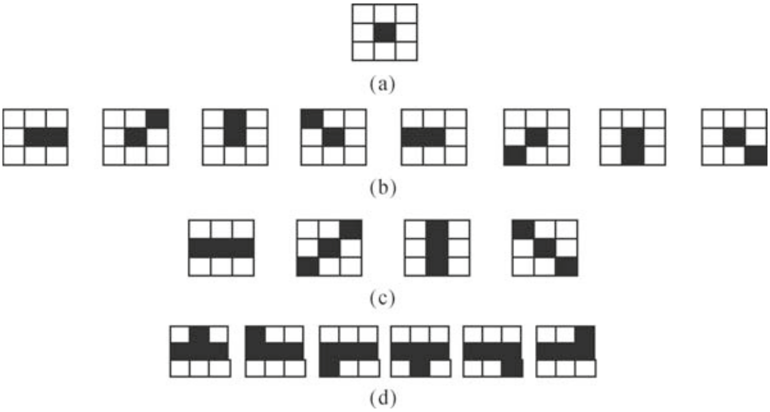


Fig. 5.17. (a)-(d) showing the pruning templates

After image pruning, the center position of the laser strip is detected, as shown in Fig. 5.18.

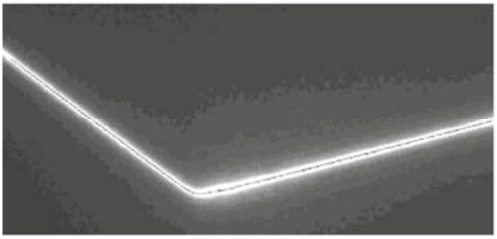


Fig. 5.18. The result after pruning

5.2.2 Gray Scale Gravity Center Algorithm

Fig. 5.19 shows the captured image of the laser line and its profile along the vertical direction. Using Eq. (5.12), the position of the laser strips can be detected using the gravity center algorithm.

$$X_j = \frac{\sum_{i=1}^{N} x_i I(x_i, y_j)}{\sum_{i=1}^{N} I(x_i, y_j)}, \qquad Y_j = y_j \tag{5.12}$$

where (X_j, Y_j) is the gravity center of the profile, $I(x_i, y_j)$ is the gray scale intensity of pixel (x_i, y_i), N is the number of pixels of the profile.

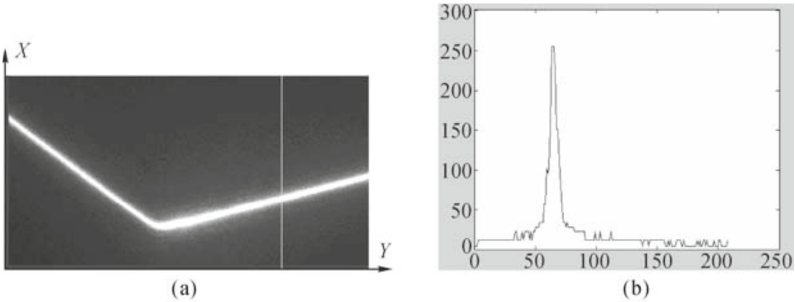


Fig. 5.19. (a) Image of laser line; (b) Profile along the vertical direction

If the laser line direction is not towards the vertical or horizontal, the intensity distribution along the horizontal or vertical directions is far from the Gaussian distribution, resulting in inaccurate laser stripe positioning. A gravity center algorithm along the direction of the laser profile can solve this problem (Wu, 2007).

The following is the procedure:

(1) Detect the skeleton of the laser line.

(2) Calculate the normal direction of each point on the skeleton.

(3) Apply the gravity algorithm along the normal direction of the laser profile.

A skeleton of the laser line has been discussed in the previous section. The key step is to calculate the normal direction of each point on the skeleton.

5.2.2.1 Calculation of Normal Direction with Directional Template

One simple method is to use the direction templates. The normal direction of the laser line can be simplified into four types, including vertical, horizontal, 45° right and 45° left directions. According to these types, four directional templates can be set, as shown in Fig. 5.20.

By comparing a laser line image with these four directional templates, the normal direction can be determined.

$$
\begin{array}{|c|c|c|c|c|}
\hline
0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 \\ \hline
1 & 1 & 1 & 1 & 1 \\ \hline
0 & 0 & 0 & 0 & 0 \\ \hline
0 & 0 & 0 & 0 & 0 \\ \hline
\end{array}
\quad
\begin{array}{|c|c|c|c|c|}
\hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
\end{array}
\quad
\begin{array}{|c|c|c|c|c|}
\hline
1 & 0 & 0 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 0 & 0 & 1 & 0 \\ \hline
0 & 0 & 0 & 0 & 1 \\ \hline
\end{array}
\quad
\begin{array}{|c|c|c|c|c|}
\hline
0 & 0 & 0 & 0 & 1 \\ \hline
0 & 0 & 0 & 1 & 0 \\ \hline
0 & 0 & 1 & 0 & 0 \\ \hline
0 & 1 & 0 & 0 & 0 \\ \hline
1 & 0 & 0 & 0 & 0 \\ \hline
\end{array}
$$

Fig. 5.20. Directional templates

The comparison is based on the correlation between the template and the image as given by

$$H_k(i, j) = \sum_{u=1}^{11} \sum_{v=1}^{11} I(i - 6 + u, j - 6 + v) T_k(u, v) \quad k = 1, 2, 3, 4 \tag{5.13}$$

where $I(i, j)$ is the image intensity value on the skeleton of the laser stripe and $T_k(u, v)$ (k =1, 2, 3, 4) are the four directional templates. For each pixel (i, j) on the skeleton, its direction is determined based on the calculated value $H_k(i, j)$. If $H(k)$ = max(H), it indicates that the k-th template has the largest correleation value with the image and the laser stripe direction is closest to the k-th template.

After the normal direction of the laser stripe is determined, its gravity center is calculated by selection of the pixels along the normal direction.

5.2.2.2 Calculation of Normal Direction with Sobel Operator

For more accurate calculation of the normal direction of the laser stripe, a Sobel operator can be used (Bazen, 2002; Xiong, 2009). The normal direction of the pixel (i, j) is calculated as

$$n(i, j) = t(i, j) + \frac{\pi}{2} \tag{5.14}$$

where $t(i, j)$ is the tangential direction of the pixel that is calculated as

$$t(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{V_y(i, j)}{V_x(i, j)}\right) \tag{5.15}$$

where

$$
\begin{aligned}
V_x(i, j) &= \sum_{u=i-w/2}^{u=i+w/2} \sum_{v=j-w/2}^{v=j+w/2} (G_x^2(u, v) - G_y^2(u, v)) \\
V_y(i, j) &= \sum_{u=i-w/2}^{u=i+w/2} \sum_{v=j-w/2}^{v=j+w/2} 2G_x(u, v) G_y(u, v)
\end{aligned}
\tag{5.16}
$$

where w is the widow size, G_x and G_y are Sobel derivative operators as defined in Eq. (5.3) and its digital implementation is shown in Fig. 5.4.

The gravity center calculated along the normal direction is given by

$$
\begin{aligned}
X_c(i, j) &= \frac{\sum_{u=-D/2}^{u=D/2} I(i+u, j+n(i, j)u)u}{\sum_{u=-D/2}^{u=D/2} I(i+u, j+n(i, j)u)} \\
Y_c(i, j) &= \frac{\sum_{u=-D/2}^{u=D/2} I(i-u/n(i, j), j+u)u}{\sum_{u=-D/2}^{u=D/2} I(i-u/n(i, j), j+u)}
\end{aligned}
\tag{5.17}
$$

where D is the width of the laser stripe along the normal direction.

5.3 Range Image Registration with the ICP Algorithm

The goal of registration is to find a transformation between two sets of range images (3D data points) of a given object taken from different coordinate frames in order to represent them all with respect to a reference frame. It has been used in a robot vision system for work object calibration, localization, and shape inspection. For shape inspection, for instance, the measured data points of the work piece need to be aligned with its original CAD model or a reference data points so that comparison can be made for error check. The alignment of two sets of data points is conducted through the registration algorithms.

The iterative closest point (ICP) algorithm (Besl and Mckay, 1992) is a well-known standard algorithm to solve the registration problem. Several improvements to the ICP algorithm have been proposed, such as the iterative closest compatible point (Godin, 1995) and the iterative closest points using invariant features (Sharp, 2002). The ICP algorithm requires a good first approximation in order to converge to a global minimum. However, even if there is considerable overlap, convergence to a global minimum is not guaranteed. The ICP algorithm can also be computationally intensive and time-consuming in its search for corresponding points in two sets of data points. Comparison of various registration algorithms can be found from a review paper (Salvi, 2007).

The ICP algorithm can be primarily classified to two stages for the original algorithm and its variants:

(1) Determination of corresponding points from two sets of data.

(2) Calculation of transform matrix.

5.3.1 Determination of Corresponding Points

Basically there are three approaches to find the corresponding points from two sets of data.

Point to Point Approach

As shown in Fig. 5.21, there are two sets of data points indicated by ***P*** and ***Q***, respectively.

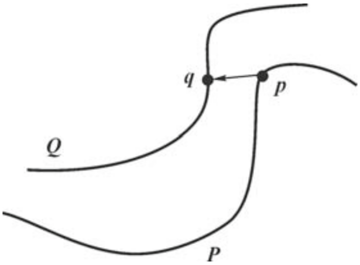


Fig. 5.21. Find the corresponding points by point to point method

P is the source point cloud and ***p*** is a sample point on it. ***Q*** is the destination point cloud and ***q*** is the corresponding point of ***p*** on it. ***q*** is searched and determined based on the criteria that the distance between ***p*** and ***q*** is the shortest one. This method is relatively simple and straight forward. However when one of the data sets has noise it will result in a false matching.

Normal Shooting Approach

The corresponding point can be found by using the intersection of the ray originating at the source point in the direction of the destination point's normal with the destination surface.

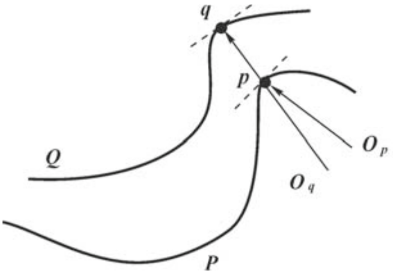


Fig. 5.22. Find the corresponding points using normal shooting method

As shown in Fig. 5.22, $\boldsymbol{p}$ is one point on the source point cloud, $\boldsymbol{O_P}$ is the normal through $\boldsymbol{p}$ for the source point cloud, $\boldsymbol{O_q}$ is the normal for the destination point cloud through $\boldsymbol{p}$. $\boldsymbol{q}$ is the intersection of the ray originating at $\boldsymbol{p}$ in the direction $\boldsymbol{O_q}$. Therefore $\boldsymbol{p}$ is the corresponding point of $\boldsymbol{p}$ on destination point cloud. This method is much faster than the first one. However it is not very accurate for registration.

Point to Surface Approach

As shown in Fig. 5.23, $\boldsymbol{q}$ is the intersection of the ray originating at the source point $\boldsymbol{p}$ in the direction of the source point's normal with the destination surface, $\boldsymbol{S}$ is the tangent plane through $\boldsymbol{q}$. $\boldsymbol{q}$ is the intersection of the ray originating at the source point $\boldsymbol{p}$ in the direction of the S with the S surface. $\boldsymbol{q}$ is the corresponding point of $\boldsymbol{p}$ by using point to surface method. This method is fast, accurate and stable, combining the advantages of previous two methods.

When the registration is applied to two sets of point clouds, one of them needs to become the surface so the point to surface algorithm can be used.

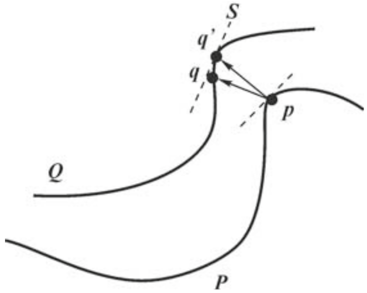


Fig. 5.23. Find the corresponding points using point to surface method

5.3.2 Calculation of Transformation Matrix

The ICP method is used to obtain a rigid body transformation by minimizing the distance between point correspondences, known as closest point. When an initial estimation is known, all the points are transformed to a reference system applying the Euclidean motion. Then every point in the source point cloud is taken into consideration to search for its closest point in the destination point cloud, so that the distance between these correspondences is minimized, and the process is iterated until the following error converges

$$E(\boldsymbol{R},\boldsymbol{t}) = \frac{1}{N}\sum_{k=1}^{N}\left\|\boldsymbol{q}_k - (\boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t})\right\|^2 \tag{5.18}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ are a pair of corresponding points from the source and destination point cloud, respectively. ***R*** and ***t*** are the rotation matrix and translation vector of the transformation matrix.

This is a nonlinear optimization problem. Given two sets of corresponding data points $(\boldsymbol{p}_k, \boldsymbol{q}_k)$ $k=1$, ..., *N*, we want to find ***R*** and ***t*** to minimize the objective function *E*(***R***,***t***). There are several numerical approaches to solve this kind of optimization problem such as the steepest descent method and Newton-Raphson method (please see Appendix for detail).

For a closed-form solution of the transformation matrix, four methods are reported, including SVD method (Arun, 1987), orthonormal matrix method (Horn, 1988), unit quaternion method (Horn, 1987), and dual quaternion method (Walker, 1991). They all have almost the same accuracy. SVD and unit quaternion methods are more stable than orthonormal matrix and dual quaternion methods. However orthonormal matrix and dual quaternion methods have higher efficiency in terms of actual processing time. We will give a brief discussion for a unit quaternion based method. A unit quaternion is a four component vector $\boldsymbol{v} = [q_0 \; q_1 \; q_2 \; q_3]^{\mathrm{T}}$ where $q_0 \geq 0$, $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. It describes a rotation axis and an angle to rotate around that axis.

First calculate the centroid vectors of the source and destination point clouds

$$\boldsymbol{\mu}_p = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{p}_i$$
$$\boldsymbol{\mu}_q = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{q}_i \tag{5.19}$$

Then two new sets of data points are defined by subtracting the centroid vectors

$$\boldsymbol{p} \; = \boldsymbol{p} - \boldsymbol{\mu}_p$$
$$\boldsymbol{q} \; = \boldsymbol{q} - \boldsymbol{\mu}_q \tag{5.20}$$

A 4×4 cross-covariance matrix ***S*** is constructed whose elements are combinations of sums of products of corresponding coordinates of the points

$$\boldsymbol{S} = \begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix} \tag{5.21}$$

where $S_{xx} = \sum_{i=1}^{N} x'_{p,i} x'_{q,i}$, $S_{xy} = \sum_{i=1}^{N} x'_{p,i} y'_{q,i}$ and so on; and $(x'_{p,i}, y'_{p,i}, z'_{p,i})$ and $(x'_{q,i}, y'_{q,i}, z'_{q,i})$ are the coordinates of the corresponding point pair from those two set of points as indicated in Eq. (5.20). To find the eigen-values, an equation has to be solved whose coefficients are sums of products of the elements of the matrix. The minimum number of the corresponding point pairs required to determine the transformation is three, and the points should be non-collinear.

The unit eigen-vector $\boldsymbol{q}_R = [q_0\ q_1\ q_2\ q_3]^{\mathrm{T}}$ corresponding to the maximum eigen-values of the cross-covariant matrix $\boldsymbol{S}$ is selected as the optimal rotation expressed in quaternion. Therefore rotation matrix can be calculated by using its quaternion notation as

$$\boldsymbol{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{5.22}$$

Therefore the translation vector $\boldsymbol{t}$ is calculated by the following equation

$$\boldsymbol{t} = \boldsymbol{\mu}_q - \boldsymbol{R}\boldsymbol{\mu}_p \tag{5.23}$$

The transformation is applied to the first set of data points and the process is repeated until distances between corresponding points decrease below a threshold. ICP obtains good results even in the presence of Gaussian noise.

Illustrations of the two kinds of registration processes are shown in Fig. 5.24 and Fig. 5.25 for the registration of a point cloud with a CAD model, and the registration of two sets of point clouds, respectively.

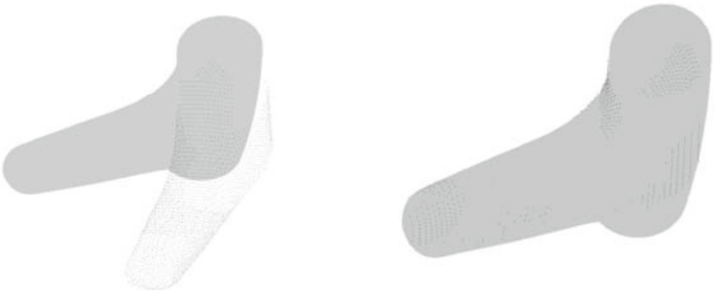


Fig. 5.24. Registration process with a point cloud and a CAD model. (a) A point cloud and a CAD model before registration; (b) The CAD model and point cloud are superimposed after registration

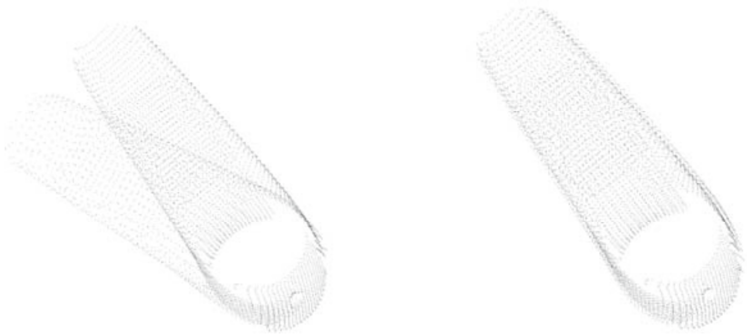


Fig. 5.25. Registration process with two point clouds. (a) Two point clouds before registration; (b) Two point clouds are superimposed after registration

References

Arun KS, Huang TS, Blostein SD (1987).Least-Squares fitting of two 3-D point sets. IEEE Transactions on Pattern Analysis and Machine intelligence, 9(5):698-700.

Bazen AM, Gerez SH (2002) Systematic methods for the computation of the directional fields and singular points of fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7):905-918.

Besl P., Mckay N (1992) A method for registration of 3-d shapes. IEEE Transactions on Pattern Analysis and machine intelligence, 14(2): 239-256.

Canny JA (1986) Computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679-698.

Deriche R (1987) Using Canny's criteria to derive a recursively implemented optimal edge detector. International Journal of Computer Vision, 1:167-187.

Faugeras O (1993) Three-Dimensional Computer Vision. MIT Press, London.

Godin G, Boulanger P (1995) Range image registration through viewpoint invariant computation of curvature. In: International Archives of Photogrammetry and Remote Sensing, Zurich, Switzerland, 30(5/W1): 170-175.

Gonzalez R, Woods R (1992) Digital Image Processing, Addison Wesley, 414-428.

Heikkila J, Silven O (1997) A four-step calibration procedure with implicit image correction. IEEE Computer Society. Conference on Computer Vision and Pattern Recognition pp 1106.

Horn B (1987) Closed-Form solution of absolute orientation using unit quaternions. Journal of the Optical Society of America A, 4:629-642.

Horn B, Hilden HM, Negahdaripour S (1988). Closed form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America A,5(7):1127-1135.

Pedersini F, Sarti A, Subaro S (1997) Estimation and compensation of sub pixel edge location error. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI, 19 (11): 1278-1284.

Salvi J, Matabosch C, Fofi D, and Forest J (2007) A review of recent range image registration methods with accuracy evaluation. Image and vision Computing, 25(5):578-596.

Sharp GC, Lee SW, Wehe DK (2002) ICP registration using invariant features. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):90-102.

Sui LS, Li B, Jiang ZD (2003) Research on method of laser knife center position extracting based on NURBS interpolation. Chinese Journal of Lasers, 30(10): 933-936.

Walker MW, Shao L, Volz RA (1991). Estimating 3-D location parameters using dual number quaternions. CVGIP: Image Understand, 54(3):358-367.

Wu QY, Su XY, Song F(2006) A pruning algorithm for the skeleton of line-structure-light pattern. Journal of Sichuan University (Engineering Science Edition), 38(5):148-151 (In Chinese).

Wu QY, Su XY, Li JZ and Hui B (2007)A New Method for Extracting the Centre-Line of Line Structure Light-Stripe. Journal of Sichuan University (Engineering Science Edition), 39(4):151-155 (In Chinese).

Xiong HY, Zong ZJ, Chen CH (2009) Accurately extracting full resolution centers of structured light stripe. Optics and Precision Engineering, 17(5): 1057-1062 (In Chinese).

6

Robot Kinematic Calibration

When an industrial robot is integrated with a laser sensor, it can be used as a coordinate measuring machine. We call the system a robot coordinate measuring machine (RCMM). To have RCMM achieve its highest possible accuracy, a trio of calibrations are introduced. In order for the robot to use a vision system to measure the 3D pose of an object relative to its own base, it is necessary to know the relative position and orientation between the hand and the robot base (robot kinematic calibration), between the eye and the hand (TCP calibration), and between the object and the eye (sensor calibration). For a laser sensor based robot vision system we have addressed its component vision system in previous chapters. We discussed the formation and calibration of the vision system and calibration of its TCP position. Robot calibration is a natural extension of TCP calibration.

In this chapter, we first provide an introduction to robot calibration in Section 6.1. We address the general model function of robots and review the D-H model in Section 6.2. Determination of independent error parameters is discussed in Section 6.3. Error budget analysis and error parameters solutions are presented in Sections 6.4 and 6.5, respectively. To calibrate the error parameters separately, a circular fitting based robot calibration is presented in Section 6.6. In Section 6.7, detailed discussion of TAU robot calibration, including forward and inverse kinematic modeling with and without all error parameters, and the Jacobian matrix with all error parameters, is presented. Simulation and experimental results are also presented.

6.1 Background

Robot kinematic calibration is defined as a technique or process by which the accuracy of a robot manipulator can be enhanced through modification of the control software. Kinematic calibration is a critical issue in industrial robot applications. It provides the foundation for accurate kinematic control. Position

Z. Gan et al., *Visual Sensing and its Applications*

error mostly arises for three reasons: kinematic error including manufacturing and assembly error, compliance (deformation) error and dynamic error. Each of them roughly contributes one third of the total error. Kinematic error is also the base error when considering the other two errors.

An overview of robot calibration can be found in (Elatta, 2004; Roth, 1987). The typical kinematic calibration used is described in the follwoing procedures. First, for a robot or any other structure, its kinematic model is formulated mathematically based on its mechanical structure. Then all factors that affect kinematic accuracy are parameterized and put into equations. The measured poses are selected systematically. This process is very critical, as it significantly affects the parameter errors that contribute to the robot pose errors. Afterwards, robot model parameters are estimated based on the poses by using a numerical method such as singular value decomposition (SVD) or nonlinear optimization tools. The new robot model parameters are inserted into the kinematic program to update all parameters.

In summary, robot calibration consists of four phases:

(1) Selection of a suitable robot kinematic model.

(2) Measurement of robot end-effector's poses (position or orientation or both).

(3) Estimation of the robot model parameters.

(4) Compensation of robot pose errors by modifying the kinematic parameters.

In terms of modeling, a kinematic model is a mathematical description of the geometry and motion of a robot. A number of different approaches have been developed for the kinematic modeling of a robot. The most popular and effective one is the D-H model (Denavit, 1955). D-H modeling of a robot has been studied extensively and its limitations are discussed (Tsai, 1999; Raghavan, 1993; Abderrahim, 2000). The singularity problem of the D-H model has been identified (Hayati, 1983) and the model is modified (Ibara, 1986) to overcome its drawbacks. Apart from the D-H model, other models have been established to represent robot kinematics, including the S-model (Stone, 1986), which uses 6 parameters for each link, and the Zero-reference model (Mooring, 1983) which does not use a common normal as a link parameter to avoid model singularity.

On the subject of forward kinematics, the focus has been on finding closed form solutions based on various robotic configurations, and numerical solutions for difficult configurations of robots (Dhingra, 2000a; Dhingra, 2000b; Shi, 1994; Didrit, 1998; Zhang, 1991; Nanua, 1990; Sreenivasan, 1994; Griffis, 1989; Lin, 1992). On the subject of error analysis, forward solutions and the Jacobian matrix are used to obtain errors (Wang, 1993; Gong, 2000a; Patel, 2000).

Typically, in the robot kinematic calibration process, absolute measurement of the robot end-effector is needed by using an external measurement tool like a CMM or a high precision laser tracker for the purpose of kinematic identification (Ye, 2006; Alici, 2005). When the robot is integrated with a sensor, like a vision system, a relative calibration is applied that is based on the measurement of distance rather than the absolute position measurement in the reference frame (Gong, 2000b; Zhuang, 1996; Meng, 2007). The unique feature of this calibration is that it can calibrate the robot without calibrating the transformation from the world coordinate system to the robot base coordinate system. Robot kinematic calibration can be done in conjunction with the TCP calibration.

For the vision based robot system, Zhuang (1995) proposed a method that calibrates a robot and a monocular camera simultaneously. Unlike the Tsai and Lenz's method (Tsai, 1989) that calibrates robot and camera individually, Zhuang's algorithm solves the kinematic parameter of the parameters of the robot and camera in one stage, thus eliminating error propagation and improving noise sensitivity.

6.2 Model Function of Robots

The general model of the robot kinematics is represented by the function

$$\boldsymbol{X} = f\left(\boldsymbol{\theta}, \boldsymbol{\rho}\right) \tag{6.1}$$

where $\boldsymbol{X} = (X,\ Y,\ Z,\ \alpha,\ \beta,\ \gamma)^{\mathrm{T}}$ is the position and orientation of the robot TCP pose; $\boldsymbol{\theta}$ is the vector of the robot joint readings; $\boldsymbol{\rho}$ represents the vector of error parameters.

Due to the parameter errors, the calculated robot TCP positions deviate from their actual positions. This can be described by the robot error model that relates the parameter error with the robot end-effector position and orientation error. Assume that there are n error parameters. At the i-th robot measurement pose we have

$$(\delta \boldsymbol{X})_i = \boldsymbol{J}_i \delta \boldsymbol{\rho} \quad (i = 1,\ 2,\ \ldots,\ m) \tag{6.2}$$

where m is the number of measurements; $\delta \boldsymbol{X} = (\delta X,\ \delta Y,\ \delta Z,\ \delta\alpha,\ \delta\beta,\ \delta\gamma)^{\mathrm{T}}$ is the error vector, that is the difference between measured value and calculated value with error parameters set to zero; $\boldsymbol{J}_i$ is the error model Jacobian that is defined as

$$\boldsymbol{J}_i = \begin{bmatrix} \dfrac{\partial f_x(\theta)}{\partial \rho_1} & \dfrac{\partial f_x(\theta)}{\partial \rho_2} & \cdots & \dfrac{\partial f_x(\theta)}{\partial \rho_n} \\ \vdots & \vdots & \cdots & \vdots \\ \dfrac{\partial f_\gamma(\theta)}{\partial \rho_1} & \dfrac{\partial f_\gamma(\theta)}{\partial \rho_2} & \cdots & \dfrac{\partial f_\gamma(\theta)}{\partial \rho_n} \end{bmatrix} \tag{6.3}$$

and $\delta \boldsymbol{\rho} = (\delta\rho_1,\ \delta\rho_2,\ \ldots,\ \delta\rho_n)^{\mathrm{T}}$ is the vector of error increments.

For m robot measurements, Eq. (6.2) can be rewritten as a compact form

$$\boldsymbol{E} = \boldsymbol{J} \delta \boldsymbol{\rho} \tag{6.4}$$

where

$$\boldsymbol{E}_{6m\times 1}=\begin{pmatrix}(\delta \boldsymbol{X})_1\\(\delta \boldsymbol{X})_2\\\vdots\\(\delta \boldsymbol{X})_m\end{pmatrix},\ \boldsymbol{J}_{6m\times n}=\begin{pmatrix}\boldsymbol{J}_1\\\boldsymbol{J}_2\\\vdots\\\boldsymbol{J}_m\end{pmatrix},\ \delta\boldsymbol{\rho}_{n\times 1}=\begin{pmatrix}\delta\rho_1\\\delta\rho_2\\\vdots\\\delta\rho_n\end{pmatrix} \tag{6.5}$$

For different robot kinematics models there are different forms of $\boldsymbol{J}$ and $\delta\boldsymbol{\rho}$.

For a serial robot, the D-H model is used for the following purposes:

(1) Fully describing the kinematic positional relationship among all the links and joints.

(2) Accurately and easily integrating the error model into a full parameter model.

(3) Standardizing and parameterizing the model to establish a dynamic coupling control model.

With the parameters defined in Fig. 6.1, the D-H model transformation matrix for an ABB robot can be obtained as follows

$$\begin{aligned}\boldsymbol{A}_i &= \mathrm{Rot}(Z,\ \theta_i)\mathrm{Trans}(Z,\ d_i)\mathrm{Trans}(X,\ a_i)\mathrm{Rot}(X,\ \alpha_i)\\ &=\begin{bmatrix}\cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i\\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i\\ 0 & \sin\alpha_i & \cos\alpha_i & d_i\\ 0 & 0 & 0 & 1\end{bmatrix}\end{aligned} \tag{6.6}$$

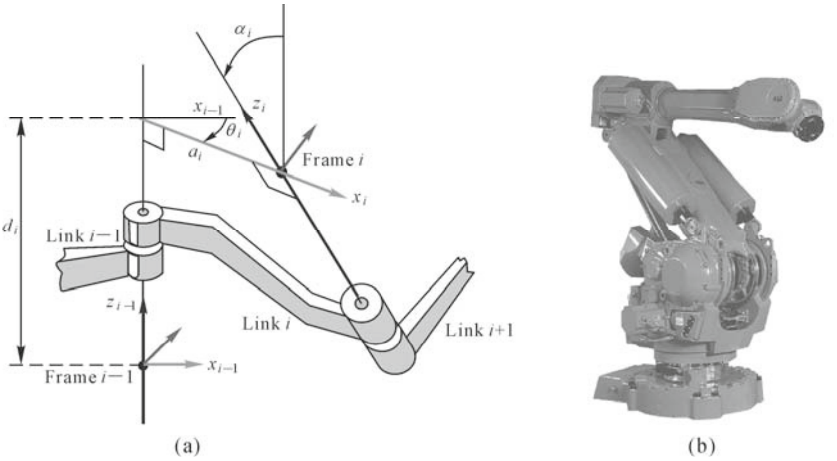


Fig.6.1. (a) Parameter definition of the D-H model; (b) an ABB serial robot

As an example, D-H parameters of ABB 4400 are listed in Table 6.1.

Table 6.1 D-H parameters of ABB 4400 robot

i	a_i	α_i	d_i	θ_i
1	0.2	−90	0.68	0
2	0.89	0	0	−90
3	0.15	−90	0	0
4	0	−90	0.88	180
5	0	90	0	0
6	0	0	0.14	0

6.3 Determination of Independent Error Parameters Using SVD Method

To obtain those parameters in the error model in Eq. (6.4) that are linearly dependent and those parameters that are difficult to observe for certain sets of joint angles, the Jacobian needs to be analyzed. A powerful tool of SVD is used. For different error parameters ρ_i $(i = 1,\ 2,\ ...,\ n)$ they have different scales. Before performing the SVD the Jacobian should be scaled. The matrix is scaled with respect to columns, i.e

$$a_{m,n} = \boldsymbol{a}_{m,n} \Big/ a_{\max,n}\text{; } a_{\max,n} = \max(\boldsymbol{a}_{m,n}) \tag{6.7}$$

where $a_{m,\,n}$ is a Jacobian element. This operation is performed for all elements in the matrix. The vector of scaling factors $a_{\max,\,n}$ is saved for later use during the identification phase.

To ensure that the Jacobian matrix is not singular, the condition number is calculated. The condition number of a matrix is defined as cond $(\boldsymbol{J})=\sigma_{\max}/\sigma_{\min}$, where σ are the singular values. If $\sigma_{\min}$ is close to zero the Jacobian is practically singular. This implies that model redundancies exist or that some parameters are difficult to observe. It is therefore necessary to decrease the number of error parameters (i.e. reduce number of columns in the Jacobian). A way of determining which parameters that are redundant is to investigate the singular vectors. The SVD of the Jacobian $\boldsymbol{J}$ is

$$\left.\begin{aligned}
&\boldsymbol{J}_{m\times n} = \boldsymbol{U}_{m\times m}\boldsymbol{S}_{m\times n}\boldsymbol{V}^{\mathrm{T}}_{n\times n}\\
&\boldsymbol{U} = (\boldsymbol{u}_1,\ \boldsymbol{u}_2 ... \boldsymbol{u}_m),\ \boldsymbol{V} = (\boldsymbol{v}_1,\ \boldsymbol{v}_2 ... \boldsymbol{v}_n)\\
&\boldsymbol{S} = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix},\ \ \Sigma_r = \mathrm{Diag}(\sigma_1,\ \sigma_{2,} ... \sigma_p)\\
&p = \min(m,\ n),\ \ \sigma_1 \geq \sigma_2 ... \geq \sigma_p \geq 0\\
&\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{I}_m,\ \ \boldsymbol{V}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{I}_n
\end{aligned}\right\} \tag{6.8}$$

Suppose that the rank of $\boldsymbol{J}$ is r, we have

$$\left.\begin{aligned} &\boldsymbol{J} = \begin{bmatrix} \boldsymbol{U}_r & \boldsymbol{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_r^{\mathrm{T}} \\ \boldsymbol{V}_2^{\mathrm{T}} \end{bmatrix} = \boldsymbol{U}_r \Sigma_r \boldsymbol{V}_r^{\mathrm{T}} \\ &\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0 \text{ and } \sigma_{r+1} = \ldots = \sigma_p = 0 \end{aligned}\right\} \tag{6.9}$$

An investigation of the last column of the $\boldsymbol{V}$ vector will reveal that some elements are dominant in order of magnitude. This implies that corresponding columns in the Jacobian matrix are linearly dependent. The work of reducing the number of error parameters must continue until no singularities exist and the condition number has reached an acceptable value.

The following procedure is used to identify redundant error parameters and trim the Jacobian matrix until its condition number has reached an acceptable value (approximately 50):

(1) Scale the Jacobian matrix $\boldsymbol{J}$ with respect to columns;

(2) Decompose $\boldsymbol{J}$ into $\boldsymbol{U}$, $\boldsymbol{S}$, and $\boldsymbol{V}$ matrices;

(3) Calculate condition number of $\boldsymbol{J}$;

(4) Look into $\boldsymbol{S}$ matrix and search for the lowest singular value (normally it is located in last row/column);

(5) Identify a column of $\boldsymbol{J}$ corresponding to lowest singular value by searching the last column of $\boldsymbol{V}$ matrix and order of element that has the highest $\boldsymbol{V}$ element value. The order represents column of $\boldsymbol{J}$ that needs to be eliminated. For example, if the second element in the last column of $\boldsymbol{V}$ matrix is the largest one, it indicates that the second column of $\boldsymbol{J}$ matrix contributes most to the singular value. This column of $\boldsymbol{J}$ matrix corresponds to the change in the cost function with respect to the second error parameter;

(6) Eliminate that particular $\boldsymbol{J}$ column.

After independent parameters are identified and then solved, those trimmed parameters in the previous procedure can be solved by inserting the values of the independent parameters into Eq. (6.4).

6.4 Error Budget Analysis

When the SVD is completed and a linearly independent set of error model parameters determined, the error budget can be determined. The error budget analysis is used to determine the error tolerance of all the robot error parameters for a given end effector error. The mathematical description of the error budget is as follows

$$\boldsymbol{J} = \boldsymbol{USV}^{\mathrm{T}} \Rightarrow \mathrm{d}\boldsymbol{X} = \boldsymbol{J}\mathrm{d}\boldsymbol{\rho} = \boldsymbol{USV}^{\mathrm{T}}\mathrm{d}\boldsymbol{\rho} \Rightarrow \boldsymbol{U}^{\mathrm{T}}\mathrm{d}\boldsymbol{X} = \boldsymbol{SV}^{\mathrm{T}}\mathrm{d}\boldsymbol{\rho} \Rightarrow \mathrm{d}\boldsymbol{\rho} = \boldsymbol{V}\frac{1}{\boldsymbol{S}}\boldsymbol{U}^{\mathrm{T}}\mathrm{d}\boldsymbol{X} \tag{6.10}$$

Thus, if d$\boldsymbol{X}$ is given as the accuracy of the robot end effector, the error budget d$\boldsymbol{\rho}$ can be determined.

6.5 Solving the Error Parameters

When the analysis phase is completed and a linearly independent set of error model parameters determined, the error parameters for a specific manipulator can be determined by using SVD analysis.

From Eq. (6.4) we have

$$\begin{aligned} &\boldsymbol{J}\delta\boldsymbol{\rho} = \boldsymbol{E} \Rightarrow \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\delta\boldsymbol{\rho} = \boldsymbol{E} \Rightarrow \boldsymbol{S}\boldsymbol{V}^{\mathrm{T}}\delta\boldsymbol{\rho} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{E} \\ &\Rightarrow \boldsymbol{V}^{\mathrm{T}}\delta\boldsymbol{\rho} = \frac{1}{\boldsymbol{S}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{E} \Rightarrow \delta\boldsymbol{\rho} = \boldsymbol{V}\frac{1}{\boldsymbol{S}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{E} \end{aligned} \tag{6.11}$$

By updating the error parameters with $\delta\boldsymbol{\rho}_1$, the new error vector and a new Jacobian can be determined and thus $\delta\boldsymbol{\rho}_2$ can be calculated. This iteration continues until a certain convergence criterion is met.

6.6 Circle Fitting Based Calibration

As described in previous sections, error parameters are calculated based on a nonlinear optimization method by minimizing the error vector. They are calibrated simultaneously. Usually the numerical optimization involves high order nonlinear fitting, which is difficult for a convergence when all parameters are included. Depending on the robot pose during the measurement, some parameters are not able to be calibrated since there is no significant contribution to the error vector.

Instead of calibrating all the error parameters together, an individual joint is considered and calibrated separately. Therefore, real physical errors can be locally presented. Since a whole high-order nonlinear fitting breaks down into several independent loops, all or most parameters on a joint can be included. Also, the order and degree of nonlinear fitting is reduced.

Building up an accurate kinematic model with all error parameters is very difficult, especially if the model must work in an industrial environment. The first step towards a kinematic solution is to establish the frame system by attaching one frame to each rigid body at its joint position, as described by, for example, the D-H model. Then, between every two adjacent frames, parameters are used to describe their relationship. The frame system can be reconstructed directly from the measurement and then their parameters are obtained. The reason why this is possible and applicable in industrial robot calibration is because most industrial robots use rotational motors as actuators and are connected by revolution joints. The path in motion of any point on a revolution rigid body is circular. Then, by taking measurements on the circular path, its rotational axis can be found by curve fitting. Once all rotational axes are obtained, the frame system can be reestablished according to certain selected rules, such as D-H model. The following is the procedure for an industrial robot with rotational actuators and joints.

(1) Measurement. Take one set of measurements while only driving one joint and keeping the others fixed. This should give a circular or arc path in 3D. Repeat this for each joint.

(2) Determination of rotational axes. For each set of measurements, do circular fitting to get the orientation vector and center and radius of the circle, as shown in Fig. 6.2.

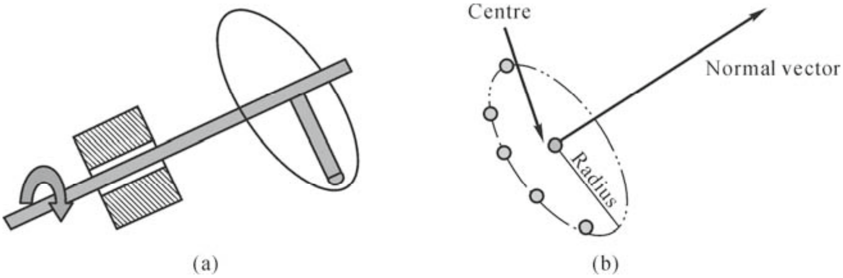


Fig. 6.2. Determination of rotational axis. (a) A fixed point on the rotational part is followed and measured; (b) The circle fitting is used to determine the normal vector of the circle plane that is the rotational axis

(3) Setup of frame system. For the first frame, the Z axis is aligned with the rotational axis. Although the X axis can be arbitrary, usually it is set up in such a way that it is aligned in the direction where the joint 1 angle is zero and the Y axis is determined by the right hand rule. The origin of the first frame is located along the orientation vector where its Z position is zero. For the second frame, the Z axis is aligned with the rotational axis. The X axis is setup by the D-H model rule, e.g., it is the common normal of the first and second frames' Z axis. The intersection point of the common normal and the second frame's Z axis is the origin of the second frame. The Y axis is governed by the right hand rule. The rest of the frames are set up in the same way as the second frame.

A validation test is performed on an ABB IRB6400-25 robot, a well used standard 6-axis industrial robot. It has 6-DOF in its workspace and is driven by 6 rotational motors through gearboxes. The 4-bar linkage used to pass the driving motion to joint 3 is not taken into consideration as a validation test. Joint 3 is treated as being directly driven by the motor like other joints. A 3D digitizer Romer 3000i with an accuracy of 30 mm is used as the measurement tool. To calibrate the rotational axis by using the digitizer, a fixed point on the rotational part is followed and its position is measured for different rotation angles. The initial position of the robot is 0°, 0°, 0°, 0°, 90°, −90° of encoder readings for joints 1 to 6, respectively. Each joint rotates 2° for each measurement point and up to 30° while other joints remain at their initial angles.

The D-H model is used as the rule when the frame system is setup. The D-H model uses 4 parameters, including 2 translational parameters *a* and *d*, and 2 rotational parameters α and θ to describe a 1 DOF rotational or translational joint. Fig. 6.3 shows the frame system created from the measurement.

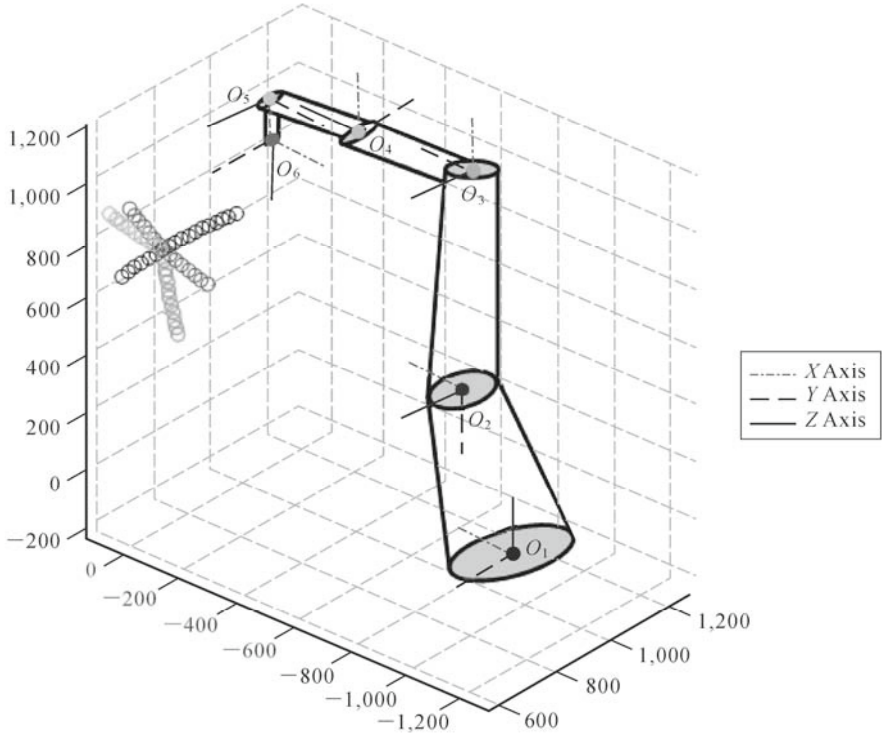


Fig. 6.3. Robot D-H model frame system created from the measurement

The preliminary test results are shown in Table 6.2.

Table 6.2 D-H model parameters for nominal value and calculated value

	a	d	α	θ
	240	800	−90	0
	1,050	0	0	−90
Nominal	225	0	−90	0
	0	1,175	−90	180
	0	0	90	0
	252.46	39.61	−89.57	−16.17
	1,056.5	5.96	−0.34	−89.64
Calculated	224.55	−12.69	−89.63	−0.43
	4.21	1,176.2	−89.84	179.23
	−2.76	5.64	88.71	2.33

The RMS difference between measurement data and calculated data based on the nominal value is 12.23 mm, 18.29 mm, 5.67 mm, in *X*, *Y* and *Z* directions, respectively. The RMS difference between measurement data and calculated data based on calibrated value is 0.99 mm, 0.91 mm, 2.55 mm in *X*, *Y* and *Z* directions, respectively. For comparison, both results before and after robot parameter calibrations are listed in Table 6.3.

Table 6.3 RMS error comparison before and after calibration

	X	*Y*	*Z*
RMS difference between measurement and calibrated value	12.23	18.29	5.67
RMS difference between measurement and nominal value	0.99	0.91	2.55

6.7 TAU Parallel Robot Calibration

TAU parallel configuration (Cui, 2005) is rooted in a series of inventions (Brogangrdh, 2002a; Brogangrdh, 2002b). The configuration of the robot simulates the shape of τ” like the name of the Delta after the “Δ” shape configuration of another parallel robot. As shown in Fig. 6.4, the basic TAU configuration consists of three driving axes, three arms, six linkages, twelve joints and a moving (tool) plate. There are six chains connecting the main column to the end-effector in the TAU configuration. The TAU robot is a typical 3/2/1 configuration. There are three parallel and identical links of lower arm 1 and another two parallel and identical links of lower arm 2. Six chains will be used to derive all kinematic equations. Table 6.4 highlights the features of the TAU configuration.

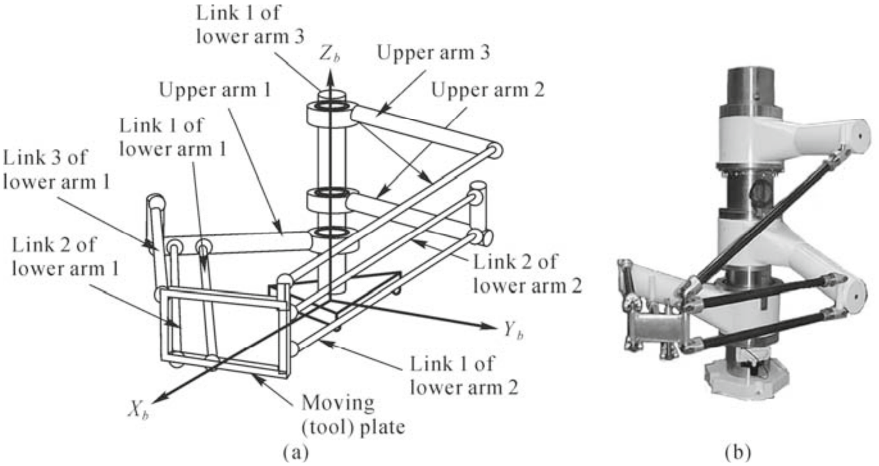


Fig. 6.4. (a) ABB TAU robot configuration; (b) Prototype

Table 6.4 Comparison of TAU and other robots

	Serial robot	Stewart platform	TAU configuration
Stiffness	Low	High	High (simulation)
Accuracy	Low	High	High (simulation)
Workspace	Large	Small	Large
Footprint	Small	Large	Small
Inverse solution in general	Easy	Easy	Difficult
Analytical inverse solution	Easy	Easy	Difficult
Forward solution in general	Easy	Difficult	Easy
Analytical forward solution	Easy	Difficult	Easy

In this Section, the D-H model is used to define the TAU robot and a complete set of parameters are included in the modeling process. Kinematic modeling and error modeling are established with all errors using Jacobian matrix method for the TAU robot. Meanwhile, a very effective Jacobian approximation method is introduced to calculate the forward kinematic problem instead of the Newton-Raphson (N-R) method. It denotes that a closed form solution can be obtained instead of a numerical solution.

A full size Jacobian matrix is used in carrying out error analysis, error budget and model parameter estimation and identification. Simulation results indicate that both the Jacobian matrix and Jacobian approximation method are correct and have an accuracy of micron meters. ADAMS simulation results are used in verifying the established models.

6.7.1 Kinematic Modeling

The D-H Model of TAU Robot

With the parameters defined in Fig. 6.1, the D-H model transformation matrix for the TAU robot can be obtained as follows

$$\boldsymbol{A}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 & -a_i \\ -\cos\alpha_i \sin\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i & -d_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & -\cos\alpha_i \sin\theta_i & \cos\alpha_i & -d_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.12}$$

Inverse Kinematics and Forward Kinematics

For the TAU robot, the inverse kinematics and forward kinematics are relatively simple. Of the six equations of kinematic chains, 3 remain, as shown in Fig. 6.5, based on the condition of parallel and identical links.

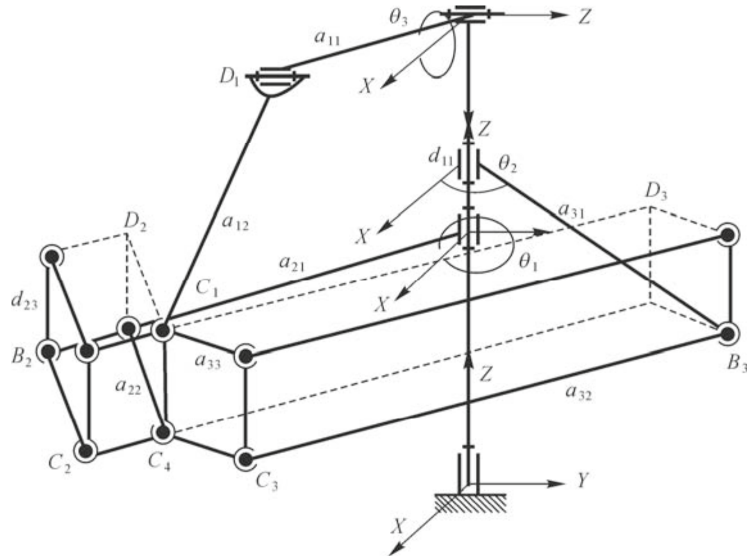


Fig. 6.5. TAU parallel mechanism

Coordinates of D_1 are obtained as

$$\left.\begin{aligned} d_{1x} &= a_{11}\cos((\theta_1+\theta_2)/2)\cos\theta_3 \\ d_{1y} &= a_{11}\cos((\theta_1+\theta_2)/2)\sin\theta_3 \\ d_{1z} &= -a_{11}\sin\theta_1 + d_{11} \\ c_{1x} = p_x,\ c_{1y} &= p_y,\ c_{1z} = p_z \\ \mathrm{dist}(d_1 - c_1) &= a_{12} \end{aligned}\right\} \tag{6.13}$$

where p_x, p_y, and p_z are the coordinates of c_1.
Coordinates of D_2 are obtained as

$$\left.\begin{aligned} d_{2x} &= a_{21}\cos\theta_1 \\ d_{2y} &= a_{21}\sin\theta_1 \\ d_{2z} &= d_{21} + d_{23} \\ c_{2x} = p_x,\ c_{2y} &= p_y,\ c_{2z} = p_z - d_{23} \\ \mathrm{dist}(d_2 - c_1) &= a_{22} \end{aligned}\right\} \tag{6.14}$$

Coordinates of D_3 are obtained as

$$\left.\begin{array}{c} d_{3x} = a_{31}\cos\theta_2 - a_{33}\cos(120+\theta_1) \\ d_{3y} = a_{31}\sin\theta_2 - a_{33}\sin(120+\theta_1) \\ d_{3z} = d_{31} \\ \mathrm{dist}(d_3 - c_1) = a_{32} \end{array}\right\} \quad (6.15)$$

For inverse kinematics, assuming the next expressions

$$\cos\delta = \frac{p_x}{\sqrt{p_x^2 + p_y^2}}, \quad \sin\delta = \frac{p_y}{\sqrt{p_x^2 + p_y^2}} \quad (6.16)$$

and then simply from Eq. (6.14) one can obtain

$$\begin{array}{c} 2a_{21}\sqrt{p_x^2 + p_y^2}\left(\dfrac{p_x}{\sqrt{p_x^2 + p_y^2}}\cos\theta_1 + \dfrac{p_y}{\sqrt{p_x^2 + p_y^2}}\sin\theta_1\right) \\ = a_{21}^2 + (p_x^2 + p_y^2 + p_z^2) - a_{22}^2 \end{array} \quad (6.17)$$

Substitution of Eq. (6.16) by Eq. (6.17) results in

$$\cos(\theta_1 - \delta) = \frac{a_{21}^2 + (p_x^2 + p_y^2 + p_z^2) - a_{22}^2}{2a_{21}\sqrt{p_x^2 + p_y^2}} \quad (6.18)$$

Thus we have

$$\theta_1 = \cos^{-1}\left[\frac{a_{21}^2 + (p_x^2 + p_y^2 + p_z^2) - a_{22}^2}{2a_{21}\sqrt{p_x^2 + p_y^2}}\right] + \delta \quad (6.19)$$

where

$$\delta = \tan^{-1}\left(\frac{p_y}{p_x}\right)$$

Assume that

$$\left.\begin{array}{l} \cos\gamma = \dfrac{p_{\,x}}{\sqrt{p_{\,x}^2 + p_{\,y}^2}}, \sin\gamma = \dfrac{p_{\,y}}{\sqrt{p_{\,x}^2 + p_{\,y}^2}} \\ p_{\,x} = p_x - a_{33}\cos(\theta_1 + 120) \\ p_{\,y} = p_y - a_{33}\sin(\theta_1 + 120) \end{array}\right\} \quad (6.20)$$

Substitution of Eq. (6.20) by Eq. (6.15) results in

$$\theta_2 = \cos^{-1}\left[\frac{a_{31}^2 - a_{32}^2 + (p'^2_x + p'^2_y + p'^2_z) - a_{22}^2}{2a_{31}\sqrt{p_x^2 + p_y^2}}\right] + \gamma \tag{6.21}$$

where

$$\gamma = \tan^{-1}\left(\frac{p_y}{p_x}\right)$$

Substitution of Eq. (6.19) and Eq. (6.21) by Eq. (6.13) results in

$$\theta_3 = \cos^{-1}\left[\frac{a_{11}^2 + p_x^2 + p_y^2 + (p_z - d_{11})^2 - a_{12}^2}{2\sqrt{[a_{11}\cos(\frac{\theta_1+\theta_2}{2}) + a_{11}\sin(\frac{\theta_1+\theta_2}{2})]^2 + (p_z - d_{11})^2}}\right] - \varphi \tag{6.22}$$

where

$$\varphi = \tan^{-1}\left[\frac{p_z - d_{11}}{a_{11}\cos(\frac{\theta_1+\theta_2}{2}) + a_{11}\sin(\frac{\theta_1+\theta_2}{2})}\right]$$

For forward kinematics, it is relatively easy. Subtract Eq. (6.14) from Eq. (6.13) by eliminating the square items $(p_x^2,\ p_y^2,\ p_z^2)$, then do the same procedure to Eqs. (6.14) and (6.15), and finally three linear equations can be obtained. The three length equations are applied to solve inverse and forward problems. A closed form solution can be obtained from the three equations for both inverse and forward problems.

6.7.2 Jacobian Matrix of TAU Robot with All Error Parameters

In error analysis, error sensitivity is represented by the Jacobian matrix. Derivation of the Jacobian matrix can be carried out after all the D-H models are established. For the TAU robot, the 3-DOF kinematic problem will become a 6-DOF kinematic problem. The kinematic problem becomes more complicated. In fact, the error sensitivity is formulated through $\partial x / \partial g_i$, $\partial y / \partial g_i$, $\partial z / \partial g_i$, where *x*, *y*, *z* represent the position of the tool plate and dg_i is the error source for each component. So

$$\mathrm{d}x = \sum_i^N \frac{\partial x}{\partial g_i}\mathrm{d}g_i,\ \mathrm{d}y = \sum_i^N \frac{\partial y}{\partial g_i}\mathrm{d}g_i,\ \mathrm{d}z = \sum_i^N \frac{\partial z}{\partial g_i}\mathrm{d}g_i \tag{6.23}$$

The error model is actually a 6-DOF model, since all error sources have been considered. It includes both the position variables *x*, *y*, *z* and also the rotational angles *α*, *β*, *γ*.

6.7.2.1 Formulation of Forward Jacobian Matrix

From the six kinematic chains, equations established based on D-H models are

$$\left.\begin{aligned} f_1 &= f_1(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}) = 0 \\ f_2 &= f_2(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}) = 0 \\ &\vdots \\ f_6 &= f_6(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}) = 0 \end{aligned}\right\} \tag{6.24}$$

Differentiate all the equations against all the variables x, y, z, α, β, γ and $\boldsymbol{g}$, where $\boldsymbol{g}$ is a vector including all geometric parameters

$$\frac{\partial f_i}{\partial x}\cdot \mathrm{d}x + \frac{\partial f_i}{\partial y}\cdot \mathrm{d}y + \frac{\partial f_i}{\partial z}\cdot \mathrm{d}z + \frac{\partial f_i}{\partial \alpha}\cdot \mathrm{d}\alpha + \frac{\partial f_i}{\partial \beta}\cdot \mathrm{d}\beta + \frac{\partial f_i}{\partial \gamma}\cdot \mathrm{d}\gamma + \sum_j \frac{\partial f_i}{\partial g_j}\cdot \mathrm{d}g_j = 0 \tag{6.25}$$

Rewrite it in the matrix form as

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ & & \cdots & & & \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix}_{6\times 6} \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \\ \mathrm{d}\alpha \\ \mathrm{d}\beta \\ \mathrm{d}\gamma \end{bmatrix}_{6\times 1} = \begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j}\mathrm{d}g_j \\ \cdots \\ \sum_j \frac{-\partial f_6}{\partial g_j}\mathrm{d}g_j \end{bmatrix}_{6\times 1} \tag{6.26}$$

In a compact form, it becomes

$$\boldsymbol{J}_J \mathrm{d}\boldsymbol{X} = \mathrm{d}\boldsymbol{G} \tag{6.27}$$

where

$$\mathrm{d}\boldsymbol{G} = \begin{bmatrix} \sum_j \frac{-\partial f_1}{\partial g_j}\mathrm{d}g_j \\ \cdots \\ \sum_j \frac{-\partial f_6}{\partial g_j}\mathrm{d}g_j \end{bmatrix}_{6\times 1} = \begin{bmatrix} -\frac{\partial f_1}{\partial g_1} & -\frac{\partial f_1}{\partial g_2} & \cdots & -\frac{\partial f_1}{\partial g_N} \\ & & \cdots & \\ -\frac{\partial f_6}{\partial g_1} & -\frac{\partial f_6}{\partial g_2} & \cdots & -\frac{\partial f_6}{\partial g_N} \end{bmatrix}_{6\times N} \begin{bmatrix} \mathrm{d}g_1 \\ \mathrm{d}g_2 \\ \vdots \\ \mathrm{d}g_N \end{bmatrix}_{N\times 1} \tag{6.28}$$

From Eq.(6.28) we have

$$\mathrm{d}\boldsymbol{G} = \boldsymbol{J}_2 \mathrm{d}\boldsymbol{g} \tag{6.29}$$

Substitution of Eq. (6.27) by Eq. (6.29) yields

$$\boldsymbol{J}_1 \mathrm{d}\boldsymbol{X} = \boldsymbol{J}_2 \mathrm{d}\boldsymbol{g} \tag{6.30}$$

That is

$$\mathrm{d}\boldsymbol{X} = (\boldsymbol{J}_1^{-1}\boldsymbol{J}_2)\mathrm{d}\boldsymbol{g} \tag{6.31}$$

The Jacobian matrix is obtained as $\boldsymbol{J}_1^{-1}\boldsymbol{J}_2$

$$\boldsymbol{J}_{\text{forward}} = \boldsymbol{J}_1^{-1}\boldsymbol{J}_2$$

$$= \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} & \dfrac{\partial f_1}{\partial z} & \dfrac{\partial f_1}{\partial \alpha} & \dfrac{\partial f_1}{\partial \beta} & \dfrac{\partial f_1}{\partial \gamma} \\ & & \cdots & & & \\ \dfrac{\partial f_6}{\partial x} & \dfrac{\partial f_6}{\partial y} & \dfrac{\partial f_6}{\partial z} & \dfrac{\partial f_6}{\partial \alpha} & \dfrac{\partial f_6}{\partial \beta} & \dfrac{\partial f_6}{\partial \gamma} \end{bmatrix}^{-1} \begin{bmatrix} -\dfrac{\partial f_1}{\partial g_1} & -\dfrac{\partial f_1}{\partial g_2} & \cdots & -\dfrac{\partial f_1}{\partial g_N} \\ & & \cdots & \\ -\dfrac{\partial f_6}{\partial g_1} & -\dfrac{\partial f_6}{\partial g_2} & \cdots & -\dfrac{\partial f_6}{\partial g_N} \end{bmatrix} \tag{6.32}$$

For a prototype of the TAU robotic design, the dimension of the Jacobian matrix is 6 by 71. An analytical solution can be obtained and is used in our analysis.

6.7.2.2 Jacobian Matrix in Case a Tool is Attached

If a tool is attached to the robot, the end effector of the robot system extends to the TCP of the attached tool. The coordinate of TCP in the robot base frame $(X_p, Y_p, Z_p)^{\mathrm{T}}$ is given by

$$\begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ z_L \\ 1 \end{bmatrix} \tag{6.33}$$

where $(x_L, y_L, z_L)^{\mathrm{T}}$ is the TCP value of the tool, $(r_{11}, \ldots, r_{33})$ and (x, y, z) are the rotation and translation components of $Tool_0$ frame.

Taking the differentiation to Eq. (6.33) we have

$$\begin{bmatrix} \mathrm{d}X_p \\ \mathrm{d}Y_p \\ \mathrm{d}Z_p \end{bmatrix} = \boldsymbol{DR} \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} + \boldsymbol{R} \cdot \begin{bmatrix} \mathrm{d}x_L \\ \mathrm{d}y_L \\ \mathrm{d}z_L \end{bmatrix} + \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \end{bmatrix} \tag{6.34}$$

where

$$\left.\begin{aligned} DR_{i,j} &= R_{i,j}^{\alpha} \cdot \mathrm{d}\alpha + R_{i,j}^{\beta} \cdot \mathrm{d}\beta + R_{i,j}^{\gamma} \cdot \mathrm{d}\gamma \\ R_{i,j}^{\alpha} &= \frac{\mathrm{d}R_{i,j}}{\mathrm{d}\alpha}, R_{i,j}^{\beta} = \frac{\mathrm{d}R_{i,j}}{\mathrm{d}\beta}, R_{i,j}^{\gamma} = \frac{\mathrm{d}R_{i,j}}{\mathrm{d}\gamma} \end{aligned}\right\} \tag{6.35}$$

Rewriting Eq. (6.34) yields

$$\begin{bmatrix} \mathrm{d}X_P \\ \mathrm{d}Y_P \\ \mathrm{d}Z_P \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} \mathrm{d}x \\ \mathrm{d}y \\ \mathrm{d}z \\ \mathrm{d}\alpha \\ \mathrm{d}\beta \\ \mathrm{d}\gamma \end{bmatrix} + \boldsymbol{R} \cdot \begin{bmatrix} \mathrm{d}x_L \\ \mathrm{d}y_L \\ \mathrm{d}z_L \end{bmatrix} \tag{6.36}$$

where

$$M_{ij} = \boldsymbol{DR} \cdot \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \tag{6.37}$$

Substituting Eq. (6.31) by Eq. (6.36) we obtain

$$\begin{bmatrix} \mathrm{d}X_P \\ \mathrm{d}Y_P \\ \mathrm{d}Z_P \end{bmatrix} = \left[\begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \cdot \boldsymbol{J} \quad \boldsymbol{R} \right] \cdot \begin{bmatrix} \mathrm{d}L_1 \\ \mathrm{d}L_2 \\ \vdots \\ \mathrm{d}L_N \\ \mathrm{d}x_L \\ \mathrm{d}y_L \\ \mathrm{d}z_L \end{bmatrix} \tag{6.38}$$

The forward Jacobian matrix with TCP is obtained as

$$J_{\mathrm{TCP}} = \left[\begin{bmatrix} 1 & 0 & 0 & M_{11} & M_{12} & M_{13} \\ 0 & 1 & 0 & M_{21} & M_{22} & M_{23} \\ 0 & 0 & 1 & M_{31} & M_{32} & M_{33} \end{bmatrix} \cdot \boldsymbol{J} \quad \boldsymbol{R} \right] \tag{6.39}$$

6.7.2.3 Formulation of Inverse Jacobian Matrix

Kinematic equations Eq. (6.24) can be rewritten as the following to explicitly include joint angles (θ_1, θ_2, θ_3)

$$\left. \begin{aligned} & f_1 = f_1(\theta_1,\ \theta_2,\ \theta_3,\ x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_1) = 0 \\ & f_2 = f_2(\theta_1,\ \theta_2,\ \theta_3,\ x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_1) = 0 \\ & \vdots \\ & f_6 = f_6(\theta_1,\ \theta_2,\ \theta_3,\ x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_1) = 0 \end{aligned} \right\} \tag{6.40}$$

With the probe as the end effector, its position (x_p, y_p, z_p) in the robot base frame is given by

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = T_{Tool_0} \begin{bmatrix} x_L \\ y_L \\ z_L \end{bmatrix} \tag{6.41}$$

where (x_L, y_L, z_L) is the probe coordinate relative to the robot $Tool_0$ frame. It can be rewritten in the general form

$$\begin{aligned} f_7 &= f_7(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_2) = 0 \\ f_8 &= f_8(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_2) = 0 \\ f_9 &= f_9(x,\ y,\ z,\ \alpha,\ \beta,\ \gamma,\ \boldsymbol{g}_2) = 0 \end{aligned} \tag{6.42}$$

Differentiating all the equations in Eq. (6.40) against all the variables θ_1, θ_2, θ_3, x, y, z, α, β, γ and $\boldsymbol{g}$, one can get

$$\begin{aligned} &\frac{\partial f_i}{\partial \theta_1} \cdot \mathrm{d}\theta_1 + \frac{\partial f_i}{\partial \theta_2} \cdot \mathrm{d}\theta_2 + \frac{\partial f_i}{\partial \theta_3} \cdot \mathrm{d}\theta_3 + \frac{\partial f_i}{\partial x} \cdot \mathrm{d}x + \frac{\partial f_i}{\partial y} \cdot \mathrm{d}y + \frac{\partial f_i}{\partial z} \cdot \mathrm{d}z \\ &+ \frac{\partial f_i}{\partial \alpha} \cdot \mathrm{d}\alpha + \frac{\partial f_i}{\partial \beta} \cdot \mathrm{d}\beta + \frac{\partial f_i}{\partial \gamma} \cdot \mathrm{d}\gamma + \sum_j \frac{\partial f_i}{\partial g_{1_j}} \cdot \mathrm{d}g_{1_j} = 0 \end{aligned} \tag{6.43}$$

It can be rewritten in matrix format

$$\boldsymbol{J}_1 \mathrm{d}\boldsymbol{\theta} + \boldsymbol{J}_2 \mathrm{d}\boldsymbol{X} = -\mathrm{d}\boldsymbol{G}_1 \tag{6.44}$$

where

$$\begin{aligned} \boldsymbol{J}_1 &= \begin{bmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial \theta_3} \\ & \cdots & \\ \frac{\partial f_6}{\partial \theta_1} & \frac{\partial f_6}{\partial \theta_2} & \frac{\partial f_6}{\partial \theta_3} \end{bmatrix}, \ \mathrm{d}\boldsymbol{G}_1 = \begin{bmatrix} \sum_j \frac{\partial f_1}{\partial g_{1_j}} \mathrm{a}g_{1_j} \\ \cdots \\ \sum_j \frac{\partial f_6}{\partial g_{1_j}} \mathrm{d}g_{1_j} \end{bmatrix} \\ \boldsymbol{J}_2 &= \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} & \frac{\partial f_1}{\partial \alpha} & \frac{\partial f_1}{\partial \beta} & \frac{\partial f_1}{\partial \gamma} \\ & & \cdots & & & \\ \frac{\partial f_6}{\partial x} & \frac{\partial f_6}{\partial y} & \frac{\partial f_6}{\partial z} & \frac{\partial f_6}{\partial \alpha} & \frac{\partial f_6}{\partial \beta} & \frac{\partial f_6}{\partial \gamma} \end{bmatrix} \end{aligned} \tag{6.45}$$

Differentiating all the equations in Eq. (6.42) against all the variables x, y, z, α, β, γ and $\boldsymbol{g}$, one can get

$$\boldsymbol{J}_3 \mathrm{d}\boldsymbol{X} = -\mathrm{d}\boldsymbol{G}_2 \tag{6.46}$$

where

$$\boldsymbol{J}_3 = \begin{bmatrix} \frac{\partial f_7}{\partial x} & \frac{\partial f_7}{\partial y} & \frac{\partial f_7}{\partial z} & \frac{\partial f_7}{\partial \alpha} & \frac{\partial f_7}{\partial \beta} & \frac{\partial f_7}{\partial \gamma} \\ \frac{\partial f_8}{\partial x} & \frac{\partial f_8}{\partial y} & \frac{\partial f_8}{\partial z} & \frac{\partial f_8}{\partial \alpha} & \frac{\partial f_8}{\partial \beta} & \frac{\partial f_8}{\partial \gamma} \\ \frac{\partial f_9}{\partial x} & \frac{\partial f_9}{\partial y} & \frac{\partial f_9}{\partial z} & \frac{\partial f_9}{\partial \alpha} & \frac{\partial f_9}{\partial \beta} & \frac{\partial f_9}{\partial \gamma} \end{bmatrix}, \ \mathrm{d}\boldsymbol{G}_2 = \begin{bmatrix} \sum_j \frac{\partial f_7}{\partial g_{2j}} \mathrm{d}g_{2j} \\ \sum_j \frac{\partial f_8}{\partial g_{2j}} \mathrm{d}g_{2j} \\ \sum_j \frac{\partial f_9}{\partial g_{2j}} \mathrm{d}g_{2j} \end{bmatrix} \tag{6.47}$$

From Eq. (6.44) we have

$$\mathrm{d}\boldsymbol{X} = -\boldsymbol{J}_2^{-1} \mathrm{d}\boldsymbol{G}_1 - \boldsymbol{J}_2^{-1} \boldsymbol{J}_1 \mathrm{d}\boldsymbol{\theta} \tag{6.48}$$

Substituting Eq. (6.48) by Eq. (6.46) one can get

$$\boldsymbol{J}_3 \cdot \boldsymbol{J}_2^{-1} \cdot \boldsymbol{J}_1 \cdot \mathrm{d}\boldsymbol{\theta} = \boldsymbol{J}_5 \cdot \begin{bmatrix} \mathrm{d}x_L \\ \mathrm{d}y_L \\ \mathrm{d}z_L \end{bmatrix} - \boldsymbol{J}_3 \cdot \boldsymbol{J}_2^{-1} \boldsymbol{J}_4 \cdot \begin{bmatrix} \mathrm{d}g_1 \\ \mathrm{d}g_2 \\ \vdots \\ \mathrm{d}g_N \end{bmatrix} \tag{6.49}$$

where

$$\boldsymbol{J}_4 = \begin{bmatrix} \frac{\partial f_1}{\partial g_1} & \cdots & \frac{\partial f_1}{\partial g_N} \\ & \cdots & \\ \frac{\partial f_6}{\partial g_1} & \cdots & \frac{\partial f_6}{\partial g_N} \end{bmatrix}, \ \boldsymbol{J}_5 = \begin{bmatrix} \frac{\partial f_7}{\partial x_L} & \frac{\partial f_7}{\partial y_L} & \frac{\partial f_7}{\partial z_L} \\ \frac{\partial f_8}{\partial x_L} & \frac{\partial f_8}{\partial y_L} & \frac{\partial f_8}{\partial z_L} \\ \frac{\partial f_9}{\partial x_L} & \frac{\partial f_9}{\partial y_L} & \frac{\partial f_9}{\partial z_L} \end{bmatrix} \tag{6.50}$$

Therefore we have

$$\mathrm{d}\boldsymbol{\theta} = [\boldsymbol{J}_3 \boldsymbol{J}_2^{-1} \boldsymbol{J}_1]^{-1} [-\boldsymbol{J}_3 \boldsymbol{J}_2^{-1} \boldsymbol{J}_4 \vdots \boldsymbol{J}_5] \cdot \begin{bmatrix} \mathrm{d}g_1 \\ \vdots \\ \mathrm{d}g_N \\ \mathrm{d}x_L \\ \mathrm{d}y_L \\ \mathrm{d}z_L \end{bmatrix} \tag{6.51}$$

$$\boldsymbol{J}_{\text{inverse}} = [\boldsymbol{J}_3 \boldsymbol{J}_2^{-1} \boldsymbol{J}_1]^{-1} [-\boldsymbol{J}_3 \boldsymbol{J}_2^{-1} \boldsymbol{J}_4 \vdots \boldsymbol{J}_5]$$

6.7.3 Kinematic Modeling with all Error Parameters

N-R Numerical Method

Because of the number of parameters involved, as well as the number of error sources involved, the kinematic problem becomes very complicated. No analytical solution can be obtained but a numerical solution. The TAU configuration, being a special case of parallel robots, its forward kinematic problem is therefore very complicated. The N-R method as an effective numerical method can be applied to calculate the forward problem of the TAU robot, with an accurate Jacobian matrix obtained.

The N-R method is represented by

$$\boldsymbol{X}_{n+1} = \boldsymbol{X}_{n} - [F'(\boldsymbol{X}_{n})]^{-1} \cdot F(\boldsymbol{X}_{n}) \tag{6.52}$$

With the six chain equations obtained before, the following can now be obtained

$$[F'(\boldsymbol{X}_{n})]^{-1} = Inv \begin{bmatrix} \dfrac{\partial f_1}{\partial x} & \dfrac{\partial f_1}{\partial y} & \dfrac{\partial f_1}{\partial z} & \dfrac{\partial f_1}{\partial \alpha} & \dfrac{\partial f_1}{\partial \beta} & \dfrac{\partial f_1}{\partial \gamma} \\ & & \cdots & & & \\ \dfrac{\partial f_6}{\partial x} & \dfrac{\partial f_6}{\partial y} & \dfrac{\partial f_6}{\partial z} & \dfrac{\partial f_6}{\partial \alpha} & \dfrac{\partial f_6}{\partial \beta} & \dfrac{\partial f_6}{\partial \gamma} \end{bmatrix} \tag{6.53}$$

This equation is used later to calculate the forward kinematic problem, and it is also compared with the method described in the next section.

Jacobian Approximation Method

A quick and efficient analytical solution is still necessary, even though an accurate result has been obtained by the N-R method. The N-R result is produced based on iteration of the numerical calculation, instead of that from an analytical closed form solution. The N-R method is too slow in calculation to be used in online real time control. No certain solution is guaranteed by the N-R method. So a Jacobian approximation method is needed.

The Jacobian approximation method is established. Using this method, error analysis, calibration, compensation and an online control model can be established. As the TAU robot is based on a 3-DOF configuration, instead of a general Stewart platform, the Jacobian approximate modification can be obtained based on the 3-DOF analytical solution without any errors. The mathematical description of the Jacobian approximation method can be described as follows.

For forward kinematics

$$\left.\begin{aligned} \boldsymbol{X} &= F(\boldsymbol{\theta},\ \boldsymbol{\varepsilon}) \\ \boldsymbol{X} &= F(\boldsymbol{\theta},\ 0) + \boldsymbol{J}_{\text{Forward}} \cdot \mathrm{d}\boldsymbol{\varepsilon} \end{aligned}\right\} \tag{6.54}$$

where $J_{\text{forward}} = F(\boldsymbol{\theta}, \boldsymbol{\varepsilon})$ and $\boldsymbol{\varepsilon}$ represents error.

Thus, the analytical solution $F(\boldsymbol{\theta}, 0)$ and $F(\boldsymbol{X}, 0)$ is obtained. Therefore, the Jacobian approximation as an analytical solution is obtained and solving nonlinear equations using the N-R method is not necessary in this case.

6.7.4 Determination of Independent Design Variables

With the reality that all the parts of a robot have manufacturing errors and misalignment errors as well as thermal errors, errors should be considered for any of the components in order to accurately model the accuracy of the robot. This is realized through the established Jacobian matrix.

A total of 40 redundant design variables of the 71 design parameters are eliminated by observing the numerical Jacobian matrix obtained, based on SVD analysis as described previously. Table 6.5 shows the remaining independent design variables.

Table 6.5 List of independent design variables

Parameter number	Parameter definition	Parameter	Parameter number	Parameter definition	Parameter
16	Height of the TCP	a	52	Joint_link 31p	y_{22}
23	Joint 3	a_6	54	Joint_link 21p	x_{33}
23	Arm3	a_7	55	Joint_link 21p	y_{33}
24	Joint 1 & arm 1	d_1	56	Joint_link 21p	z_{33}
25	Short arm 1	d_3	57	Joint_link 21p	x_{44}
28	Joint 3	d_6	58	Joint_link 21p	y_{44}
31	Joint_link 11_arm 1	y_1	59	Joint_link 21p	z_{44}
34	Joint_link 21_arm 1	y_2	60	Joint_link 22p	x_{55}
37	Joint_link 31_arm 1	y_3	61	Joint_link 22p	y_{55}
40	Joint_link 12_arm 2	y_4	62	Joint_link 22p	z_{55}
43	Joint_link 22_arm 2	y_5	63	Joint_link 13p	x_{66}
46	Joint_link 13_arm 3	y_6	67	link 11	L_1
48	Joint_link 11p	x_{11}	68	link 31	L_2
49	Joint_link 11p	y_{11}	69	link 21	L_3
51	Joint_link 31p	x_{22}	70	link 22	L_4

6.7.5 Error Budget Analysis

A general form of error budget is expressed by Eq. (6.10) and it can be applied to the TAU robot structure.

Given the D-H parameters for all three upper arms and the main column, the locations of the joints located at each of the three upper arms are known accurately. The six chain equations are created for the six link lengths, as follows

$$F=\begin{bmatrix} f_1(\text{upperarm\_point, TCP\_point}) \\ \cdots \\ f_6(\text{upperarm\_point, TCP\_point}) \end{bmatrix} \tag{6.55}$$

where upperarm_point=$f(\varepsilon)$, TCP_point=$f(p_x,\ p_y,\ p_z,\ \alpha,\ \beta,\ \gamma)$ and ε is a collection of all the design parameters. Thus

$$F=\begin{bmatrix} F_1(\varepsilon,\ p_x,\ p_y,\ p_z,\ \alpha,\ \beta,\ \gamma) \\ \cdots \\ F_6(\varepsilon,\ p_x,\ p_y,\ p_z,\ \alpha,\ \beta,\ \gamma) \end{bmatrix} \tag{6.56}$$

An error model is developed based on the system of equations as described above. A total of 71 parameters are defined to represent the entire system. The 71 parameters include all the D-H parameters for the three upper arms, as well as the coordinates (x, y, z) of the six points at both ends of the six links, respectively.

Table 6.6 lists the error budget for each design variable.

Table 6.6 Error budget for all design parameters

Variable No.	Description	Name	Budget range	No.	Variable No.	Description	Name	Budget range	No.
1	Drive 1	Joint 1	32 arcsec		38		z_3	D	
2	Drive 2	Joint 2	1.17 arcsec	M	39	Joint_link12_ arm2	x_4	D	
3	Drive 3	Joint 3	1.2 arcsec	M	40		y_4	1.28 mm	
17	Joint 1 and arm 1	a_1	1.62 μm	M	41		z_4	D	
24		d_1	363 μm	16	42	Joint_link22_ arm2	x_5	2.6 mm	
4		θ_1	10.4 arcsec		43		y_5	68.2 μm	18
10		α_1	110 arcsec		44		z_5	D	
18	Joint_link 11_arm 1	a_2	373 μm		45	Joint_link13_ arm3	x_6	D	

(to be continued)

(Table 6.6)

Variable No.	Description	Name	Budget range	No.	Variable No.	Description	Name	Budget range	No.
19	Short arm 1	a_3	174 μm		46		y_6	21.6 μm	1
25		d_3	449 μm		47		z_6	213 μm	
5		θ_3	9.24 arcsec		48	Joint_link11_platform	x_{11}	50 μm	
11		α_3	9.45 arcsec		49		y_{11}	50 μm	4
20	Joint 2 and arm 2	a_4	1.9 mm		50		z_{11}	D	
26		d_4	485 μm		51	Joint_link31_platform	x_{22}	50 μm	5
6		θ_4	1.22 arcsec	M	52		y_{22}	50 μm	15
12		α_4	38.5 arcsec		53		z_{22}	D	
21	Short arm 2	a_5	430 μm		54	Joint_link21_platform	x_{33}	50 μm	14
27		d_5	D		55		y_{33}	50 μm	13
7		θ_5	11.2 arcsec		56		z_{33}	13.3 μm	10
13		α_5	D		57	Joint_link12_platform	x_{44}	50 μm	
22	Joint 3	a_6	0	M	58		y^{44}	50 μm	
28		d_6	D		59		z_{44}	37.9 μm	
8		θ_6	4.64 arcsec		60	Joint_link22_platform	x_{55}	50 μm	7
14		α6	D		61		y_{55}	50 μm	8
23	Arm 3	a_7	0	M(11)	62		z_{55}	398 μm	
29		d_7	D		63	Joint-link13_platform	x_{66}	50 μm	2
9		θ_7	6.14 arcsec		64		y_{66}	50 μm	3
15		α_7	D		65		z_{66}	50 μm	
30	Joint_link11_arm1	x_1	D		16	Height of the TCP	A	436 μm	
31		y_1	43 μm	19	66	Link 13	L_0	0	M(12)
32		z_1	123 μm		67	Link 11	L_1	88 μm	9
33	Joint_link21_arm1	x_2	D		68	Link 31	L_2	151 μm	17
34		y_2	49.4 μm	6	69	Link 21	L_3	54.3 μm	
35		z_2	D		70	Link 22	L_4	213 μm	
36	Joint_link31_arm1	x_3	115 μm		71	Link 12	L_5	1.47 mm	
37		y_3	108 μm						

6.7.6 Simulation Results

The Jacobian approximation method is verified by two different approaches:

(1) 6-DOF forward kinematic analysis (N-R method);

(2) ADAMS simulation results.

Fig. 6.6 shows the error between Jacobian approximation method and ADAMS simulation results. In Fig. 6.6, the maximum error is 1.53 μm with an input error of 1 mm. The Jacobian approximation method has a very high accuracy compared with simulation results.

It can be observed from the results that the Jacobian matrix is effective with an accuracy of up to 1.53 μm with an input error of 1mm (Link 1 of lower arm 1). This was verified using ADAMS simulation results.

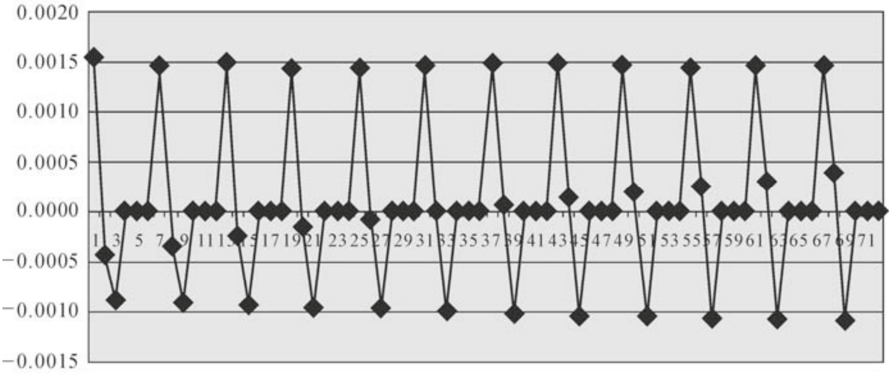


Fig. 6.6. Error between Jacobian approximation method and ADAM simulation results

6.7.7 Experimental Results

To verify the mathematical analysis, the above experiments are conducted for a 2D simplified TAU robot as a test bench, as show in Fig. 6.7, and a digitizer ROMER 3000i of accuracy 30 μm is used as the measurement tool to measure the robot end effector position. The test results are shown in Tables 6.7 and 6.8.

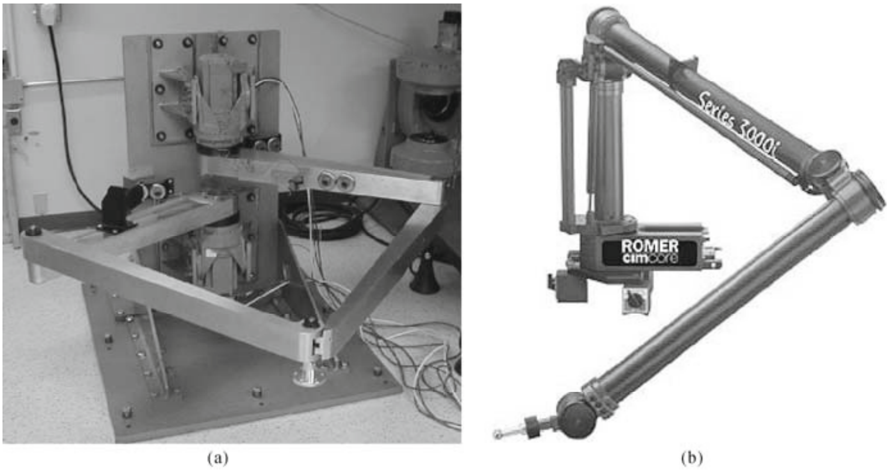


Fig. 6.7. (a) Simplified 2D TAU robot as a test bench; (b) A measurement device

Table 6.7 Experimental results of calibration for a test bench

Parameter number	Parameter name	Measured parameter errors	Calibration results with SVD (mm/deg)
1	θ_1	3	−0.731778837
2	θ_2		2.934613648
3	L_1	0.246	−0.065823708
4	α_1	/	0.005595871
5	β_1	/	0.009767543
6	a_{20}	0.639	−0.600433798
7	α_{20}	0.004297187	−0.054380834
8	L_2	0.022	−0.652730647
9	α_3	/	0.100085204
10	R_x	NA	0.237556976
11	R_y	NA	−0.297084061
12	R_z	NA	86.49124257
13	t_x	NA	−61.06910063
14	t_y	NA	−1934.277556
15	t_z	NA	510.5174107
16	x_{p1}	NA	22.96695136
17	y_{p1}	NA	−56.41477281

Table 6.8 Comparison of results with various optimization approaches

	SVD	LM nonlinear optimization	Gauss N-R nonlinear optimization
Average Absolute Accuracy (mm)	0.11718325	0.11395309	0.11395309
Average Standard Deviation (mm)	0.04774522	0.04849159	0.04849159
Elapsed Time (s)	300	175	175

From the experimental results it has been indicated that the average absolute accuracy after calibration is 0.117 mm and the standard deviation is 0.047 mm.

In this section, based on the D-H model and an accurate Jacobian matrix, a series of results have been presented including error analysis, forward kinematics, redundant variable determination, error budget and Jacobian approximation method. The Jacobian approximation method can be used in online control of the robot. For the TAU robot, a closed form solution of a forward kinematic problem is obtained with a high accuracy instead of an N-R numerical solution. The simulation results are almost perfect compared with those from ADAMS. The experiment results for a test bench show a promising average absolute accuracy of 0.117 mm.

References

Abderrahim M, Whittaker AR (2000) Kinematic model identification of industrial manipulators. Robotics and Computer Integrated Manufacturing, 16:1-8.

Alici G, Shirinzadeh B (2005) A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. Mechanism and Machine Theory, 40:879-906.

Brogangrdh T, et al. (2002a) Device for relative movement of two elements. United States Patent 6425303.

Brogangrdh T, et al. (2002b) Device for displacement of two elements. United States Patent 6336374.

Cui H, Zhu Z, Gan ZX, Brogardh T (2005) Robotics and Computer-Integrated Manufacturing, 21:497-505

Denavit, J, Hartenberg RS (1955) A kinematic notation for lower-pair mechanisms based on matrices. ASME Journal of Applied Mechanics, 5:215-221.

Dhingra AK, Almadi AN, Kohli D, Grobner A (1998) Sylvester hybrid method for closed-form displacement analysis of mechanisms. 1998 ASME Design Engineering Technical Conference, Atlanta, GA.

Dhingra AK, Almadi AN, Kohli D (2000a) Closed-Form displacement analysis of 8, 9 and 10-link mechanisms. Mechanism and Machine Theory, 35:821-850.

Dhingra AK, Almadi AN, Kohli D (2000b) Closed-Form displacement analysis of 8, 9 and 10-link mechanisms, Part II. Mechanism and Machine Theory, 35: 851-869.

Didrit O, Petitot M, Walter E(1998) Guaranteed solution of direct kinematic problems for general configurations of parallel manipulators. IEEE Trans. Robot Automat, 14(2):259-266.

Elatta AY, Li PG, Fan LZ, Yu DY, Lou F (2004) An overview of robot calibration. Information Technology Journal, 3(1):74-78.

Geng JZ, Haynes LS (1994) A 3-2-1 kinematic configuration of a Stewart platform and its application to six degree of freedom pose measurements. Robotics and Computer-Integrated Manufacturing, 11(1):23-34.

Gong C, Yuan J, Ni J (2000a) Non-Geometric error identification and compensation for robotic system by inverse calibration. International Journal Machine Tools & Manufacture, 40:2119-2137.

Gong C, Yuan J, Ni J (2000b) A self-calibration method for robotic measurement system. Journal of Manufacturing Science and Engineering, 122:174-181.

Griffis M, Duffy J (1989) A forward displacement analysis of a class of Stewart platform. J. Robot Syst, 6(6):703-720.

Hayatti SA (1983) Robot arm geometric link parameter estimation. In: Proceedings of the 22nd IEEE Conference on Decision and Control, 1477-1483.

Ibara R, Perreira ND (1986) Determination of linkage parameter and pair variable errors in open chain kinematic linkage, using a minimal set of pose measurement data. Journal of Mechanisms, Transmission and Automation in Design, 159-166.

Lin W, Griffis M, Duffy J (1992) Forward displacement analyses of the 4-4 Stewart platforms. Trans, ASME, 114:444-450.

Meng Y, Zhuang H (2007) Autonomous robot calibration using vision technology. Robotics and Computer-Integrated Manufacturing, 23:436-446.

Mooring BW, Tang GR (1983) An improved method for identifying the kinematic parameters in a six axis robot. In: Proceedings of the 1983 ASME Computers in Engineering Conference, pp.79-84.

Nanua P, Waldron KJ, Murthy V (1990) Direct kinematic Solution of a Stewart platform. IEEE Transactions on Robotics and Automation, 6(4):431-437.

Patel AJ, Ehmann KF (2000) Calibration of a hexapod machine tool using a redundant leg. International Joural of Machine Tools and Manufact, 40:489-512.

Raghavan M (1993) The stewart platform of general geometry has 40 configurations. Joural of Mechanical Design, 115:277-282.

Roth ZS, Mooring BW, Ravani B (1987) An overview of robot calibration. IEEE Journal of Robotics and Automation, 3(5):377-385.

Shi X, Fenton RG (1994) A complete and general solution to the forward kinematics problem of platform-type robotic manipulators. IEEE, 1050-4729: 3055-3062.

Sreenivasan SV, Waldron KJ, Nanua P (1994) Closed-Form direct displacement analysis of a 6-6 Stewart platform. Mechanism and Machine Theory, 29(6):855-864.

Stone HW, Sanderson AC, Neuman CP (1986) Arm signature identification. In: Proceedings of IEEE International Conference on Robotics and Automation, 41-48.

Tsai LW (1999) Robot Analysis: The Mechanics of Serial and Parallel Manipulators. Wiley, New York.

Tsai RY, Lenz RK (1989) A new technique for fully autonomous and efficient 3D robotics hand/Eye calibration. IEEE Transactions on Robotics and Automation, 5(3):345-358.

Wang J, Masory O (1993) On the accuracy of a Stewart platform—Part I: The effect of manufacturing tolerances. IEEE, 1050-4729:114-120.

Ye SH, Wang Y, Ren YJ, Li DK. (2006) Robot calibration using iteration and differential kinematics. Journal of Physics: Conference Series, 48:1-6.

Zhang CD, Song SM (1991) Forward kinematics of a class of parallel (Stewart) platform with closed-form solutions. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation, CA, USA. 2676-2681.

Zhuang H, Roth ZS (1996) Camera Aided Robot Calibration. CRC Press, Inc.

Zhuang H, Wang K. Roth ZS (1995) Simultaneous calibration of a robot and a hand-mounted camera. IEEE Transactions on Robotics and Automation, 11(5): 649-660.

7

Visual Sensing and Control-Laser Sensor Based Robot Applications

In this chapter various laser sensors based robot vision systems and their applications are presented. Three kinds of visual sensing and control applications are discussed including measurement and inspection, identification and localization, and visual servoing. In the area of measurement and inspection, the method for detecting the position and orientation of holes or channels in a 3D space is presented in Section 7.1, where a camera and a laser point sensor are used. A robotic grinding system for a complex surface work piece is presented in Section 7.2 where the laser stripe sensor is used to measure and locate the work piece. A profile modeling based grinding approach is presented and discussed in Section 7.3 where a 3D laser stripe sensor and a LVDT are used for measuring, fine-tune and on-line quality control. A flexible robotic machining system that can compensate for parts shape variations is presented in Section 7.5, where the individual parts are measured by a laser stripe sensor and the measured data are used as feedback for the robot controller. A highly accurate relative measurement robot system is presented in Section 7.6, which is used to measure the material removal of a free-form work piece in the grinding process.

Regarding identification and localization applications, in Section 7.4, a sand core handling/assembly system is presented, which handles and assembles sand cores to sand boxes to form sand molds for sand casting with an accuracy requirement of ±0.3 mm. In Section 7.8, two pick and place robot systems are presented where point laser sensors are used to identify edges of parts.

For visual servoing applications, a general robotic seam tracking system which can tune robotic poses with 6 degrees of freedom is presented in Chapter 7.7. This includes the architecture of the system, the welding joint detection, the path generation algorithm, and computer-robot communication.

Z. Gan et al., *Visual Sensing and its Applications*

7.1 Automatic Inspection of Holes in 3D Space

In certain industrial applications, there are parts with many tiny or large holes or tunnels of various shapes (for example, cooling holes on the gas turbine blades or vanes), and the orientation and position of each of the holes needs to be inspected automatically. The method for detecting the position and orientation of holes or channels in a 3D space using a robot vision system is presented (Wang, 2003). It includes the approaches and algorithms to detect the hole position, size and orientation by using a vision system mounted on the robot arms. The hole orientation is determined based on the alignment of the vision system and the hole axis. The position of the hole is the intersection between the hole axis and the surface region around the hole opening. Experimental results have indicated that the concept of cooling hole identification is feasible. It has been shown that the reproducible detection of thecooling channel position is within an accuracy of ±0.15 mm and cooling channel orientation is within an accuracy of ±3°.

7.1.1 Introduction

Gas turbines are extensively used in flight propulsion, electrical power generation and other industrial applications. Since turbine engines operate at a very high temperature (1,200 °C – 1,400 °C), it is very important to cool the turbine blades/vanes to reduce the thermal stress. Cooling holes/channels are passages for the coolant on the blade/vane for this purpose. During its life span, a turbine blade is taken out periodically for repair and maintenance. This includes re-coating the blade surface and re-drilling the cooling holes (Hoebel, 2010). A successful laser re-drilling requires the measurement of a hole within an accuracy of ±0.15 mm in position and ± 3° in orientation. Conventionally, this measurement is done on a coordinate measurement machine (CMM) using a cylindrical gauging pin. The pin is first inserted into a hole on a blade with the coating stripped. A number of points on the pin are then measured using a CMM to construct a cylinder. The position of the cylinder gives the location of the hole. This is a manual process and it is time consuming, considering that there are hundreds of cooling holes on one blade, and the measurement of a single hole takes about 2 min. It is also error prone due to the difficulty in tightly fitting the pin into the hole. It is obviously costly due to the lengthy use of a CMM. Automation of the measurement process is therefore a very demanding task in the industry. It is preferable to use non-contact measuring systems to improve the efficiency and make the inspection process fully automatic. Such a system can be realized by an industrial robot or other multi-axis CNC machines in conjunction with the vision system.

In this section, a robotic vision system is presented as the solution. Section

7.1.2 introduces the overall system configuration. Section 7.1.3 is devoted to the calibration of the vision system. Section 7.1.4 focuses on the measurement process. The test results and conclusion are then presented in Section 7.1.5.

7.1.2 System Overview

The implemented workcell for cooling hole measurement is basically a robotic vision system, consisting of an industrial robot, a CCD camera and a laser displacement sensor, as shown in Fig. 7.1. The CCD camera and the laser sensor are mounted on the end effector of the robot. The images of cooling holes are acquired from an image grabber. The laser displacement sensor measures 1D linear displacement based on the triangulation principle. Its reading can be obtained from an A/D card. A remote PC is used for the graphic interface and the calculation. This PC controls the robot through a network using communication software. The calibration and measurement software is the core of the system. On the one hand, it communicates with the robot and the laser sensor to acquire the current robot and tool positions. On the other hand, it processes the cooling hole images to identify the hole position and orientation. The workpiece to be measured is placed on a fixture and kept stationary during the operation.

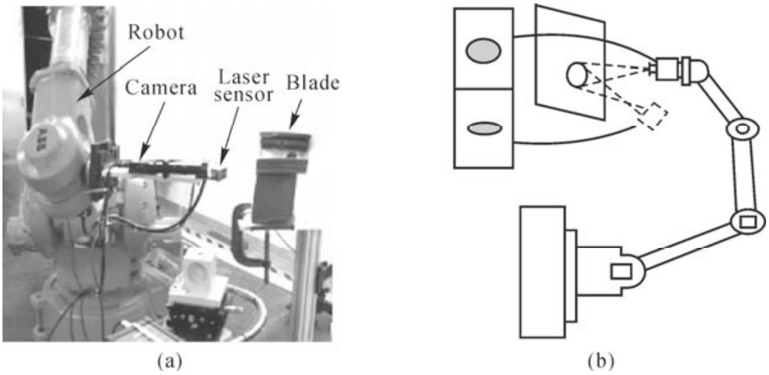


Fig. 7.1. (a) Setup for detection of the hole orientation and position by using robot vision system; (b) Hole alignment principle

The basic idea for the determination of the cooling hole orientation is to align the camera optical axis with the hole axis. This is done by sweeping the camera around the hole axis and searching for the alignment as indicated in Fig. 7.1(b). The criterion for the alignment is the maximization of an image feature function. When alignment is achieved, the image center of the hole opening is then detected. The orientation of the hole can be readily obtained as the ray connecting the image center and the camera lens center. The position of the hole opening is simply the intersectional point between the hole axis and the surface around the hole opening. This surface is measured by the laser displacement sensor.

7.1.3 System Calibrations

In order to get accurate results, the vision system, laser and workpiece must be calibrated (Schreoder, 1999).

7.1.3.1 Calibration of Vision System

For the vision system, the following parameters need to be calibrated.

(1) T_v^i : Camera intrinsic parameters like focal length, center of optical axis and coefficient of the lens distortion. That will establish the relationship between the image coordinate $P_i(x, y)$ and the 3D position in the vision system $P_v(X, Y, Z)$, that is $P_v(X,\ Y,\ Z) = T_v^i \cdot P_i(x,\ y)$.

(2) T_t^v : Transformation from the camera system to the robot tool mounting flange ($Tool_0$) coordinate system, also called the TCP of the camera system.

(3) T_t^l : Mapping between the reading from the laser displacement sensor and its 3D position in the robot tool mounting flange ($Tool_0$) coordinate system. It is called the TCP of the laser system.

The camera calibration is based on a well-known RAC algorithm (Tsai, 1987). This is to calibrate the camera intrinsic parameters and establish the relationship between the image coordinate (x, y) and a reference 3D coordinate (X, Y, Z). To calibrate the above parameters, a set of calibration points are used. This set of points have known positions in the reference coordinates. An accurate target is used and placed near the workpiece to be measured, as shown in Fig. 7.2. This is a commercially available distortion target. The dot center-to-center spacing accuracy is within ±0.0025 mm.

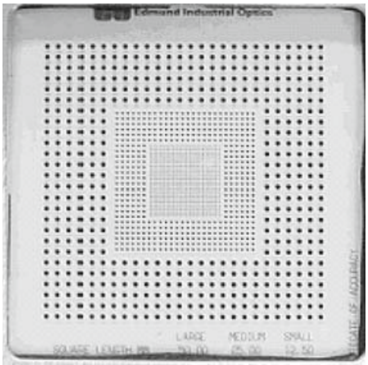


Fig. 7.2. Calibration target

Calibration of the Camera TCP

There are two steps in calibrating the camera position relative to the robot mounting flange (*Tool*$_0$), also called camera TCP. In the first step, the camera position relative to a calibration target can be determined with Tsai's coplanar RAC algorithm, as discussed in the previous section. In the second step, the camera TCP can be obtained from the chained relationship between the camera the coordinate, the calibration target coordinate and the robot coordinate.

Jog the robot to make the vision system see the calibration target so that the camera to calibration target position can be determined first (T_e^v) using the RAC algorithm. Also record the corresponding robot mounting flange position (T_0). The vision system TCP can be determined by the following relationship

$$T_0 T_t^v T_v^e = T_b^e \tag{7.1}$$

then

$$T_t^v = (T_0)^{-1} T_b^e (T_v^e)^{-1} = (T_0)^{-1} T_b^e (T_e^v) \tag{7.2}$$

where T_b^e is the calibration target position relative to the robot base that can be measured with a calibrated laser tool. The calibration of the laser tool will be addressed in the following section.

Calibration of the Laser TCP

To correctly define the laser coordinate system, with respect to the robot base coordinate system, the laser TCP must be accurately calibrated relative to the robot mounting flange.

The laser sensor is a relative one-dimensional measurement tool. It determines relative displacement using a CCD-array. Its output (in volts) indicates the displacement of the surface relative to the reference position (TCP). Fig. 7.3 shows the triangulation principle of the laser sensor with the TCP positioned at the zero output voltage reference position. The output scale is 0.1 V to 0.1 mm, giving the sensor a range of ±5 mm (±5 V) from the TCP. The position on the CCD array indicates the relative displacement.

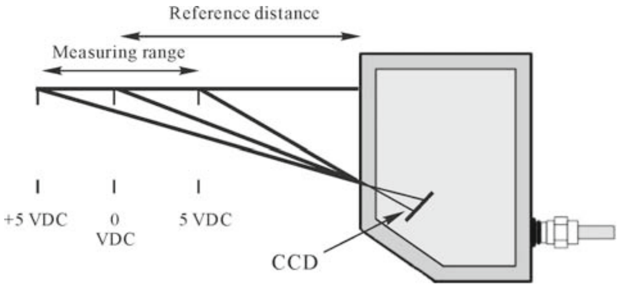


Fig. 7.3. Laser sensor and its TCP position used in the robot vision system

Two steps are required to calibrate the TCP with respect to the robot mounting flange. One is to determine the beam orientation and the other is to calibrate the TCP position (Zhu, 2004). The detail is described in Section 4.2.

It is convenient to have the laser beam aligned with the *Z* axis of the robot mounting flange coordinate frame. To verify the alignment, a small target sheet is placed on the workbench. The laser sensor is moved 100 mm away from the target and pointed at the target center. The robot is then moved back from the target in the negative *Z* direction. If the position of the laser spot remains unchanged on the center of the target, the laser beam is collinear with the *Z* axis (normal to the mounting surface). If the laser spot moves from the target center, adjustment is required. Using this technique, it is possible to orient the laser to within ±0.3° of the desired orientation.

Another method for determining the TCP position is to use the conventional 4-point calibration method that is a built-in function of a robot controller. A small sphere of diameter 6 mm is placed in the position of the robot workcell where it can be accessed from different robot poses, as shown in Fig. 7.4. The laser TCP is focused onto the center of the sphere from four different approach orientations.

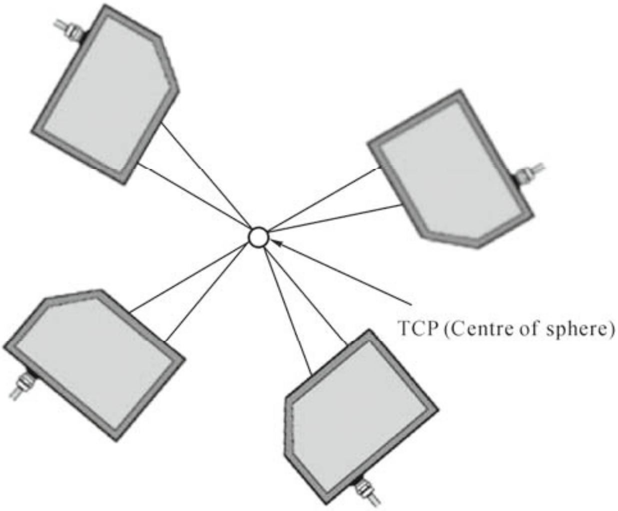


Fig. 7.4. Four-point calibration setup

To ensure the TCP is focused through the center of the sphere, the laser beam moves through the sphere surface in the *X* direction of the $Tool_0$ frame. When the TCP is focused on the center of the sphere, the sensor reading is minimized. Another method for obtaining a correct TCP position is to use a back-reflection technique to verify that the laser beam is incident normal to the sphere surface. If the incident beam is correctly oriented normal to the surface, it will be reflected directly back to the source. But if it is oriented incorrectly, it will be reflected at an angle. Because of the small diameter of the sphere, a slight mis-orientation of the incident angle will produce a much larger angle of reflection. The robot position can be manually adjusted so that the incident beam is positioned for reflection directly back into the source. With the beam correctly focused through the center of the sphere, the next step is to ensure that the TCP is positioned at the sphere center. The sensor TCP position is where the sensor reading is 0 mm. When the sphere of radius 3 mm is used, the TCP position is corresponding to the sensor reading of –3 mm. To position the TCP 3 mm into the sphere, the robot is moved in the tool *Z* axis so that the sensor reading is −3 mm. This correctly places the laser TCP at the center of the sphere. At this point, the robot pose is entered into the built-in TCP calibration program. The process is then repeated three more times from various robot poses and the sensor TCP can be calculated from the built-in program. Repeatability results from the calibration have shown that the position of the tool centre point can be determined to within ±50 μm precision. Combined with ±0.3° precision in orientation, the tool calibration is within the expected degree of accuracy.

7.1.3.2 Calibration of Work Object

This is to calibrate the workpiece coordinate system with respect to the robot base coordinate system, which is T_b^w.

After the laser displacement sensor has been calibrated previously, it can be used as a measurement tool. Jog or program the robot with the laser sensor to measure three feature planes of the workpiece as indicated in Fig. 7.5. This requires measuring 5 points on each plane. Those three planes will form a coordinate system and determine T_b^w.

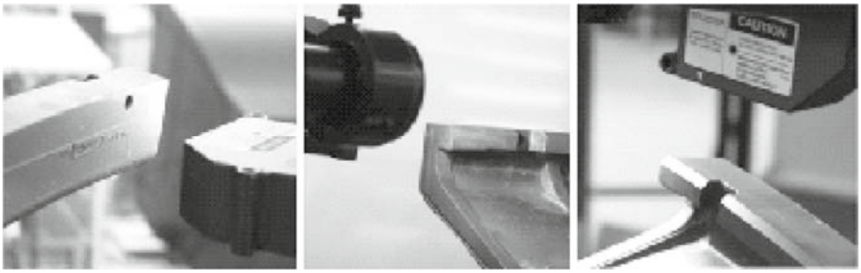


Fig. 7.5. Creation of the coordinate system using three feature planes on the object. The planes are measured using a laser displacement sensor

The definition of a coordinate system from three planes is as follows. The intersection point from three planes will uniquely determine the origin of the coordinate system. The plane normal of the first measured plane will be used as the *Z* direction of the coordinate system. The cross product of the first plane normal and the third plane normal will be used to determine the *X* direction of the coordinate system. Then the *Y* direction of the coordinate system will be the cross product of *Z* and *X* directions. Note that the three planes may not be perpendicular to each other.

In this way we can define a coordinate system that is unique and orthogonal.

7.1.4 Inspection Procedure

With the current accuracy of the industrial robot, it is extremely difficult to achieve the 0.15 mm measurement error. Instead, a relative measurement strategy is adopted which relies only on the high repeatability of the robot (0.07 mm). For each type of turbine blade, a reference blade (called master blade) is first measured with the traditional method using CMM. Then this master blade is measured again by the robot vision system. An error compensation scheme is determined. During the measurement of the actual blades, the robot moves along the same path, and the measurement results are compensated according to the identified error scheme.

7.1.4.1 Determine the Hole Orientation

The orientation of the hole in 3D space can be represented by a straight line equation with parameters (n_x, n_y, n_z, x_0, y_0, z_0), where (n_x, n_y, n_z) represents the orientation and (x_0, y_0, z_0) represents any point on the line.

Step 1: *Alignment of the Vision System with the Hole Axis*

Orient the robot to get into the pose where the hole axis is roughly aligned with the camera. Rotate the camera orientation vertically and horizontally and take a snapshot of the hole opening images during the robotic searching process. In order to obtain a high contrast image, an illumination system is used which can be mounted on the robot arm. For each image, the pattern of the hole-opening cross-section looks like an elliptical shape, as shown in Fig. 7.6(a). But it may not be symmetrical. The opening portion has low optical intensity since most of the incident light has been absorbed from the inner surface of the hole. The outside has high optical intensity (relatively white) due to the surface reflection of the high illumination. Calculate the image area of the hole-opening cross-section. The alignment position is determined based on the fact that the image area of the hole opening cross section is maximized, as shown in Fig. 7.6(b). This criterion is independent of the real shape of the hole-opening cross-section. Some other criterion like roundness and pattern match may apply, depending on the real shape of the hole-opening cross-section.

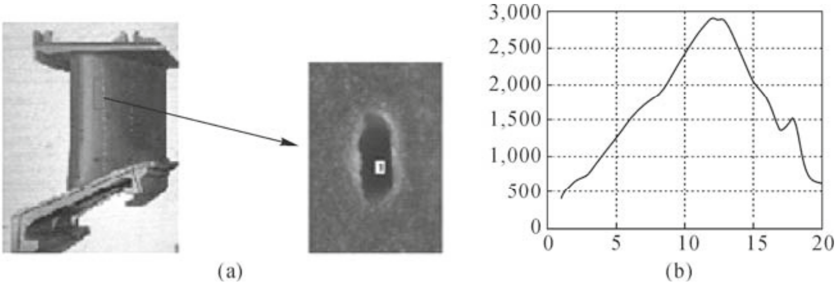


Fig. 7.6. Determination of alignment position based on the maximum area of the cross section of the hole opening image. (a) Typical image of the hole opening; (b) The image area of the hole-opening cross-section varies when the hole which the camera aiming at is rotated and gets maximized when the camera axis is aligned with the hole axis

Step 2: Determination of the Hole Orientation in the Camera Coordinate System

After the hole-opening is aligned with the camera, an image processing algorithm is applied to detect the center position of the hole opening (u, v) in the image coordinate system, as shown in Fig. 7.6(a). The hole orientation in the camera coordinate system will be determined by the following line equations (image project relation)

$$u = f_x \frac{x}{z},\ v = f_y \frac{y}{z} \tag{7.3}$$

where (f_x, f_y) is the camera focal length in horizontal and vertical directions, respectively. Those parameters are camera parameters and can be calibrated previously. Eq. (7.3) actually represents a ray that connects the image center and the lens center. From Eq. (7.3), the normal (n_x, n_y, n_z) of the line can be easily derived as

$$n_x = \frac{u}{f_x d},\ n_y = \frac{v}{f_y d},\ n_z = \frac{1}{d} \tag{7.4}$$

where $d = \sqrt{(\frac{u}{f_x})^2 + (\frac{v}{f_y})^2 + 1}$.

The point on the line can be set as (x_0, y_0, z_0) = (0, 0, 0) that is the origin of the camera coordinate system.

Step 3: Convert the Hole Orientation in a Part Coordinate System

It is more convenient to have the hole orientation described in the workpiece coordinate system. Assume they are $(n_x', n_y', n_z', x_0', y_0', z_0')$. The transformation can be done with the robot kinematic equation. Define a transformation describing the relationship between the camera coordinate system and workpiece coordinate system through the robot kinematics

$$\boldsymbol{T}_w^v = (\boldsymbol{T}_b^w)^{-1} \boldsymbol{T}_0 \boldsymbol{T}_t^v \tag{7.5}$$

where $\boldsymbol{T}_b^w$ is the transformation from the workpiece coordinate system to the robot base, which defines the position of the workpiece relative to the robot base that can be determined by using the laser sensor, as described previously. $\boldsymbol{T}_t^v$ is the camera TCP as determined in Section 7.1.3. $\boldsymbol{T}_0$ is the position matrix of the robot mounting flange coordinate system (i.e, *Tool*$_0$) in the robot base. The value can be obtained from the robot control system and varies when the robot moves.
Then

$$(x_0', y_0', z_0', 1)' = \boldsymbol{T}_w^v \cdot (x_0, y_0, z_0, 1) \tag{7.6}$$

$$(n_x', n_y', n_z')' = \boldsymbol{R}_w^v \cdot (n_x, n_y, n_z) \tag{7.7}$$

where $\boldsymbol{R}_w^v$ is the rotation matrix of the transformation matrix $\boldsymbol{T}_w^v$.

7.1.4.2 Determination of the Hole Position

In order to determine the hole-opening position, a laser displacement sensor is used to measure the surface of the part that is around the hole-opening. The laser displacement sensor has to be calibrated previously, as described in Section 7.1.3. 5 points around the hole opening are measured. Since the reading of the laser scanner is based on the robot mount flange coordinate system *Tool*$_0$, it has to be converted into the workpiece coordinate system by using the following relationship

$$(x', y', z', 1)_w' = (\boldsymbol{T}_b^w)^{-1} \cdot \boldsymbol{T}_0 \cdot (x, y, z, 1)' \tag{7.8}$$

where $\boldsymbol{T}_b^w$ is the workpiece coordinate that can be obtained by using the approach described in Section 7.1.3.

Do surface fitting to determine the surface equation. For simplicity, we assume that the surface can be modeled as a plane as an approximation. It can be described by the following plane equation

$$n_x * x' + n_y * y' + n_z * z' = D \tag{7.9}$$

where (n_x, n_y, n_z) is the normal of the plane and D is the plane offset that is determined by the least square plane fitting algorithm.

The intersection of the hole axis described by Eqs. (7.6) and (7.7) and the surface plane described by Eq. (7.9) around the hole-opening gives the hole-opening position (x, y, z).

7.1.5 Experimental Results and Conclusion

The robot used for the tests is an IRB 4400 from ABB. The laser displacement sensor is an Opto NCDT 1800 from Micro-Epsilon, having a measurement range of ±5 mm, a working distance of 25 mm and a resolution of 1 μm. The camera used for image acquisition is a Sony XC-ST 50 with a resolution of 640×480 pixels. The lens used (VZM 200i from Edmund Industrial optics) has a working distance of 152 mm and focus depth of about 3 mm. In order to have adequate illumination, a ring light fiber optics illuminator is attached to the head of the lens. The laser tool and camera tool are calibrated according to the procedures stated in Section 7.1.3. Limited tests have been performed on the turbine vanes following the measurement procedure in section 7.1.4. Table 7.1 shows the measurement result for a single hole on the turbine vane. The position and orientation of the hole as indicated from its CAD are $X = 91.694$ mm, $Y = 99.404$ mm, $Z = -24.440$ mm; $nx = 0.0000$, $ny = -0.123412$, $nz = -0.992355$.

The results show that the measurement repeatability is about 100 μm for hole positions and 2.6° for the orientations. The error compensation scheme is simply an offset compensation. This offset is a function of the hole position and orientation.

Table 7.1 Position and orientation of the hole on the test piece

Test number	X	Y	Z	n_x	n_y	n_z
1	89.209	99.314	−26.011	0.005122	−0.069185	−0.997595
2	89.213	99.322	−26.002	0.005167	−0.069111	−0.997600
3	89.190	99.246	−26.082	0.005023	−0.052214	−0.998627
4	89.272	99.174	−26.153	0.005418	−0.070100	−0.997527
5	89.272	99.176	−26.150	0.005415	−0.070099	−0.997595
6	89.192	99.106	−26.225	0.005088	−0.070431	−0.997595
Mean	89.224	99.223	−26.104	0.0052055	−0.066857	−0.99773
Std. Dev.	0.038	0.086	0.087	0.0001500	0.007193	0.000441
Max. Dev.	0.047	0.1169	0.1206	0.0002125	0.01464	0.0008968

It has been indicated from the analysis and experimental results that the concept and algorithms for the measurement of the hole orientation and position using the robot vision system are feasible.

In this section a robot vision system is presented to automate the measurement of turbine blade cooling holes. To meet the high accuracy requirements, a relative measurement strategy is adopted in conjunction with the sophisticated calibration of individual components in the vision system. Limited test results show that the measurement repeatability for the hole position is within ±0.15 mm, and for the orientation is within ±3°, in laboratory test conditions.

7.2 Robotic Grinding System of Free-Form Workpieces

This section presents a robotic grinding system for workpieces with free-form geometries. In robotic grinding systems, smooth finishing of the complex workpiece requires highly accurate motion control of the robot. To achieve this, not only should the workpiece be located properly, to close the kinematic chain of the robotic system, but also the robotic system error should be taken into consideration and well compensated for. A 3D laser scanner is used to obtain the point clouds of the workpiece surface and a registration algorithm is adopted to locate the workpiece. And two novel error compensation methods are proposed to enhance the accuracy of the robotic grinding system. Experimental results indicate the effectiveness of the methods.

7.2.1 Introduction

Grinding broadly refers to the final machining or finishing process that removes material utilizing hard abrasive particles as the cutting medium. As an important machining method, grinding has been investigated for decades. Studies have been made concerning grinding force, energy, thermal model, friction, vibration and material removing processes, among others.

The grinding of a part with free form geometries is typically time and labor consuming. In recent years, work was presented about robotic grinding systems (Gunnarsson, 1987; Chen, 1999; Huang et al., 2003; Vergeest, 2003; Sun, 2004; Sun, 2009). In these kinds of systems, path generation is one of the key issues that have to be addressed. Some studies (Chen, 1999; Huang et al., 2003) present a path generation method based on the sensory measurement data.

Fig. 7.7 indicates the architecture and function modules of the robotic grinding system. It consists of offline programming, calibration, process modeling and online compensation modules.

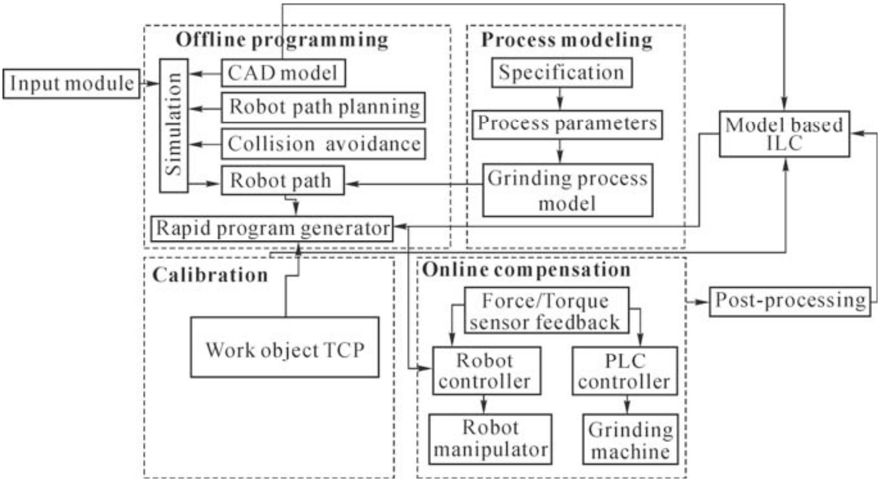


Fig. 7.7. Overview of the architecture of the robotic grinding system

7.2.2 Offline Programming

Due to the complexity of the geometry of the free-form workpiece, offline programming is usually used for path generation for robotic grinding (Sun, 2004). The idea of offline programming is to generate a sequence of coordinates (defined as targets) on the workpiece based on its CAD model, which the machining tool is to reach (Craig, 1989). Fig. 7.8 shows an example in which targets are created on the surface of a blade.

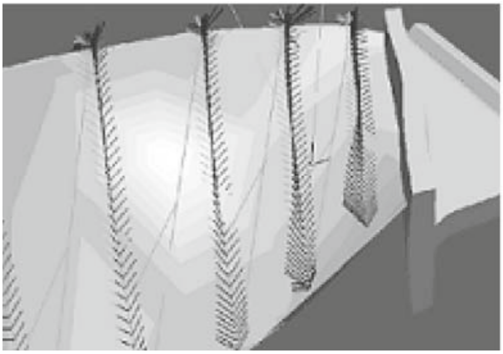


Fig. 7.8. Target generation on a CAD model

For a NURBS surface $S(u, v)=(x(u, v), y(u, v), z(u, v))$, U/V curves can be generated by sampling along U/V directions. After that, points can be further generated by sampling along the curve. The surface normal can be calculated as a cross product of the tangential vectors along U and V.

$$N(u, v) = T_u(u, v) \times T_v(u, v) \tag{7.10}$$

where

$$\left.\begin{aligned} T_u(u, v) &= \frac{\partial}{\partial u} S(u, v) \\ T_v(u, v) &= \frac{\partial}{\partial v} S(u, v) \end{aligned}\right\} \tag{7.11}$$

Therefore, targets can be generated with respect to the coordinate of the CAD model. With the generated targets, the motion of the robot during the grinding can be determined by closing the kinematic chain, as shown in Fig. 7.9, with equation

$$\boldsymbol{Tool} = \boldsymbol{Tool}_0 \cdot \boldsymbol{Wobj} \cdot \boldsymbol{Target} \tag{7.12}$$

where ***Tool*** is the 4×4 homogeneous position/orientation matrix of the tool with respect to the base frame of the robot, which is pre-calibrated; ***Tool***$_0$ is the 4×4 position/orientation matrix of the flange plate with respect to the base frame of the robot, which can be controlled by the robot controller; ***Wobj*** is the 4×4 position/

orientation matrix of the workpiece with respect to the flange plate frame $\boldsymbol{Tool}_0$; ***Target*** is the 4×4 position/orientation matrix of the target with respect to the ***Wobj*** frame, which is generated from offline programming.

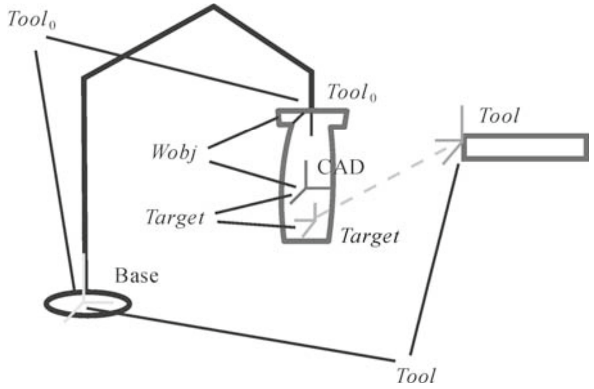


Fig. 7.9. Kinematic chain

7.2.3 Workpiece Calibration

In order to effectively use the path generated by offline programming, the workpiece has to be calibrated. Methods using 2D sectional profile matching and 3D registration (Chen, 1999; Huang et al., 2003; Sun, 2004) have been proposed to locate a workpiece in the real world. Compared with the 2D sectional profile matching methods, the 3D registration methods, which calibrate the workpiece by finding the transformation matrix between the 3D point cloud of the actual workpiece and its CAD model positioned at an initial pose, are capable of locating all 6-DOF of the workpiece (Gunersson, 1987; Vergeest, 2003; Pottmann, 2006). To obtain the point cloud of a workpiece, a 3D laser scanner is used, as shown in Fig. 7.10. The point cloud construction can be done as following

$$\boldsymbol{P}_T = \boldsymbol{Tool}_0^{-1} \cdot \boldsymbol{T}_{\text{Sensor}} \cdot \boldsymbol{P}_S \tag{7.13}$$

where P_T is the position (4×1 matrix) of the measured point with respect to the $Tool_0$ frame; $\boldsymbol{Tool}_0$ is the 4×4 position/orientation matrix of the flange plate with respect to the base frame of the robot, which can be obtained from the robot controller; $\boldsymbol{T}_{\text{Sensor}}$ is the homogeneous 4×4 position/orientation matrix of the sensor with respect to the base frame of the robot, which is pre-calibrated; $\boldsymbol{P}_S$ is the position (4×1 matrix) of the measured point with respect to the sensor frame, which can be obtained from the scanner .

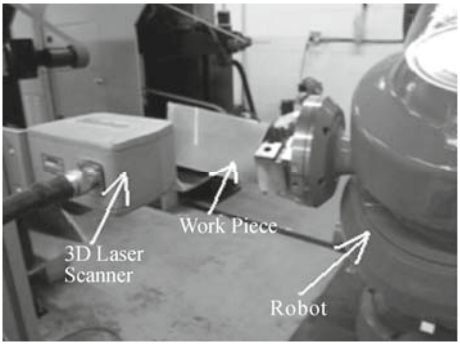


Fig. 7.10. Robotic measurement system

From the measured point cloud, the location of the workpiece can be obtained by the following steps:

(1) Locating the CAD model of the workpiece to a nominal position/orientation ($Wobj_{\mathrm{norm}}$);

(2) Using 3D registration algorithms to solve the transformation matrix (Tt) from the scanned point cloud to the pre-located CAD model;

(3) Calculating Wobjreal (the real Wobj) with equation

$$\boldsymbol{Wobj}_{\mathrm{real}} = \boldsymbol{T}_t^{-1} \cdot \boldsymbol{Wobj}_{\mathrm{norm}} \tag{7.14}$$

Fig. 7.11 shows the convergence processes of the registration.

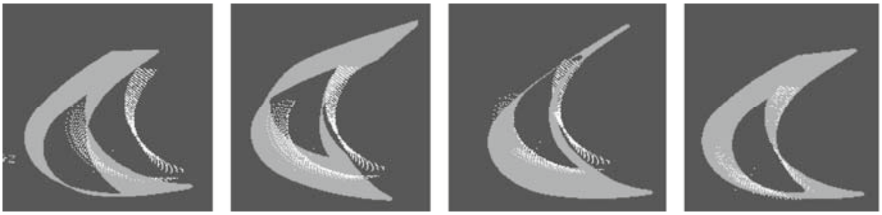


Fig. 7.11. Convergent process of registration (CAD model in gray; measured point cloud in white)

7.2.4 Robotic System Error Compensation

Due to the errors in the robotic systems, which consist of joint level error, kinematic model error, and non-kinematic error (Roth, 1987), the robot cannot precisely reach the position/orientation required, making the calibrated ***Wobj*** inaccurate. In the grinding process, the erroneous ***Wobj*** can lead to unexpected contact (shown in Fig. 7.12) between the grinding wheel and the workpiece.

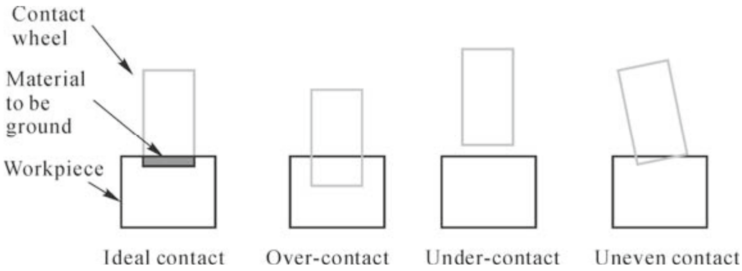


Fig. 7.12. Contact between grinding wheel and workpiece

The calibration procedures of robotic systems are usually time-consuming (Gong, 2000; Gursel, 2005). Expensive coordinate measurement devices such as laser trackers are often involved in the calibration process. In industry, easily implemented methods are highly demanded for the routine maintenance of robotic systems. In this section, instead of calibrating the absolute accuracy of the robot, a novel method for system error compensation is presented.

A compliant tool may be able to compensate for the position error, but it can hardly do anything about the orientation error, which is usually more critical for grinding. In the following, two novel methods are introduced to compensate for both the position and the orientation errors.

Position Error Based Workpiece Calibration

Fig. 7.13 shows a system setup for robotic error compensation, in which a virtual tool is placed in front of a sensor. By controlling the robot to make the surface of the workpiece pass by the virtual tool, the error for each target (the coordinate on the surface generated by offline programming) can be measured by the laser sensor. The error can be considered as the *Z* component of the position vector of the virtual tool, with respect to the target frame on the actual workpiece. With this constraint, an equation can be given as

$$\boldsymbol{Offset} - \boldsymbol{E}_m = \{(\boldsymbol{Tool}_0 \cdot \boldsymbol{Wobj}_{act} \cdot \boldsymbol{Target})^{-1} \cdot \boldsymbol{Tool}\}_3 \tag{7.15}$$

where $\boldsymbol{E}_m$ is the position error of the virtual tool along its measurement direction; $\boldsymbol{Tool}_0$ is the 4×4 position/orientation matrix of the flange plate when the target reaches the virtual tool; $\boldsymbol{Wobj}_{\text{act}}$ is the actual $\boldsymbol{Wobj}$; $\boldsymbol{Target}$ is the position/orientation matrix of the target with respect to the workpiece frame; $\boldsymbol{Tool}$ is the position/orientation matrix of the virtual tool with respect to the robot base frame and $\{\}_3$ means the third element of the inside vector. With several measurements, the $\boldsymbol{Wobj}_{\text{act}}$ and $\boldsymbol{E}_m$ can be solved with mathematical methods such as nonlinear least squares.

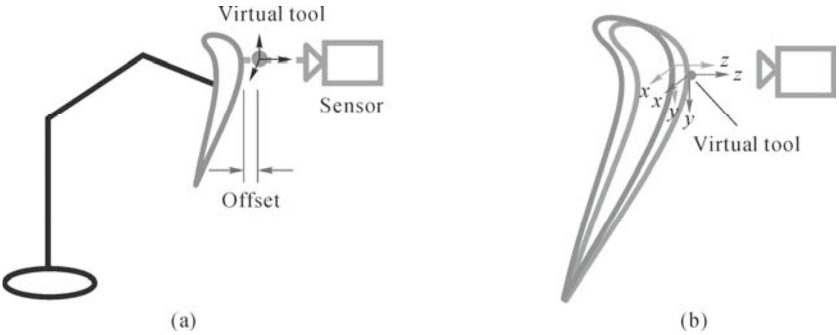


Fig. 7.13. (a) Error measurement; (b) Error compensation

Position/Orientation Error Based Fine Tuning Method

Besides the global compensation of the system error with *Wobj* re-calibration, target by target compensation, which has more degrees of freedom to play with, is also feasible. A fine tuning device is developed for that purpose.

As shown in Fig. 7.14, the device consists of:

(1) A dummy contact wheel;

(2) A digital angular gauge which can measure the rotation of the dummy contact wheel;

(3) An LVDT which can measure the shift of the dummy contact wheel in the direction perpendicular to the contact surface;

(4) Other components.

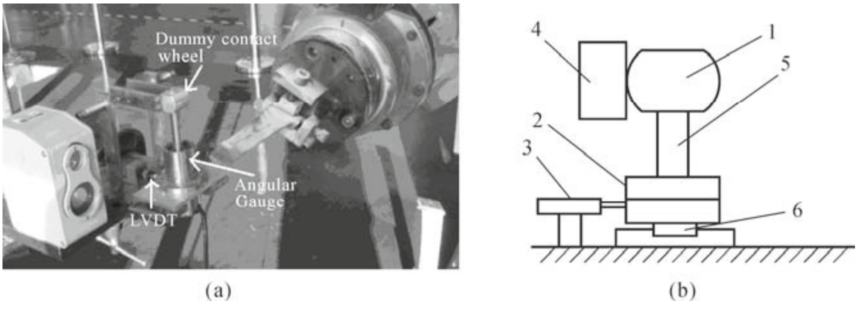


Fig. 7.14. Fine tuning device. (a) System layout; (b) Side view

A side view of the device is shown in Fig. 7.14(b). When the workpiece (component 4) contacts the dummy contact wheel (component 1), if orientation error exists, a torque along the axis (component 5) will be generated. And the rotation angle can be detected by the angular gauge (component 2). If position error exists, the component will be pushed along the slider (component 6) and the offset can be measured by the LVDT (component 3). With the measured data, a transformation matrix, which does not depend on the absolute accuracy of the robot, can be generated for each target on the workpiece, to compensate for the error. Since the process is feedback based compensation, the limit of the accuracy is the repeatability of the robot.

One disadvantage of target by target compensation is its efficiency. An improvement to this method is to conduct the compensation process once on a master workpiece. For other workpieces of the same type, only the overall *Wobj* difference is calibrated.

7.2.5 Experimental System

An actual system consisting of an ABB-4400 robot, a 3D laser scanner, a force sensor and a belt grinder is built, as shown in Fig. 7.15. In the system, the point cloud of the workpiece was scanned by the laser scanner and the quadratically convergent 3D registration method (Pottmann, 2006) was adopted to locate the pose of the workpiece. Targets on two paths (shown in Fig. 7.16) were measured by the laser scanner for the compensation of the robotic error by the method described previously. As shown in Fig. 7.17, after the compensation the error can be reduced from several millimeters to below 0.1 mm.

Fig. 7.18 shows the position/orientation error before and after the fine-tune compensation. Targets on five paths are fine tuned. As shown in Fig. 7.18, after the compensation, the orientation error drops from 2° (max) to below 0.2°, while the position error is reduced from about 1.5 mm (max) to below 0.1 mm.

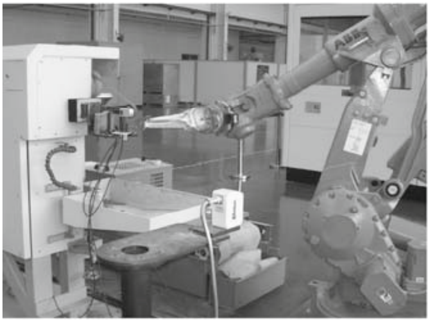


Fig. 7.15. Robotic grinding system

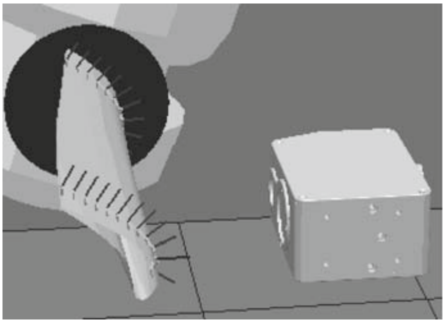
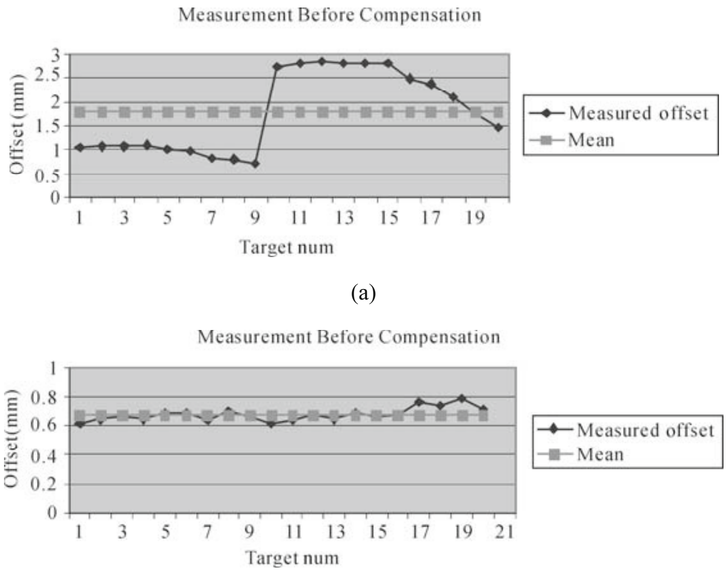


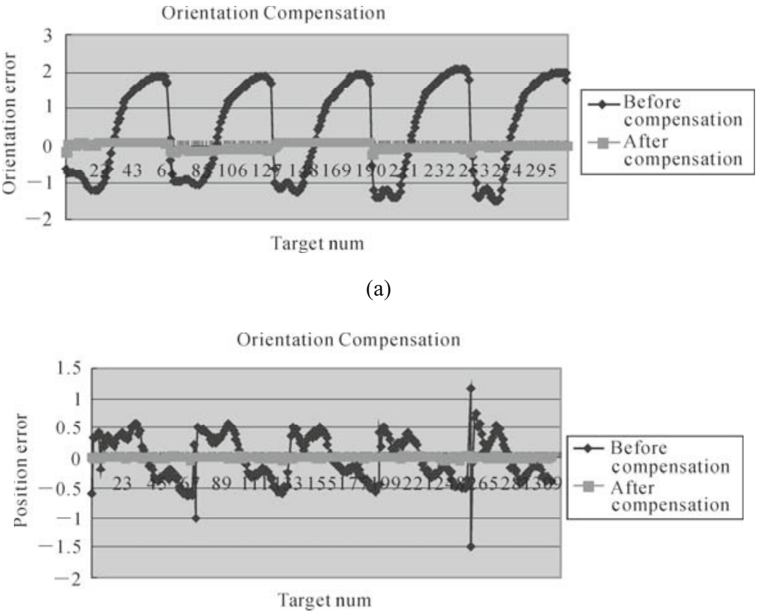
Fig. 7.16. Measured targets/paths



(a)

(b)

Fig. 7.17. Offset measured before/after compensation. (a) measurement before compensation; (b) measurement after compensation



(a)

(b)

Fig. 7.18. (a) orientation error before/after fine tuning; (b) Position error before/after fine tuning

7.2.6 Conclusion and Remarks

In this section, a robotic grinding system for a free-form workpiece is presented. The robotic kinematic chain is closed with offline programming and the in-process calibration method. A 3D laser scanner is used to measure the surface of the workpiece and the registration method is adopted to locate the workpiece. Instead of calibrating the robotic system itself, two novel easily implemented task-oriented methods are introduced to compensate for the system error in the grinding process. Experiments are conducted to demonstrate the effectiveness of the methods.

7.3 Robotic Remanufacturing of Blade Tip Refurbishment

A profile modeling based grinding approach is presented and discussed in this section. This approach is applied to blade tip profile refurbishment in the aerospace overhaul industry. Online profile calibration and fine-tuning methods are adopted to generate an accurate processing path for different parts that are deformed after a few years service under severe conditions. Demonstration experiments are developed with a robotic grinding workcell. A 3D laser scanner and LVDT in the robot workcell are used for measuring fine-tuning and online quality control. Experimental results indicate that this profile modeling based grinding approach is acceptable for the blade tip refurbishment task.

7.3.1 Introduction

Turbine blades/vanes are key parts in the aeronautic, astronautic and power generation fields. After service at high temperature and in high pressure environments, the blades/vanes are severely worn and distorted and cracks due to heat fatigue often form on their airfoils. Repairing the old blades/vanes significantly saves costs when compared to replacement with new parts. Hence blades/vanes refurbishment is the method of choice in the repair process. This field is called the aerospace overhaul industry.

7.3.1.1 Traditional Grinding Process

Grinding is a key process in the refurbishing procedure that is used to remove the extra materials after covering a layer of braze material by welding. To date, manual operations are dominant in the grinding of turbine blades/vanes. Skilled workers remove the excessive braze material using abrasive belt grinding and polishing to restore the original profile of the airfoil manually. Poor efficiency and inconsistent product quality are inherent in the manual grinding process, and this manual grinding operation exposes operators to high noise levels and an environment full of metal dust.

To improve the product quality, some researchers apply CMM for measuring and the CNC machine for repairing. This approach can achieve high measurement precision but it is excessively time consuming and the process is inflexible. In addition, tool changing introduces uncertain errors that lower the performance of this approach. To reduce the disadvantages mentioned above, a robotic grinding process is applied by researchers and manufacturers. This process has numerous advantages, such as it can improve the working environment, guarantee the safety of the human operators, provide highly consistent and repeatable quality, and save time and money.

7.3.1.2 Robot Grinding

The stiffness of a robot is significantly lower when compared with a CNC machine. This may result in unacceptable quality and lower productivity, hence it is very important to design an approach to compensate for this effect.

Force control is adopted by most researchers when applying robot grinding. Zeng (1997) summarized most existing robot force control algorithms and indicated that robot force control involved the integration of task goals like modeling the environment, position, velocity, force feedback, and adjustment of the applied torque to the robot joints. Giblin (2007) applied target tracking theory, or combined force and position control, in open and closed loop manipulators, and demonstrated the theory in simulation experiments with both serial and parallel manipulators. Wang (2001) monitored torque by an external DC observer-motor in polishing applications and determined an ideal grinding condition for this experimental setup.

The force control approach focuses on a more efficient filter and estimates, better feedback strategy choices, faster learning capabilities and stronger robustness. This proves the theory and achieves effective simulation but it is hard to put into practice in an industrial environment because it is very hard to model a real grinding process.

Beside force control, specially designed tools with passive compliance are applied on a robot wrist or grinder to fit the position error of the parts. For example, some grinders and milling motors (spindle) can provide a consistent preset contact force by an air floating mechanism.

7.3.1.3 Turbine Blade/Vane Refurbishment

Refurbishment is a kind of difficult grinding application, due to the difference between parts. Serving in high-temperature and high-pressure environments will lead to severely worn and distorted parts. And after welding process, the difference becomes huge and the surface features are hard to measure. Fig. 7.19 shows the blades and vane to be ground. Braze material covers the surface of the parts and the thickness is not consistent over different areas.

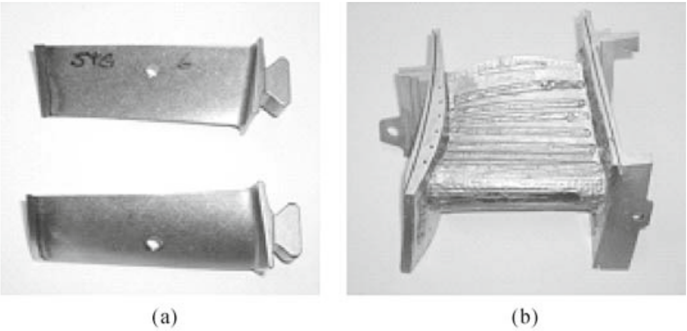


Fig. 7.19. Blade (a) and vane (b) to be ground

To grind the blade and vane, there are three items that should be established: position, orientation and finishing condition. Normally, position and orientation can be determined by measuring and fitting the workpiece to the CAD model, and the finishing condition is based on either knowledge or online quality testing. For a vane, Huang et al. (2003) applied an LVDT to measure the surface position and match the CAD model with a template-based optimal profile fitting algorithm. Then, a knowledge database was adopted to achieve process control and determine the finishing condition.

For a blade, it is harder than a vane based on the following effects:

(1) The whole aerofoil surface of a vane must be ground. Grinding to a consistent profile base on a CAD model will lead to a perfect aerofoil. But for a blade, only tips and edges are covered by a braze which needs to be ground, hence the grinding positions are based on the uncovered surface instead of on the CAD model. It's hard to deal with the boundary between the part to be ground and the part left untouched.

(2) The blades are cantilevers. They have less stiffness and it is hard to apply process control and determine the finishing condition.

To meet the demands of blade refurbishment, a profile modeling based grinding process is presented in this section. The approach of this process will be introduced in the next sub-section. Then the setup of a demo workcell will be described in Section 7.3.3. In Section 7.3.4, the experimental results, conclusions and future work will be presented.

7.3.2 Profile Modeling Based Grinding

After investigation of the used blades, it can be seen that blade tips get distorted and twisted and local material get lost but the original profile can be retained.

Hence the profile model of a used blade tip can be obtained by comparing the measured value with the CAD model. And then a precise process target and path can be obtained by a fine tuning process. Fig. 7.20 shows the processes involved in robotic blade refurbishment.

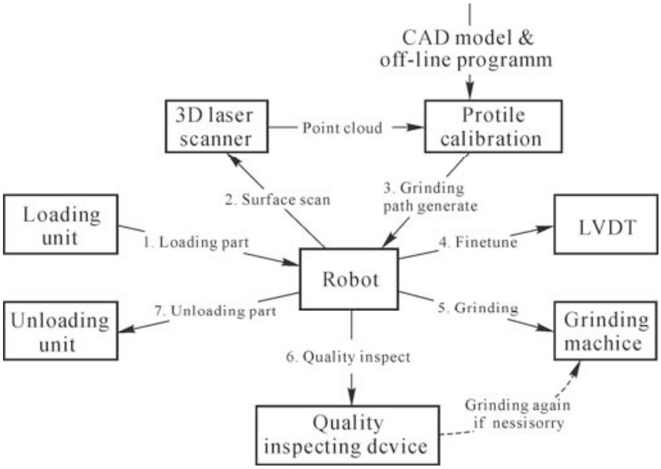


Fig. 7.20. Robotic blade refurbishment processes

(1) The blade tip surface is scanned by using a 3D laser scanner and the point clouds are obtained;

(2) The CAD model is aligned with measured point clouds by using global registration algorithm, resulting in a transformation matrix that applies to the CAD model;

(3) Reference profile is generated from the CAD model.

The reference profile is used to generate the robot grinding path. It includes two separate paths, the concave path and the convex path to enclose the blade cross sectional profile. On each path, 50 – 100 targets will be sampled. Then the robot grinding path will be generated by combining all the targets in both paths (Fig. 7.21).

For a blade tip, an original surface is a smooth extension of the uncovered surface by brazing. This is what we need to re-generate by using grinding process. Normally, this surface is not the same as in the CAD model due to the material lost after usage. Hence, a fine tuning process should be applied to compensate for the position error.

The robot follows the grinding path by setting the LVDT as a dummy grinding tool. With the values from LVDT, an error map can be drawn. Then, a final processing path is generated by compensating the original path with an error map.

The robot follows the grinding path several times to the grinding belt until there is no more material that can be removed. Grinding times are preset as finishing conditions based on prior testing, and an online quality control process is performed by scanning the grinded surface and by re-grinding if necessary to guarantee good product quality.

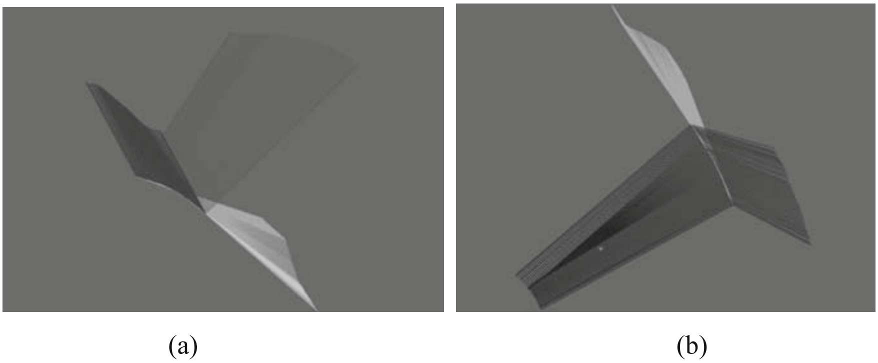


Fig. 7.21. Robot grinding path for (a) convex part and (b) concave part

7.3.3 Experimental Setup

A robotic grinding workcell is set up to demonstrate a profile modeling based grinding approach. A series of small turbine blades (shown in Fig. 7.19(a)) will be processed. And the parameters of the blade are listed in Table 7.2.

Table 7.2 Parameters of turbine blade

Parameters	Values (mm)
Width	30
Length	50 – 80
Thickness	<0.4
Tip width for grinding	3 – 5
Material removal	<0.2

An ABB140 robot is adopted to hold the blades by a pneumatic quick change gripper, as shown in Fig. 7.22. With this gripper and a tool stand, the operator can load and unload parts without interrupting the robot grinding process. A belt grinder, as shown in Fig. 7.23, is used, which provides the contact force adjusted by an air floating mechanism and supports the function of continuously adjusting the grinding belt speed and the contact force by robot controller.

A 3D laser scanner, as shown in Fig. 7.24(a), is used to create 3D cloud points, as shown in Fig. 7.24(b). All these equipments are set up in an enclosure with a light and vacuum system. With this system, a blade can be processed in less than 5 min. And the final quality meets the manufacturing demand very well.

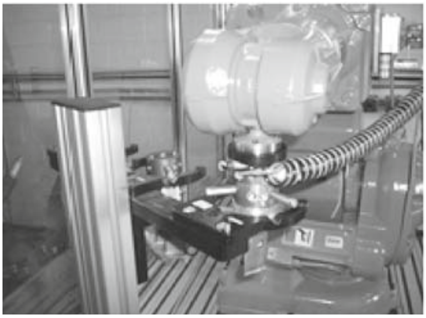


Fig. 7.22. Robot with gripper

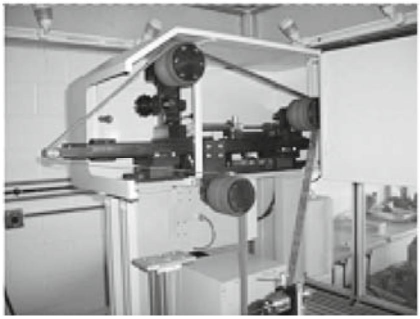
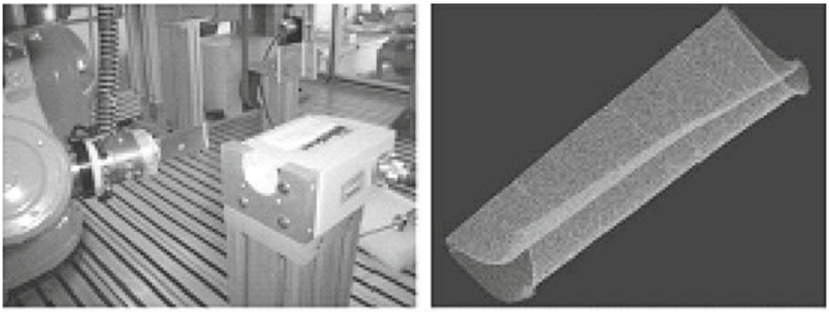


Fig. 7.23. Grinding machine



(a)

(b)

Fig. 7.24. (a) 3D laser scanner is used to scan the surface of the blade; (b) Create point of cloud data

Fig. 7.25 shows the repaired blade after grinding. It indicates from the experimental results that this approach works well for the blade tip refurbishing task. However, there is still some additional work to be done to improve the quality in the future.

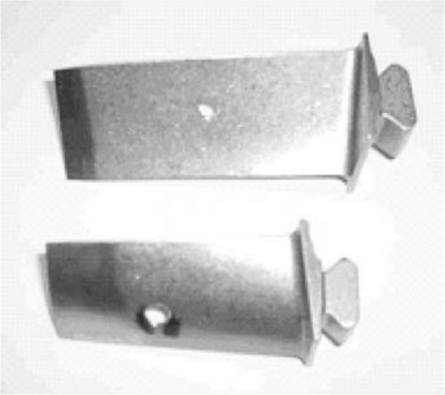


Fig. 7.25. Refurbished blades after grinding

7.3.4 Conclusion and Future Work

With high precision, robotic milling is an alternative way for refurbishing blades and vanes. Since the robot cannot provide good stiffness for the milling process, real time stiffness is a good topic for enhancing robotic applications. Some researchers have already studied this issue (Zhang, 2005) with a novel methodology that consists of stiffness modeling and real-time deformation compensation. And a parallel robot is another way to provide good stiffness and precision.

7.4 Robotic Materials Handling System for Complex Parts

Robots in materials handling have been used in industry to increase productivity and profitability. The typical examples include robotic palletizing systems.

Workpieces are laid in certain pre-defined loading places with the accuracy usually provided by workpiece loading locators. Robots usually repeat the pre-taught program for handling the workpieces using fixed procedures. In such kinds of applications, to ensure a certain materials handling accuracy, the accuracy of locating workpieces, grippers and the loading operations need to be all guaranteed, which requires high accuracy in the machining and assembling of mechanical parts.

Some intelligent robotic systems have been developed to enhance the robot performance in materials handling. In such systems, the strict restriction on the position of the workpiece is no longer required. Vision sensors are working with the robotic system to compensate for workpiece positioning error. Examples are presented in the following sections.

7.4.1 System Overview

This section presents a sand core handling/assembly system, which assembles sand cores in sand boxes to form sand molds for sand casting with an accuracy requirement of ±0.3 mm. In this system, the sand core and the sand box are loaded onto two conveyors, with no strict constraints being enforced for the position of parts. The sand core and box are transported to the working positions for a robot to pick up the sand core and then assemble it in the sand box. Because of loading and transportation error of the conveyors, a positioning error for the core/box in the working position can easily reach ±50 mm. To compensate for the error, a laser based 3D vision system is integrated in the robotic system to guarantee high accuracy materials handling and assembly.

Here is an example of ideal workpiece location and handling:

(1) Load a master workpiece onto the working position and then instruct a robotic handling program based on the master workpiece. Scan the feature area of the master workpiece to obtain the point cloud data for later use.

(2) When the other workpieces are loaded, the workpieces are scanned to get the point cloud data and then compared to the point cloud data of the master workpiece to determine the positioning error of the current workpiece.

(3) The robotic handling program for the master workpiece is then adjusted with positioning error compensation to generate a handling program for the current workpiece.

Due to the large variation in the position of the workpieces (cores/boxes), the positioning error is compensated for by two steps:

(1) A point laser is used to approximately locate the core/box by scanning the edge of their bases (refer to next section for details).

(2) A 3D laser scanner is used to scan the feature area of the core/box. By using 3D point cloud registration algorithms the position error between the current core/box and the master core/box can be determined. The reference core/box is used for the teaching of robotic materials handling programs. Once the error is known, the workpiece pick-up and drop-down robotic programs for the master core/box can be adjusted for the current core/box.

Fig. 7.26 shows a sand core handling and assembly system that consists of a robot, a gripper and a conveyor.

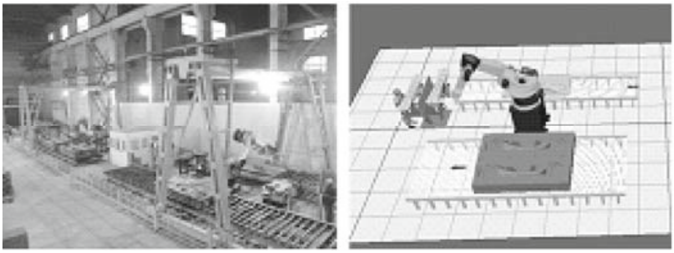


Fig. 7.26. A sand core handling/assembly system. (a) Photo of an installed system; (b) Materials handling workcell

7.4.2 Approximately Locating Workpieces

To locate the cores/boxes, their feature areas are scanned and the corresponding point clouds are compared to the point clouds of the master core/box. To obtain point clouds, a 3D laser scanner is integrated in the robotic system. However, because of the large variation in the core/box position (i.e. ±50 mm) and the limitation of the field of view of the 3D laser scanner (several centimeters), the same scanning program for workpieces located in different positions is not able to scan the same feature area of the core/box. Therefore, the scanning program should be adjusted before it is used to scan different core/box.

For different core/box loadings, because of the positioning error, the position/orientation of the core/box base coordinate is changing. To obtain the position/orientation of the core/box, a point laser is applied for the roughly positioning.

During the core/box base locating process, the robot holds the point laser and moves through three pre-defined paths with the point laser projecting to the surface of the core/box base, as indicated in Fig. 7.27. When the robot moves to points 1, 2 and 3, as shown in Fig. 7.27, an I/O signal “1” will be triggered by the point laser. At that time, the robot positions can be recorded. If the robot keeps the same orientation while moving and the scanning paths 1, 2 and 3 are on the same *Z* plane parallel to the top surface of the core/box base (assume it is a *Z* plane), the coordinate of the core/box base can be easily defined, based on the 3 recorded robot positions.

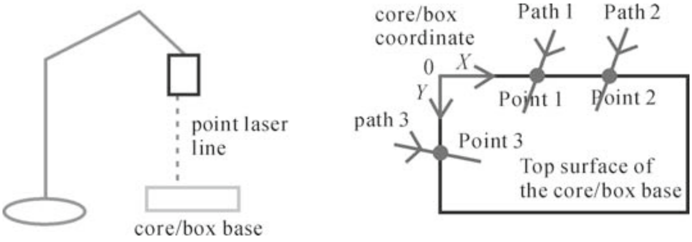


Fig. 7.27. Core/box base rough location

Notice that we do not know the exact plane of the surface of the core/box base in each loading, the 3 paths can only be approximately taught. Furthermore, other errors, such as the edge detection error and the error caused by placing the core/box onto the base, exist. Therefore, the overall accuracy of the core/box location after the approximate location can only reach several millimeters.

7.4.3 Precisely Locating Workpieces

Since the approximate location of the core/box can reduce the positioning error from about 50 mm to several millimeters, after the approximately locating the scanning program can be updated to scan the feature area of the core/box. With the scanning program, point clouds of the current core/box can be obtained which are reconstructed to the robotic base coordinate with a kinematic relationship

$$P_b = Tool_U \cdot T_{\text{Scanner}} \cdot P_S \tag{7.16}$$

where P_S is the coordinate of the scanned 3D points in the scanner coordinate, which can be obtained from the scanner; T_{Scanner} is the position/orientation of the sensor with respect to the base coordinate of the robot, which is calibrated in advance; P_b is the measured point in the robot base coordinate.

The positioning error of the current core/box with respect to the master core/box can be obtained by comparing the 3D point clouds. Registration algorithms (Chen, 1991; Pottmann, 2006) can be applied to calculate the transformation matrix between the two sets of point clouds, as indicated in Fig. 7.28.

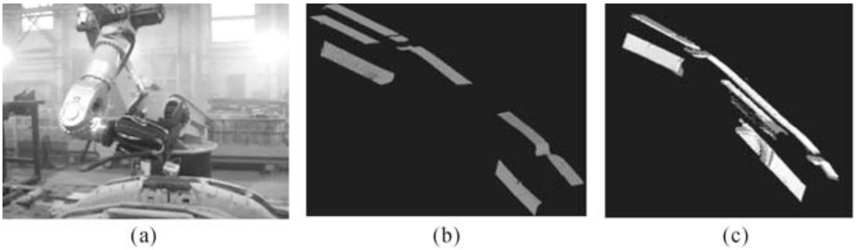


Fig. 7.28. (a) Feature area scanning for sand; (b) Scanned point clouds; (c) Registration of two sets of point clouds

After registration, the calculated transformation matrix can be applied to update the pick-up/drop-down program for the robot to pick up the cores and drop the cores into the boxes, as indicated in Fig. 7.29.

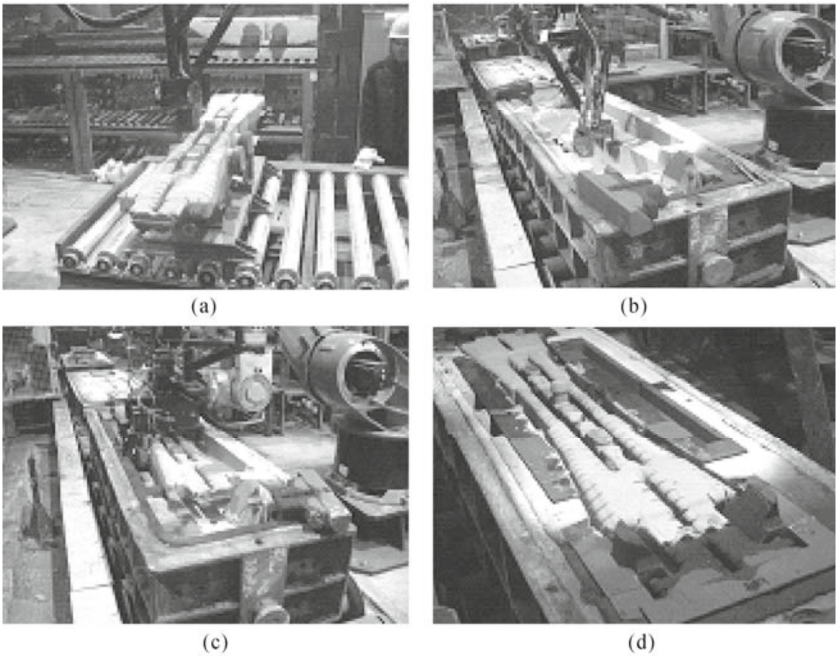


Fig. 7.29. Process to assemble sand cores in sand boxes. (a) Scan of the sand core; (b) Scan of the sand box; (c) Sand core drop-down to the sand box; (d) Assembled sand mold

7.4.4 Another Example

Fig. 7.30 shows another materials handling robotic system. It is a robotic workcell for pulling the hoses off the mandrel. Besides other processes, the key point here is to determine the hose position and orientation so that the gripper can engage with the hose. This can be done by using a 3D laser scanner to generate the 3D model for every mandrel to be used in this cell. At the same time, the robot path will be generated based on the scanning results. A tool station holds different grippers designed for different hoses. Quick changers will be used in order to change the tools easily. A rotary table, which can rotate through 180°, will transport the cylinders which hold the hoses for the loading and unloading. The other two indexing tables will be used to index the hoses to the position for the robot to reach.

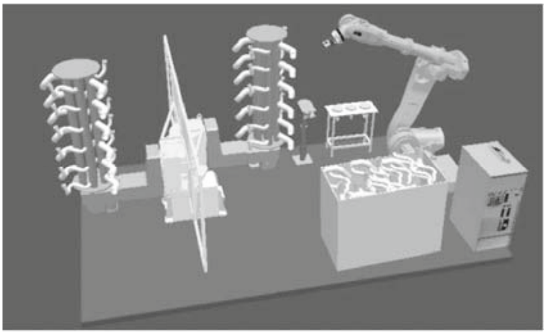


Fig. 7.30. Robot workcell to pull the hoses off the mandrel

Fig. 7.31 shows the process to pull the hose off the mandrel including laser scanning, 3D modeling and path generation, and pulling operations.

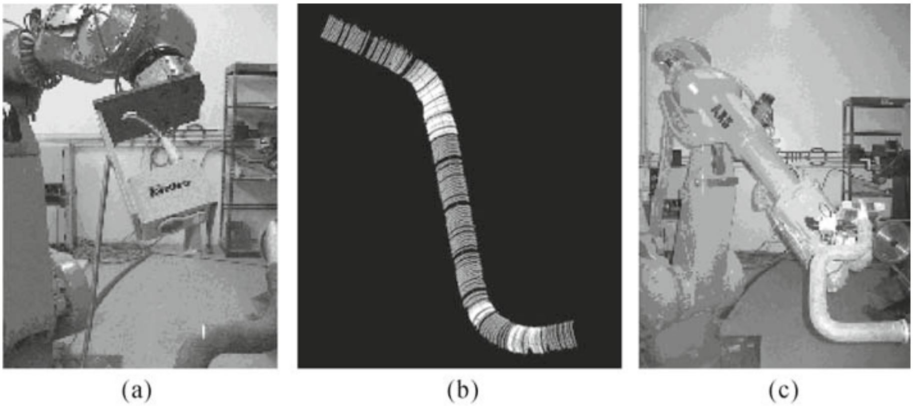


Fig. 7.31. Robot hose pulling process. (a) Laser scanning; (b) 3D modeling and path generation; (c) Pulling operation

7.4.5 Summary and Remarks

In this section a robotic vision system based work piece handling and assembly system is presented. The key task here is how to accurately locate the work piece that lacks of geometry feature, such that the gripper can pick the part. The point laser sensor and the 3D laser scanner are integrated with the robot to identify and handle the work piece. The point laser sensor is used to roughly locate the part on the conveyor to ensure the scanning takes place for the desired feature surface of the part. The laser scanner is used to obtain point clouds of the feature surface of the part and the global registration algorithm is used to accurately locate the work piece by comparing the point clouds data for the current work piece and the master piece. It has been demonstrated that global registration algorithm is very effective approach of identification and localization for the work piece that lacks of geometry feature and where feature extraction algorithm fails.

7.5 Robot Machining System with Visual Feedback

In traditional robot applications, workpieces to be machined are considered to be the same in geometric shape. Normally, “teach and play” schemes are used in such applications: a robot program is taught based on a master workpiece and then stored in the robotic controller. During the real production, the taught robot program is executed again and again to repeat the machining process. However, these kinds of schemes fail when applied to workpieces that vary in geometric shape, even when the variation is just on a small scale. To compensate for the shape variation error, the application of feedback is a good choice. With the feedback, it becomes possible for the robotic system to generate the machining program in process for each individual workpiece. This section will present a visual feedback based robotic solution for workpieces with geometric shape variation.

7.5.1 Introduction

To decorate and protect the edge of a guitar, ribbons are usually embedded in the perimeter of a guitar frame. Before embedding the ribbon, a groove needs to be cut around the perimeter to make room for the ribbon to fit in. The accuracy requirement for the groove is normally 0.15 mm. Because the shape of the border is like a free form curve, to cut a groove along it is not an easy job. Usually, border cutting work requires very skillful workers with complex cutting machines. The challenge in developing an automation system for this process lies in the fact that the workpieces vary one from another in geometric shape. Therefore, machines like traditional CNC, which do not have feedback for the workpiece, do not seem to be capable of the cutting work. A robotic system with visual feedback will be presented in the following sections. It applies a 3D laser scanner to scan the border of the guitar frame to obtain the 3D datum and then generate a robotic program based on the scanned data to cut the guitar. In the system, the geometric variation of the workpiece is well compensated for by the visual feedback which guarantees the quality of the manufacture.

7.5.2 System Overview

Fig. 7.32 shows the layout of the robotic guitar cutting system. A laser scanner and a cutting machine are fixed in the workcell, whose position/orientation relative to the base frame of the robot is pre-calibrated. A guitar is gripped by the robot. A pre-taught scanning program is run by the robot after it picks up the guitar. In this process, the 3D information along the border (corner) of the guitar frame is scanned by the laser scanner. After the scanning, path generation algorithms can be used to generate a machining path for the robot to cut the guitar.

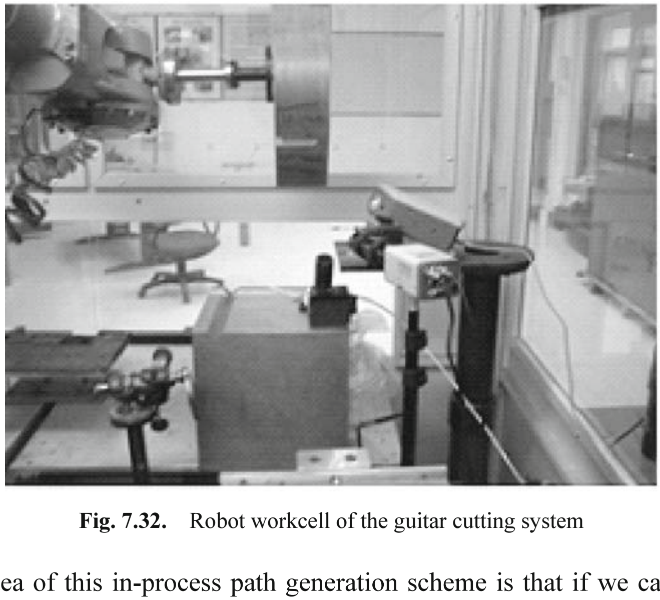


Fig. 7.32. Robot workcell of the guitar cutting system

The idea of this in-process path generation scheme is that if we can scan the 3D points along the corner of the guitar and then detect the edge point, then we can calculate the targets (coordinates with respect to the work object coordinate to be reached by the tool during the manufacturing process) for the cutting machine to reach.

7.5.3 Scanning and Edge Detection

In the scanning process, the laser scanner and the robot are synchronized to do the measurement. During the running of the robot, the robot *Tool*$_0$ (the position/orientation of the flange plate coordinate with respect to the robot base coordinate) is recorded and, at the same time, the 3D points on a laser line are detected and reconstructed by the laser scanner. With the following equation, the scanned 3D points can be restored into the *Tool*$_0$ coordinate.

$$P_T = Tool_0^{-1} \cdot T_{\text{Scanner}} \cdot P_S \qquad (7.17)$$

where P_S is the coordinate of the scanned 3D points in the scanner coordinate, which can be obtained from the scanner; T_{Scanner} is the position/orientation of the sensor with respect to the base coordinate of the robot, which is calibrated in advance; P_T is the measured point in the *Tool*$_0$ coordinate.

As such scanning goes on, the 3D points along the corner of the guitar frame can be scanned and stored. Fig. 7.33 shows the point cloud of the guitar corner that is obtained by the scanning.

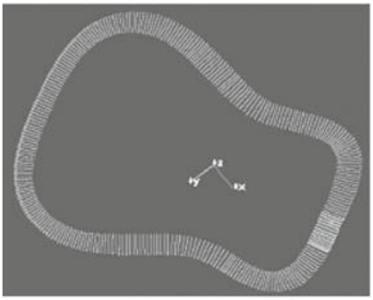


Fig. 7.33. Scanned point cloud of the corner of the guitar frame

The process of scanning is to obtain the point cloud of the corner. The step that follows is then for detecting the edge of the guitar, based on which the robot program can be generated. The edge point on the guitar can be obtained by calculating the intersection point of the two line section of one laser line, as indicated in Fig. 7.34.

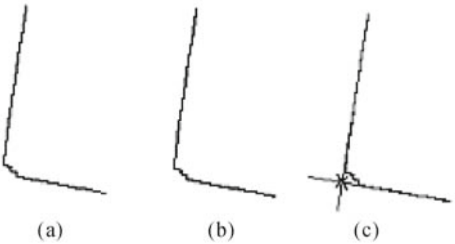


Fig. 7.34. Edge point detection. (a) Points on a laser line; (b) Two line segments extracted from the laser line; (c) Intersection of the two line segments

7.5.4 Path Smoothing Based on the B-Spline

After the detection of the edge points, the targets on the work object for the cutting process can be generated by these rules:

(1) The positions of the edge points are set as the positions of the targets.

(2) The vector calculated by the two adjacent edge points ($X_{0i} = \dfrac{(P_{i+1} - P_{i-1})}{\left\| P_{i+1} - P_{i-1} \right\|}$) is considered as the initial *X direction* of the targets.

(3) The normal of the surface containing the reconstructed 3D points on the top/back board is considered as the *Y direction* of the targets.

(4) The Z direction of the target can be calculated as $Z_i = X_{0i} \times Y_i$ and the X direction can be finally updated as $X_i = Y_i \times Z_i$.

However, because of the presence of the measurement error and edge point detection error, the program generated following the rules mentioned above may not be a path smooth enough for the cutting process. Algorithms should be adopted to improve the path smoothness.

The B-spline refers to a spline curve parameterized by spline functions that are expressed as linear combinations of basic spline curves. It is used to express the smooth free-form curve. In the robotic application, to improve the path smoothness, B-spline curve fitting and point re-sampling algorithms can be adopted (Piegl, 2000). For detail please see Appendix.

After the B-spline fitting, 3D points can be sampled from the curve to create the smooth robot path, as shown in Figs. 7.35(a) and (b). And instead of calculating the X axis of the targets with the two adjacent edge points as mentioned above, tangent directions can be calculated from the B-spline curve and used as the X axis. Fig. 7.35(c) shows the guitar with the edge grooved by using the generated robot program.

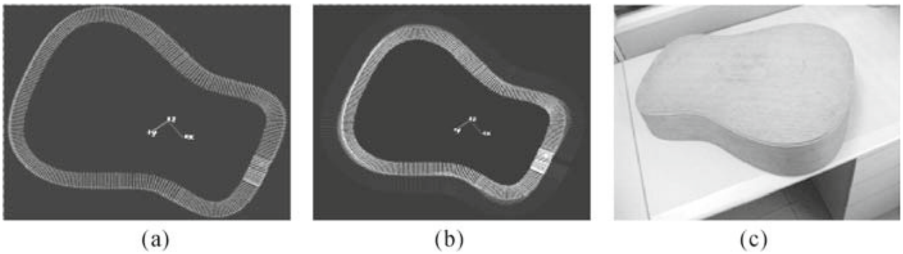


Fig. 7.35. (a) B-Spline curve fitting based on the measured edge points; (b) Target generation from the fitted B-spline; (c) Guitar with edge that has been grooved by using the generated program

7.5.5 Other Examples

The following two examples demonstrate how the laser scanner is used to determine the work object position for robot machining applications.

The first one is a robot water jet cutting system (as shown in Fig. 7.36), which is used to remove the extra casting materials of the blade. The system basically consists of an industrial robot, a water jet cutting system and fixtures. A water jet cutter is held by the robot and used to cut the casting blade. The tip of the cutter needs to keep the same distance from the non-planar surface during the cutting process and the cutting path is a 3D path, as shown in Fig. 7.36(b). The laser scanner is used to scan the surface of the blade in order to determine its position before the cutting process takes place. The scanned point of clouds data of the feature surface is compared with the reference point of clouds data, or CAD model, to calculate the transform of the object relative to the reference position through the range data registration algorithm. The process is illustrated in Fig. 7.28

The other example is a robot fender paint system that is used to paint a closed strip (two parallel lines) on the surface of a motorcycle fender, as shown in Figs. 7.38 (a) and (b). The robot holds a paint tool and moves along a curved path in a non-stop continuous motion to finish the painting in Fig. 7.38(b). On the tip of the tool, there are two parallel painting wheels, which allow two parallel lines of the strip to be finished simultaneously. The paint tank, which is located above the paint wheels, will continuously feed paint to paint the wheels under the effect of gravity. Therefore, the paint tool should be held close to normal of the surface. The important factor here is the orientation accuracy, because inaccurate orientation will cause the paint tool to tilt unevenly and therefore the two parallel paint lines will have an uneven finish. A laser scanner is first used to determine the work object position through the global registration algorithm, as shown in Fig. 7.38(c). This is to scan the surface of the fender and compare the scanned point of clouds data with the referenced one to calculate the position of the fender. Another task of the scanner is to measure the normal of the local surface along the painting path so that the paint wheel direction can be kept normal to the surface.

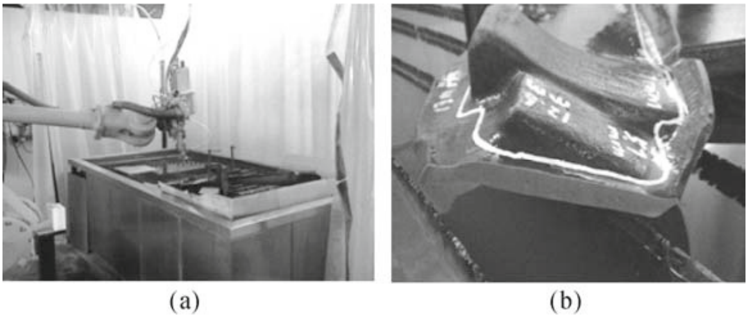


Fig. 7.36. (a) Water jet cutting system that consists of a robot and a water jet cutter; (b) Object (casting blade) to be processed and a 3D cutting path

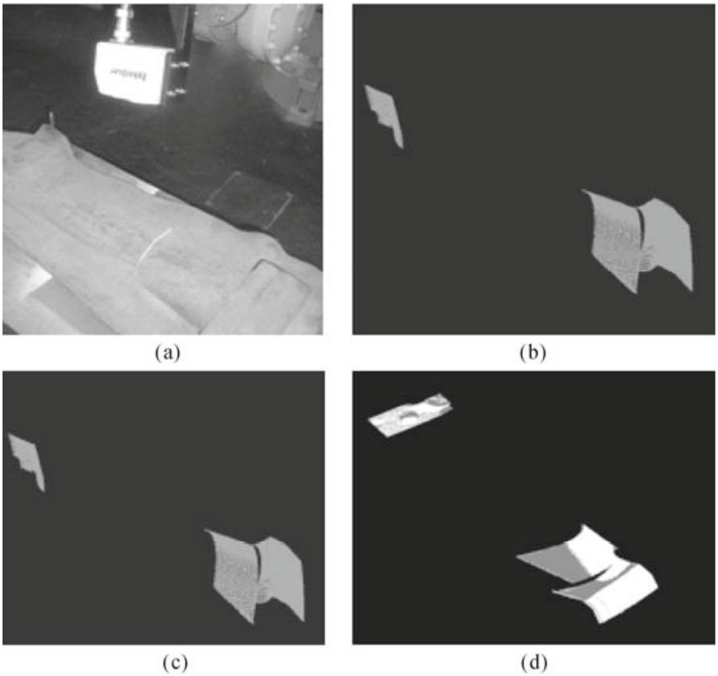


Fig. 7.37. Registration algorithm is used to determine the object position. (a) Surface scanning; (b) Scanned point of clouds of the object; (c) Point clouds of reference object; (d) Transform between the current blade position and the reference position is determined by using registration algorithm

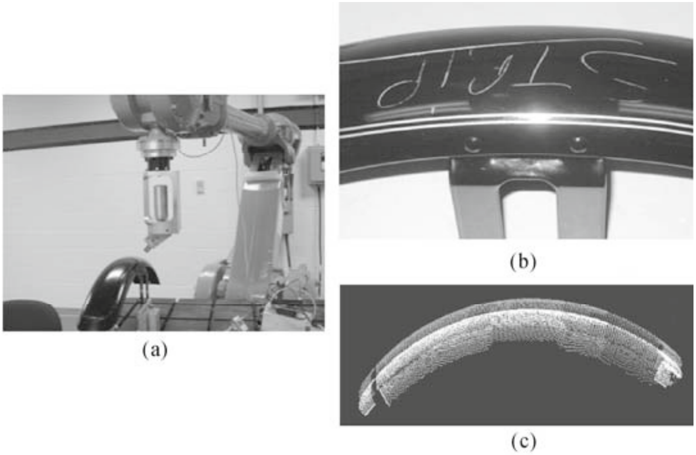


Fig. 7.38. (a) Robot fender paint strip system that consists of a robot and a paint tool; (b) Finished painting pattern that is a closed strip on the surface with two parallel lines; (c) Comparison of scanned point of clouds data with reference data to calculate the fender position

7.5.6 Summary and Remarks

In this section a vision based robotic machining system is presented and discussed that can compensate for the shape variation of the part. With the visual feedback, the robot generates the machining program in process for each individual workpiece. Integration of the laser sensor to the robot system makes the robot more flexible and operative as a machining tool as well as a measurement tool.

7.6 Robotic Measurement and Inspection System for Quality Control

Robots are normally used as executors which execute the commands (programs) from operators. However, during the execution, robots can also provide feedbacks on their joint positions and, furthermore, the Cartesian position of their tips ($Tool_0$). These feedbacks enable robots to be used as a measurement system for production inspection and quality control (Pastorius, 2009; Reinhart, 2009).

The robotic measurement system usually consists of a robot and a sensor (or several sensors). The sensor(s) can be mounted on the robot tip or fixed in a certain position in the world coordinate. On the other hand, the workpiece to be measured can be fixed in a certain position in the world coordinate (in the sensor-held-by-robot case) or mounted on the robot tip (in the sensor-fixed case). During the measurement, the workpiece is measured by the sensor and at the same time the position of the robot is recorded. With the pre-calibrated position/ orientation information from the sensor, the coordinate of the workpiece can be reconstructed with the kinematic relationship. However, the relatively low absolute accuracy of the robot system itself limits the performance of the robotic measurement system. It is not surprising to see an error of one millimeter or even several millimeters appear in the measurement results.

As we know, the robot has much better repeatability than accuracy. Taking advantage of its repeatability (around 50 μm for some commercial robots) makes it possible to develop relative measurement systems with high accuracy for some applications. In the following section, a high accuracy relative measurement robot system is presented, which is used to measure the material removal of free-form workpieces (faucets) in the grinding process.

7.6.1 System Overview

The presented robotic measurement system is used to measure the material removal on the surface of a faucet (free-form surface) after it has been ground.

The measured material removal data is useful for the adjustment of the grinding parameters for the faucet grinding. The material removal data is calculated based on two measurements of the same surface before and after grinding. Therefore, if the robot programs for these two measurements of the surface are the same, the robot system error involved in one measurement tends to cancel the other one out, which makes the measurement essentially a relative measurement.

As shown in Fig. 7.39, the system consists of a robot (which holds the faucet), a point laser sensor and a 3D laser scanner. The measurement operation is as follows:

(1) Before grinding a faucet is picked up by the robot. Feature points of the faucet are measured by the point laser. The point clouds of the faucet surface are obtained by using the laser scanner. The point laser can provide high accuracy measurement to ensure the accurately locationing. The laser scanner provides fast surface measurement.
(2) After the first measurement, the faucet is dropped and then ground by the faucet grinding system.
(3) After the grinding, the faucet is picked up by the robot and measured again.
(4) Based on the measured feature points, a transformation matrix between the two mounted positions of the faucet can be calculated. With the transformation matrix, the two measured point clouds can be put into the same coordinate and finally the material removal can be calculated.

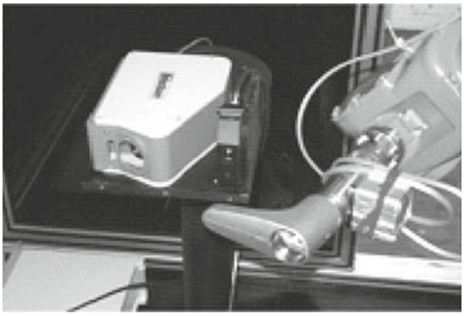


Fig. 7.39. A robotic system for material removal measurement

7.6.2 Pick-up Error Compensation

As mentioned above, the faucet is picked up twice (before and after grinding) during the whole measurement process. Therefore, the pick-up error should be compensated before the two measured point clouds of the faucet can be compared.

A coordinate can be defined on the faucet, based on the area that is untouched during the grinding process. The two holes shown in Fig. 7.40 are a good choice, since they are untouched during grinding and can also provide enough information for coordinate setup. The coordinate system can be defined by using two points and a directional vector. The two points are the centers of the two holes on the faucet and the directional vector is the normal of the plane where the larger hole intersects.

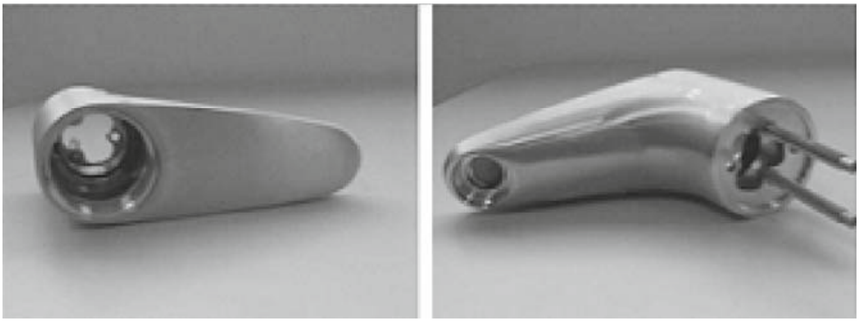


Fig. 7.40. Features on the faucet

Assume the center points of the small hole and large hole are P_1 and P_2, the normal of the plane where the larger hole intersects is $\boldsymbol{V}$, then the coordinate on the faucet can be set up as:

$$P = P_2, \quad \boldsymbol{X} = \frac{P_1 - P_2}{|P_1 - P_2||}, \quad \boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{V}, \quad \boldsymbol{Z} = \boldsymbol{X} \times \boldsymbol{Y} \tag{7.18}$$

where P is the position of the origin of the coordinate system; $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ are the $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$ axes of the coordinate system.

During the measurement, the faucet is held by the robot and the measurement data is converted to $Tool_0$ coordinate with the kinematic relationship

$$P_T = Tool_0^{-1} \cdot T_{\text{Sensor}} \cdot P_S \tag{7.19}$$

where P_S is the coordinate of the measured point in the sensor coordinate (either point laser or laser scanner), which can be obtained from the sensor; T_{Sensor} is the position/orientation of the sensor with respect to the base coordinate of the robot, which is calibrated in advance; $Tool_0$ is the position/orientation of the flange plate with respect to the base coordinate of the robot, which can be read from the robot controller; P_T is the measured point in the $Tool_0$ coordinate.

A point laser with resolution of 2 μm is used for the measurement of the area with features (holes and the plane normal) to provide information for high accuracy part locationing. With the coordinate setup method mentioned above, two coordinates can be defined on the faucet after it is gripped, before and after grinding as

$$^{Tool_0}\boldsymbol{T}_{fi} = \begin{bmatrix} \boldsymbol{X}_i & \boldsymbol{Y}_i & \boldsymbol{Z}_i & \boldsymbol{P}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (i = 1,2). \tag{7.20}$$

where $^{Tool_0}T_f$ is the transformation matrix from the faucet coordinate; $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$

are the ***X***, ***Y*** and ***Z*** axes of the faucet coordinate.

Assuming the two point clouds of the faucet scanned before and after grinding (reconstructed respective to the *Tool*$_0$ coordinate) are PC_{t1} and PC_{t2}, they can be transformed to the faucet coordinate with equation

$$PC_{fi} = {}^{Tool_0}\boldsymbol{T}_f^{-1} \cdot PC_{ti} \quad (i = 1, 2) \tag{7.21}$$

where PC_{fi} is the point cloud of the faucet in the faucet coordinate.

Since the faucet coordinate is on the faucet itself, which remains unchanged before and after the grinding, the point clouds PC_{f1} and PC_{f2} are then put in the same coordinate for comparison.

7.6.3 Feature Based Workpiece Locationing

To compensate for the pick-up error, a coordinate should be set up on the faucet. Two feature positions and one directional vector can provide enough information for the coordinate setup. In the project, a feature area with two holes and one plane is scanned to obtain the two feature positions and one directional vector by feature detection.

Fig. 7.41 shows the scanned feature areas. Feature detection methods can be further used to determine the edge point of the holes. Circle fitting algorithms can be adopted to obtain the center points of the two holes, while a plane fitting algorithm can help the determination of the normal of the plane.

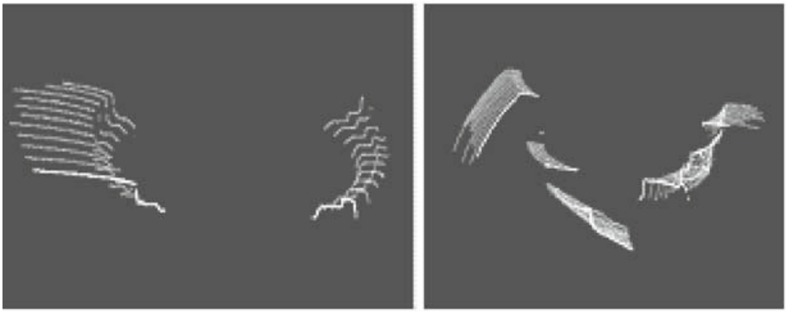


Fig. 7.41. Feature areas scanned by the point laser

7.6.4 Point Cloud Comparison

Once the pick-up error is compensated and the two sets of point clouds data of the faucet surface that are scanned before and after grinding are transformed to the same coordinate (the faucet coordinate), the point clouds are ready to be compared.

To calculate the difference between two point clouds, a relatively simple method is to reconstruct a surface from one of these point clouds, then calculate the distance between the surface (consisting of triangles) and the points in the other point cloud (Fig. 7.42).

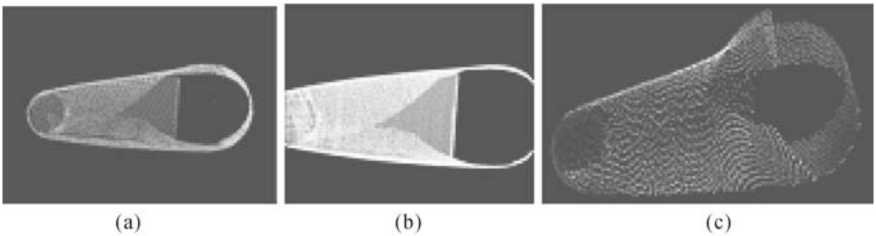


Fig. 7.42. (a) Point cloud of a faucet; (b) Surface reconstructed; (c) Calculated materials removal

7.6.5 Summary and Remarks

In this section a robot measurement and inspection system is presented in which a point laser sensor and a laser scanner are integrated into the robot work cell. Two issues in a typical robotic measurement and inspection system are addressed. One is how to reduce the robot system error and the other is to make coordinate alignment for data comparison. The measurement is based on the relative measurement principle in order to reduce the impact of the robot kinematic error. Alignment of two sets of measurement data is based on the coordinate transform where the coordinate system is generated by using the geometry feature.

7.7 Robot Weld System with Seam Tracking Sensors

In the automatic welding process, issues like part variation, part loading error and thermal distortion may largely affect the quality of the welding result, especially for long seam welding. Sensors have been integrated to enhance the system performance. Laser vision systems are examples of the sensors. These kinds of systems can detect the position of the welding seam and then, based on the detected position of the welding joint, the adjustment data is sent to the robot system to tune the robot welding program in real time (Agapiou, 1999; Fridenfalk, 2003). From the task level point of view, the laser sensing systems provide the vision feedback to the robotic system and close the control loop.

In the following subsections, a general robotic seam tracking system which can tune robotic poses with 6-DOF is presented and explained in detail, including the architecture of the system, the welding joint detection, the path generation algorithm, computer-robot communication, etc. A tube panel welding system which integrates a seam tracking system is also introduced, as an example of the commercial applications of seam tracking systems.

7.7.1 System Overview

Fig. 7.43 shows a robotic seam tracking system.

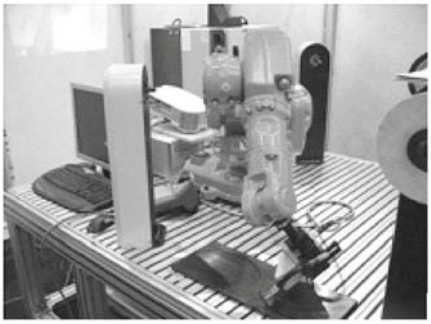


Fig.7.43. A robotic seam tracking demo system

In the system, the robot holds a 3D laser scanner as well as a welding gun. During the welding, the welding joint is detected by the laser scanner and then the robotic welding path is updated, based on the joint data. The seam tracking principle is illustrated in Fig. 7.44.

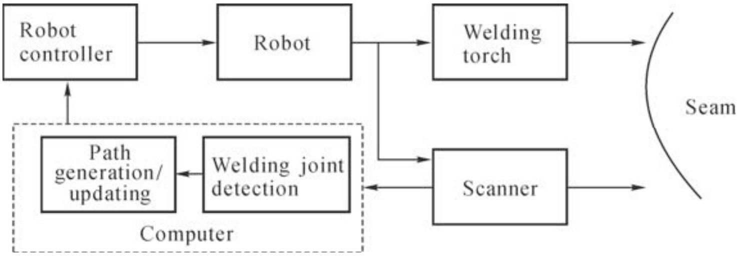


Fig. 7.44. Seam tracking principle

7.7.2 Welding Joint Detection

A low-power laser beam is projected onto the surface of the part and the reflected light is picked up by a CCD or CMOS sensor, after the laser line image is captured by the camera. Feature detection algorithms can be adopted to extract the welding joint. An example is shown in Fig. 7.45, which finds the V-type joint by calculating the intersection point of the two detected lines.

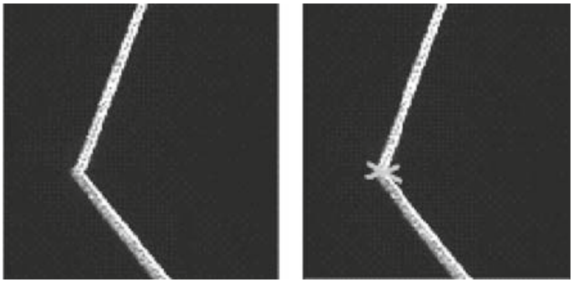


Fig. 7.45. V-type joint by calculating the intersection point

7.7.3 Path Generation

A robotic seam tracking system, which can tune the robotic welding path with 6-DOF, is presented. In this system, the laser scanner is used as a vision sensor to guide the welding torch. Once a laser line is projected to the welding joint, the coordinates of the points on the welding joint can be reconstructed on the robot base frame with the following equation

$$P_w = Tool_0 \cdot T_{\text{Scanner}} \cdot P_S \tag{7.22}$$

where P_S is the coordinate of the scanned 3D points in the scanner coordinate, which can be obtained from the scanner; T_{Scanner} is the position/orientation of the sensor with respect to $Tool_0$ coordinate of the robot, which is calibrated in advance; P_w is the coordinate of the measured points in the robot base coordinate.

The welding joint can be detected with feature detection algorithms. After that, smoothing algorithms can be adopted to smooth the path of the welding joint. Besides the B-spline based algorithm, the polynomial curve fitting algorithm is another option. With this algorithm, one can assume that the path is a polynomial curve

$$\boldsymbol{X}(t) = \boldsymbol{a}_n t^n + \boldsymbol{a}_{(n-1)} t^{(n-1)} + \ldots + \boldsymbol{a}_1 t + \boldsymbol{a}_0 \tag{7.23}$$

where $\boldsymbol{X}$ is the coordinate of the welding joint; t is the parameter of the welding joint; $\boldsymbol{a}_i$ are the control points to be calculated.

The scanned welding joints of number M can be parameterized with equations

$$\left.\begin{aligned} t_0 &= 0 \\ t_i &= \sum_{j=1}^{i} d(j-1,\ j),\ (i = 1,\ 2,\ \ldots,\ M-1) \end{aligned}\right\} \tag{7.24}$$

$$S = \sum_{i=1}^{M} d(i-1,\ i) \tag{7.25}$$

where $d(i-1,\ i)$ is the distance between welding joint i-1 and joint i.

With the parameterization, one can obtain a linear equation concerning $\boldsymbol{a}_i$

$$\begin{bmatrix} t_0^n & \cdots & t_0 & 1 \\ & \cdots & & \\ t_{M-1}^n & \cdots & t_{M-1} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{a}_n \\ \vdots \\ \boldsymbol{a}_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{X}_0 \\ \vdots \\ \boldsymbol{X}_M \end{bmatrix} \tag{7.26}$$

If $M > n$, the control points $\boldsymbol{a}_i$ can be solved.

After the path is smoothed, the targets can be generated along the path following certain rules such as:

(1) The sampled positions along the curve are set as the positions of the targets.

(2) The tangential direction obtained by

$$X'(t) = n\boldsymbol{a}_n t^{(n-1)} + (n-1)\boldsymbol{a}_{(n-1)} t^{(n-2)} + \ldots + \boldsymbol{a}_0$$

is considered as the initial X direction of the targets.

(3) The normal of the local surface of the sampled position is considered the Y direction of the targets.

(4) The Z direction of the targets can be calculated as $Z_i = X_{0i} \times Y_i$ and the X direction can be finally updated as $X_i = Y_i \times Z_i$.

7.7.4 Computer-Robot Communication

To track the welding joint in real time, a computer-robot communication protocol has to be defined. A dual-buffer method can be used to ensure the continuity of the robot welding path, as indicated in Fig. 7.46.

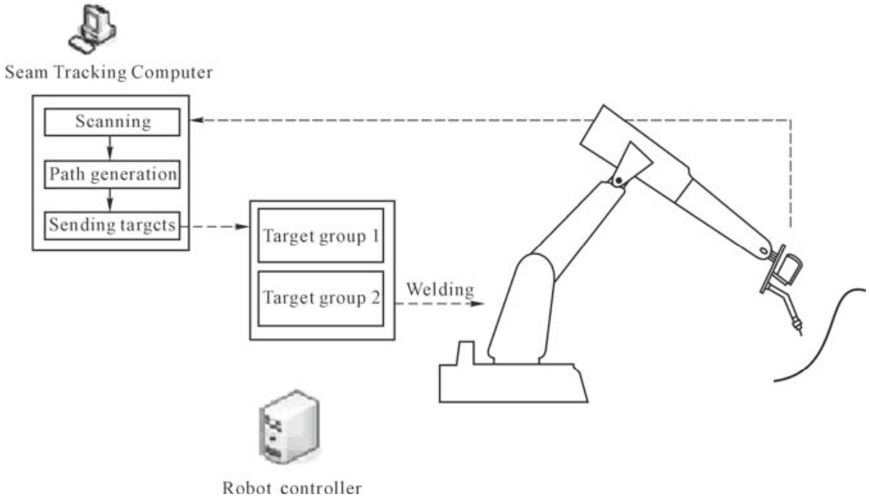


Fig. 7.46. Dual-buffer computer-robot communication

In this method, two groups of targets are defined in the robotic controller. After the first section of the path is scanned, the generated targets are sent to the robot and saved in the first group of targets. The robot follows the targets in the first group to weld the part and at the same time the second section of the path is scanned and the generated targets are sent to the other group of targets. After the robot reaches the last target in the first group of targets, it continues the path by switching to the other group of targets. The two groups of targets swap their roles and the procedure proceeds.

7.7.5 A Robotic Tube Panel Weld System

Fig. 7.47 shows a robotic weld system used to join fins and tubes to form a boiler water wall panel. The system consists of a weld platform, a robot system, an arc weld station, and a laser seam tracking system.

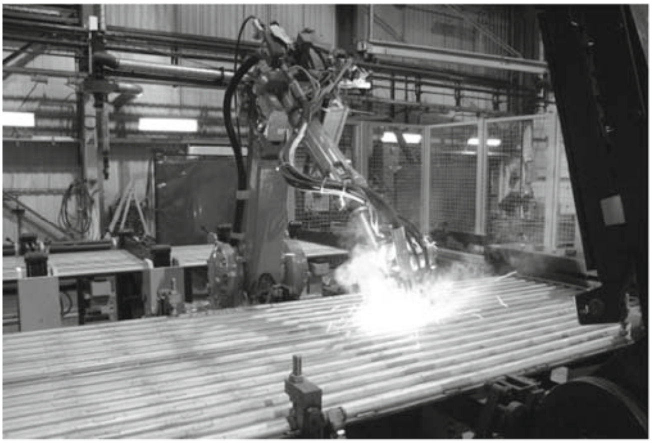


Fig. 7.47. A robotic tube panel weld system

Because the tube has a length of about 20 feet, the thermal distortion of the tube during welding is large. Besides using clamps to physically fix the ends of the tube, a seam tracking system is also integrated to compensate for the position error occurring in the welding process. Instead of generating the welding path in real time with 6-DOF as mentioned above, the system simplifies the problem to a position tuning problem.

A master workpiece can be used for teaching the robotic welding program. Considering the straight seam, if a robotic program for the seam is taught to have the same robot orientation, then when the torch aligns the seam, the position of the welding joint detected by the laser sensing system will remain constant (since the position difference between the torch and the laser sensor remains constant). This constant position is considered a reference point for the welding operation. During the welding, if errors occur, the detected welding joint will not have the same position as the reference point. Then a difference is obtained in the laser sensor coordinate. With the pre-calibrated information, a difference in $Tool_0$ or tool (welding torch) coordinate can be calculated and sent to the robot controller to shift the welding path to compensate for the errors, as indicated in Fig. 7.48.

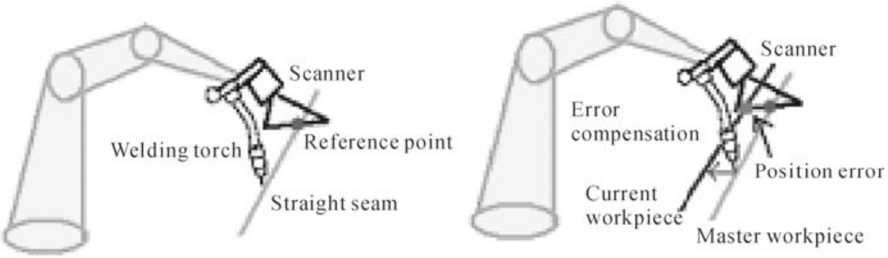


Fig. 7.48. Position error compensation

We can see that this kind of position error compensation is based on the assumptions of a straight seam and small orientation change. When the assumptions are violated, a second consideration should be made to evaluate the validity of the compensation. In some of these assumption-violated cases, with proper selection of the parameters, the error after compensation may still meet the accuracy requirement of the welding, though the compensation is not mathematically "perfect". A rule of thumb is that the shorter the look-ahead distance (the distance between the reference point and the welding torch), the smaller the after-compensation error will be, as shown in Fig. 7.49.

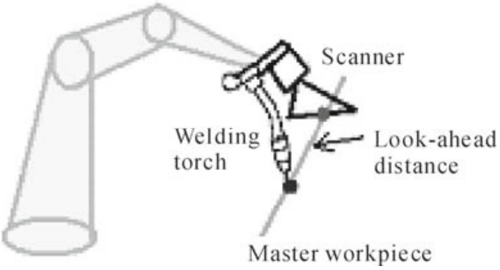


Fig. 7.49. Look-ahead distance

Another similar robotic welding system with a seam tracking sensor includes a weld station that is used to weld an alumina case, as shown in Fig. 7.50, where two welding robots are used simultaneously to increase the efficiency.

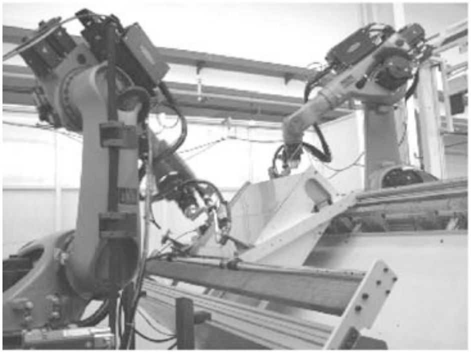


Fig. 7.50. Robot welding system with seam tracking sensors

7.7.6 Summary and Remarks

In this section a general robotic seam tracking system is discussed which can tune robotic poses with 6 degrees of freedom instead of only modifying the offset of the original path as in traditional seam tracking system. A tube panel welding system which integrates a seam tracking system is presented as a commercial example to compensate for the in process variation of the weld seam.

7.8 Robotic Pick and Place System with Point Lasers

Compared with the laser stripe sensor, the point laser is cheaper, has higher precision and is more robust to environmental lighting. A proper point laser positioning process not only improves the precision of positioning but also makes the positioning process more convenient.

In this section, a few engineering applications of the robot visual system with point laser sensors are presented.

7.8.1 Robot Logs Pick and Center System

The laser sensor has been widely used in a robot workcell and assembly line for parts identification and localization. An example here is a log unloading workcell, as shown in Fig. 7.51(a). logs coming out from the calciner are picked up by an inverted robot and placed on the other log conveyor to the next log cutting station. Two sets of grippers are mounted on the robot arm to pick up the two logs sequentially, as indicated in Fig. 7.51(b). The gripper is also equipped with vacuum sensing to insure that enough vacuum is present to pick up a log. Also, the gripper is equipped with a contact sensor to detect whether the robot has a log. If a log is dropped, this sensor will be activated to warn of the situation. Logs that cannot be picked, or that are dropped, will have to be manually removed and the robot reset. Three point lasers are mounted on each gripper to locate the log position and orientation. The sensors are used to find the front and side edges of the log and then allow the robot to pick the log at its center.

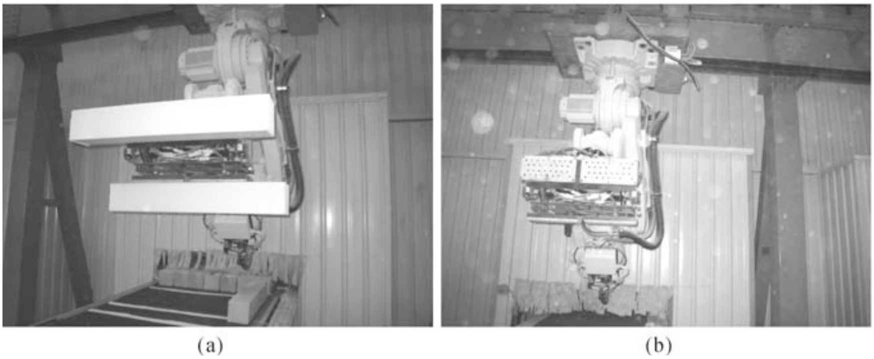


Fig. 7.51. Log unloading workcell. (a) Two logs coming out from a calciner conveyor are picked up by an inverted robot; (b) Two sets of grippers are mounted on the robot arm to pick up the log and three laser point sensors are mounted on each gripper to locate the log position and orientation

7.8.1.1 Laser Sensor

The laser sensor used to detect the edge is a convergent and retroreflective mode laser sensor, like a PicoDot laser sensor from Banner Engineering Corp. that provides retroreflective sensing. The edge is detected by moving the laser sensor across the edge. When the sensor is held by the robot, the edge position relative to the robot can be detected, as shown in Fig. 7.52.

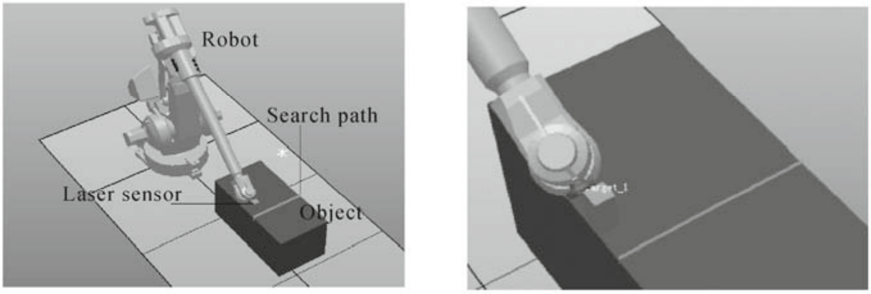


Fig. 7.52. Principle of edge detection by using retroreflective laser sensor

7.8.1.2 Edge Detection and Localization

Firstly, the robot with the laser sensor moves along the X direction to find x_1 point on the side edge of the log, as shown in Fig. 7.53. Due to the function of robot grippers, strict centering of the X direction is not necessary for pick-up purposes. Hence the orientation error can be ignored and the center point of the X direction can be calculated by point x_1 and the length of the log. Secondly, the robot moves along Y direction to find edge points y_1, y_2 on the log. Therefore, the orientation and center of the log can be calculated by detected points (x_1, y_1, y_2) and the known log length. There are three laser point sensors mounted on the gripper to detect edge points x_1, y_1, and y_2, respectively.

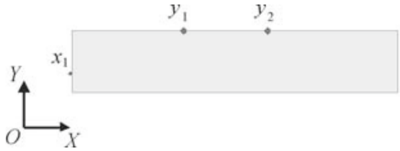


Fig. 7.53. Localization of a log by detecting edges using point laser sensors

7.8.1.3 Log Centering

In this scenario, the log is held by the robot and is required to be placed in the center of a fixture, as shown in Fig. 7.54. In this case, a search for the side edge positions in the X direction is needed. Since the log length is unknown, the center of the log in the X direction can be determined by using two laser sensors fixed at the station and the log held by the robot moves along the X direction. The positioning process is as follows:

(1) The robot moves to the initial position between two laser sensors S_1 and S_2.

(2) The robot moves along $\boldsymbol{S}_1\boldsymbol{S}_2$ direction and the right edge of the log is determined by the laser sensor $\boldsymbol{S}_2$. Record the distance ($L_{\text{right})}$ that the robot moves from its initial position.

(3) The robot moves along $\boldsymbol{S}_2$ $\boldsymbol{S}_1$ direction and the left edge of the log is determined by the laser sensor. Record the distance (L_{left}) that the robot moves from its initial position.

(4) The center point offset is calculated as $C_{\text{Offset}} = \frac{1}{2} \times (L_{\text{right}} - L_{\text{left}})$.

(5) The robot moves a distance C_{Offset} along S_1 S_2 if $C_{\text{Offset}} > 0$, otherwise a distance C_{Offset} along $\boldsymbol{S}_2$ $\boldsymbol{S}_1$ to place the log on the fixture. The log will be in the center position on the fixture.

The centering principle is shown in Fig. 7.55.

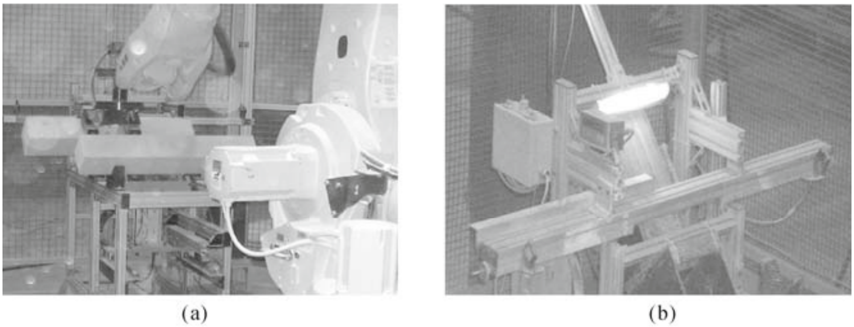


Fig. 7.54. Log centering station. (a) The log is held by a robot and the edge of the log is scanned by two laser sensors mounted on; (b) The fixture of the centering station

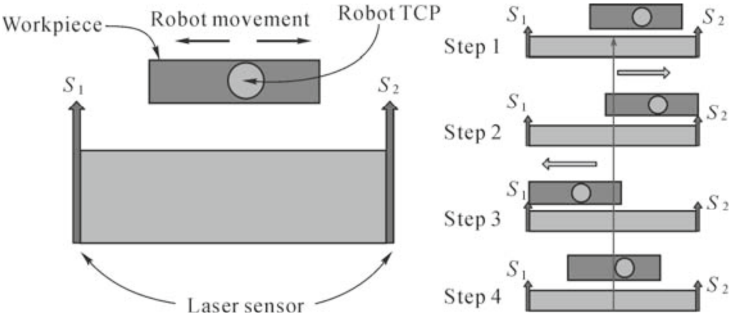


Fig. 7.55. Illustration of centering principle by using laser sensors mounted on the fixture

7.8.1.4 Locate the Object with Laser Sensors

As an other example of using the laser sensor to compensate for the variation of the object location on the production line, a paper roll packing and labeling production line is investigated. In order to increase the quality and efficiency of paper roll packing and labeling, a robotic system is used for a pick and place task. The robot is used to pick up the cover from a cover stack and to place it on the paper roll at both ends. The robot is also used to pick up labels from the printer and to place them on the cylindrical surface of the paper roll. Since the position and radius (size) of the paper roll on the conveyor vary, the robot system is equipped with laser sensors to detect the position of the paper roll before placing the label on it. The laser sensor used is an optical distance sensor. The position of the paper roll can be easily detected by the sensor, in conjunction with robot search movement, as shown in Fig. 7.56. The gripper for pick and place consists of vacuum cups to pick up paper labels and also laser sensors to detect the distance of the paper roll from the gripper. During the operation the robot moves towards the target position with slow search speed until the laser sensor is activated when the target is within the sensing range. When the target position is determined, the robot approaches the paper roll and places the label on the surface at high speed.

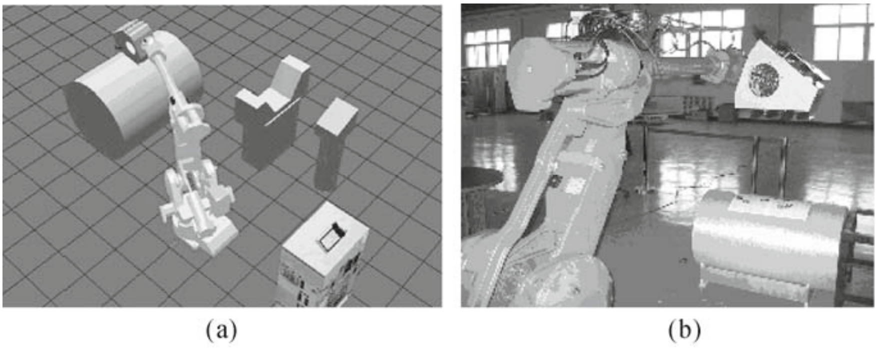


Fig. 7.56. Robot paper roll labeling system. (a) Simulation workcell to indicate the paper labeling process; (b) Experimental system

7.8.2 Robot Solar Panel Installation System

As shown in Fig. 7.57, a robotic system is designed to automatically install large size solar panels in the field. A robot is loaded on a truck or caterpillar that moves along the aisle of racks that have been installed previously. Solar panels are loaded on another truck that moves along with the robot truck and tries to keep the same distance from the first robot truck. When the two trucks reach the location where the panels need to be installed, the robot in the first truck picks up the panel in the second truck and places it onto the desired rack. Each solar panel to be installed can be as large as 2.6 m×2.2 m, with a weight of 120 kg. Because the position of the robot relative to the panels varies, due to the uneven floor of the installation field and control accuracy of the truck position, a laser sensor that is mounted on the robot arm is used. That detects and locates the panel position so the robot can pick up the panels properly. The truck that loads the panels has to be positioned relative to the robot truck by a driver within a certain range so the robot can reach and detect the panels. Because the accurate position of the rack is unknown, the same laser sensor is used to detect and locate the rack position and orientation. This is implemented by using the laser sensor to scan the edge of the rack. The following subsections will give details of the positioning of the panels and racks.

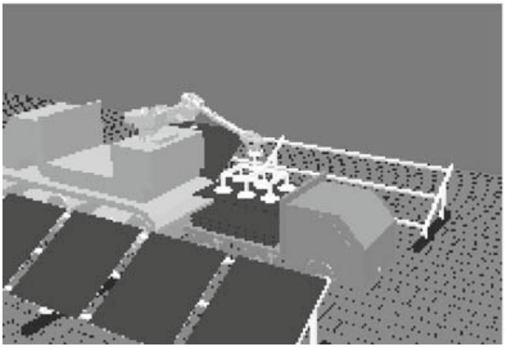


Fig. 7.57. Robotic solar panel installation system

7.8.2.1 System Layout and Positioning Process

The experimental setup to demonstrate the pick up process is shown in Fig. 7.58. The robot holds a gripper and three point laser sensors on the end of the arm, as shown in Fig. 7.58. The gripper with the vacuum sucker is used to pick up the solar panels. The three laser distance sensors are used to detect the panel position and rack position. After two vehicles get into the installation position, the robot scans the rack first and then scans the solar panel to determine the position of the rack and panels, respectively. The robot then picks up the panel loaded on the other truck and places it on the rack to finish the installation process for a single panel.

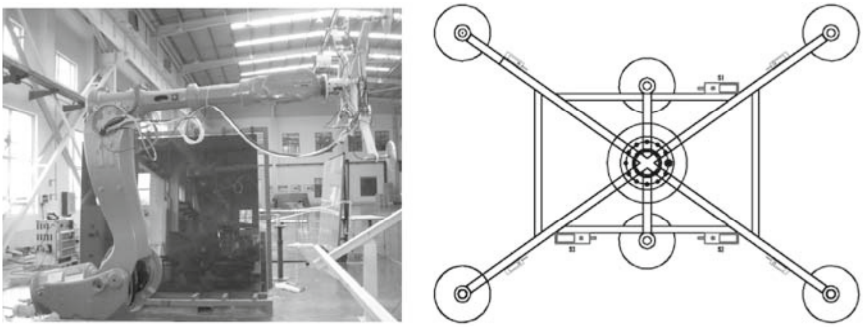


Fig. 7.58. Gripper with laser sensors is used to pick up solar panels

7.8.2.2. Rack Positioning

The position of the rack is determined by scanning and measuring two edges of the frame.

Assume that the positions of three laser distance sensors in the tool coordinate system are $S_1(X_1, Y_1, 0)$, $S_2(X_2, Y_2, 0)$ and $S_3(X_3, Y_3, 0)$, respectively. The direction of all the laser sensors is perpendicular to the XY plane of the tool coordinate. That means the measuring distance is aligned with the Z axis of the tool coordinate frame. Three laser distance sensors S_1, S_2, and S_3 are used to determine three points P_1, P_2, and P_3 on the frame, respectively. As shown in Fig. 7.59, assume that P_h is the initial robot position, and the robot searches the edges of the rack frame in the following steps:

(1) The robot moves into the initial position $P_h(x_h, y_h, z_h)$ and then searches along the Y direction of the tool coordinate and the edge point P_1 is found when the tool moves a distance $Y_{P1.}$ The depth of the point P_1 is measured as Z_{P1}. The edge point P_1 is determined when the laser distance sensor is activated from out of range status that indicates there is no object in front of the sensor. Similarly, the edge point P_2 is determined when the tool travels a distance Y_{P2} along the Y direction of the tool coordinate, and its depth is measured as Z_{P2}.

(2) The robot moves back to P_h position, and then searches along the X direction of the tool coordinate. The edge point P_3 on the rack is measured when the robot moves a distance X_{p_3} along the X direction of the tool coordinate. Its depth is measured as Z_{p_3}

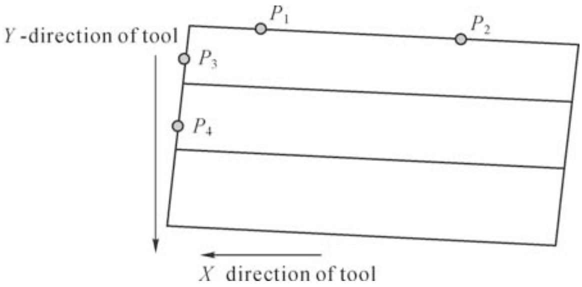


Fig. 7.59. Rack edge searching process

Therefore, the positions P_1, P_2, and P_3 can be calculated as $P_1(X_1, Y_1+Y_{p_1}, Z_{p_1})$, $P_2(X_2, Y_2+Y_{p_2}, Z_{p_2})$, and $P_3(X_3, Y_3+Y_{p_3}, Z_{p_3})$.

The rack pose is determined by at least three points on the rack.

The position of the solar panel can be determined by using the same method as the one used for rack positioning.

7.8.3 Summary and Remarks

In this section two examples are presented to indicate how the point laser sensors are effectively used to identify edges of the part. Comparing with a laser scanner a point laser sensor is cost effective and with higher operation speed. It is more suitable for the identification and localization of the part with a regular geometry shape. In contrast a laser scanner is for the part with a complex and irregular surface.

References

Agapiou G, Kasiouras C, and Serafetinides AA (1999) A detailed analysis of the MIG spectrum for the development of laser-based seam tracking sensors. Optics & Laser Technology, 31:157-161.

Besl PJ, Mckay ND (1992) A Method for Registration of 3-D Shapes. IEEE Transactions on Pattern Analysis and Machine Intelligence, 14(2):239-256.

Chen X, Gong Z, Huang H, Zhou L, Ge SS, Zhu Q, and Woon LC (1999) An automated 3D polishing robotic system for repairing turbine airfoils. Industrial Automation Journal, 8(2):6-11.

Chen Y, Medioni G (1991) Object modeling by registration of multiple range images. Proceedings of the 1991 IEEE International Conference on Robotics and Automation. Sacramento, CA. 2724-2729.

Craig JJ (1989) Introduction to Robotics: Mechanics and Control. Addison-Wesley Publishing Company, Boston.

Fridenfalk M, Bolmsjö G (2003) Design and validation of a universal 6D seam tracking system in robotic welding using laser scanning. Journal of Industrial Robot, 30(5):437-448.

Giblin D, Zongliang M, Gan Z, Kazerounian K (2007) Target tracking manipulation theories for combined force and position control in open and closed loop manipulators. Journal of Mechanical Design, 129(3):326-334.

Gong C, Yuan J, Ni J (2000) Non-geometric error identification and compensation for robotic system by inverse calibration. International Journal of Machine Tools & Manufacture, 40:2119-2137.

Gunnarsson KT, Prinz FB (1987) CAD model-based localization of parts in manufacturing. Computer, 20(8):66-74.

Gursel A, Bijan S (2005) A systematic technique to estimate positioning errors for robot accuracy improvement using laser interferometry based sensing. Mechanism and Machine Theory, 40(8): 879-906.

Hoebel M, Fehrmann B, Kohles C, Franitza K (2010) Method of processing turbine components. US Patent 7,725,219 B2

Huang H, Zhou L, Chen XQ, Gong ZM (2003) SMART robotic system for 3D profile turbine vane airfoil repair. International Journal of Advanced Manufacturing Technology, 21(4):275-283.

Pastorius W (2009) 3-D vision for process monitoring in assembly operations. Quality Digest, 05/20/2009.

Piegl LA, Tiller W (2000) Least squares B-spline curve approximation with arbitrary end derivatives. Engineering with Computers, 16:109-116.

Pottmann H, Huang QX, Yang YL, Hu SM (2006) Geometry and convergence analysis of algorithms for registration of 3D shapes. International Journal of Computer Vision 67(3):277-296.

Reinhart G, Tekouo W (2009) Automatic programming of robot-mounted 3D optical devices to easily measure parts in high-variant assembly. Manufacturing Technology, 58:25-28.

Roth ZS, Mooring BW, Ravani B (1987) An overview of robot calibration. IEEE Journal of Robotics and Automation, 3(5):377-385.

Schreoder K (1999) Handbook of Industrial Robots, John Wiley & Sons Inc., New York. 795-810.

Sun Y (2004). Development of a unified flexible grinding process. ETD Collection for University of Connecticut. Paper AAI3123470.

Sun Y, Giblin DJ, Kazerounian K (2009) Accurate robotic belt grinding of workpieces with complex geometries using relative calibration techniques. Robotics and Computer-Integrated Manufacturing, 25(1):204-210.

Tsai RY (1987) A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-shelf TV camera and lenses. IEEE Journal of Automation, 3(4): 323-334.

Vergeest JSM, Song Y, and Hartge D (2003), Freeform Object Positioning by 3D Shape Matching without Artificial Future Points. Journal of WSCG, 12(1-3).

Wang JJ, Tang Q, Gan Z (2003) A robotic vision system for turbine blade cooling hole detection. In: 2003 International Conference on Control, Automation, and Systems (ICCAS 2003), October, South Korea.

Wang YT, Jan YJ (2001) Path Planning for Robot-Assisted Grinding Processes, Robotics and Automation. In: Proceedings 2001 ICRA IEEE International Conference. 1: 331-336.

Zeng GW and Hemami A (1997) An overview of robot force control, Robotica, 15:473-482.

Zhang H, Wang JJ, Zhang G, Gan Z (2005) Machining with flexible manipulator: Toward improving robotic machining performance. In: Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics Monterey, California, USA. 24-28.

Zhang W. et al (2007) Seam Tracking of Articulated Robot for Laser Welding Based on Visual Feedback Control. Springer, New York.

Zhu Z, Tang Q, Li J, Gan Z (2004) Calibration of laser displacement sensor used by industrial robots. Optical Engineering, 43:12-13.

Appendix

A.1 Linear Equations and Its Solution

Given an over-determined system

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ \vdots \\ \vdots \\ b_n \end{bmatrix} \tag{A.1}$$

or its compact format

$$\boldsymbol{Ax} = \boldsymbol{b}$$

There are n equations and m variables. When $n \geq m$, the solution of normal equation is that which minimizes the sum of the square differences between left and right sides

$$\min \left\| \boldsymbol{Ax} - \boldsymbol{b} \right\|^2 \tag{A.2}$$

Assuming that

$$\begin{aligned} E(\boldsymbol{x}) &= \left\| \boldsymbol{Ax} - \boldsymbol{b} \right\|^2 \\ &= \left(\boldsymbol{Ax} - \boldsymbol{b} \right)^{\mathbf{T}} \left(\boldsymbol{Ax} - \boldsymbol{b} \right) \\ &= \left(\boldsymbol{x}^{\mathrm{T}} \boldsymbol{A}^{\mathrm{T}} - \boldsymbol{b}^{\mathrm{T}} \right) \left(\boldsymbol{Ax} - \boldsymbol{b} \right) \\ &= \boldsymbol{x}^{\mathbf{T}} \boldsymbol{A}^{\mathbf{T}} \boldsymbol{Ax} - \boldsymbol{b}^{\mathbf{T}} \boldsymbol{Ax} - \boldsymbol{x}^{\mathbf{T}} \boldsymbol{A}^{\mathbf{T}} \boldsymbol{b} + \boldsymbol{b}^{\mathbf{T}} \boldsymbol{b} \\ &= \boldsymbol{x}^{\mathbf{T}} \boldsymbol{A}^{\mathbf{T}} \boldsymbol{Ax} - \left(\boldsymbol{A}^{\mathbf{T}} \boldsymbol{b} \right)^{\mathbf{T}} \boldsymbol{x} - \left(\boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{b} \right)^{\mathbf{T}} \boldsymbol{x} + \boldsymbol{b}^{\mathbf{T}} \boldsymbol{b} \end{aligned} \tag{A.3}$$

The minimal value of the function is taken place when the divertive is equal to zero.

$$\frac{\partial E}{\partial \boldsymbol{x}} = 2A^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} - 2A^{\mathrm{T}}\boldsymbol{b} = 0 \tag{A.4}$$

Therefore

$$A^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) = \boldsymbol{0} \tag{A.5}$$

and **x** can be solved as

$$\boldsymbol{x} = \left(A^{\mathrm{T}}A\right)^{-1} A^{\mathrm{T}}\boldsymbol{b} \tag{A.6}$$

This is the analytical solution of the linear equations by using least squares method.

A.2 Singular Value Decomposition (SVD)

For any matrix **A** with dimension *m*×*n* and rank *r* there exists a factorization

$$\boldsymbol{A}_{m\times n} = \boldsymbol{U}_{m\times m}\boldsymbol{S}_{m\times n}\boldsymbol{V}^{\mathrm{T}}_{n\times n} \tag{A.7}$$

where the columns of $\boldsymbol{U}$ are orthogonal eigenvectors of $\boldsymbol{AA}^{\mathrm{T}}$, the columns of $\boldsymbol{V}$ are orthogonal eigenvectors of $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$.

$$\boldsymbol{S} = \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix}, \Sigma_r = \mathrm{Diag}(\sigma_1, \sigma_2, ..., \sigma_r) \tag{A.8}$$

$$\begin{aligned} \boldsymbol{U} &= (\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_m) \\ \boldsymbol{V} &= (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_n) \end{aligned} \tag{A.9}$$

where diagonal entries $\sigma_1 \geq \sigma_2, ..., \geq \sigma_r > 0, \sigma_{r+1} = ... = \sigma_{\min(m,n)} = 0$ are called the singular values of $\boldsymbol{A}$. σ_i^2 $(i = 1, ...r,)$ are the eigenvalues of symmetric matrices $\boldsymbol{AA}^{\mathrm{T}}$ and $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}$. u_i and v_i are corresponding eigenvectors, respectively. Decomposition (A.7) is called singular value decomposition (SVD).

If $\boldsymbol{U}_r = [\boldsymbol{u}_1, ..., \boldsymbol{u}_r]$, $\boldsymbol{V}_r = [\boldsymbol{v}_1, ..., \boldsymbol{v}_r]$, we have

$$\boldsymbol{A} = \boldsymbol{U}_r\, \Sigma_r\, \boldsymbol{V}_r^{\mathrm{T}} = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^{\mathrm{T}} \tag{A.10}$$

It decomposes the matrix $\boldsymbol{A}$ of rank r into sum of r matrices of rank 1.

A.3 Nonlinear Optimization Algorithms

Minimization of the nonlinear least squares objective function $C(\gamma)$ is achieved by iterative calculation of a correction parameter vector ($\delta\gamma$) using various approaches: Steepest decent, Newton approach, Gauss-Newton method, and modified Gauss-Newton method.

(1) Steepest decent

$$\delta\gamma = \frac{-\mathrm{grad}(C)}{\left\|\mathrm{grad}(C)\right\|}\Delta \tag{A.11}$$

where Δ is the step size. The disadvantage of this method is that the speed of convergence may become prohibitively slow for certain shapes of the hyper surface of C.

(2) Newton's method

$$\delta\gamma = -\left(\frac{\partial^2 C}{\partial\gamma^2}\right)^{-1}\left(\frac{\partial C}{\partial\gamma}\right)^{\mathrm{T}} E \tag{A.12}$$

where E is the error vector between the measurement and modeling data. The main advantage of Newton's method is its rapid, quadratic convergence near the minimum. However it needs to calculate the second derivative (Hessian matrix) which is a relatively large computational effort.

(3) Gauss-Newton method

$$\delta\gamma = \left(\left(\frac{\partial C}{\partial\gamma}\right)^{T}\left(\frac{\partial C}{\partial\gamma}\right)\right)^{-1}\left(\frac{\partial C}{\partial\gamma}\right)^{\mathrm{T}} E \tag{A.13}$$

Gauss-Newton method is derived from an approximation of Hessian matrix.

(4) Modified Gauss-Newton method (Levenberg-Marquardt Algorithm)

$$\delta\gamma = \left(\left(\frac{\partial C}{\partial\gamma}\right)^{\mathrm{T}}\left(\frac{\partial C}{\partial\gamma}\right) + \lambda I\right)^{-1}\left(\frac{\partial C}{\partial\gamma}\right)^{\mathrm{T}} E \tag{A.14}$$

where I is the identity matrix. The factor $\lambda > 0$ determines the tendency of Levenberg-Marquardt step towards either steepest decent or Gauss-Newton. For large λ the step will approach the direction of steepest decent whereas for small λ it will approach that of Gauss-Newton. The strategy will therefore be to use a large λ initially (far from the minimum) with λ decreasing as the minimum is approached. Furthermore, as long as $\lambda \neq 0$, numerical stability will be guaranteed because any potential ill-determinacy in the Jacobian ($\partial C/\partial \gamma$) is eliminated by the addition of λI. This overcomes numerical instability problems when the Jacobian has a singularity for Eq. (A.13).

Covariance matrix of the standard errors in the fitted parameters will be used to estimate the confidence limits of the optimization

$$\text{Cov} = \left(\left(\frac{\partial C}{\partial \gamma} \right)^{\mathrm{T}} \left(\frac{\partial C}{\partial \gamma} \right) \right)^{-1} \tag{A.15}$$

A.4 B-Spline Data Fitting

A B-spline curve can be expressed as

$$p(u) = \sum_{j=0} d_j N_{j,k}(u), u \in [0,1] \qquad \text{A.16}$$

where, d_j are the control points, $N_{j,k}$ are the basic functions that can be calculated recursively with the following equations

$$N_{i,0}(u) = \begin{cases} 1 & u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{A.17}$$

$$N_{i,j}(u) = \frac{u - u_i}{u_{i+j} - u_i} N_{i,j-1}(u) + \frac{u_{i+j+1} - u}{u_{i+j+1} - u_{i+1}} N_{i+1,j-1}(u) \tag{A.18}$$

where, u_i is referred to as the knots.

Given a set of points $\{q_i\}$ and associating parameters $\{\tilde{u}_i\}$ (i = 0, ..., m), the approximated curve $p(u)$ in the least squares sense is defined by minimizing the object function:

$$f = \sum_{i=0}^{m} \left\| q_i - p(\tilde{u}_i) \right\|^2 \tag{A.19}$$

To fit a group of ordered data, one can use the following steps:

(1) In order to do B-spline fitting upon a group of ordered data, such as the edge points detected in the application, the first step is to select the proper method to parameterize the 3D points. The popular chord length parameterization method is found to be quite adequate for engineering applications here, i.e.,

$$q_i = p(\tilde{u}_i) \tag{A.20}$$

where

$$\begin{cases} \tilde{u}_0 = 0 \\ \tilde{u}_i = \tilde{u}_{i-1} + \left|\Delta q_{i-1}\right|, i = 1, 2, ..., n \end{cases} \tag{A.21}$$

Δq_i | is obtained by $\dfrac{q_i - q_{i-q} \mid}{\sum_{j=1}^{n} q_j - q_{j-1} \mid}$; n is the number of the measurement data points.

(2) Set the knot vector (u_i) of the B-spline curve with equations

$$\left.\begin{aligned} & u_0 = u_1 = ... = u_k = 0 \\ & u_{k+j} = (1-\alpha)\tilde{u}_i - 1 + \alpha\tilde{u}_i, \ j = 1, 2, ..., n-k \end{aligned}\right\} \tag{A.22}$$

where $\alpha = j \cdot \dfrac{m+1}{n-k+1} - \mathrm{int}(j \cdot \dfrac{m+1}{n-k+1})$; m is the number of the measurement data points; n is the number of the control points; k is the degree of the B-spline curve.

By setting

$$\left.\begin{aligned} & d_0 = p(0) = q_0 \\ & d_{n-1} = p(1) = q_{m-1} \end{aligned}\right\} \tag{A.23}$$

where, d_0 and $d_{n\text{-}1}$ are the first and the last control points; q_0 and $q_{m\text{-}1}$ are the first and the last data points, the curve passes the first and the last data points. In this case, the fitting of the B-spline curve is to solve control points $d_1 ... d_{n\text{-}2}$ to minimize the objective function

$$f = \sum_{i=1}^{m-1} [q_i - p(\tilde{u}_i)]^2 = \sum_{i=1}^{m-1} \lfloor r_i - \sum_{j=1}^{n-1} d_j N_{j,k}(\tilde{u}_i)]^2 \tag{A.24}$$

where

$$r_i = q_i - q_0 N_{0,k}(\tilde{u}_i) - q_m N_{n,k}(\tilde{u}_i), i = 1, 2, \cdots, m-1 \tag{A.25}$$

By setting the derivative

$$\frac{\partial f}{\partial d_l} = \sum_{i=1}^{m-1} \lfloor -2 r_i N_{l,k}(\tilde{u}_i) + 2 N_{l,k}(\tilde{u}_i) \sum_{j=1}^{n-1} d_j N_{j,k}(\tilde{u}_i)] \tag{A.26}$$

to zero, we can obtain

$$(N^{\mathrm{T}}N)D = R \tag{A.27}$$

where

$$N = \begin{bmatrix} N_{1,k}(\tilde{u}_1) & \cdots & N_{n-1,k}(\tilde{u}_1) \\ \vdots & \vdots & \vdots \\ N_{1,k}(\tilde{u}_{m-1}) & \cdots & N_{n-1,k}(\tilde{u}_{m-1}) \end{bmatrix} \tag{A.28}$$

$$R = \begin{bmatrix} N_{1,k}(\tilde{u}_1)r_1 + & \cdots & +N_{1,k}(\tilde{u}_{m-1})r_{m-1} \\ \vdots & \vdots & \vdots \\ N_{n-1,k}(\tilde{u}_1)r_1 + & \cdots & +N_{n-1,k}(\tilde{u}_{m-1})r_{m-1} \end{bmatrix}, D = \begin{bmatrix} d_1 \\ \vdots \\ d_{n-1} \end{bmatrix} \tag{A.29}$$

Therefore D or $d_1, \ldots, d_{n-2}$ can be solved with a linear least squares algorithm.

Index