

## DOCUMENT IMAGE PROCESSING

A digital document image is a scanned representation of a paper document, such as a page of facsimile transmission. Document image processing consists of processes for taking a document through various representations: from scanned image to semantic representation. This article describes algorithmic processes involved in document image processing. The current state-of-the-art and the future directions on each of them are indicated. The topics described are: system architecture, decomposition and structural analysis, text recognition, table, diagram and image understanding, and performance evaluation.

The need to process documents on paper by computer has led to an area of information technology that may be referred to as document image processing (DIP). The goal of a DIP system is to convert a scanned representation of a document into an appropriate symbolic form. For instance, a DIP system takes as input raster images of typed hard-copy text—as captured by an imaging camera—and produces as output an editable file in a standard word processing format. Another example is one where the input is an image of a postal envelope with a handwritten address and the output is its postal zip code.

DIP consists of several subprocesses necessary to take a document from scanned gray-scale or color images to editable or high-level semantic descriptions of the document. Thus it involves techniques from many subdisciplines of information processing including image processing, pattern recognition, natural language processing, artificial intelligence, and database systems.

The symbolic representation desired as output of a DIP system can take one of several forms: an editable description, a representation from which the document can be (exactly)

reconstructed, a semantic description useful for document sorting/filing, etc. Representation schema that are useful for editing and exact reproduction are standards for electronic document description, such as SGML. Deriving a semantic representation, which may be referred to as document image interpretation, will involve using extensive problem knowledge. For instance, the task of assigning a postal zip code to a postal envelope involves looking-up directories for recognizing individual city, state, and street address words.

DIP system design involves the following issues:

*System Architecture.* The complexity of the DIP task leads to modularization into distinct subprocesses. Because of the interdependency of processes, issues of how to maintain communication and integrate results from each process arise.

*Decomposition and Structural Analysis.* Documents consist of text (machine-printed and hand-written), line drawings, tables, maps, half-tone pictures, icons, etc. It is necessary to decompose a document into its component parts in order to process these individual components. Their structural analysis, in terms of spatical relationships and logical ordering, is necessary to invoke modules in appropriate order and to integrate the results of the appropriate modules.

*Text Recognition and Interpretation.* It is necessary to recognize words of text, often using lexicons and higher level linguistic and statistical context. The necessity for contextual analysis arises from the fact that it is often impossible to recognize characters and words in isolation, particularly with handwriting and degraded print. Preprocessing of character images is an important step in recognition. It is often necessary to enhance binary character images to assist the subsequent recognition process in both handwritten and machine-printed documents.

*Tables, Graphics, and Halftone Recognition.* Specialized subsystems are necessary for processing a variety of nontext or mixed entities, such as recognizing tabular data, converting graphical drawings into vector representation, and extracting objects from half-tone photographs.

*Databases and System Performance Evaluation.* Methods for determining data sets on which evaluation is based and the metrics for reporting performance.

### System Architecture

The architecture of a typical DIP system is illustrated in Fig. 1. A DIP system should be capable of handling documents with varied layouts, containing text, graphics, line drawings, and half-tones. Several special-purpose modules are required to process different types of document components. It is essential to have a global data representation to facilitate communication between processes. This also allows independent subsystem development without being concerned with communication protocol.

As the figure shows, the image scanner optically captures text images to be recognized. Text images are processed with DIP software and hardware. The process involves three operations: document analysis (extracting individual character images), recognizing these images (based on shape), and con-

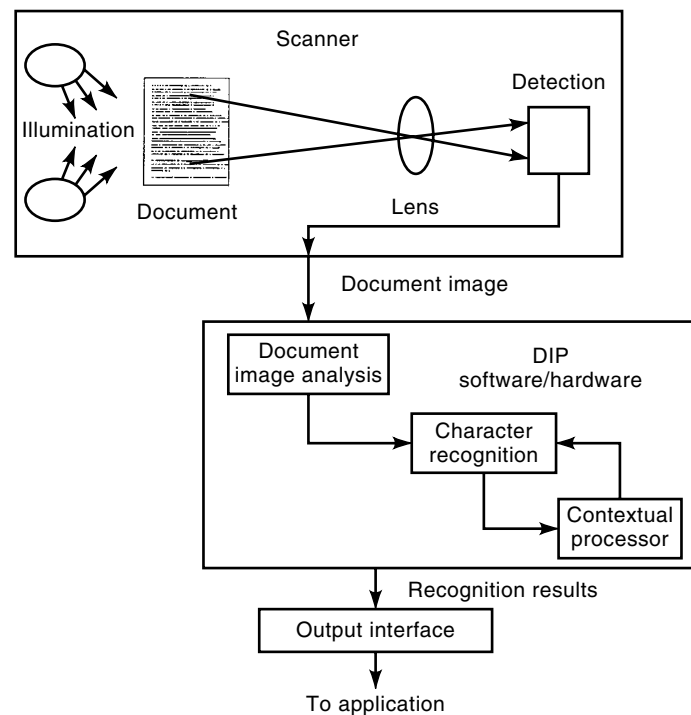


Figure 1. A typical document image processing system.

textual processing (either to correct misclassifications made by the recognition algorithm or to limit recognition choices). The output interface is responsible for communication of DIP system results to the outside world.

Four basic building blocks form functional image scanners: a detector (and associated electronics), an illumination source, a scan lens, and a document transport. The document transport places the document in the scanning field, the light source floods the object with illumination, and the lens forms the object's image on the detector. The detector consists of any array of elements each of which convert incident light into a charge, or analog signal. These analog signals are then converted into an image. Scanning is performed by the detector and the motion of the text object with respect to the detector. After an image is captured, the document transport removes the document from the scanning field.

Recent advances in scanner technology have made available higher resolution, often in the range of 300 pixels per inch (ppi) to 400 ppi. Recognition methods that use features (as opposed to template matching) use resolutions of at least 200 ppi and careful consideration of gray-scale. Lower resolutions and simple thresholding tend to break thin lines or fill gaps, thus invalidating features.

### Document Image Analysis

Text is extracted from the document image in a process known as document image analysis. Reliable character segmentation and recognition depend on both original document quality and registered image quality. Processes that attempt to compensate for poor quality originals or poor quality scanning include image enhancement, underline removal, and noise removal. Image enhancement methods emphasize character versus noncharacter discrimination. Underline removal

erases printed guidelines and other lines which may touch characters and interfere with character recognition and noise removal erases portions of the image that are not part of the characters.

Prior to character recognition it is necessary to isolate individual characters from the text image. Many DIP systems use connected components for this process. For those connected components that represent multiple or partial characters, more sophisticated algorithms are used. In low-quality or nonuniform text images these sophisticated algorithms may not correctly extract characters and thus, recognition errors may occur. Recognition of unconstrained handwritten text can be very difficult because characters cannot be reliably isolated especially when the text is cursive handwriting.

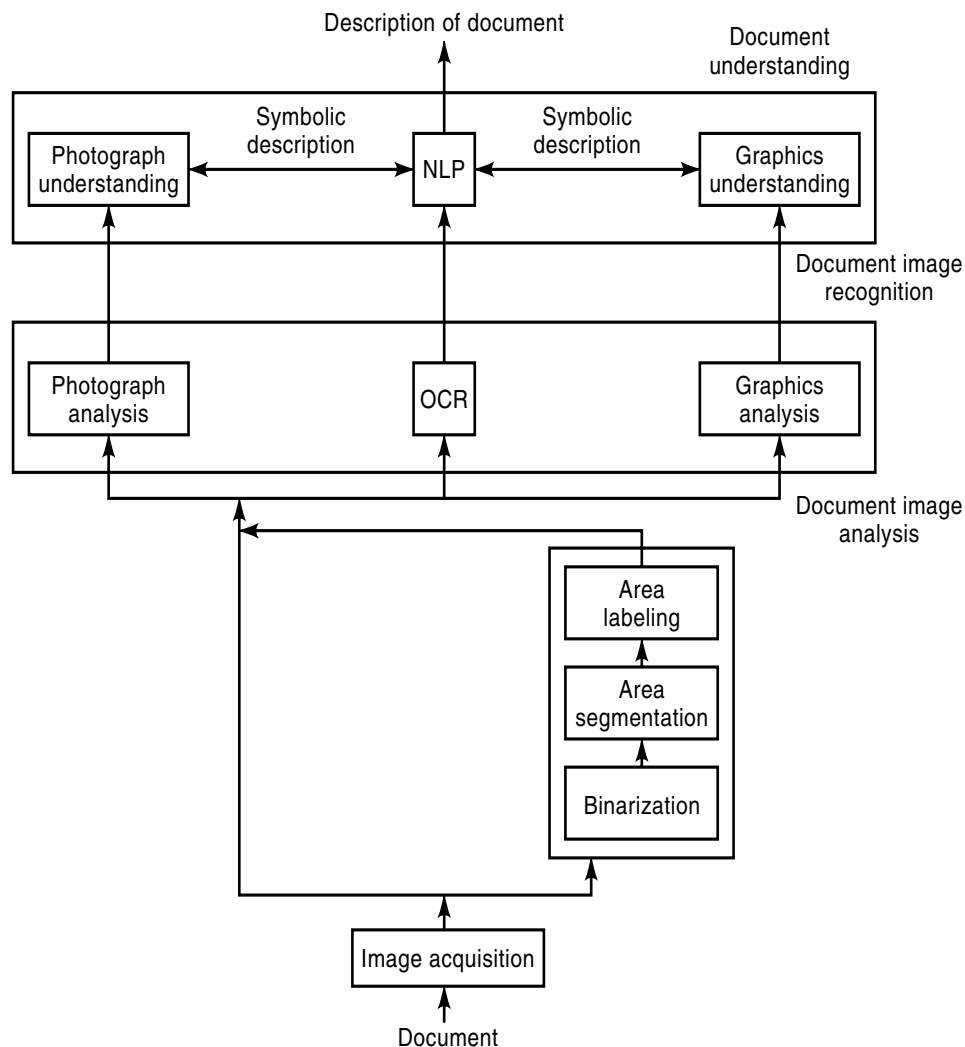
### Document Understanding

Document understanding (DU) is the goal-oriented task of deriving a symbolic representation of the contents of a document image, which involves detecting and interpreting different blocks (like photographs, text, and drawings), accounting for the interactions of the different components, and coordinating the interpretations to achieve an end result.

A functional architecture for document understanding specifies the major functional components without concerning

itself with practical considerations such as shared resources. The functional modules and interactions of a document understanding system are shown in Fig. 2. The *document understanding* task is divided into three conceptual levels: document image analysis, document image recognition, and document image understanding. Within these levels there are several processing modules (or tools): binarization, area-segmentation, area-labeling, optical character recognition (OCR), photograph analysis, graphics analysis, picture understanding, natural language processing, and graphics understanding. The interaction between modules allows for the interpretation of individual subareas to be combined to form a higher level of representation: for example, the interpretation of a photograph caption by a natural language processing module and objects in a photograph located by a photograph analysis module can be used by a photograph understanding module to label the objects' identity.

The input to the system is a high-resolution color, gray-scale, or binary image. The output of the system is a description of the contents of the document components. An editable description should contain the following entries: (i) component identities and locations on the document, such as text, graphics, and half-tones, (ii) spatial relationships between components, (iii) layout attributes such as component size



**Figure 2.** Functional architecture for document understanding.

and number of lines in text blocks, and (iv) logical grouping of components.

The level of document understanding is determined by the level of representation that the DU system can derive from the document image. Three types of information can be derived from a document: (i) *layout (geometric) structure* is a physical description of document regions, such as size, location, and spatial relationships between regions, (ii) *logical structure* is a grouping of layout components based on human interpretation of the content of the components and the spatial constraints between components, and (iii) *content interpretation* contains coded data of a component which can be used to derive logical structure or can be stored for later access. A system whose output contains only layout structure is a *document layout analysis system* and a system whose output contains all three types of information is a document understanding system. From this perspective, layout analysis is an intermediate step of document understanding.

## DECOMPOSITION AND STRUCTURAL ANALYSIS

A document image is basically a visual representation of a printed page such as a journal article page, a facsimile cover page, a technical document, or an office letter. Typically, it consists of blocks of text, such as letters, words, and sentences, that are interspersed with tables and figures. The figures can be symbolic icons, gray-level images, line drawings, or maps. A digital document image is a two-dimensional representation of a document image obtained by optically scanning and digitizing a hardcopy document. It may also be an electronic version that was created for publishing or drawing applications available for computers.

Methods of deriving the blocks can take advantage of the fact that the structural elements of a document are generally laid down in rectangular blocks aligned parallel to the horizontal and vertical axes of the page. The methods can also use several types of knowledge including visual, spatial, and linguistic. Visual knowledge is needed to determine objects from the background. Labeling blocks involves the use of spatial knowledge, such as layout of a typical document. Determining the font and identity of characters also involves spatial knowledge. Reading words in degraded text is a process that involves spatial as well as linguistic knowledge like a lexicon of acceptable words. Determining the role of a block of text, "is this a title?", is a process requiring spatial, syntactic, as well as semantic knowledge. Considerable interaction among different types of knowledge is necessary. For instance, assigning a role to a textual region may require not only knowledge of the spatial layout, but also an analysis of its textual syntax and semantics, and an interpretation of neighboring pictorial regions.

The document decomposition and structural analysis task can be divided into three phases. Phase 1 consists of *block segmentation* where the document is decomposed into several rectangular blocks. Each block is a homogeneous entity containing one of the following: text of a uniform font, a picture, a diagram, or a table. The result of phase 1 is a set of blocks with the relevant properties. A textual block is associated with its font type, style, and size; a table might be associated with the number of columns and rows, and so on. Phase 2 consists of *block classification*. The result of phase 2 is an

assignment of labels (title, regular text, picture, table, etc.) to all the blocks using properties of individual blocks from phase 1, as well as spatial layout rules. Phase 3 consists of *logical grouping and ordering of blocks*. For OCR, it is necessary to order text blocks. Also, the document blocks are grouped into items that "mean" something to the human reader (author, abstract, date, etc.), and is more than just the physical decomposition of the document. The output of phase 3 is a hierarchical tree-of-frames, where the structure is defined by the shape of the tree and the content is stored entirely in the leaves. The tree-of-frames can be converted to the SGML representation to ensure portability, easy information retrieval, editability, and efficient semantic indexing of the document.

## Block Segmentation

Approaches for segmenting document image components can be either top-down or bottom-up. Top-down techniques divide the document into major regions which are further divided into subregions based on knowledge of the layout structure of the document. Bottom-up methods progressively refine the data by layered grouping operations.

**Top-Down Methods.** Approaches to document segmentation can be grouped into four categories.

*Smearing* is based on the run-length smoothing algorithm (RLSA). It merges any two black pixels which are less than a threshold apart, into a continuous stream of black pixels. The method is first applied row-by-row and then column-by-column, yielding two distinct bit maps. The two results are then combined by applying a logical AND to each pixel location. The output has a smear wherever printed material (text, pictures) appears on the original image.

The *X-Y tree* method assumes that a document can be represented in the form of nested rectangular blocks. A "local" peak detector is applied to the horizontal and vertical "projection profiles" to detect local peaks (corresponding to thick black or white gaps) at which the cuts are placed; it is local in that the width is determined by the nesting level of recursion: that is, gaps between paragraphs are thicker than those between lines.

The *Hough transform* approach exploits the fact that documents have significant linearity. There exist straight lines in tables and diagrams. Centroids of connected components corresponding to text also line up. Columns of text are separated by straight rivers of white space. Text itself can be viewed as thick textured lines. The Hough transform is a technique for detecting parametrically representable forms like straight lines in noisy binary images.

*Activity identification* is a method for the task of locating destination address blocks on mail-pieces. The goal is to use a simple technique for documents with simple layout structure (few blocks). It quickly closes in on an area of maximum activity by identifying regions with a high density of connected components. The method is useful for goal-oriented tasks like finding the destination address block on a facsimile cover page which is predominantly empty.

**Bottom-Up Methods.** The image is first processed to determine the individual connected components. At the lowest level of analysis, there would be individual characters and large figures. In the case of text, the characters are merged

into words, words are merged into lines, lines into paragraphs, etc.

The application of different operators for bottom-up grouping can be coordinated by a rule-based system. Most characters can be quickly identified by their component size. However, if characters touch a line, as is often the case in tables and annotated line-drawings, the characters have to be segmented from the lines. One technique for segmenting characters from line structures is to determine the high neighborhood line density (NLD) areas in the line structures. The following is an example of the type of rules that can be used.

```

if    connected component size is larger than a threshold
then  the connected component is a figure (with likelihood
       $L_i$ )
if    neighborhood line density (NLD) of a figure is high
then  the high NLD area is a character area (with likelihood
       $L_j$ )

```

In certain document segmentation tasks it is only necessary to extract a given block of interest. An example of this is locating an address block on a mail piece. Several top-down and bottom-up tools are used to segment candidate blocks. They are resolved by a control structure to determine the destination address block by considering spatial relationships between different types of block segments.

### Block Classification

Block classification involves the categorization of blocks by assigning them appropriate labels, based on the features extracted from the block. The basic block type, such as text versus graphics, can be determined by performing text/graphics separation. The exact label, however, may vary depending on the type of document. For example, newspaper pages may have blocks labeled as headline, photograph, caption, etc., whereas a postal mailpiece may have blocks labeled destination address, postage mark, etc.

**Text/Graphics Separation.** Separating machine-printed text from handwritten annotations is necessary for invoking appropriate recognition algorithms. One method that performs this discrimination well with postal addresses is based on computing the histogram of heights of the connected components. Handwritten components tend to have a wider distribution in heights than print.

**Figure Classification.** A figure block can belong to one of the following categories: half-tone picture, line-drawing (diagrams), and table. Pictures are gray-scale images and can be separated from tables and diagrams (binary images) by a histogram analysis of the gray-level distributions. Although both diagrams and tables predominantly comprise straight lines and interspersed text, the fact that the straight lines in tables run only vertically and horizontally can be used to advantage. Analysis of the Hough transform accumulator array is used to separate tables from diagrams.

### Logical Grouping

It is necessary to provide a logical ordering/grouping of blocks to process them for recognition and understanding. Textual

blocks corresponding to different columns have to be ordered for performing OCR.

Journal pages can have complicated layout structures. They are usually multicolumned and can have several *sidebars*. Sidebars are explicit boxes enclosing text and figures used for topics that are not part of the mainstream of text. They can either span all or some of the columns of text. Readers wanting a quick overview usually read the main text and avoid the sidebars. The block classification phase labels both the mainstream of text and the sidebars as textual blocks. It is up to the logical grouping phase to order the blocks of the main text body into a continuous stream by ignoring the sidebars.

Another grouping task pertinent to this phase is matching titles and highlight boxes to the corresponding text blocks in the mainstream. When a title pertains to a single-columned block, associating the title with the corresponding text is straightforward. However, titles can span several columns, and sometimes can be located at the center of the page without aligning with any of the columns, making the task of logical grouping challenging.

One method of logical grouping is to use rules of the layout structure of the document. Following are some examples of rules derived from the literature on "newspaper design" used in the spatial knowledge module of this approach.

- R21: Headlines occupying more than one printed line are left-justified
- R32: Captions are always below photographs, unless two or more photographs have a common caption
- R43: Explicit boxes around blocks signify an independent unit.

Blocks with different labels (photograph and text) that are not necessarily adjacent might have to be grouped together. For instance, a photograph and its accompanying caption together form a logical unit and must be linked together in the output representation.

## TEXT RECOGNITION AND INTERPRETATION

Character recognition, also known as optical character recognition or OCR, is concerned with the automatic conversion of scanned and digitized images of characters in running text into their corresponding symbolic forms. The ability of humans to read poor quality machine print as well as text with unusual fonts and handwriting is far from matched by today's machines.

The task of pushing machine-reading technology to reach human capability calls for developing and integrating techniques from several subareas of artificial intelligence research. Since machine reading is a continuum from the visual domain to the natural language understanding domain, the subareas of research include: computer vision, pattern recognition, natural language processing, and knowledge-based methods.

### Recognition Without Context

The task is associated with the image of a segmented, or isolated, character its symbolic identity. However, segmentation

of a field of characters into individual characters may well depend on preliminary recognition.

Although there exists a large number of recognition techniques, the creation of new fonts, the occasional presence of decorative or unusual fonts, and degradations caused by faxed and multiple generation of copies continues to make isolated character recognition a topic of importance.

Recognition techniques involve feature extraction and classification. The extraction of appropriate features is the most important subarea. Character features that have shown great promise are strategically selected pixel pairs, features from histograms, features from gradient and structural maps, and morphological features. Features derived from gray-scale imagery is a relatively new area of OCR research. Gray-scale imagery is not traditionally used in OCR due to the large increase in the amount of data to be used: a restriction that can be removed with current computer technology.

Classification techniques that have found promise are a polynomial classifier, neural networks based on backpropagation, and Bayes classifiers that assume feature independence and binary features. Orthogonal recognition techniques can be combined in order to achieve robustness over a wide range of fonts and degradations.

### Recognition with Context

The problem of character recognition is a special case of the general problem of reading. While characters occasionally appear in isolation, they predominately occur as parts of words, phrases, and sentences. Even though a poorly formed or degraded character in isolation may be unrecognizable, the context in which the character appears can make the recognition problem simple. The utilization of model knowledge about the domain of discourse as well as constraints imposed by the surrounding orthography is the main challenge in developing robust methods. Several approaches to utilize models at the word level and at a higher linguistic level are known.

### Character Recognition

Two essential components in a character recognition algorithm are the feature extractor and the classifier. Feature analysis determines the descriptors, or feature set, used to describe all characters. Given a character image, the feature extractor derives the features that the character possesses. The derived features are then used as input to the character classifier.

Template matching, or matrix matching, is one of the most common classification methods. In template matching, individual image pixels are used as features. Classification is performed by comparing an input character image with a set of templates (or prototypes) from each character class. Each comparison results in a similarity measure between the input character and the template. One measure increases the amount of similarity when a pixel in the observed character is identical to the same pixel in the template image. If the pixels differ the measure of similarity may be decreased. After all templates have been compared with the observed character image, the character's identity is assigned as the identity of the most similar template.

Template matching is a trainable process because template characters may be changed. In many commercial systems, PROMs (programmable read-only memory) store templates

containing single fonts. To retain the algorithm the current PROMs are replaced with PROMs that contain images of a new font. Thus, if a suitable PROM exists for a font then template matching can be trained to recognize that font. The similarity measure of template matching may also be modified, by commercial OCR systems typically do not allow this.

Structural classification methods utilize structural features and decision rules to classify characters. Structural features may be defined in terms of character strokes, character holes, or other character attributes such as concavities. For instance, the letter P may be described as a vertical stroke with a hole attached on the upper right side. For a character image input, the structural features are extracted and a rule-based system is applied to classify the character. Structural methods are also trainable but construction of a good feature set and a good rule-base can be time-consuming.

Many character recognizers are based on mathematical formalisms that minimize a measure of misclassification. These recognizers may use pixel-based features or structural features. Some examples are discriminant function classifiers, Bayesian classifiers, artificial neural networks (ANN), and template matchers. Discriminant function classifiers use hypersurfaces to separate the feature description of characters from different semantic classes and in the process reduce the mean-squared error. Bayesian methods seek to minimize the loss function associated with misclassification through the use of probability theory. ANNs, which are closer to theories of human perception, employ mathematical minimization techniques. Both discriminant functions and ANNs are used in commercial OCR systems.

Character misclassifications stem from two main sources: poor quality character images and poor discriminatory ability. Poor document quality, image scanning, and preprocessing can all degrade performance by yielding poor quality characters. On the other hand, the character recognition method may not have been trained for a proper response on the character causing the error. This type of error source is difficult to overcome because the recognition method may have limitations and all possible character images cannot possibly be considered in training the classifier. Recognition rates for machine-printed characters can reach over 99% but handwritten character recognition rates are typically lower because every person writes differently. This random nature often manifests itself by resulting in misclassifications. Figure 3 shows several examples of machine-printed and handwritten capital O's. Each capital O can be easily confused with the numeral 0 and the number of different styles of capital O's demonstrates the difficulties recognizers must cope with.

### Image Enhancement

A character image is usually a bilevel image produced by an imperfect binarization process which results in fragmentation. It is nontrivial to choose a threshold so that the connectivity of character strokes is preserved. Traditional ways of image enhancement include filtering methods such as low pass filters and high pass filters using Fourier transforms. They are suitable for general enhancement such as smoothing and edge detection. However, for the purpose of character image enhancement and subsequent recognition these methods are not adequate.

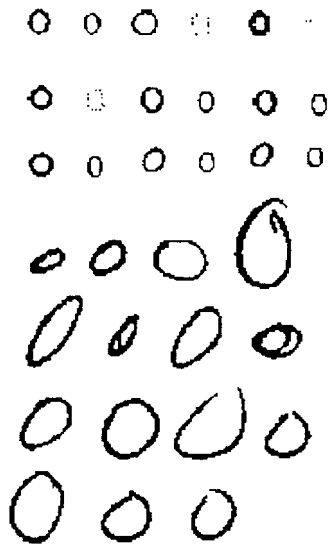


Figure 3. Machine-printed and handwritten capital O's.

Neighborhood operators have been widely used for image enhancement of general scenes as well. Although their implementation on general purpose computers is slow, their simplicity renders them suitable for customized hardware. While they have been found to be suitable for enhancing character images, typically they have failed to reconstruct broken strokes.

One method of performing character image enhancement is by using a neighborhood operator. The method applies to both handwritten and machine-printed binary images efficiently. It emphasizes stroke connectivity while at the same time conservatively checks aggressive "over-filling". It is implemented using a binary tree structure that ensures an efficient single pass algorithm.

Reconstruction of character strokes requires filling in gaps in the broken strokes caused by imperfect binarization. It is only local connectivity information that is available for the reconstructing procedure. The task can be explained in terms of the task of making a map of a forest based on a series of local views. Each local view is centered upon a tree and few of its immediate neighbors. One can go around a tree and mark the neighboring trees. The procedure is repeated at the boundary of the marked neighborhood to reach unmarked trees until all trees in the forest are marked. While preparing the map, small gaps between trees are also marked.

Figure 4 illustrates the processing on a character image extracted from an address block. The strokes from adjacent characters are in close proximity which makes it difficult to reconstruct broken strokes using traditional methods (such as Fourier-based methods). The method described in this article is selective in its choice of neighborhoods that undergo reconstruction. This is achieved by centering the neighborhood on black pixel areas. As is indicated in the figure, the width of character strokes is unchanged in the enhanced image. The selective mechanism has the added advantage of being fast as enhancement is invoked only in selected regions of the image.

#### TABLES, GRAPHICS, AND HALFTONE RECOGNITION

Tables, diagrams, and images are often integral components of documents. Each of these document component types share

the characteristic of being diagrammatic representations. Their interpretation is related to the human brain's capacity for analogical (or spatial) reasoning. They are used to explicitly represent information (and thus permit direct retrieval) of information which can be expressed implicitly using other representations. Furthermore, there may be a considerable cost in converting from the implicit to the explicit representation.

The interpretations of tables, diagrams, and images also share the characteristic that it is usually necessary to integrate visual information (e.g., lines separating columns, arrows, photographs) with information obtained from text (e.g., labels, captions). The central issue here is defining a static representation of meaning associated with each of these document component types. If such a meaning can be captured, it can be incorporated into the data structure representing the overall understanding of the document. This would allow for interactive user-queries at a later time. Hence, document understanding deals with the definition of meaning and how to go about extracting such a meaning.

For each of the three component types—tables, diagrams, and images—we address the following issues:

1. meaning
2. complexity of visual processing
3. knowledge required
4. system architecture (data and control structures).

Tables are usually stand-alone units that have little interaction with accompanying text (except for table captions), diagrams need a higher level of interaction, and photographs usually need a textual explanation. We consider the integration with accompanying text only in our discussion on image understanding.

#### Engineering Drawing and Map Interpretation

Diagrams are used to convey information which is understood by humans more readily when presented visually. This category is very diverse and includes domains such as maps, engineering drawings, and flowcharts. The purpose of diagram understanding can be either one or both of the following: (i) to transform the paper representation into a more useful target

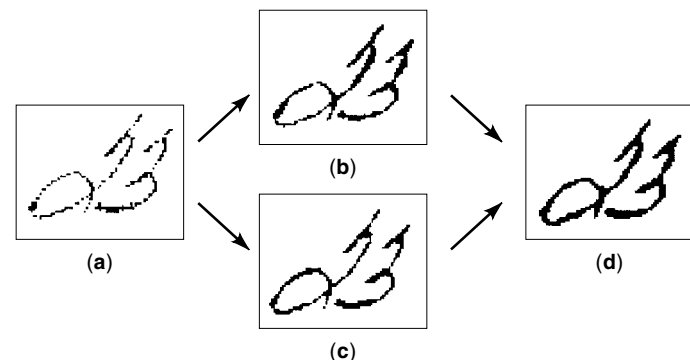


Figure 4. Character image enhancement: (a) Original image, (b) row major processing with single run, (c) column major processing with single run, and (d) combining row major and column major processing.

representation (such as commands to a graphics program, new entries in an integrated text-graphic database), and (ii) to derive a more compact representation of the data for the purpose of creating an archive.

A simplistic view of diagram understanding is the conversion of a raster representation to a vector representation: that is, to convert a binary pixel representation of line-work into a connected set of segments and nodes. Segments are typically primitives such as straight lines, parametric curves, domain-specific graphical icons, and text. In addition, portions of the drawing containing text must be converted to ASCII, and graphical icons must be recognized and converted to their symbolic representation. Line segments have parameters such as start positions, extent, orientation, line width, and pattern associated with them. Similar features are associated with parametric curves. The connections between segments represent logical (typically, spatial) relationships.

A deeper level of understanding can be attained if groups of primitives (lines, curves, text, icons) are combined to produce an integrated meaning. For example, in a map, if a dotted line appears between two words (representing city names), and the legend block associates a dotted line with a two-lane highway, it means that a two-lane highway exists between the two cities. It is possible to define meanings for documents such as maps, engineering drawings, and flow-charts. The definition of meaning is somewhat ambiguous in diagrams such as those found in a physics textbook.

It is necessary to have a priori knowledge of the types and meanings of primitives for a given context. In the case of line drawings, it is necessary to represent the various types of lines and curves that may appear. This could include higher level knowledge such as angles (formed by the intersection of two lines) and directed arrows. In the domain of geographic maps for example, the symbol resembling a ladder represents a railroad. It is also necessary to have a lexicon of typical textual primitives along with their meaning. This aids both the text recognition process as well as the later stages of understanding.

In addition to the knowledge used for diagram analysis, domain-specific knowledge regarding the interpretation of higher level units must also be represented. Finally, information contained in an accompanying caption block may be used to determine the type of diagram, or additionally, to be used in the process of understanding. This area is addressed extensively in the discussion of image understanding.

Visual processing includes the following:

- Separating text from the image: This is a nontrivial process since text may be touching lines. In such cases, a technique is to determine the high neighborhood line density (NLD) areas in the line structures. Based on the size of connected components and values of NLD, it is possible to separate the characters from the line.
- Vectorization: There have been several approaches to vectorization including: pixel-based thinning, run-length-based vectorization, and contour-based axis computation. Parametric curves can be approximated either through curve-fitting algorithms or piecewise linear approximation.
- Graphical icons are best detected through template-matching procedures.
- Some other problems that need to be specifically examined are handling of dotted lines, determining end points of lines, effectiveness of thinning, and determination of vector intersections.

A hierarchical, modular system architecture is employed. The lower level modules perform the tasks of contour and text extraction. At the next level, intermediate-level primitives such as polygons, circles, and angles are determined. Finally, high-level domain-specific knowledge is employed to derive a conceptual understanding of a diagram. This information is then converted to the target representation, such as commands in a graphics language, icon database, or a CAD/CAM file. Once again, bidirectional flow of control is employed whereby domain-specific knowledge is used in a top-down manner and inconsistencies are propagated in a bottom-up manner.

### Form Processing Systems

A paper form is any preprinted document designed to elicit information. The problem of automatically interpreting an optically scanned and digitized image of a filled-out paper form containing typed and handwritten entries is considered in this chapter. The goal of forms analysis is to segment the form into the basic components of text (characters and words), line segments, and boxes, and then analyze the components to extract relevant information from the form.

Forms analysis has grown to be a major component of document image understanding. Imaging-based technology, as well as several innovative forms analysis techniques, have made it much easier, quicker, and cheaper to process the hundreds of different types of forms that typically need to be processed.

Most common documents that are processed are forms. For example, bank checks are small, simple forms. All kinds of organizations use forms: finance, education, retail, health care, shipping, and transportation. Forms hasten market research, order entry, payment, and delivery. Governments, especially, deal in a vast number of forms.

In forms processing, one usually handles completed forms that arrive via fax or scanned documents. After cleanup and processing, the form images can be routed on a network or stored. After extracting the data from a form and verifying its accuracy, one does not need to keep the form. (One part, the signature, might need to be archived, but there is no need to record indefinitely whether someone filled in a box with a checkmark, an X, or a solid circle.) Once the correctly extracted data enters the database, the data become the important resource to safeguard and process, not the form from which they came.

Forms are more complex and harder to work with than other imaging documents due to several reasons. First, forms can contain more than machine-readable text—they often come with checkmarks, handprint, and signatures, all of which the system must recognize and process. Second, forms processing systems need to OCR text in discrete chunks surrounded by graphic elements, such as lines and boxes and “combs.” Third, forms contain lines, fancy type, logos, and guide text (instructions that help users to fill the form), all of which hinder compression. Fourth, because it contains a lot of extra elements, a compressed form can be four to 10 times



the size of the same data without the form. Fifth, with forms, one might perform different OCR operations on them to extract different types of data based on preliminary data that is read from the form (e.g., type of form). Sixth, forms are designed to be filled in by hand, but handprinting is more difficult to read than machine-generated text, since there is more variation. Fortunately, forms present blocks or fields to constrain the handprint.

Current research and development in forms processing includes the use of forms processing techniques in reading IRS tax forms and for extracting information from business reply cards. In the first application, address block labels are extracted from IRS tax forms. Only the address block is received from the IRS tax forms, and data are extracted from these address blocks. All but one of the forms have “drop-out” guidelines, which allow a person to fill out the form within the boundaries marked, but also allow the form to be processed electronically without the guidelines being present to interfere with the recognition of the characters. The only exception to this is the IRS Form 1040EZ, which does not have drop-off guidelines. Therefore, for this form, the guidelines have to be first removed before any further processing. In the second application, the system analyzes certain types of forms, such as postal replay cards. The system is designed to acquire the images of the information-bearing side of the replay cards; determine the recipient’s ID to determine the required service(s) and to retrieve relevant a priori information regarding that particular reply card; extract the information contents (both address and non-address) automatically; forward the image to a site for further manual processing, if automated processing fails; organize the extracted information in a format specified by the recipient; and aggregate the processing results of the reply cards destined for the same recipient. The system has the capability of processing both handwritten and machine-printed reply cards.

## DATABASES AND SYSTEM PERFORMANCE EVALUATION

Performance evaluation of document understanding systems can be an important guide to research. The results of performance evaluation could be used to allocate resources to difficult, unsolved problems that are significant barriers to achieving high performance.

A precise definition of the goal of the system being measured and any intermediate steps that lead to a solution of that goal are essential to achieve a useful performance evaluation. A model for system performance should be developed and the importance of each intermediate step in achieving the overall goal of the system should be defined. This is to ensure that research efforts are properly directed.

A representative image database and indicative testing procedure should also be defined. The database should reflect the environment in which the system will be applied and the testing procedure should fairly determine the performance of the intermediate steps as well as the final goal of the system. The image database should contain a mixture of stress cases designed to determine the response of the system to various potential problems as well as a random sample of the images the system is expected to encounter. A selective set of stress cases are useful for initial system development. A large ran-

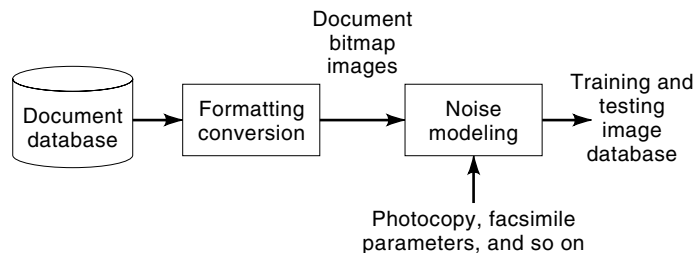


Figure 5. Database and performance testing.

dom sample, perhaps on the order of tens of thousands of images, is useful for testing a mature system.

## Performance Model Design

A systematic definition of the desired performance at each of the intermediate levels in a document analysis system is necessary to predict overall system performance. An analytic model based on the subcomponents of a system should be derived. The model should be validated by run-time observations.

## Database and Testing Procedure

The database used for performance evaluation has two essential components: images and ASCII truth. Traditionally, images for a testing database are generated by scanning selected documents. Truth values are applied by a manual process that includes “boxing” regions in the documents and typing in the complete text within the document. This can be labor intensive and error prone. Furthermore, multiple iterations of manual truthing may be needed as system requirements change.

An alternative is to generate test data directly from the ASCII truth. An application of a similar methodology to document analysis is shown in Fig. 5. An SGML representation for a set of documents would be input. The necessary macros would be defined to provide the physical realization of the document. After running a formatting package, the resultant bitmap image of the document would be saved. Such images could then be corrupted with noise to generate test data. Models for different noise sources, such as facsimile or photocopy processes, could be used. It is interesting to note that a similar strategy for developing OCR algorithms based on synthetic noise models has been successful.

The advantages of this approach for database generation include the flexibility it provides in the use of different formatting macros. Versions of the same logical document generated in a range of fonts, sizes, styles, and so on, could be utilized. This would allow for the testing of any format-dependent characteristics of the system. Examples of different document formats would not need to be found by an exhaustive search. If it is desired to provide a system with the capability to recognize a certain format of document, that format could be generated synthetically from the database and it would not be necessary to encounter a large number of examples of that format a priori. An example of this would be if a large number of open-source documents printed in 10-point type with three columns per page, where each column had a

ragged left edge, were going to be seen. If a sufficient number of documents in that format were not actually available, they could be generated synthetically. After appropriate training, the system would be ready to process the actual documents when they arrived.

An additional advantage of synthetic data generation is the ability to model noise sources and simulate various reasons for errors. Models for noise caused by repeated photocopying or facsimile transmission would be quite valuable. The performance of document analysis systems operating under various levels of noise could then be characterized. Together with the ability to change formats at will, noise modeling would provide an ideal method for testing a document analysis system under a variety of constraints. Everything from document decomposition to any associated OCR processes could be stress tested.

The procedure used for any comparative testing should be carefully considered. A substantial set of training data, representative of any test data that will be processed, should be provided to all concerned parties. Each group should develop their system on this data and demonstrate performance under a variety of conditions.

One scenario for testing would include the distribution of a quantity of document images without truth. A limited time would be provided for the testing and return of results. A neutral third party would evaluate the performance. Only enough time would be provided for one round of testing.

Another scenario for testing would require participants to install copies of the code for their systems at a neutral location. This party would perform tests on a common database and evaluate results in a standard format. This methodology would eliminate any of the natural bias that occurs when the developers of a system test it themselves.

## SUMMARY

The major processes needed to develop a DIP system have been described. The processes contained in the system architecture are: decomposition and structural analysis; text recognition and interpretation; and table, diagram, and image understanding. The system architecture provides a computational framework to integrate and regulate activities needed in document layout analysis and content interpretation. Decomposition and structural analysis is responsible for decomposing a document into several regions, each of which contains homogeneous entities. These regions are then grouped into logical units to form a high-level interpretation of the document structure. Current OCR technology has performed well with limited fonts but has limited success in recognizing unusual fonts and poor quality text. The use of contextual information, such as lexicon and syntax, has shown promising results in degraded text recognition. Preprocessing of character images for image enhancement assists the subsequent recognition process in both handwritten and machine-printed documents.

As can be seen from these descriptions, document image processing is a complex field of research with many challenging topics. It has also given rise to innovations in modern OCR technology and the advent of several leading-edge products.

## Reading List

- H. Bunke and P. S. P. Wang (eds.), *Handbook of Character Recognition and Document Image Analysis*, River Edge, NJ: World Scientific, 1997.
- R. G. Matteson, *Introduction to Document Image Processing Techniques*, Boston: Artech House, 1995.

SARGUR N. SRIHARI  
DEBASHISH NIYOGI  
State University of New York at  
Buffalo