Sādhanā Vol. 27, Part 1, February 2002, pp. 3–22. © Printed in India

Document image analysis: A primer

RANGACHAR KASTURI[1], LAWRENCE O'GORMAN[2] and VENU GOVINDARAJU[3]

[1] Department of Computer Science & Engineering, The Pennsylvania State University, University Park, PA 16802, USA
[2]Avaya Labs, Room 1B04, 233 Mt. Airy Road, Basking Ridge, NJ 07920, USA
[3]CEDAR, State University of New York at Buffalo, Amherst, NY 14228, USA
e-mail: kasturi@cse.psu.edu; logorman@avaya.com; govind@cedar.buffalo.edu

Abstract. *Document image analysis* refers to algorithms and techniques that are applied to images of documents to obtain a computer-readable description from pixel data. A well-known document image analysis product is the Optical Character Recognition (OCR) software that recognizes characters in a scanned document. OCR makes it possible for the user to edit or search the document's contents. In this paper we briefly describe various components of a document analysis system. Many of these basic building blocks are found in most document analysis systems, irrespective of the particular domain or language to which they are applied. We hope that this paper will help the reader by providing the background necessary to understand the detailed descriptions of specific techniques presented in other papers in this issue.

Keywords. OCR; feature analysis; document processing; graphics recognition; character recognition; layout analysis.

1. Introduction

The objective of document image analysis is to recognize the text and graphics components in images of documents, and to extract the intended information as a human would. Two categories of document image analysis can be defined (see figure 1). Textual processing deals with the text components of a document image. Some tasks here are: determining the skew (any tilt at which the document may have been scanned into the computer), finding columns, paragraphs, text lines, and words, and finally recognizing the text (and possibly its attributes such as size, font etc.) by optical character recognition (OCR). Graphics processing deals with the non-textual line and symbol components that make up line diagrams, delimiting straight lines between text sections, company logos etc. Pictures are a third major component of documents, but except for recognizing their location on a page, further analysis of these is usually the task of other image processing and machine vision techniques. After application of these text and graphics analysis techniques, the several megabytes of initial data are culled to yield a much more concise semantic description of the document.

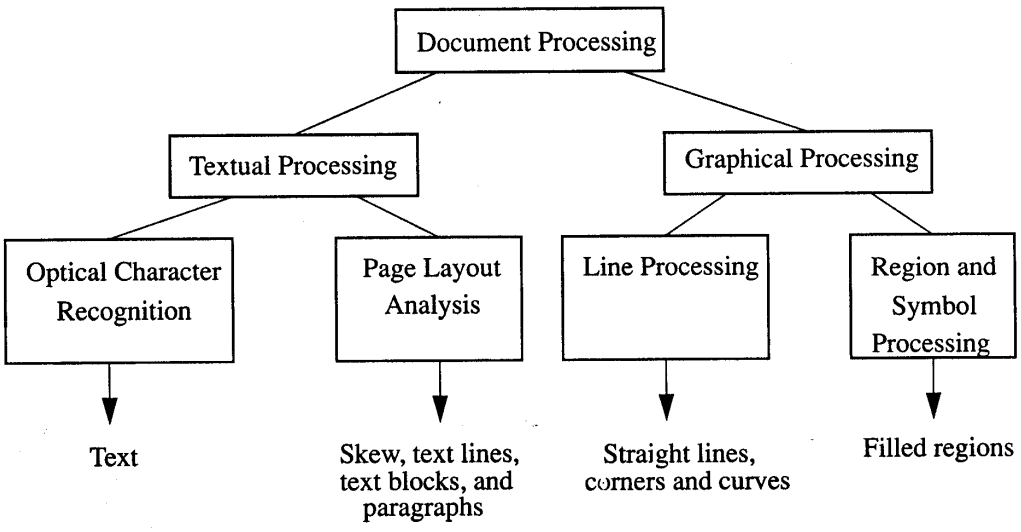

Figure 1. A hierarchy of document processing subareas listing the types of document components dealt within each subarea. (Reproduced with permission from O'Gorman & Kasturi 1997.)

Consider three specific examples of the need for document analysis presented here.

(1) Typical documents in today's office are computer-generated, but even so, inevitably by different computers and software such that even their electronic formats are incompatible. Some include both formatted text and tables as well as handwritten entries. There are different sizes, from a business card to a large engineering drawing. Document analysis systems recognize types of documents, enable the extraction of their functional parts, and translate from one computer generated format to another.
(2) Automated mail-sorting machines to perform sorting and address recognition have been used for several decades, but there is the need to process more mail, more quickly, and more accurately.
(3) In a traditional library, loss of material, misfiling, limited numbers of each copy, and even degradation of materials are common problems, and may be improved by document analysis techniques. All these examples serve as applications ripe for the potential solutions of document image analysis.

Document analysis systems will become increasingly more evident in the form of everyday document systems. For instance, OCR systems will be more widely used to store, search, and excerpt from paper-based documents. Page-layout analysis techniques will recognize a particular form, or page format and allow its duplication. Diagrams will be entered from pictures or by hand, and logically edited. Pen-based computers will translate handwritten entries into electronic documents. Archives of paper documents in libraries and engineering companies will be electronically converted for more efficient storage and instant delivery to a home or office computer. Though it will be increasingly the case that documents are produced and reside on a computer, the fact that there are very many different systems and protocols, and also the fact that paper is a very comfortable medium for us to deal with, ensures that paper documents will be with us to some degree for many decades to come. The difference will be that they will finally be integrated into our computerized world.

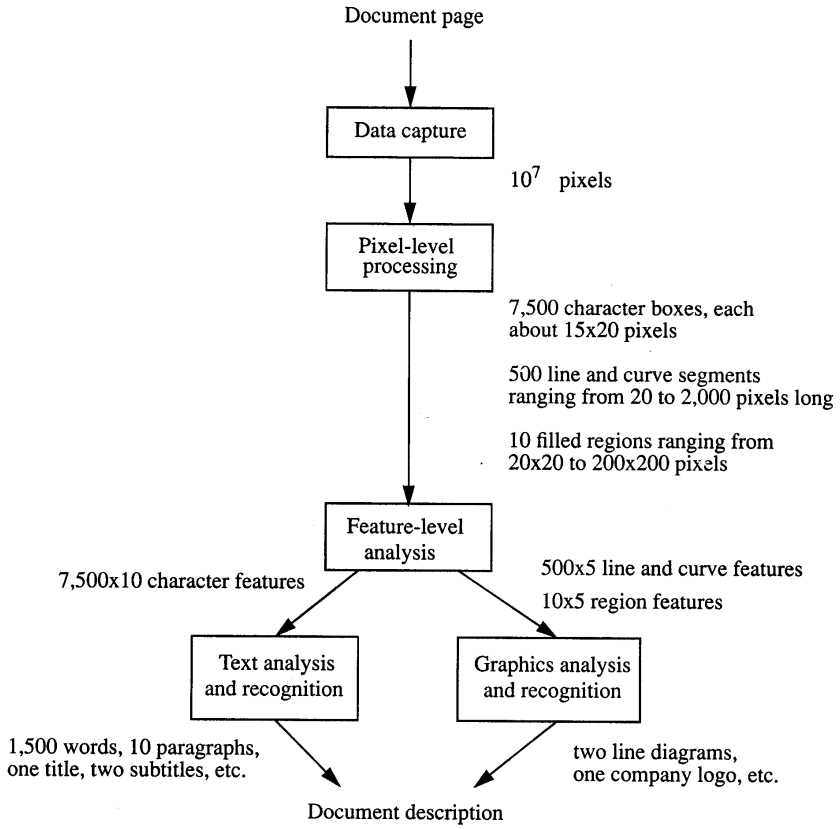

Figure 2. A typical sequence of steps for document analysis, along with examples of intermediate and final results and the data size. (Reproduced with permission from O'Gorman & Kasturi 1997.)

Figure 2 illustrates a common sequence of steps in document image analysis. After data capture, the image undergoes pixel-level processing and feature analysis and then text and graphics are treated separately for the recognition of each. We describe these steps briefly in the following sections; the reader is referred to the book, *Document image analysis*, for details (O'Gorman & Kasturi 1997). We conclude this paper by considering the challenges in analysing multilingual documents which is particularly important in the context of Indian language document analysis.

2. Data capture

Data in a paper document are usually captured by optical scanning and stored in a file of picture elements, called pixels, that are sampled in a grid pattern throughout the document. These pixels may have values: OFF (0) or ON (1) for binary images, 0–255 for gray-scale images, and 3 channels of 0–255 colour values for colour images. At a typical sampling resolution of 120 pixels per centimetre, a 20 x 30 cm page would yield an image of 2400x3600 pixels. When the document is on a different medium such as microfilm, palm leaves, or fabric, photographic methods are often

used to capture images. In any case, it is important to understand that the image of the document contains only raw data that must be further analysed to glean the information.

3. Pixel-level processing

The next step in document analysis is to perform processing on the captured image to prepare it for further analysis. Such processing includes: Thresholding to reduce a gray-scale or colour image to a binary image, reduction of noise to reduce extraneous data, segmentation to separate various components in the image, and, finally, thinning or boundary detection to enable easier subsequent detection of pertinent features and objects of interest. After such processing, data are often represented in compact form such as chain-codes and vectors. This pixel-level processing (also called preprocessing and low-level processing in other literature) is the subject of this section.

3.1 *Binarization*

For gray-scale images with information that is inherently binary such as text or graphics, binarization is usually performed first. The objective of binarization is to automatically choose a threshold that separates the foreground and background information. Selection of a good threshold is often a trial and error process (see figure 3). This becomes particularly difficult in cases where the contrast between text pixels and background is low (for example, text printed on a gray background), when text strokes are very thin resulting in background bleeding into text pixels during digitization, or when the page is not uniformly illuminated during data capture. Many methods have been developed for addressing these problems including those that model the background and foreground pixels as samples drawn from statistical distributions and methods based on spatially varying (adaptive) thresholds. Whether global or adaptive thresholding methods are used for binarization, one can seldom expect perfect results. Depending on the quality of the original, there may be gaps in lines, ragged edges on region boundaries, and extraneous pixel regions of ON and OFF values. This fact, that processed results are not perfect, is generally true with other document processing methods, and indeed image processing in general. The recommended procedure is to process as well as possible at each step of processing, but to defer decisions that do not have to be made until later, to avoid making irreparable errors. In later steps, there is more information as a result of processing up to that point, and this provides greater context and higher level descriptions to aid in making correct decisions, and ultimately recognition. Some good references for thresholding methods are Reddi *et al* (1984), Tsai (1985), Sahoo *et al* (1988), O'Gorman (1994), and Trier & Taxt (1995).

3.2 *Noise reduction*

Document image noise is due to many sources including degradation due to aging, photocopying, or during data capture. Image and signal processing methods are applied to reduce noise. After binarization, document images are usually filtered to reduce noise. Salt-and-pepper noise (also called impulse and speckle noise, or just dirt) is a prevalent artifact in poorer quality document images (such as poorly thresholded faxes or poorly photocopied pages). This appears as isolated pixels or pixel regions of ON noise in OFF backgrounds or OFF noise (holes) within ON regions, and as rough edges on character and graphics components.

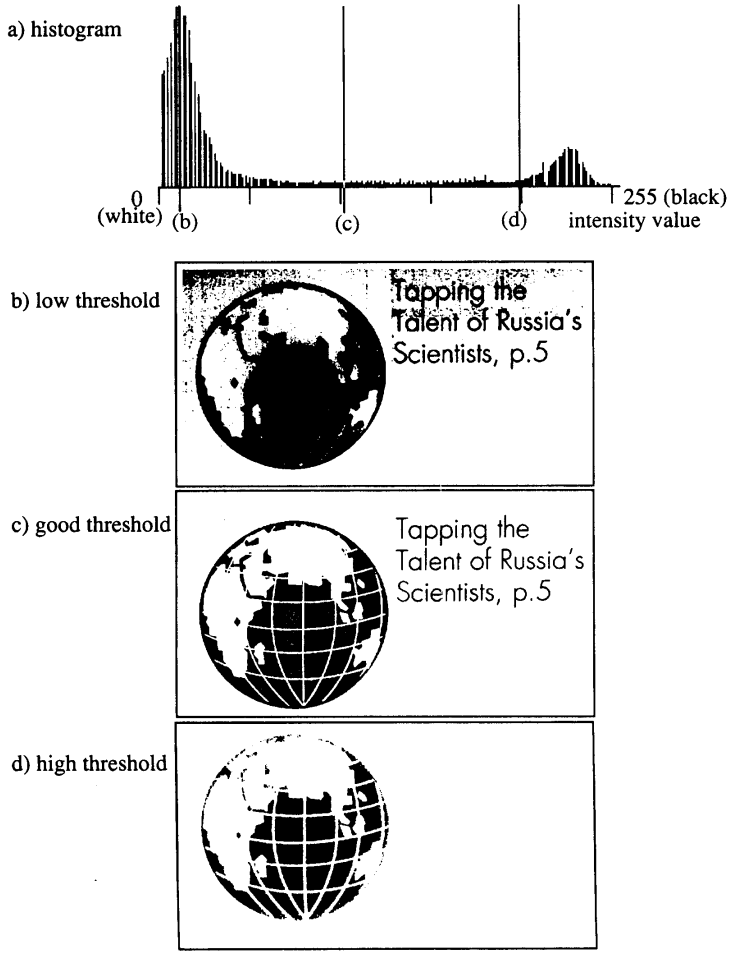

Figure 3. Image binarization. **(a)** Histogram of original gray-scale image. Horizontal axis shows markings for threshold values of images below. The lower peak is for the white background pixels, and the upper peak is for the black foreground pixels. Image binarized with: **(b)** too low a threshold value, **(c)** a good threshold value, and **(d)** too high a threshold value. (Reproduced with permission from O'Gorman & Kasturi 1997.)

The process of reducing this is called "filling". The most important reason to reduce noise is that extraneous features otherwise cause subsequent errors in recognition. The objective in the design of a filter to reduce noise is that it remove as much of the noise as possible while retaining all of the signal. Morphological (Serra 1982; Haralick *et al* 1987; Haralick & Shapino 1992) methods are frequently used for noise reduction. The basic morphological operations are erosion and dilation. Erosion is the reduction in size of ON-regions. This is most simply accomplished by peeling off a single-pixel layer from the outer boundary of all ON regions on each erosion step. Dilation is the opposite process, where single-pixel, ON-valued layers are added to boundaries to increase their size. These operations are usually combined and applied iteratively to erode and dilate many layers. One of these combined operations is called opening, where one or more iterations of erosion are followed by the same number of iterations of dilation. The result of opening is that boundaries can be smoothed,

narrow isthmuses broken, and small noise regions eliminated. The morphological dual of opening is closing. This combines one or more iterations of dilation followed by the same number of iterations of erosion. The result of closing is that boundaries can be smoothed, narrow gaps joined, and small noise holes filled. For documents, more specific filters can be designed to take advantage of the known characteristics of the text and graphics components. In particular, we desire to maintain sharpness in these document components, not to round corners and shorten lengths, as some noise reduction filters do. For the simplest case of single-pixel islands, holes, and protrusions, these can be found by passing a 3 x 3 sized window over the image that matches these patterns (Shih & Kasturi 1988), then filled. For noise larger than one-pixel, the kFill filter can be used (O'Gorman 1992).

3.3 *Segmentation*

Segmentation occurs on two levels. On the first level, if the document contains both text and graphics, these are separated for subsequent processing by different methods (Wong 1982; Fletcher 1988; Jain 1992). On the second level, segmentation is performed on text by locating columns, paragraphs, words, and characters; and on graphics, segmentation usually includes separating symbol and line components. For instance, in a page containing text and some illustrations similar to the pages of this journal, text and graphics are first separated. Then the text is separated into its components down to individual characters. The graphics is separated into its components such as rectangles, circles, connecting lines, symbols etc. After this step an image is typically broken down into its basic components such as an individual character or a graphical element.

3.4 *Thinning and region detection*

Thinning is an image processing operation in which binary valued image regions are reduced to lines that approximate the centre lines, or skeletons, of the regions. The purpose of thinning is to reduce the image components to their essential information so that further analysis and recognition are facilitated. For instance, a line drawing can be handwritten with different pens giving different stroke thicknesses, but the information presented is the same. In figure 4, some images are shown whose contents can be analysed well due to thinning, and their thinning results are also shown here. Note should be made that thinning is also referred to as skeletonizing and core-line detection in the literature. We will use the term "thinning" to describe the procedure, and thinned line, or skeleton, to describe the results. A related term is the "medial axis". This is the set of points of a region in which each point is equidistant to its two closest points on the boundary. The medial axis is often described as the ideal that thinning approaches. However, since the medial axis is defined only for continuous space, it can only be approximated by practical thinning techniques that operate on a sampled image in discrete space. Several thinning algorithms have been described by Arcelli & Sanniti di Baja (1985, 1993), and Sanniti di Baja (1994) and an algorithm that thins by several pixels in each pass is described by O'Gorman (1990). Reviews of thinning methods have been given by Lam *et al* (1992) and Lam & Suen (1995).

Note that a circle or a square that is completely filled with black pixels results ideally in a single pixel at the centre of the shape, irrespective of the size of the original. Clearly, it is more useful to detect the boundary of such objects. In general, for large blob-like objects, region-boundary detection is preferred and for objects made up of long connected strokes, thinning is the method of choice. Thinning is commonly used in the preprocessing stage of such document analysis applications as diagram understanding and map processing. For

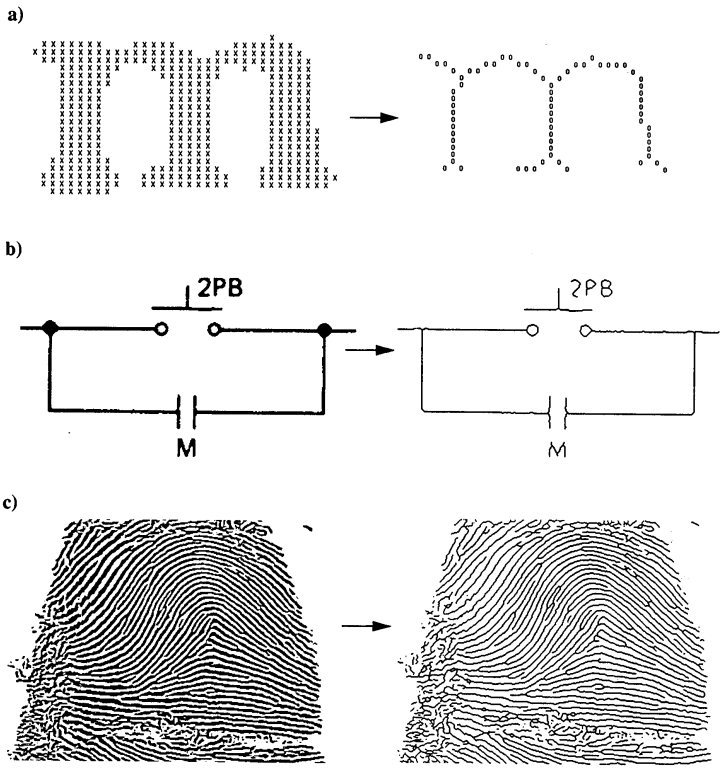

Figure 4. Original images on left and thinned image results on right. **(a)** The letter "m". **(b)** A line diagram. **(c)** A fingerprint image. (Reproduced with permission from O'Gorman & Kasturi 1997.)

recognition of large graphical objects with filled regions which are often found in logos, region boundary detection is useful. But for small regions such as those which correspond to individual characters, neither thinning nor boundary detection is performed and the entire pixel array representing the region is forwarded to the subsequent stage of analysis.

3.5 *Chain coding and vectorization*

When objects are described by their skeletons or contours, they can be represented more efficiently than simply by ON and OFF valued pixels in a raster image. One common way to do this is by chain coding (Freeman 1974), where the ON pixels are represented as sequences of connected neighbours along lines and curves. Instead of storing the absolute location of each ON pixel, the direction from its previously coded neighbour is stored. A neighbour is any of the adjacent pixels in the 3 x 3 pixel neighbourhood around that centre pixel (see figure 5). There are two advantages of coding by direction versus absolute coordinate location. One is in storage efficiency. For commonly

3	2	1
4	X	0
5	6	7

Figure 5. For a 3 x 3 pixel region with centre pixel denoted as X, figure shows codes for chain directions from centre pixel to each of eight neighbours: 0 (east), 1 (northeast), 2 (north), 3 (northwest), etc. (Reproduced with permission from O'Gorman & Kasturi 1997.)

sized images larger than 256x256, the coordinates of an ON-valued pixel are usually represented as two 16-bit words; in contrast, for chain coding with eight possible directions from a pixel, each ON-valued pixel can be stored in a byte, or even packed into three bits. A more important advantage in this context is that, since chain coding contains information on connectedness within its code, this can facilitate further processing such as smoothing of continuous curves, and approximation of smoothed straight lines by vectors.

After pixel-level processing, the raw image data is converted to a higher level of abstraction; viz., regions representing individual characters, chain codes or vectors representing curve and straight line segments, and boundaries representing large solid objects.

4. Feature-level analysis

After pixel-level processing has prepared the document image, intermediate features are found from the image to aid in the final step of recognition. At the feature level, thinned and chain-coded data is analysed to detect straight lines, curves, and significant points along the curves. This is a more informative representation that is also closer to how humans would describe the diagram – as lines and curves rather than as ON and OFF points. Curved lines are often approximated by polygonalization. Critical points such as corners and points of high curvature are determined to assist in subsequent analysis for shape recognition. For regions corresponding to individual characters or graphical symbols, local features such as aspect ratio, compactness (ratio of area to square of perimeter), asymmetry, black pixel density, contour smoothness, number of loops, number of line crossings and line ends etc. are computed for input to object recognition stage.

4.1 *Line and curve fitting*

A simple way to fit a straight line to a sequence of points is to just to specify the endpoints as the straight line endpoints. However, this may result in some portions of the curve having large error with respect to the fit. A popular way to achieve the lowest average error for all points on the curve is to perform a least-squares fit of a line to the points on the curve. Especially for machine-drawn documents such as engineering drawings, circular curve features are prevalent. These features are described by the radius of the curve, the centre location of the curve, and the two transition locations where the curve either ends or smoothly makes the transition to one or two straight lines around it. One sequence of procedures in circular curve detection begins by finding the transition points of the curve, that is the locations along the line where a straight line makes a transition into the curve, and then becomes a straight line again. Next, a decision is made on whether the feature between the straight lines is actually a corner or a curve. Finally, the centre of the curve is determined. If a higher order fit is desired, splines can be used to perform piecewise polynomial interpolations among data points (Pavlidis 1982; Medioni & Yasumoto 1987). B-splines are piecewise polynomial curves that are specified by a guiding. Given two endpoints and a curve represented by its polygonal fit (described next), a B-spline can be determined that performs a close but smooth fit to the polygonal approximation. The B-spline has several properties that make it a popular spline choice. A different approach for line and curve fitting is by the Hough transform (Illingworth & Kittler 1988). This approach is useful when the objective is to find lines or curves that fit groups of individual points which are not necessarily connected in the image plane. For example, pixels belonging to different line segments of a dashed line are grouped together by the Hough transform (Lai & Kasturi 1991).

4.2 *Polygonalization*

Polygonal approximation is one common approach to obtaining features from curves. The objective is to approximate a given curve with connected straight lines such that the result is close to the original, but that the description is more succinct. The user can direct the degree of approximation by specifying some measure of maximum error from the original. In general, when the specified maximum error is smaller, a greater number of lines is required for the approximation. The effectiveness of one polygonalization method compared to another can be measured in the number of lines required to produce a comparable approximation, and also by the computation time required to obtain the result. The iterative endpoint fit algorithm (Ramer 1972) is a popular polygonalization method whose error measure is the distance between the original curve and the polygonal approximation. One potential drawback of polygonal approximation is that the result is not usually unique, that is, an approximation of the same curve that begins at a different point on the curve will perhaps yield different results. The lack of a unique result will present a problem if the results of polygonalization are to be used for matching. Besides polygonal approximation methods, there are higher order curve-and spline-fitting methods that achieve closer fits. These are computationally more expensive than most polygonalization methods, and can be more difficult to apply.

4.3 *Critical point detection*

The concept of "critical points" or "dominant points" derives from the observation that humans recognize shape in large part by curvature maxima in the shape outline. The objective in image analysis is to locate these critical points and represent the shape more succinctly by piecewise linear segments between critical points, or to perform shape recognition on the basis of the critical points. Critical point detection is especially important for graphics analysis of man-made drawings. Since corners and curves have intended locations, it is important that any analysis precisely locates these. It is the objective of critical point detection to do this – as opposed to polygonal approximation, where the objective is only a visually close approximation. One approach for critical point detection begins with curvature estimation. A popular family of methods for curvature estimation is called the k-curvatures approach (also the difference of slopes, or DOS, approach) (Freeman & Davies 1977; O'Gorman 1988). For these methods, curvature is measured as the angular difference between the slopes of two line segments fit to the data around each curve point. Curvature is measured for all points along a line, and plotted on a curvature plot. For straight portions of the curve, the curvature is low. For corners, there is a peak of high curvature that is proportional to the corner angle. For curves, there is a curvature plateau whose height is proportional to the sharpness of the curve (that is, the curvature is inversely proportional to the radius of curvature), and the length of the plateau is proportional to the length of the curve. To locate these features, the curvature plot is thresholded to find all curvature points above a chosen threshold value; these points correspond to features. The corner is then parameterized by its location, the curve by its radius of curvature and bounds, and the beginning and end transition points from straight lines around the curve into the curve. The user must choose one method parameter, the length of the line segments, to be fit along the data to determine the curvature. There is a tradeoff in the choice of this length. It should be as long as possible to smooth out effects due to noise, but not so long so as to also average out features. That is, the length should be chosen as the minimum arclength between critical points. The paper by (Wu & Wang 1993) is a good reference on critical point detection. The approach in this paper begins with polygonal approximation, then

refines the corners to coincide better with true corners. This paper also uses k-curvature in the preprocessing stage, so it provides a good description of this method and its use as well.

5. Text document analysis

There are two main types of analysis that are applied to text in documents. One is optical character recognition (OCR) to derive the meaning of the characters and words from their bit-mapped images, and the other is page-layout analysis to determine the formatting of the text, and from that to derive meaning associated with the positional and functional blocks (titles, subtitles, bodies of text, footnotes etc) in which the text is located. Depending on the arrangement of these text blocks, a page of text may be a title page of a paper, a table of contents of a journal, a business form, or the face of a mail piece. OCR and page layout analysis may be performed separately, or the results from one analysis may be used to aid or correct the other. OCR methods are usually distinguished as being applicable for either machine-printed or handwritten character recognition. Layout analysis techniques are applied to formatted, machine-printed pages, and a type of layout analysis, forms recognition, is applied to machine-printed or handwritten text occurring within delineated blocks on a printed form. In some cases it is necessary to correct the skew of the document which is typically a result of improper paper feeding into the scanner. Skew estimation and layout analysis are discussed briefly in this section. General approaches to OCR are presented in the next section.

5.1 *Skew estimation*

A text line is a group of characters, symbols, and words that are adjacent, relatively close to each other, and through which a straight line can be drawn (usually with horizontal or vertical orientation). The dominant orientation of the text lines in a document page determines the skew angle of that page. A document originally has zero skew, where horizontally or vertically printed text lines are parallel to the respective edges of the paper, however when a page is manually scanned or photocopied, non-zero skew may be introduced. Since such analysis steps as OCR and page layout analysis most often depend on an input page with zero skew, it is important to perform skew estimation and correction before these steps. Also, since a reader expects a page displayed on a computer screen to be upright in normal reading orientation, skew correction is normally done before displaying scanned pages. A popular method for skew detection employs the projection profile. A projection profile is a histogram of the number of ON pixel values accumulated along parallel sample lines taken through the document. The profile may be at any angle, but often it is taken horizontally along rows or vertically along columns, and these are called the horizontal and vertical projection profiles respectively. For a document whose text lines span horizontally, the horizontal projection profile has peaks whose widths are equal to the character height and valleys whose widths are equal to the between-line spacing. For multi-column documents, the vertical projection profile has a plateau for each column, separated by valleys for the between-column and margin spacing. The most straightforward use of the projection profile for skew detection is to compute it at a number of angles close to the expected orientation (Postl 1986). For each angle, a measure is made of the variation in the bin heights along the profile, and the one with the maximum variation gives the skew angle. At the correct skew angle, since scan lines are aligned to text lines, the projection profile has maximum height peaks for text and valleys for between-line spacing. Modifications and improvements can be made to this general technique

to iterate more quickly to the correct skew angle and to determine it more accurately. Several other methods for skew correction have also been proposed (Baird 1987; Akiyama & Hagita 1990; Srihari & Govindaraju 1989; Hashizume 1986).

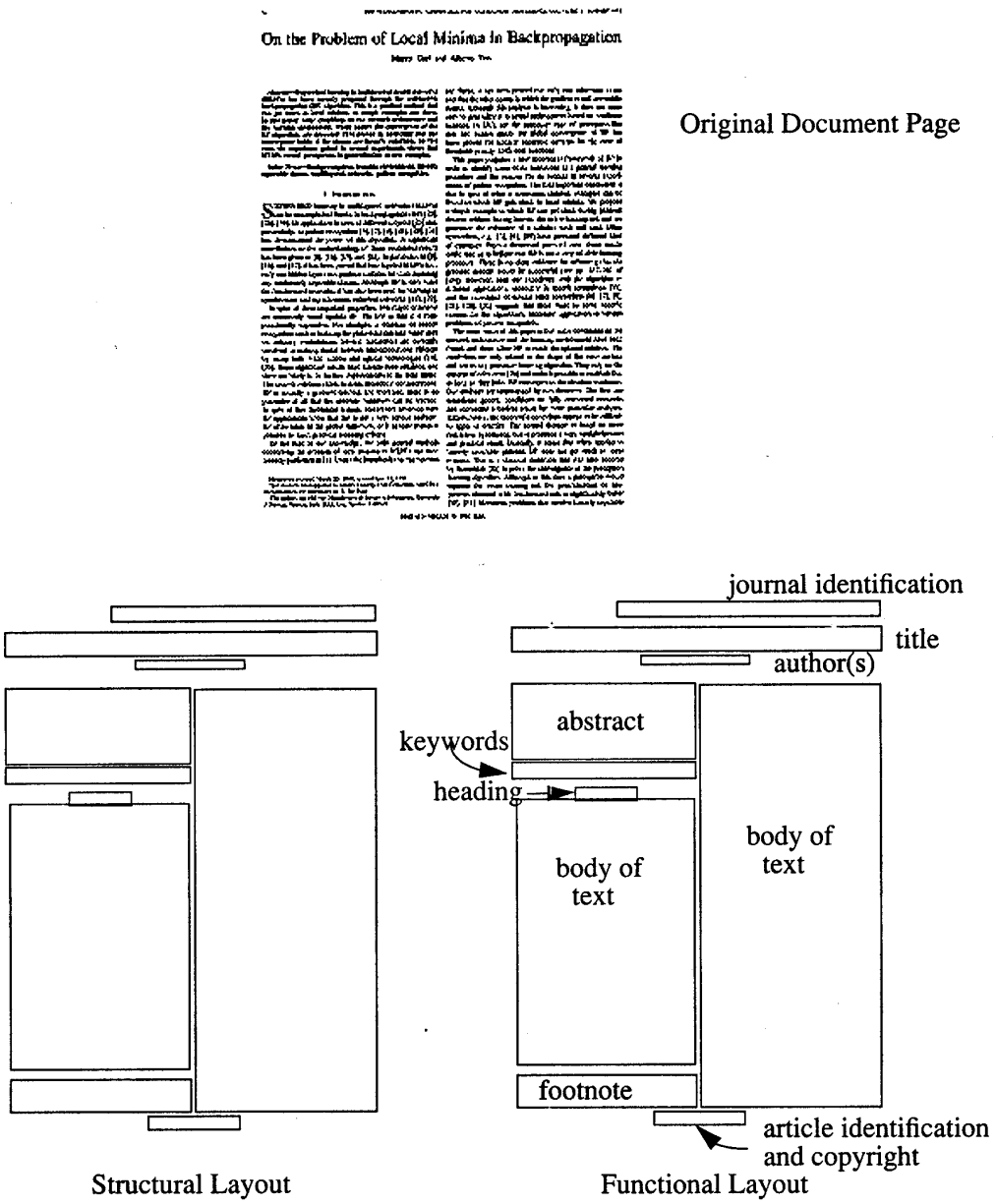

Figure 6. The original document page is shown with results from structural and functional layout analysis. The structural layout results show blocks that are segmented on the basis of spacing in the original. The labeling in the functional layout results is made with the knowledge of the formatting rules of the particular journal (*IEEE Trans. Pattern Anal. Machine Intell.)*. (Reproduced with permission from O'Gorman & Kasturi 1997.)

5.2 *Layout analysis*

After skew detection, the image is usually rotated to zero skew angle, and then layout analysis is performed. Structural layout analysis (also called physical and geometric layout analysis in the literature) is performed to obtain a physical segmentation of groups of document components. Depending on the document format, segmentation can be performed to isolate words, text lines, and structural blocks (groups of text lines such as separated paragraphs or table of contents entries). Functional layout analysis (Dengel *et al* 1992) (also called syntactic and logical layout analysis in the literature) uses domain-dependent information consisting of layout rules of a particular page to perform labeling of the structural blocks giving some indication of the function of the block. (This functional labeling may also entail splitting or merging of structural blocks.) An example of the result of functional labeling for the first page of a technical article would indicate the title, author block, abstract, keywords, paragraphs of the text body, etc. See figure 6 for an example of the results of structural analysis and functional labeling on a document image. Structural layout analysis can be performed in top-down (Pavlidis & Zhon 1991) or bottom-up (O'Gorman 1993) fashion. When it is done top-down, a page is segmented from large components to smaller sub-components, for example, the page may be split into one or more column blocks of text, then each column split into paragraph blocks, then each paragraph split into text lines, etc. For the bottom-up approach, connected components are merged into characters, then words, then text lines, etc. Alternately, top-down and bottom-up analyses may be combined.

6. Optical character recognition

Optical Character Recognition (OCR) lies at the core of the discipline of pattern recognition where the objective is to interpret a sequence of characters taken from an alphabet. Characters of the alphabet are usually rich in shape. In fact, the characters can be subject to many variations in terms of fonts and handwriting styles. Despite these variations, there is perhaps a basic abstraction of the shapes that identifies any of their instantiations. Developing computer algorithms to identify the characters of the alphabet is the principal task of OCR. The challenge to the research community is the following – while humans can recognize neatly handwritten characters with 100% accuracy, there is no OCR that can match that performance.

OCR difficulty can increase on several counts. Increase in fonts, size of the alphabet set, unconstrained handwriting, touching of adjacent characters, broken strokes due to poor binarization, noise etc. all contribute to the difficulty. figure 7 shows a sample of 0's and 6's that are easily confused by a handwritten digit recognizer. There are many applications that require the recognition of unconstrained handwriting. A word can be either purely numeric as in the case of a Zip code, or purely alphabetic as in the case of US state abbreviations or mixed as in the number of an apartment (e.g., 1A).

The task becomes particularly challenging when adjacent characters in a character string are touching as shown in figure 8. Unlike purely alphabetic strings where joining of the characters is natural and takes place by means of ligatures, the joining of numerals in a numeric word and the upper-case characters in an abbreviation are accidental. There are various ways in which two digits can touch. Some of the categories lend themselves to natural segmentation, whereas for some a holistic approach is the only option available.

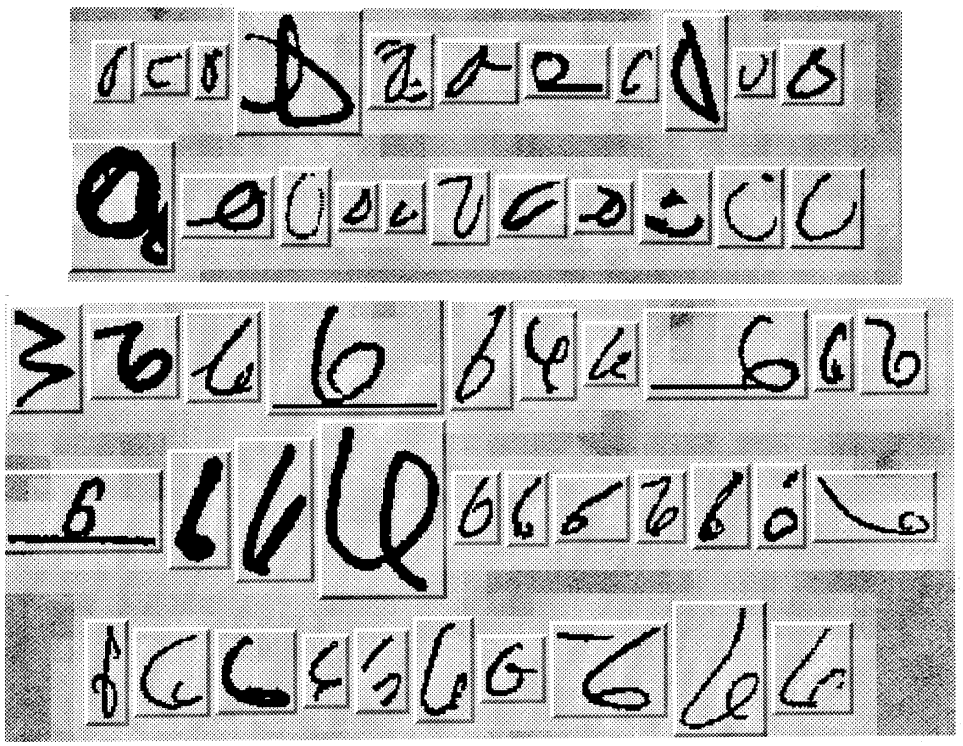

Figure 7. Handwritten digits are easily confused.

6.1 *Methodology*

OCR algorithms have two essential components: (i) Feature extraction, and (ii) classification. Ultimately, the OCR process assigns a character image to a class by using a classification algorithm based on the features extracted and the relationships among the features. Since members of a character class are equivalent or similar in as much as they share defining attributes, the measurement of similarity, either explicitly or implicitly, is central to any classifier. There is often a third component of *contextual processing* to correct OCR errors using contextual knowledge. We will describe the three components briefly.

6.1a *Feature extraction:* Feature extraction is concerned with recovering the defining attributes obscured by imperfect measurements. To represent a character class, either a prototype (an ideal form on which all member patterns are based, the class “essence") or a set of samples must be known. The feature selection process attempts to recover the pattern attributes characteristic of each class. Global features, such as the number of holes in

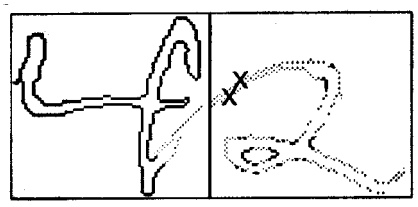

Figure 8. There is no simple splitting line that will segment the two touching digits without leaving artifacts on the separated digits.

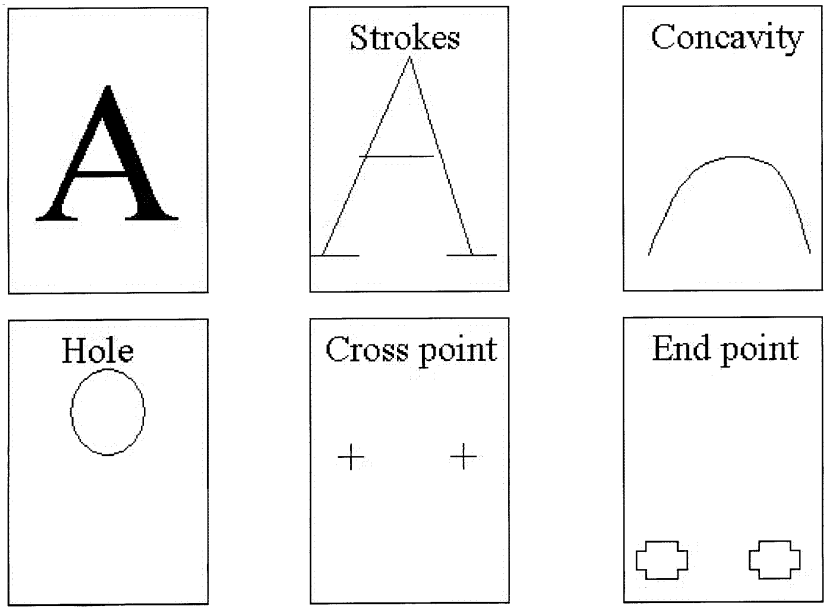

Figure 9. Characters are separated into categories by partioning the feature space.

the character, the number of concavities in its outer contour, and the relative protrusion of character extremities, and local features, such as the relative positions of line-endings, line crossovers, and corners are commonly used. The classification stage identifies each input character image by considering the detected features.

6.1b *Classification:* In the *statistical classification* approaches, character image patterns are represented by points in a multidimensional *feature space*. Each component of the feature space is a measurement or feature value, which is a random variable reflecting the inherent variability within and between classes. A classifier partitions the feature space into regions associated with each class, labeling an observed pattern according to the class region into which it falls.

An example of feature space partitioning used to classify a set of 50 characters into 5 different classes {C,E,T,X,Y} is illustrated in figure 9. The feature space is based on two features, the percentage of black pixels belonging to the vertical (S_V) and horizontal stroke (S_H). Strokes are classified as horizontal and vertical. Pixels can be part of one or both strokes, hence $S_V + S_H$ can be greater than 100%. Characters E and T are expected to have large values of S_V and S_H; character C is expected to have both S_V and S_H at about 50%; characters X and Y are expected to have small values of S_H. The decision functions for each class are measures of distance from the classes, and the dotted lines are equidistant from the classes they separate.

Template matching is a natural approach to classification and is one of the most commonly used. Individual pixels are directly used as features. A similarity measure instead of a distance measure is defined. One can count the number of agreements (black pixels in test pattern matching black pixels in template and white pixels in test pattern matching white pixels in template). The template class that has the maximum number of agreements can be chosen as the class of the test pattern. Such an approach is called

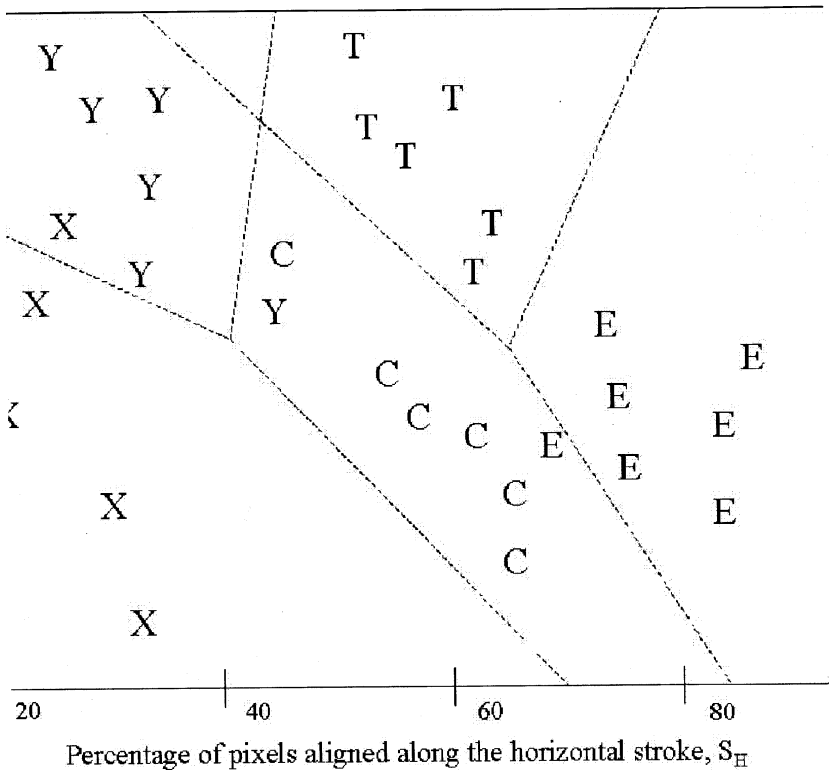

Figure 10. Structural features of *stroke, concavity, hole, cross points & end points* can be used as the dimensions of a feature space to classify characters; the locations of these features for 'A' are illustrated.

the maximum correlation approach. Alternatively, one could count the number of disagreements (black pixel in test pattern where it is white in template and vice versa). The class with the minimum number of disagreements can be chosen as the class of the test pattern. Such an approach is called the minimum error approach. The agreements and disagreements can be weighted in order to derive a suitable similarity measure. Template matching is effective when the variations within a class are due to "additive noise" only and test patterns are free of noise due to rotation, shearing, warping, or occlusion.

The K nearest neighbour (K-NN) rule is a well-known decision rule used extensively in pattern classification problems. The misclassification rate of the K-NN rule approaches the optimal Bayes error rate asymptotically as K increases (Fukunaga & Hostetler 1975]. The K-NN rule is particularly effective, when probability distributions of the feature variable are not known. Selection of templates is an important part of the nearest neighbour (1-NN) rule (Hart 1968). Templates are selected so that classifications obtained using any proper subset of the initial template set lead to gradual degradation in recognition accuracy.

Although many problems are successfully dealt with using the statistical approach, it is often more appropriate to represent patterns explicitly in terms of the structure or arrangement of components or primitive elements taken as the defining attribute of the pattern. The *structural* approach to OCR represents character pattern images in

terms of *primitives* and *relations* among primitives in order to describe pattern structure explicitly.

When asked to describe an alphanumeric character, people are most likely to use structural features (figure 10). For example, an upper-case 'A' has two straight lines (strokes) meeting with a sharp point (endpoint) at the top, and a third line crossing the two at approximately their midpoint (cross points), creating a gap in the upper part (hole). The basis of any structural technique is the representation of the pattern with a set of feature primitives that are able to describe all encountered patterns and discriminate between them.

6.1c *Contextual processing:* These techniques utilize knowledge at the word level to correct errors committed by OCR. These methods use information about other characters that have been recognized in a word as well as knowledge about the text in which the word occurs to carry out the task. Typically, the knowledge about the text takes the form of a lexicon (a list of words that occur in the text). For example, a character recognizer may not be able to reliably distinguish between an *u* and a *v* in the second position of *qXeen*. A contextual postprocessing technique would determine that *u* is correct since it is very unlikely that *qveen* would occur in the English language dictionary. Furthermore, one could use knowledge of the English language by which a *q* is almost always followed by an *u*.

There has been systematic study of OCR performance for English (Rice *et al* 1992). A complete review of OCR products for machine printed documents with benchmarks has been published by the University of Nevada (Nartker *et al* 1994). On good quality, machine-printed text data the correct recognition rate ranged from 99.13% to 99.77%. For poor quality machine-printed text data, the correct recognition rate ranged from 89.34% to 97.01%. The drop in recognition performance on poor quality data was primarily due to broken strokes in characters and touching of adjacent characters. More recently, OCR in the presence of graphics in complex documents has also been reported (Sawaki & Hagita 1998). Wilson *et al* (1996) have published a comprehensive report on the use and evaluation of OCR technology for form-based applications.

There is abundant literature describing methods for OCR. OCR is perhaps the most researched area of the pattern recognition discipline. While research in OCR for printed Roman script has reached a point of diminishing returns, OCR for handwriting and for non-Roman scripts continues to be a very active field. The reader is referred to the proceedings of conferences and workshops such as the International Conferences on Document Analysis and Recognition and the International Workshop on Frontiers in Handwriting Recognition.

7. Analysis of graphical documents

Graphics recognition and interpretation is an important topic in document image analysis since graphics elements pervade textual material, with diagrams illustrating concepts in the text, company logos heading business letters, and lines separating fields in tables and sections of text. The graphics components that we deal with are the binary-valued entities that occur along with text and pictures in documents. We also consider special application domains in which graphical components dominate the document; these include symbols in the forms of lines and regions on engineering diagrams, maps, business charts, fingerprints, musical scores etc. The objective is to obtain information to semantically describe the contents within images of document pages.

Document image analysis can be important when the original document is produced by computer as well. Anyone who has dealt with transport and conversion of computer files knows that compatibility can rarely be taken for granted. Because of the many different languages, proprietary systems, and changing versions of CAD and text formatting packages that are used, incompatibility is especially true in this area. Because the formatted document – that viewed by humans – is semantically the same independent of the language of production, this form is a "protocol-less protocol". If a document system can translate between different machine-drawn formats, the next objective is to translate from hand-drawn graphics. This is analogous to handwriting recognition and text recognition in OCR. When machines can analyse complex hand-drawn diagrams accurately and quickly, the graphics recognition problem will be solved, but there is still much opportunity for research before this goal will be reached.

A common sequence of steps taken for document image analysis of graphics interpretation is similar to that for text. Preprocessing, segmentation, and feature extraction methods such as those described in earlier sections are first applied. An initial segmentation step that is generally applied to a mixed text/graphics image is that of text and graphics separation. An algorithm specifically designed for separating text components in graphics regions irrespective of their orientation is described by Fletcher (1988). This is a Hough transform-based technique that uses the heuristic that text components are collinear. Once text is segmented, typical features extracted from a graphics image include straight lines, curves, and filled regions. After feature extraction, pattern recognition techniques are applied, both structural pattern recognition methods to determine the similarity of an extracted feature to a known feature using geometric and statistical means, and syntactic pattern recognition techniques to accomplish this same task using rules (a grammar) on context and sequence of features. After this mid-level processing, these features are assembled into entities with some meaning – or semantics – that is dependent upon the domain of the particular application. Techniques used for this include pattern matching, hypothesis and verification, and knowledge-based methods. The semantic interpretation of a graphics element may be different depending on domain; for instance a line may be a road on a map, or an electrical connection of a circuit diagram.

Most commercial OCR systems recognize long border and table lines as being different from characters, so no attempt to recognize them as characters is made. Graphics analysis systems for engineering drawings must discriminate between text and graphics (mainly lines). This is usually accomplished very well except for some confusion when characters adjoin lines, causing them to be interpreted as graphics; or when there are small, isolated graphics symbols that are interpreted as characters. Segmentation and analysis of colour-composite multi-layer maps, recognition of the three-dimensional object represented by its orthographic projections in a mechanical part drawing, and construction of a 3-D virtual walk-through from an architectural drawing are some examples of challenges presented to the graphics image analysis researchers. Clearly, much domain-dependent knowledge is applied in essentially all graphics analysis systems.

It is beyond the scope of this paper to describe the approaches for graphical document analysis in detail. In addition to many journal papers, the interested reader is referred to several workshops dedicated to this topic (GREC 19, 95,97,99) as well as papers in the International Conferences on Document Analysis and Recognition, and the International Conferences on Pattern Recognition.

8. Document analysis in a multilingual context

In this paper we have presented an overview of the methods applied to document images to extract meaningful semantic information from bit-mapped images of documents. Although we did not make any explicit statement, the methods described so far implicitly assumed that the documents contain a single language. In general, pixel-level processing and feature analysis methods are essentially independent of the particular language present in a document. However, particular features extracted from the image and the methods that are applied for discrimination among various characters are clearly language-dependent. When a document contains several languages within a page it is necessary to first segment it into regions containing a single language in each region.

8.1 *Multilingual document segmentation*

An interesting problem in multilingual document analysis is that of segmenting the document images into regions that contain a single language and identifying the language in each region. Spitz (1997) presented a method that performs such a classification in documents containing several Han-based scripts (Chinese, Japanese, and Korean) and 23 Latin-based languages such as English, German, and Vietnamese. His method made strong use of language-dependent features such as the distribution of optical density and the location of concavities in pixel data.

9. OCR for Indian languages

Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR 1999) has several papers dealing with OCR for Devanagari (such as Karnik 1999). OCR for Indian languages in general is more difficult than for European languages (Murthy 1998) because of the large number of vowels, consonants, and conjuncts (combination of vowels and consonants). Further, most scripts spread over several zones. Segmentation has to deal with the positioning of the conjuncts and half syllables. These factors coupled with the inflectional and agglutinative nature of Indian languages make the OCR task quite challenging. Language models and computational linguistics as it pertains to Indian languages is an area of recent research. Bharati *et al* (1998) describe machine translation systems and lexical resources for Indian languages including Sanskrit. The issues in parsing and understanding Sanskrit are discussed in detail by Ramanujam (1999).

10. Conclusions

We presented a brief summary of basic building blocks that comprise a document analysis system. We encourage the reader to refer to cited papers for more detailed descriptions. We hope that this introduction will help the reader by providing the background necessary to understand the contents of the papers that follow in this special issue.

References

Arcelli C, Sanniti di Baja G 1985 A width-independent fast thinning algorithm. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-7: 463–474

Arcelli C, Sanniti di Baja G 1993 Euclidean skeleton via center-of-maximal-disc extraction. *Image Vision Comput.* 11: 163–173

Akiyama T, Hagita N 1990 Automated entry system for printed documents. *Pattern Recogn.* 23: 1141–1154

Baird H S 1987 The skew angle of printed documents. *Proceedings of the Conference of the Society of Photographic Scientists and Engineers on Hybrid Imaging Systems* (Springfield, VA: Soc. Photogr. Sci. Eng.) pp 14-21

Bharati A, Chaitanya V, Sangal R 1998 Computational linguistics in India: An overview. Technical Report, Indian Institute of Information Technologies, Hyderabad

Dengel A, Bleisinger R, Hoch R, Fein F, Hones F 1992 From paper to office document standard representation. *IEEE Comput.* 25: 63–67

Fletcher A, Kasturi R 1988 A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-10: 910–918

Freeman H 1974 Computer processing of line drawing images. *Comput. Surv.* 6: 57–98

Freeman H, Davis L 1977 A corner-finding algorithm for chain-coded curves. *IEEE Trans. Comput.* C-26: 297–303

Fukunaga K, Hostetler L D 1975 K-nearest-neighbour Bayes-risk estimation. *IEEE Trans. Inf. Theor.* 21: 285-293

Garris M D, Dimmick D L 1996 Form design for high accuracy optical character recognition. *IEEE Trans. Pattern Anal. Machine Intel.* PAMI-18: 653–656

GREC 1995, 97, 99 Selected papers from the International Workshops on Graphics Recognition 1995, 1997, and 1999. *Lecture Notes in Computer Science* series (Springer Verlag) vols. 1072 (1996), 1389 (1998), 1941 (2000)

Haralick R M, Shapiro L G 1992 *Computer and robot vision* (Reading, MA: Addison-Wesley)

Haralick R M, Sternberg S R, Zhuang X 1987 Image analysis using mathematical morphology. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-9: 532–550

Hashizume A, Yeh P S, Rosenfeld A 1986 A method of detecting the orientation of aligned components. *Pattern Recogn. Lett.* 4: 125–132

Hart P E 1968 The condensed nearest neighbour rule. *IEEE Trans. Inf. Theor.* 14: 515–516

ICDAR 1999 *5th Int. Conf. on Document Analysis and Recognition* (Los Alamitos, CA: IEEE Comput. Soc.)

Illingworth J, Kittler J 1988 A survey of the Hough transform. *Comput. Graphics Image Process.* 44: 87–116

Karnik R P 1999 Identifying Devnagari characters. *Proc. Int. Conf. on Document Analysis and Recognition* (Los Alamitos, CA: IEEE Comput. Soc.) pp. 669–672

Jain A K, Bhattacharjee S K 1992 Text segmentation using Gabor filters for automatic document processing. *Machine Vision Appl. J.* 5: 169–184

Lai C P, Kasturi R 1991 Detection of dashed lines in engineering drawings and maps. *Proc. First Int. Conf. on Document Analysis and Recognition*, St. Malo, France, pp. 507–515

Lam L, Lee S-W, Suen C Y 1992 Thinning methodologies - A comprehensive survey. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-14: 869–885

Lam L, Suen C Y 1995 An evaluation of parallel thinning algorithms for character recognition. *IEEE Trans. Pattern Recogn. Machine Intell.* 17: 914–919

Medioni G, Yasumoto Y 1987 Corner detection and curve representation using cubic B-splines. *Comput. Vision, Graphics, Image Process.* 29: 267–278

Murthy B K, Deshpande W R 1998 Optical character recognition (OCR) for Indian languages. *Proc. Int. Conf. on Comput. Vision, Graphics, Vision, Image Process.* ICVGIP, New Delhi

Nartker T A, Rice S V, Kanai J 1994 OCR Accuracy. UNLV's Second Annual Test. Technical Journal INFORM, University of Nevada, Las Vegas

O'Gorman L 1988 Curvilinear feature detection from curvature estimation. *9th Int. Conference on Pattern Recognition*, Rome, Italy, pp 1116–1119

O'Gorman L 1990 k x k Thinning. *Comput. Vision, Graphics, Image Process.* 51: 195–215

O'Gorman L 1992 Image and document processing techniques for the right pages electronic library system. *Int. Conf. Pattern Recognition (ICPR)*, The Netherlands, pp 260–263

O'Gorman L 1993 The document spectrum for structural page layout analysis. *IEEE Trans. Pattern Anal. Machine Intelli.* PAMI-15: 1162–73

O'Gorman L 1994 Binarization and multi-thresholding of document images using connectivity. *CVGIP: Graphical Models Image Process.* 56: 494–506

O'Gorman L, Kasturi R 1997 Document image analysis. *IEEE Computer Society Press Executive Briefing Series*, Los Alamitos, CA

Pavlidis T 1982 *Algorithms for graphics and image processing* (Rockville, MD: Comput. Sci. Press)

Pavlidis T, Zhou J 1991 Page segmentation by white streams. *Proc. 1st Int. Conf. on Document Analysis and Recognition (ICDAR)*, St. Malo, France, pp 945–953

Postl W 1986 Detection of linear oblique structures and skew scan in digitized documents. *Proc. 8th Int. Conf. on Pattern Recognition (ICPR)*, Paris, France, pp 687–689

Ramanujan P 1999 Development of a general-purpose Sanskrit parser, M Sc thesis, Dept. of Computer Science & Automation, Indian Institute of Science, Bangalore

Ramer U E 1972 An iterative procedure for the polygonal approximation of plane curves *Comput. Graphics Image Process.* 1: 244–256

Reddi S S, Rudin S F, Keshavan H R 1984 An optimal multiple threshold scheme for image segmentation. *IEEE Trans. Syst. Man Cybern.* SMC-14: 661–665

Rice S V, Kanai J, Nartker T A 1992 A report on the accuracy of OCR devices. Technical Report, Information Science Research Institute of Nevada, Las Vegas

Sawaki M, Hagita K 1998 Text-line extraction and character recognition of document headlines with graphical design using complimentary similarity measure. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-20: 1103–1109

Sahoo P K, Soltani S, Wong A K C, Chen Y C 1988 A survey of thresholding techniques. *Comput. Vision, Graphics, Image Process.* 41: 233–260

Sanniti di Baja G 1994 Well-shaped, stable and reversible skeletons from the (3,4)-distance transform. *Visual Commun. Image Representation* 5: 107–115

Serra J 1982 *Image analysis and mathematical morphology* (London: Academic Press)

Shih C-C, Kasturi R 1988 Generation of a line-description file for graphics recognition. *Proc. SPIE Conf. on Applications of Artificial Intelligence* 937: 568–575

Spitz L 1997 Determination of the Script and Language Content of Document Images. *IEEE Trans. Pattern Analy. Machine Intell.* PAMI-19: 235–245

Srihari S N, Govindaraju V 1989 Analysis of textual images using the Hough Transform. *Machine Vision Appl.* 2: 141–153

Trier O D, Taxt T 1995 Evaluation of binarization methods for document images *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-17: 312–315

Tsai W-H 1985 Moment-preserving thresholding: A new approach. *Comput. Vision, Grapics, Image Process.* 29: 377-393

Wilson C L, Geist J, Garris M D, Chellapa R 1996 Design, integration, and evaluation of form-based handprint and OCR systems. Technical Report, NISTIR5932, National Institute of Standards & Technology, US; download from http://www.itl.nist.gov/iad/894.03/pubs.html

Wong K Y, Casey R G, Wahl F M 1982 Document analysis system. *IBM J. Res. Dev.* 6: 647–656

Wu W-Y, Wang M-J J 1993 Detecting the dominant points by the curvature-based polygonal approximation. *CVGIP: Graphical Models Image Process.* 55: 79–88