MVA'90 IAPR Workshop on Machine Vision Applications Nov. 28-30,1990, Tokyo

Techniques for Line Drawing Interpretation: An Overview

Rangachar Kasturi and Senthil Siva

Department of Electrical and Computer Engineering
The Pennsylvania State University
University Park, Pennsylvania, 16802, USA

Lawrence O'Gorman

AT&T Bell Laboratories, Murray Hill, New Jersey, 07974, USA

ABSTRACT

An overview is presented of algorithms and techniques for document image analysis with an emphasis on those for graphics recognition and interpretation. The techniques are derived from the fields of image processing, pattern recognition, and machine vision. The objective in document image analysis is to recognize page contents including layout, text, and figures. Although optical character recognition (OCR) falls within the context of document image analysis, we do not cover this area, since OCR techniques have been covered extensively in the literature. We also limit the focus to images containing binary information. Topics covered are segmentation of document image into text and graphics regions, vectorization to obtain lines, identification of graphical primitives, and generation of succinct image interpretations.

INTRODUCTION

Document image analysis addresses the problem of creating a higher level description of the contents of paper-based documents. In particular, efficient algorithms for intelligent interpretation and succinct description of mixed text and graphics documents have many immediate applications. For example, they permit conversion of paper-based engineering drawings into a CAD compatible form; provide a mechanism for efficiently archiving documents such as journals and newsmagazines; and facilitate integration of vector-oriented graphics databases and raster-oriented images in applications such as geographic information systems.

In a typical document image analysis system, the input is a raster image obtained from optically scanning a document page. Such documents as technical manuals, for example, may contain text, half-tone images, as well as graphical components such as block diagrams, circuit diagrams, illustration of mechanical parts, and flow charts. The objective is then to segment such an image into its meaningful text, image, and graphics regions, and further analyze each of these to generate a succinct description. Ideally, such a description should include the page structure and format information such as the size and location of various blocks and columns of text, and the size and font style of characters. It should contain adequate information to obtain a fairly accurate reconstruction of the original document. In addition, it must be at a level of description that facilitates document modification and information interchange between various systems. It is in this respect that document analysis systems differ from conventional data compression systems. An intelligent interpretation system may also combine data in the document with the knowledge about the domain to create an even higher level description. This enables graphics components to be accessed as semantically complete entities for indexing or modification. An example of how domain-specific rules can be used to obtain a semantic description from a line drawing is the following. In engineering drafting, there are rules and conventions for interpreting 3-dimensional mechanical parts from drawings of different views. A combination of machine vision and use of these rules can be employed to determine the part type from orthographic views of the mechanical drawings of the part. With this ability, computer-aided design information can be combined with results from machine vision to perform computer-aided manufacturing.

A common sequence of steps taken for document image analysis for graphics interpretation is listed here:

Data capture and preprocessing,
Region segmentation,
Vectorization,
Feature extraction, and
Graphics recognition and interpretation.

Data capture and preprocessing includes operations such as scanning, noise filtering, and thresholding for converting a paper-based document to a reasonably noise-free binary image. Scanning has been discussed in [Ejiri et al. 90, Casey and Wong 90, Kasturi et al. 90-2]. For an extensive treatment of thresholding techniques, the paper [Sahoo et al. 88] contains a recent survey. We will not discuss data capture and preprocessing in this paper but concentrate on the other steps listed above. A typical image of a document contains regions of text, graphics, and half-tone images. It is then necessary to segment a mixed image into separate regions to facilitate application of appropriate algorithms to each region type. For example, text regions can be used as input to an OCR system; graphics regions to a graphics interpretation system, and image regions to a data compression system or an image interpretation system as appropriate. These region segmentation algorithms are described in Section 1. Binary raster graphics images contain solid objects (completely filled regions) and objects made up of thin lines. While thin lines are adequately represented by their corelines, solid objects are best represented by their contours or boundaries. These vectorization and line-following operations are performed to facilitate recognition and these are discussed in Section 2. Since straight lines and curves are basic components in a graphics diagram, detection of feature points such as corners and points of transitions from straight lines to curves is an important step in graphics recognition. Locating such points is made difficult by the artifacts introduced during digitization and preprocessing. Algorithms for feature extraction are discussed in Section 3. The extracted features and the associated line and curve description files are then processed by a semantic interpretation system that includes task-dependent graphics recognition and interpretation algorithms. Some of these techniques are discussed in Section 4. Finally, a brief summary is presented in Section 5.

1. REGION SEGMENTATION

A bit mapped binary image is typically obtained by scanning a paper-based document at an appropriate sampling resolution. After thresholding and preprocessing, the document image typically consists of regions of text, graphics, and halftone images. Since different techniques are applied to process each of these, they are usually first segmented into these different regions. [Srihari and Zack 86, Casey and Wong 90] classify block segmentation algorithms into two categories: Top-down, or knowledge based methods, that work with a knowledge of the nature of the document; and bottom-up, or data driven methods, that continuously refine data by layered grouping operations. However, many of the algorithms use a combination of these two methods.

1.1. Document Structure analysis

Since most documents have rectangular block structures (rectangular regions of text and figures) separated by white regions of different widths, they can be represented as a tree of nested rectangular blocks. A projection profile is often used to aid this segmentation [Masuda et al. 85, Zen and Ozawa 85]. This is a histogram containing the sum of black pixels along all rows or all columns in the image. For instance, the projection profile along the horizontal axis of a two-column page will have two peaks separated by a valley corresponding to the white space between columns.

In most of the block segmentation techniques, correction for skew is necessary before segmentation. This is usually accomplished by determining the projection profiles at different angles. Note that when the orientation is very close to the true orientation of the page, large fluctuations in the projection profile are observed. This phenomenon is used to estimate the skew angle of the image in [Srihari and Govindaraju 89]. [Baird 90] describes an efficient algorithm in which the skew angle is determined accurately by first computing the projection profile at coarse intervals, then recalculating them at closer intervals near the correct orientation. [Okamoto and Miyazawa 90] describe a method where skew detection is done in blocks. This method is based on their segmented block (SB) coding scheme in which a contiguous black pixel area is represented by its position and a simple shape. Enclosed rectangles are used to estimate the base lines of character strings in text areas and the skew correction is done by rotating each block accordingly. [Hinds et al. 90] use Hough transform for detecting document skew. A grey scale "burst image" is created from the black run lengths that are perpendicular to the text lines by placing the length of the run in the run's bottom most pixel. This data reduction procedure is reported to reduce the processing time of the Hough transform by a factor of as much a 7.4 for documents with grey scale images.

Besides locating blocks, the type of block is important for future processing. The run-length smearing method [Wong et al. 82] broadly segments an image into regions of text, graphics, and images using a statistical feature classification technique. First the image is low-pass filtered, or "smeared". Then features are calculated that measure characteristic "texture" of each region type [Casey and Wong 90]. These features include height, eccentricity, black pixel density, and mean length of pixel runs. The smearing method fails when there is too much skew of the document, or when the pattern of characters or graphics in the image has not been anticipated in the segmentation rules. A system for segmentation which does not use document dependant parameters is described in [Okamoto and Miyazawa 90]. Their method is based on the SB coding scheme described previously. Their algorithm avoids time consuming pixel by pixel processing like a projection profile and uses the rectangles which are generated by operations on the blocks of SB code.

Another approach for block segmentation is the top down method which uses document layout knowledge. [Fujisawa and Nakano 90] describe a top down method for structured document analysis where document layout knowledge is codified by a representation language. The generic layout structure of a document class is first coded. These descriptions are then matched to input documents. By this method page classes of input images can be identified, titles and authors names can be extracted and recognized using a character recognition system and then coded. The authors report that the system recognized 104 out of 106 samples of a certain patent application document. [Viswanathan 90] presents a similar system where a block grammar is used to hierarchically describe the spatial structure of a document image.

1.2. Text-string Segmentation

Although the two methods described above are capable of broadly segmenting an image into text and non-text regions, they are unable to segment text strings that are enclosed in graphics (e.g., tables, maps, engineering drawings). [Fletcher and Kasturi 88] describe a Hough transform-based algorithm for segmenting text-strings in a mixed text/graphics image. Since the Hough transform method is invariant to document skew, text strings in any orientation are detected. The algorithm determines the thresholds adaptively, thus making it invariant to changes in size and font style of text strings. This algorithm has been enhanced to handle text strings that are connected to graphics [Kasturi et al. 90-2]. As an illustration, the images before and after applying the algorithm are shown in Figs. 1 and 2. The Hough transform has also been used by [Srihari and Govindaraju 89] for analyzing textual images. Their system detects text skew angle, determines the signature of a text line, segments text into lines, and determines whether the text is right-side-up.

A measure, called the neighborhood line density, obtained by computing the "complexity" of an image at every black pixel has been used by [Kida et al. 86] to separate characters from graphics. [Meynieux et al.86] describe clustering methods for text block segmentation of long text strings, using a method based on grouping nearby connected components. A rule-based method that merges connected components to characters, then to words, strings, paragraphs and columns, is described in [Nagy et al. 85]. [Dori 89] describes a syntactic/geometric approach for recognition of interpretation lines (e.g. dimensioning lines describing measurements of the object) and associated components in engineering drawings.

A hypothesis-driven analysis of text regions is used to extract characters and text lines from a column of text in [Kanai 90]. The TeX imaging model is used to represent document images and their text line extraction technique is almost an inverse of the TeX typesetting process. [Fisher et al 90] describe a rule based system for segmentation of document images into text and non-text regions. No *a priori* knowledge of the document structure is assumed. The segmentation rules are adaptive and most parameters are dynamically determined for each document.

After region segmentation, a document can be separated into different images, each containing only one class of objects such as text, graphics, or halftone images. Each of these images requires different techniques for further analysis. In the remainder of this paper we consider techniques for processing graphics images such as line drawings, circuit diagrams, flow charts, maps, and engineering drawings.

2. EXTRACTION AND DESCRIPTION OF LINES AND SOLID REGIONS

The amount and type of processing applied to graphics data in a raster image is usually application dependent. If the graphics image is a part of a predominantly textual document and the objective is simply data archiving for later reconstruction, then use of any one of the well known data compression techniques [Jain 89] is adequate. On the other hand, if information is to be extracted from data for such applications as indexing image components from a pictorial database, modification of graphics in a CAD system [Karima et al. 85], or determining locations in a geographical information system, then extensive processing to extract objects and their spatial relationships is necessary. Such a level of description is attained through a series of intermediate steps. At the lowest level, graphics data is processed to locate line segments and solid symbols. Extracting the location and attributes of these is an important step for graphics interpretation. This process is discussed in detail in this section.

2.1. Solid Region Segmentation

A typical graphics image contains lines and symbols. While most of the graphics symbols are formed by thin lines, some of the symbols such as arrow-heads and diode symbols are filled regions. Such symbols can be represented by their boundaries, while thin lines are used for the remaining entities.

To represent an image by thin lines and boundaries, the lines and solid regions must first be differentiated. One approach to determine solid regions is by erosion and dilation operations [Harada et al. 85, Kasturi et al. 90, Nagasamy and Langrana 90]. In this approach, the image is eroded by a predetermined number of pixels to completely remove all thin entities such as lines. The eroded image is dilated and a logical AND operation is performed to recover solid symbols whose boundaries are then found, and used in the symbol recognition phase. The difference image, obtained by subtracting solid symbols from original image, contains only thin entities, and is processed by vectorization methods described below.

There are also methods that simultaneously extract boundaries and core lines. In [Shih and Kasturi 89] several picture decomposition algorithms for locating solid symbols are evaluated. Wakayama's Maximal Square Moving (MSM) algorithm [Wakayama 82], in which black pixel regions are represented as a union of largest possible overlapping squares, was adapted by [Shih and Kasturi 89] for simultaneously locating the corelines of thin entities and boundaries of solid regions. However, the algorithm is quite complex and requires numerous data structures and pointers to obtain the desired description. Another method in which core lines are found simultaneously with boundaries is the kxk thinning method [O'Gorman 90-1]. This method is described in the next section.

2.2. Thinning

For a line whose information can be represented by its path and perhaps width and color parameters, thinning is often performed to facilitate subsequent processing. Thinned images still convey most of the information by way of shape and topology. Because topological features are more easily found by traversing lines that are a single pixel wide, thinning is a very popular preprocessing step.

A common thinning approach is to "peel" the region boundaries until the regions have been reduced to thin lines This process is performed iteratively -- on each iteration every image pixel is inspected within 3x3 windows, and single-pixel wide boundaries that are not required to maintain connectivity or endlines are erased. In [O'Gorman 90-1], the kxk thinning method is proposed as a generalization of the 3x3 method to size kxk windows. Instead of erasing only one pixel on each iteration, k-2 x k-2 pixels can be erased, and by doing this, fewer iterations are often required. This method can also be performed in parallel, and can be configured to obtain boundaries of filled regions, widths of lines, labels of lines (for instance for colored lines), as well as skeletons of thin lines.

Methods have been proposed to thin with a fixed number of steps not dependent on the maximum line thickness [Arcelli and di Baja 85, Sinha 87]. For these non-iterative methods, skeletal points are estimated from distance measurements with respect to opposite boundary points of the regions. In general, these non-iterative methods are less regularly repetitive, not limited to local operations, and less able to be pipelined compared to the iterative methods; and this makes their implementation in special-purpose hardware less appropriate. All these algorithms can be performed in sequential, raster-scan order. Some of the iterative methods can be done in parallel, where operation at any pixel does not depend on results at any other pixel on a single iteration. Previous papers are well reviewed in recent literature. For a good background on thinning techniques, see [Pavlidis 82]. For comparisons of the iterative methods, see [Tamura 78, Naccache and Shinghal 84], and for the parallel techniques, see [Guo and Hall 89].

A thinning method for line drawings, based on cell structures is described by [Suzuki and Mori 90]. An unit is set to be a cell instead of a pixel and branch points are described accurately. The output is a piecewise linear curve expressed as a set of vertices and edges. [Jang and Chin 90] present a precise definition of digital skeletons and a mathematical framework for the analysis of a class of thinning algorithms based on morphological set transformation. A necessary and sufficient condition for the thinning process in general is also derived.

2.3. Chain Codes and Vectorization

Unwanted in the thinned image are isolated lines and spurs off longer lines that are artifacts due to the thinning process or noise in the image. Many thinning methods, e.g. [Arcelli and di Baja 85], require that the binary image is noise filtered before thinning because noise severely degrades the effectiveness and efficiency of this processing. However noise can never be totally removed, and it is often difficult to distinguish noise from the signal in earlier stages. In [Jagadish and O'Gorman 89], image lines (between endpoints and junctions) are found after thinning, and descriptive parameters (length, type, location) are associated with them. This descriptive and contextual information is then used to remove the line artifacts.

After thinning, the image can be processed to extract connected chains of pixels. The popular Freeman chain code which represents a chain as a sequence of direction codes from one pixel to the adjacent one, does not give any provision for maintaining branching line structures. This is fine for compression, but for image analysis it is important to retain the complete line structure with all its branches and to know the topology at each junction. In [Harris et al. 82], the skeleton is coded with pixel values of 0, 1, 2, and 3 for background pixels, terminal (end of line) pixels, intermediate pixels, and junction pixels respectively. This, combined with chaining allows line segments and their interconnections to be easily determined. Another method that operates on the thinned image to preserve line features is described in [O'Gorman 90-2]. This Primitives Chain Code (PCC) is an extension of Freeman chain code. In addition to retaining connectivity information, PCC also preserves branching and junction topology information. With these additional features, subsequent pattern recognition steps are facilitated, and this code usually results in higher compression than the Freeman technique.

A chain code retains all the information in the thin line image and enables exact decoding of the curves. If exact decoding is not required, then the image can be approximated by fitting line segments to the curves. This vectorization allows the image to be stored and accessed more efficiently. For example, [Ramachandran 80] describes a vector extraction and coding scheme intended to code vector representations of engineering drawings. This fits piece-wise linear segments to outlines of black pixel regions. Compression ratios of up to 35:1 have been reported for this. [Bley 84] describes a picture decomposition algorithm for segmenting electrical schematics. In this algorithm black pixel regions are approximated by rectangular blocks that are then analyzed to segment lines and symbols. The maximal square moving algorithm [Wakayama 82] has also been evaluated by [Shih and Kasturi 89] for its suitability in generating vector descriptions of lines.

Another approach implements vectorization using specialized hardware. Fastrak [Fulford 81] is an interactive line following digitizer that depends upon human interaction for tracking noisy data. [Ejiri et al. 90] describes two automatic digitizers, one for monochrome and the other for color documents. Longest possible straight lines are fit to pixels of the same color without applying any line thinning operations. This method eliminates the problems of twigs and other artifacts that are generated by thinning algorithms and is efficient if the drawing contains mostly long straight lines.

[Watson et al. 84] describe a method for extracting lines from gray-level images of line drawings. The algorithm tracks the ridges of the gray-scale intensity surfaces, which are candidates for boundary regions between objects (lines) and background. [Haralick et al. 83] describe a method for topographical labeling of gray scale image characteristics, such as peaks, ridges, valleys, etc., which can be applied to line drawings to extract straight lines and curves. A vectorization method based on generic object models is proposed in [Hori and Okazaki 90]. Here a model is used to describe the generic properties of an object. The approximate line figure which is first defined in the pre-vectorization process is modified so as to meet the properties described in the object model. This eliminates distortions yielding better perceptual quality.

Variations of these vectorization techniques as well as other approaches have been described in [Parker 88, Woetzel 78, Pavlidis and Cherry 82, Landy and Cohen 85, Pavlidis 84, Merelli et al. 85, Jimenez and Navalon 82].

2.4. Dashed Line Detection

After line segments are found, it is necessary to detect broken lines such as dashed lines by linking nearby segments that satisfy certain constraints. An algorithm for detecting such lines is described in [Kasturi et al. 90] where, a dashed line may be made up of segments of equal lengths or may consist of alternating long and short segments. [Ejiri et al. 90] describe a two-stage algorithm for detecting dashed lines. Line segments are grouped together using local constraints in the first stage. Context dependent global syntax rules are then applied in the second stage to resolve conflicts. The Hough transform is also useful to group together broken straight line segments. The segments grouped together should then be examined for consistency in inter-segment gaps.

Once the lines and object boundaries are located, feature points such as corners, inflection points, and junctions can be detected. Techniques for obtaining these are discussed in the next section.

3. FEATURE EXTRACTION

An important step in the recognition of objects for computer vision tasks involves localization of feature points. On edge, contour, or thinned images, these features include high curvature points as well as inflection points. From the methods in the previous section, chains of pixels are found for curves and boundaries. In this section, methods are given to find features on these chains, from which line segmentation can be performed. In graphics recognition applications, these features are used as critical points along a curve for performing piece-wise linear approximation and curve fitting. This results in a more concise description of the chain, and this description is also "higher level", that is it is closer to our goal of recognized graphics.

3.1. Polygonalization

Polygonal approximation is one common approach to analyzing curves. In this process, connected line segments are fit to the image contour or thin lines. The fit is made to reduce a chosen error criterion between the approximation and the original curve. Note that this is subtly different from the vectorization method described before. Vectorization is in lieu of thinning, is often combined with scanning, and has as its purpose data compression. Polygonalization is performed on the thinned or vectorized image, and is more concerned with extracting correct features for subsequent recognition, rather than simply compressing.

In the iterative endpoint fit algorithm [Ramer 72, Duda and Hart 73] straight lines are iteratively fitted to the data. Iterations are performed based on the perpendicular distances from the segments to each point on the curve. In another class of methods [Williams 78, Sklansky and Gonzalez 80, Williams 81, Pavlidis 82], a straight line fit is constrained to pass within a radius around each data point. The line segment is grown from the first point, and when further extension of the line segment causes it to fall outside the radius of a point, a new line is started. In another method [Kurozumi and Davis 82], a minimax approach is employed. The line segment approximations are chosen to minimize the maximum distance between the data points and the approximating line segment. For this method and for some of the other methods, segment endpoints must be adjusted to be within a gap tolerance of connecting segment endpoints. In another class of techniques, area versus distance is used as a measure of goodness of fit. These use a scan-along technique where, if the area deviation for each line segment exceeds a preset value, then a new segment is generated. [Wall 86] uses this polygonal approximation for generating a smooth cubic curve as an approximation to the original data. Images of the coastline of Great Britain and its polygonal approximations are shown in Fig. 3.

[Leu and Chen 88] focus on uniqueness and accuracy of representation, two important issues of polygonal approximation. Uniqueness is achieved by starting the approximation simultaneously at places along the shape boundary where the arcs are closer to straight lines than their neighboring arcs. In [Imai and Iri 86], algorithms are discussed to perform piece-wise linear approximation to a curve with minimum cardinality, so as to cover a sequence of points by a minimum number of rectangles. [Sirjani and Cross 88] describe a two-pass mark and sweep algorithm for polygonal approximation. In their method, noisy curves can cause problems in the first pass, and selection of an appropriate threshold in the second pass is critical to obtaining good results.

Besides polygonal approximation methods, higher order curve and spline fitting methods are used where more precise approximations are required. These are more computationally expensive than most polygonalization methods, and can be more difficult to apply. Some of these are described in [Pavlidis 77, Pavlidis 78, Pavlidis 82]. A concise description of curves in terms of the quasi-topological structure and the structures of singular points (branch points) is proposed by [Nishida and Mori 90]. Four types of curve primitives are used and binary operators on the primitives are defined. Curves are described hierarchically in terms of these primitive sequences. [Reihani and Thompson 90] describe a two part feature extraction system that combines a Hough technique for isolating line regions with a geometrical technique for isolating arcs of different curvatures as well as identifying and deriving features corresponding to the isolated regions.

One of the drawbacks of most of the polygonal fit techniques is that the operations are not performed symmetrically with respect to curve features such as corners and the centers of curves. The result is that the computed breakpoints between segments may be at different locations depending on starting and ending locations, or direction of operation of the line fit. Extensions can be made to some of the methods to produce fewer and more consistent breakpoint locations, but these procedures are usually iterative and can be computationally expensive. In the next section, curve fitting methods are performed from a different approach, that of detecting features first, then performing fits. It will be seen that these methods often produce better approximations than polygonalization.

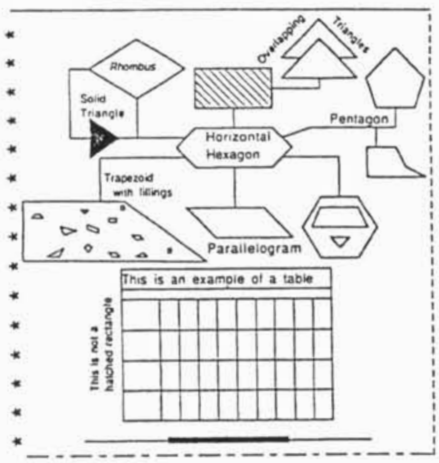

Figure 1. A typical test image.

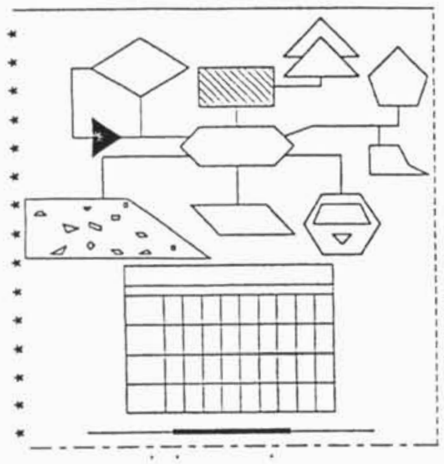

Figure 2. Fig. 1, after text string separation.

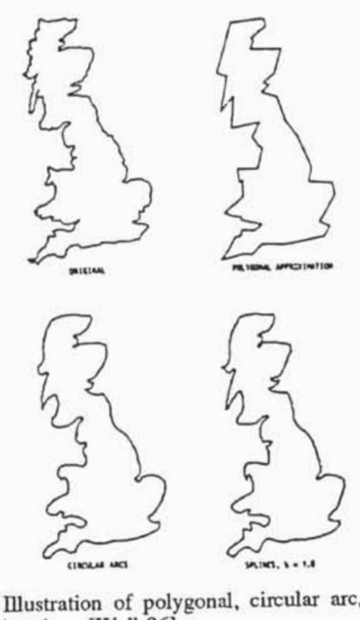

Figure 3. Illustration of polygonal, circular arc, and spline approximations [Wall 86].

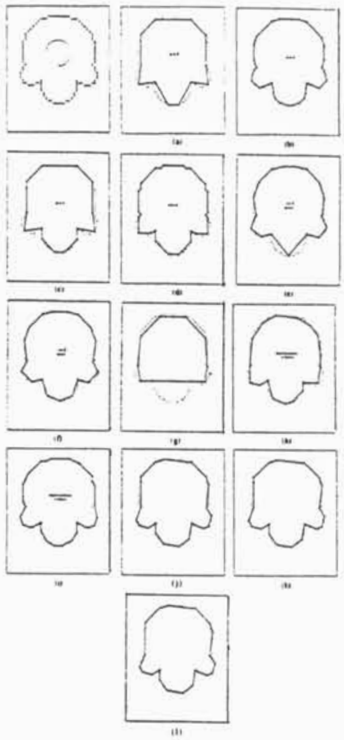

Figure 4. Comparison of various dominant point detection algorithms on a curve with four semicircles [Teh and Chin 89]. Rosenfeld-Johnston algorithm with (a) m=9 and (b) m=4. Rosenfeld-Weszka algorithm with (c) m=9 and (d) m=4. Freeman-Davis algorithm with (e) s=3, m=1 and (f) s=2, m=1. (g) Sankar-Sharma algorithm. Anderson-Bezdek algorithm with (h) m=7 and (i) m=3. Teh-Chin algorithm (j) k-cosine, (k) k curvature, and (l) 1 curvature.

Table 1. Recognized objects in Fig. 1 and their attributes.

	Object	Attributes
1	Regular Hexagon	P: (1258, 1081), L = 133, φ = 1.51
2	Parallelogram	P: (727, 1268), L1 = 292, L2 = 146, Θ = 46.1, φ = 0
3	Trapezoid	P: (73, 1081), L1 = 718, L2 = 390, H = 217, Θ = 90.68, φ = -0.16
4	Rhombus	P: (339, 1791), L = 220 Θ = 30.1, φ = -0.45
5	Trapezoid	P: (1220, 1194), L1 = 221, L2 = 150, H = 68.6, Θ = 65, φ = 0.77
6	Triangle	P1: (1295, 1153), P2: (1363, 1151), P3: (1325, 1122), Isoceles
7	Triangle	P1: (1396, 1826), P2: (1106, 1831), P3: (1256, 1971), Isoceles
8	Rectangle	P: (457, 1044), W = 835, H = 564, φ = -0.4, Table
9	Quasi-Hexagon	P: (692,1497), L1 = 440, L2 = 303, Θ = 89.1, φ = 3.0
10	Parallelogram	P: (765, 1790), L1 = 297.1, L2 = 148, Θ = 89.0, φ = -1.35, Single hatch: a1 = 135, d1 = 30
11	Traingle	P1: (1256, 1897), P2: (1399, 1755), P3: (1108, 1753), Isoceles
12	Polygon, irregular	Number of segments: 6, Center: (1556, 1706), Coordinates of vertices....

3.2. Detection of Critical Points

Most of the algorithms for detecting critical points to segment lines into shorter segments mark the local curvature maxima points as dominant. Five popular feature detection algorithms using this approach are: the angle detection procedures of [Rosenfeld and Johnston 73, Rosenfeld and Weszka 75], the corner finding algorithm of [Freeman and Davis 77], the dominant point detection method of [Sankar and Sharma 78], and the vertex detection algorithm of [Anderson and Bezdek 84]. These have been compared and evaluated in [Teh and Chin 89], who also propose a method: a parallel algorithm for detecting dominant points in a closed curve in which the region of support for each point is determined based on local properties. [Fischler and Bolles 86] detect dominant points along a curve by analyzing the deviations of the curve from a chord. The points are marked as a critical point, or belonging to a smooth or noisy interval. These markings depend on whether the curve makes a single excursion away from the chord, stays close to the chord, or makes two or more excursions away from the chord, respectively. A drawback of many of these methods based on critical points of high curvature, is that inflection points due to smooth changes between segments (such as transitions from a circular arc to a tangential line) are not found.

Another reference in which curvature methods are compared is [O'Gorman 88-1]. In this paper, chosen methods are grouped into two families, the difference of slopes (DOS) family, and the Gaussian smoothing family. For the DOS methods, curvature is measured as the angular difference between the slopes of two line segments fit to the data around each curve point. The difference between the methods within the DOS family is in the arc-length of the gap between the two segments. Many popular methods fall into this family. For the Gaussian smoothing family of methods, a local second derivative estimate of curvature is first taken for each point along the curve, then a Gaussian smoothing filter is applied to the resulting curvature plot. In both cases, features are determined from the curvature plot. In [O'Gorman 88-2], it is shown that, for the DOS methods, signal-to-noise ratio is maximized when the gap between segments is positive and equals the arc-length of the corner curvature. An advantage of both the DOS and Gaussian smoothing families of methods over many others is that their results are symmetric with respect to curve features -- an important criterion for judgement of feature detection methods.

Most of the critical point detection algorithms discussed above require parameters related to the minimum resolvable feature separation. However, features of different size and separation are usually present in a given image. Parameter values determined by the minimum size feature may not be adequate to smooth large features, and as a result, too many points may be detected as dominant points. One approach for solving this problem is to adaptively determine the parameters using local feature data -- with no required user parameters. [Teh and Chin 89] describe such an algorithm. First, the region of support is adaptively determined using local properties, and local curvature is measured within this. Then, dominant points are detected by non-maxima suppression of local curvature. Results of approximations by several algorithms on one of the test images are shown in Fig. 4. It is clear that the approximations generated by their algorithm compare well against non-adaptive methods. But the output generated by the algorithm is sensitive to the direction of travel along the curve, and this results in asymmetrical approximations to symmetrical data (Fig 4.). In addition, the effect of noise along the curve can cause the region of support to be incorrectly estimated. Of course, were it not for the tradeoff between noise reduction and signal resolution, the problem would be much easier. [Phillips and Rosenfeld 87] also discuss determination of region of support. To get feature points they use the arc-chord distance property.

The problem of feature point detection in digital curves may also be approached as a scale-space filtering problem [Witkin 83, Deguchi 88]. In [Deguchi 88], k-curvatures (for different values of k) are calculated by finding the differences in slopes of chords connecting pixels that are k pixels from the current point on either side.

An adaptive filtering algorithm to smooth noisy curves has been described by [Saint-Marc et al. 89]. To illustrate the performance of this algorithm, consider the closed curve shown in Fig. 5(a). The objective is to locate all the vertices and other critical points such as the inflection point along the curved segment and points of transition from curve to straight line segments. The curvature at each point along the core-line obtained by thinning is determined using a small region of support. Because of artifacts introduced during digitization and thinning, this curvature function in Fig. 5(b) is not smooth. Smoothing is necessary to identify features such as vertices (peaks and valleys in the curvature plot), inflection points (zero-crossings in curvature), and smooth joins (points of transition from zero curvature to a significant value). A common filter used for smoothing noisy signals is the Gaussian filter. However, Gaussian filtering smooths both noise and data points. Alternatively, an adaptive filter that emphasizes intra-region smoothing over inter-region smoothing [Saint-Marc et al. 89] is useful in this situation. The smoothed function after 75 iterations is shown in Fig. 5(c). Peaks (O's) and significant zero crossings (X's) detected are marked in Fig. 5(c) and the corresponding points on the original closed curve are shown in Fig. 5(a).

After detecting critical points, data is represented as a collection of line segments connecting feature points. Attributes of these segments are also computed. Curves are often approximated by piece-wise circular arcs. These line segment and feature point data along with their spatial relationships are retained in appropriate data structures. Information in these data structures is then used for graphics recognition. Techniques for graphics recognition and interpretation are discussed in the next section.

4. GRAPHICS RECOGNITION AND INTERPRETATION

Recognition of graphical shapes and their spatial relationships is an important final task in most document analysis systems. The recognition process essentially consists of two main steps: processing the input image to obtain representational primitives (as described in the previous sections), and matching these primitives against similar primitives derived from known models. Techniques used for graphics recognition are strongly application-dependent. To recognize isolated symbols of fixed size and orientation, simple template matching applied directly to bit-mapped image may be adequate. However for many applications, this simple technique is inappropriate, and features as described above must be extracted. In certain applications, it may be adequate to approximate closed contours by polygons. In others, more sophisticated algorithms that hypothesize possible matches, compute scene/model transformations, and verify the hypotheses are used. In more complex images such as maps and engineering drawings, context-dependent, knowledge-based graphics recognition and interpretation techniques have been used. Different algorithms exhibit varying degrees of flexibility, accuracy, and robustness.

In this section, various graphics recognition and interpretation techniques are described. The techniques described here make use of the line and feature data obtained using techniques described in the previous sections. Recognition algorithms that operate directly on bit-mapped data or those that are based on well known techniques such as signature analysis, Fourier descriptors, etc. are not described here.

4.1. Hierarchical Decomposition and Matching

Techniques such as polygonal approximations are useful for shape recognition only when the complete outline of the object is available. Although this is not a serious restriction in most of the graphics recognition applications, occasionally it would be necessary to recognize graphical shapes when some part of the symbol is missing or hidden behind other shapes. Further, in some applications in which the number of possible different shapes to be recognized is large, it may be efficient to represent complex parts as a combination of already known simple shapes along with their spatial relationships. An object recognition system that creates a library of parts by hierarchical decomposition is described in [Ettinger 88]. The library organization and indexing is designed to avoid linear search of all the model objects. The system has hierarchical organization for both structure (whole object-to-component sub-parts) and scale (gross-to-fine features). Object representation is based on the Curvature Primal Sketch. The model libraries are automatically built using the hierarchical nature of the model representations. The recognition engine is structured as an interpretation tree. A constrained search scheme is used for matching scene features to model features.

4.2. Structure Analysis Based Graphics Recognition and Interpretation

For recognition of drawings such as flow-charts, block diagrams, electrical schematics, mechanical part drawings, etc., structural analysis of lines and their interconnections is required to generate meaningful and succinct descriptions. An unconstrained algorithm could interpret a table as a collection of many rectangles, whereas it would be more useful to describe it as a rectangle with horizontal and vertical bars. In case of electrical schematics it would be necessary to separate lines belonging to symbols from connecting lines.

An algorithm for generating loops of minimum redundancy is described in [Kasturi et al. 90]. Priorities are assigned to line segments and these are used in selecting line segments to form a closed loop. The loops that are extracted are then compared with a library of *known shapes* for recognition and description. Those shapes that are not recognized are analyzed to verify if they can be described as partially overlapped *known shapes*. All other segments are described as interconnecting lines or hatching patterns. The system outputs all recognized shapes, their attributes and spatial relationships. Part of the output generated by the system corresponding to the test image of Fig. 1 is shown in Table 1. Some of these techniques have also been used in a map-based geographic information system [Kasturi et al. 89].

[Ejiri et al. 90] apply structure analysis methods for recognition of engineering drawings and maps. To process LSI cell diagrams, solid lines and two types of broken lines in any one of six colors are recognized using a color digitizer. A loop-finding routine facilitates detection of overlapped lines denoted by a special mark. Structure analysis methods have also been used to recognize characters and symbols in logic circuit diagrams, chemical plant diagrams, and mechanical part drawings.

[Fahn et al. 88] describe a topology-based component extraction system to recognize symbols in electronic circuit diagrams. The system is designed to recognize circuit symbols in four orientations, and connection lines that are horizontal, vertical or diagonal. In [Okazaki et al. 88], a loop-structure analysis system is described for recognition of circuit symbols.

A method for interpreting the 3-D shape of an object corresponding to multiple orthographic views in an engineering drawing has been described in [Lysak and Kasturi 90]. The technique is based on a bottom-up approach in which candidate vertices and edges are used to generate a set of possible faces, which are in turn assembled into enclosures representing the final object [Wesley and Markowsky 81].

4.3 Knowledge Based Systems

Rule-based or knowledge-based systems have also been used as recognition engines [Nagy et al. 85, Niyogi and Srihari 86, Bley 84, Bunke 82, Huang and Tou 86]. [Niyogi and Srihari 86] describe a production system for document understanding. Knowledge rules pertain to intrinsic properties and spatial relationships among various regions of the image. Control rules are used to decide which knowledge rules need to be executed. Strategy rules are used to determine whether a consistent interpretation has been obtained. Such production systems are difficult to develop since the compilation of a knowledge-base is a lengthy process. Since segmentation is affected by noise and viewing conditions, any error generated at this stage could be propagated all the way to final interpretation.

[Anotine et al. 90] describe REDRAW, a system for interpretation of different classes of technical documents. It uses *a priori* knowledge to achieve interpretation at a semantic level. Here the aim is to build a general model driven recognition system which can be completely parameterized. The Model contains specific knowledge for each document class. The interpretation system is then driven by this model using a general tool box for low level operations. Two applications, city map and mechanical drawing interpretation are described. The *a priori* knowledge about the domain induces particular interpretation process for each document class. Among the low level operations they describe a method for extraction of parallel lines (hatched areas). A single scan algorithm in horizontal and vertical directions determinates an order relation for each extremity point for each segment. From these relations classes of parallel lines in a same area are deduced by assuming that parallel lines from a hatched area must overlap each other in the two main directions.

[Joseph and Pridmore 90] describe a schema driven system called ANON, for the interpretation of engineering drawings. Their approach is based on the combination of schemata describing prototypical drawing constructs with a library of low-level image analysis routines and a set of explicit control rules. The system operates directly on the image without prior thresholding or vectorization, combining the extraction of primitives (low level) with their interpretation (high level). ANON integrates bottom up and top down strategies in a single framework modelled on the human "cycle of perception". The system has been successfully applied to piece-part drawings. They also give a comparison of ANON with other such image understanding systems. Another frame based system for interpretation of structured documents is described in [Bayer 90]. It consists of a declarative representation language embedded in a frame system, an interpreter and a set of knowledge sources. The language enables the definition of the document as a set of objects. The interpreter uses the contents of the frame system and the set of knowledge sources to interpret a document. The knowledge base contains information about certain classes of documents which are recognized. The interpretation is guided by a control algorithm since many hypotheses could be generated.

A model based learning system for recognizing similar hand drawn electrical circuit symbols in the absence of any information concerning the pose (translation, rotation and scale) is proposed by [Lee et al. 90]. A hybrid representation called attributed graph (AG), which incorporates structural and statistical characteristics of image patterns is used for matching an input symbol with respect to model symbols. Model AG's are created interactively using a learning algorithm. The poses of input object AG's are estimated based on a minimum square error transform and they are classified based on minimum distance. An average error rate of 7.3% is reported. [Baird and Thompson 90] use knowledge of chess rules for correcting the chess moves read by a page reader used for extracting chess games from documents. Each move is checked for legality directly in the context built up by prior moves and indirectly through the consistency of later moves. The top-down approach is used for recognizing printed piano music in [Kato and Inokuchi 90].

4.4. Other Approaches for Recognition

The hypothesis-prediction-verification paradigm combines bottom-up and top-down techniques to take advantage of the system's knowledge of the objects. [Govindaraju et al. 89] uses this to locate human faces in newspaper photographs. A Hough transform technique is used to generate candidate regions for faces. Top-down methodology employs the spatial constraints (with respect to the photograph captions) and heuristics from photo-journalism, to eliminate inconsistent candidates returned by bottom-up Hough transform analysis, and verify these predictions.

Another approach to recognition is using relaxation. [Bunke and Alleman 81] use this method to analyze electrical schematics. A vector with probabilities of possible interpretations is first assigned to each vertex in the schematic. The probabilities of the interpretations at various vertices are successively changed by the relaxation process in order to achieve unique and consistent labeling. This recognition engine depends on effective propagation along the matching vertices, good initial assignments, and convexity of the configuration space and the domain.

5. SUMMARY

In this paper we have presented an overview of many algorithms and techniques that are useful in a graphics recognition system. Since a typical graphics image contains regions of text as well as graphics, region segmentation techniques are used to separate the two. These were discussed in Section 1. Data conversion from bit-mapped form to line segment form is a necessary intermediate step in all but very simple graphics recognition systems. Vectorization by line-fitting, thinning or line tracking and other methods, were discussed in Section 2. Algorithms for detection of feature points, and segmentation of concatenated lines into straight line segments and curves were given in Section 3. Techniques for recognition and description of graphics, such as hierarchical organization and matching, structure analysis and knowledge based approaches were described in Section 4.

ACKNOWLEDGEMENTS

An earlier version of this paper was presented at the IAPR Workshop on Syntactic and Structural Pattern Recognition [Kasturi et al. 90-2]. We would like to thank R.Raman and C.Chennubhotla for their help in preparing that paper. We would also like to thank R.T. Chin and K. Wall for permitting us to use figures from their paper, in our papers.

REFERENCES

Anderson, I.M. and J.C. Bezdek, "Curvature and tangential deflection of discrete arcs: A theory based on the commutator of matrix pairs and its application to vertex detection in planar shape data", **IEEE Trans. PAMI,** 6(1):27-40, Jan. 1984

Antoine, D., S. Collin and K. Tombre,"Analysis of Technical Documents: The REDRAW System",**Pre-Proc. IAPR workshop on Structural and Syntactic Pattern Recognition**,pp.192-230, New Jersey, June 1990.

Arcelli C., G.S. Di Baja, "A width-independent fast thinning algorithm", **IEEE Trans. PAMI,** 7(4):463-474, Jul. 1985

Baird, H.S., S.E. Jones, and S.J. Fortune, "Image segmentation by shape-directed covers," **Proc. 10th Intl. Conf. on Pattern Recognition,** pp.820-825, Atlantic City, New Jersey, 1990.

Baird, H.S., and K. Thompson, "Reading Chess", **IEEE Trans. PAMI,12**(6):552-559, June 1990.

Bayer, T.,"Interpretation of structured documents in a Frame System",**Pre-Proc. IAPR workshop on Structural and Syntactic Pattern Recognition**,pp.47-56, New Jersey, June 1990.

Bley, H., "Segmentation and preprocessing of electrical schematics using picture graphs," **Computer Graphics and Image processing, 28**:271-288, 1984

Bunke, H. and G. Allerman, "Probabilistic relaxation for the interpretation of electrical schematics," **Proc. Intl. Conf. Pattern Recognition**, pp. 438-440, 1981

Bunke, H. **Automatic interpretation of lines and text in circiut diagrams** in **Pattern Recognition Theory and Applications**, J. Kittler, K.S. Fu, and L.F. Pau (eds), pp. 297-310, D. Reidal Publishing Company, 1982

Casey, R.G. and K.Y. Wong, **Document analysis systems and techniques** in **Image Analysis Applications**, R. Kasturi and M.M. Trivedi (eds), Marcel Dekker, 1990

Deguchi, K., "Multi-scale curvatures for contour feature extraction," **Proc. 9th Intl. Conf. Pattern Recognition,** pp. 1113-1115, 1988

Dori., D., "A syntactic/geometric approach to recognition of dimensions in engineering machine drawings," **Computer Vision, Graphics, and Image Processing, 47**:1-21, 1989

Duda, R.O. and P.E. Hart, **Pattern Classification and Scene Analysis**, Wiley-Interscience, New York, pp. 338-339, 1973

Ejiri, M., S. Kakumoto, T. Miyatake, S. Shimada, and K. Iwamura, **Automatic recognition of engineering drawings and maps** in **Image Analysis Applications**, R. Kasturi and M.M. Trivedi (eds), Marcel Dekker, 1990

Ettinger, G.J., "Large hierarchical object recognition using libraries of parameterized model sub-parts," **Proc. Computer Vision and Pattern Recognition**, pp. 32-41, 1988

Fahn, C.S., J.F. Wang, and J.Y.Lee, "A topology-based component extractor for understanding electronic circuit diagrams, **Computer Vilson, Graphics and Image Processing, 44**:119-138, 1988

Fisher, J.L., S.C. Hinds and D.P. D'Amato, "A rule-based system for document image segmentation", **Proc. 10th Intl. Conf. on Pattern Recognition,** pp.567-572, June 1990.

Fischler, M.A. and R.C. Bolles, "Perceptual organization and curve partitioning", **IEEE Trans. PAMI, 8**(1):100-105, Jan. 1986

Fletcher, L.A and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics image," **IEEE Trans. PAMI, 10**:(6)910-918, Nov. 1988

Freeman, H. and L.S. Davis, "corner-finding algorithm for chain-coded curves," **IEEE Trans. Computers 26**:297-303, March 1977

Fujisawa, H. and Y. Nakano,"A top-down approach for the analysis of Document Images", **Pre-Proc. IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.113-122, New Jersey, June 1990.

Fulford, N.C. "The Fastrak automatic digitizing system," **Pattern Recognition, 14**:65-74, 1981

Govindaraju V., D.B. Sher, R.K. Srihari, and S.N. Srihari, "Locating human faces in newspaper photographs" **Proc. CVPR**, pp. 549-554, San Diego, 1989

Guo, Z. and R.W. Hall, "Parallel thinning with two-subiteration algorithms," **Comm. ACM, 32**:359-373, 1989

Harada, H., Y. Itoh, and M. Ishii, "Recognition of free-hand drawings in chemical plant engineering," **Proc. IEEE Workshop on Computer Architecture for Pattern Analysis and Image Database Management**, pp. 146-153, 1985

Haralick, R.M., L.T. Watson, T.J. Laffey, "The topographic primal sketch," **Int. J. Robotics Res., 2**:50-72, 1983.

Harris, J.F., J. Kittler, B. Llewellyn, and G. Preston, A

modular system for interpreting binary pixel representations of line-structured data in **Pattern Recognition: Theory and Applications**, J. Kittler, K.S. Fu, and L.F. Pau (eds), D. Reidal Publishing Co., pp. 311-351, 1982

Hinds, S.C., J.L. Fisher and D.P. D'Amato,"A document skew detection method using run-length encoding and the Hough transform", **Proc. 10th Intl. Conf. Pattern Recognition**, pp.464-468, June 1990.

Hori, O., and A. Okazaki,"High quality vectorization based on a generic object model", **Pre-Proc. IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.137-153, New Jersey, June 1990.

Huang, C.L and J.T. Tou, "Knowledge-based functional-symbol understanding in electronic circuit diagram interpretation," **Proc. SPIE Conf. Applications of Artificial Intelligence 635**, 1986

Imai, H. and M. Iri, "Computational-geometric methods for polygonal approximations of a curve," **Computer Graphics and Image Processing, 36**:31-41, 1986

Jagdish, H.V. and L. O'Gorman, "An object model for image recognition," **IEEE Computer, 22**(12):33-41, Dec. 1989

Jang, B.K., and R.T. Chin, "Analysis of thinning algorithms using Mathematical Morphology", **IEEE Trans. PAMI,12**:(6)541-551, June 1990.

Jimenez, J. and J.L. Navalon, "Some experiments in image vetorization," **IBM J. Res. Develop., 26**(6):724-734, Nov. 1982

Joseph, S.H., and T.P. Pridmore, "A system for the interpretation of images from graphics",**Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.154-164, New Jersey,June 1990.

Kanai, J., "Text-Line extraction using character prototypes", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.182-191,New Jersey,June 1990.

Karima, M., K.S. Sadhal, and T.O. McNeil, "From paper drawings to computer aided design," **IEEE Computer Graphics and Applications**, pp. 24-39, Feb. 1985

Kasturi, R., R. Fernandez, M.L. Amlani, W.C. Feng, "Map data processing in geographical information systems," **Computer, 22**(12):10-21, 1989

Kasturi, R., S.T. Bow, W. El-Masri, J. Shah, J. Gattiker, and U. Mokate, "A system for interpretation of line drawings," **IEEE Trans. PAMI,12**(10):978-992, October 1990

Kasturi R., R. Raman, C. Chennubhotla and L. O'Gorman, "Document Image Analysis: An overview of techniques for graphics recognition",**Pre-Proc. IAPR workshop on Structural and Syntactic Pattern Recognition**,pp.192-230, New Jersey, June 1990.

Kato, O. and S. Inokuchi,"The recognition system for printed piano music using musical knowledge and constraints", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.231-248, New Jersey,June 1990.

Kida, H., O. Iwaki, and K. Kawada, "Document recognition system for office automation," **Proc. 8th ICPR**, pp. 446-448, Paris, 1986

Kurozumi, Y. and W.A. Davis, "Polygonal approximation by minimax method," **Computer Graphics and Image Processing, 19**, 248-264, 1982

Landy, M.S. and Y. Cohen, "Vectorgraph coding: Efficient coding of line drawings," **Computer Vision, Graphics and Image Processing, 30**:331-344, 1985

Lee, S.W., J.H. Kim and F. Groen, "Translation,rotation and scale-invariant recognition of hand-drawn electrical circuit symbols with Attributed Graph matching", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.273-292, New Jersey,June 1990.

Leu, J.G. and L. Chen, "Polygonal approximation of 2-D shapes through boundary merging," **Pattern Recognition Letters, 7**:231-238, 1988

Lysak, D.B. and R. Kasturi, "Interpretation of line drawings with multiple views," **Proc. 10th ICPR (Pattern Recognition Systems and Applications Sub-conference)**,pp.220-222, June 1990

Masuda, I., N. Hagita, T. Akayama, T. Takahashi, and S. Naito, "Approach to smart document reader system," **Proc. CVPR**, pp. 550-557, San Francisco, 1985

Merelli, D., F. Mussio, and M. Padula, "An approach to the defination, description, and extraction of structures in binary digital images," **Computer Vision, Graphics, and Image Processing, 30**:19-49, 1985

Meynieux, E., S. Seisen, and K. Tombre, "Bilevel information recognition and coding in office paper documents," **Proc. 8th ICPR**, pp. 442-445, Paris, 1986

Naccache, N.J. and R. Shinghal, "SPTA: A proposed algorithm for thinning binary patterns," **IEEE Trans. Systems, Man, and Cybernetics, SMC-14**(3):409-418, 1984

Nagasamy, V., and N.A. Langrana, "Engineering drawing processing and vectorization system", **Computer Vision, Graphics and Image Processing,49**:379-397, 1990.

Nagy, S., S.C. Seth, and S.D. Stoddard, "Document analysis with an expert system", **Proc. International Workshop on Pattern Recognition in Practice** 2, Amsterdam, eds: E.S. Gelsema and L.N. Kanal, Elsevier Science Publishers B.V. (North-Holland), pp. 149-159, June 1985

Nishida, H., S. Mori,"A structural description of curves by Quasi-Topological features and Singular Points", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.310-334, New Jersey,June 1990.

Niyogi, D. and Srihari S.N. "A rule-based system for document understandiing," **Proc. AAAI**, pp. 789-793, Philadelphia, 1986

O'Gorman, L., "An analysis of feature detectability from curvature estimation", **Proc. CVPR,** pp. 235-240, Ann Arbor, Michigan, June 1988

O'Gorman, L., "Curvilinear feature detection from curvature estimation", **Proc. 9th ICPR,** pp. 1116-1119, Rome, 1988

O'Gorman, L., "kxk Thinning", **Computer Vision, Graphics, and Image Processing, Vol. 51,** 195-215, 1990

O'Gorman, L. "Primitives Chain Code", **Computer Vision, Graphics, and Image Processing,** 1990 (in press)

Okamoto, M. and A. Miyazawa, "An experimental implementation of document recognition system for papers containing mathematical expressions", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.335-350, New Jersey,June 1990.

Okazaki, A., T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto, "An automatic circuit diagram reader with loop-structure-based symbol recognition" **IEEE Trans. PAMI, 10**(3):331-341, May 1988

Parker, J.R., "Extracting vectors from raster images," **Computer and Graphics, 12**(1):75-79, 1988

Pavlidis, T., **Structural Pattern Recognition,** Springer-Verlag, New York, 1977

Pavlidis, T., "Survey: A review of algorithms for shape analysis," **Computer Graphics and Image Processing, 7**:243-258, 1978

Pavlidis, T. **Algorithms for Graphics and Image Processing,** Computer Science Press, Rockville, Maryland, 1982

Pavlidis, T. and L.L. Cherry, "Vector and arc encoding of graphics and text," **Proc. 6th ICPR**, pp. 610-613, Munich, October 1982

Pavlidis, T. "A hybrid vectorization algorithm," **Proc. 7th ICPR**, pp. 490-492, Montreal, 1984

Phillips, T.Y. and A. Rosenfeld, "A method of curve partitioning using arc-chord distance" **Pattern Recog. Letters, 5**:285-288, April 1987

Ramachandran, K., "A coding method for vector representation of engineering drawings," **Proc. IEEE, 68**:813-817, 1980

Ramer, U.E., "An iterative procedure for the polygonal approximation of plane curves, **Computer Graphics and Image Processing, 1**:244-256, 1972

Reihani, K., and W.E. Thompson, "A robust technique for shape recognition", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.366-376, New Jersey,June 1990.

Rosenfeld, A. and E. Johnston, "Angle detection on digital curves," **IEEE Trans. Computers, 22**:875-878, Sept. 1973

Rosenfeld, A. and J.S. Weszka, "An improved method of angle detection on digital curves," **IEEE Trans. Computers, 24**:940-941, Sept. 1975

Sahoo, P.K., S. Soltani, A.K.C. Wong, and Y.C. Chen, "A survey of thresholding techniques," **Computer Vision, Graphics, and Image Processing, 41**(2):233-260, February 1988

Saint-Marc, P., J.S. Chen, and G. Medioni, "Adaptive smoothing: A general tool for early vision," **Proc. CVPR**, pp. 618-624, San Diego, 1989

Sankar, P.V. and C.V. Sharma, "A parallel procedure for the detection of dominant points on a digital curve," **Computer Graphics and Image Processing, 7**:403-412, 1978

Shih, C-C. and R. Kasturi, "Extraction of Graphic Primitives from Images of Paper-based Drawings," **Machine Vision and Applications, 2**:103-113, 1989

Sirjani, A. and G.R.Cross, "An algorithm for polygonal approximation of a digital object," **Pattern Recognition Letters 7**:299-303, 1988

Sklansky, J. and V. Gonzalez, "Fast polygonal approximation of digitized curves," **Pattern Recognition, 12**:327-331, 1980

Srihari, S.N. and G.M. Zack, "Document image analysis," **Proc. 8th ICPR**, pp. 434-436, Paris, 1986

Srihari, S.N. and V. Govindaraju, "Analysis of textual images using the Hough transform," **Machine Vision and Applications, 2**:141-153, 1989

Suzuki, T., and S. Mori, "A thinning method based on cell structures", **Proc. of International Workshop on Frontiers in Handwriting Recognition**, pp.39-52, Montreal,Canada, April 1990.

Tamura, H., "A comparison of line thinning algorithms from digital geometry viewpoint," **Int. Joint Conf. Pattern Recognition**, Kyoto, Japan, pp. 715-719, Nov., 1978

Teh, C.H. and R.T. Chin, "On the detection of dominant points on digital curves," **IEEE Trans. PAMI, 11**(8):859-872, 1989

Viswanathan, M., "Analysis of scanned documents- a syntactic approach", **Pre-Proc. of IAPR workshop on Structural and Syntactic Pattern Recognition**, pp.450-459, New Jersey,June 1990.

Wakayama, T., "A core line tracking algorithm based on maximal square moving," **IEEE Trans. PAMI, 4**(1):68-74, Jan. 1982

Wall, K., "Curve fitting based on polygonal approximation," **Proc. 8th ICPR**, pp. 1273-1275, Paris, 1986

Watson, L.T., K. Arvind, R.W. Ehrich, R.M. Haralick, "Extraction of lines and drawings from grey tone line drawing images," **Pattern Recognition, 17**(5):493-507, 1984

Wesley, M.A. and S. Markowsky, "Fleshing out projections," **IBM J. Research and Development, 25**(6):934-954, Nov. 1981

Williams, C.M., "Bounded straight-line approximation of digitized planar curves and lines," **Computer Graphics and Image Processing, 16**:370-381, 1981

Williams, C.M., "An efficient algorithm for the piecewise linear approximation of planar curves," **Computer Graphics and Image Processing, 8**:286-293, 1978

Witkin, A.P., "Scale-space filtering," **Proc. 8th IJCAI**, pp. 1019-1022, Karlsruhe, W. Germany, Aug. 1983

Woetzel, G., "A fast and economic scan-to-line conversion algorithm," **Computer Graphics, 12**:125-129, 1978

Wong, K.Y., R.G. Casey and F.M. Wahl, "Document analysis system," **IBM J. Res. Dev., 26**(6):647-656, 1982

Zen, H. and S. Ozawa, "Extraction of the fair document from mixed mode manuscript," **Proc. CVPR**, pp. 544-549, San Francisco, 1985

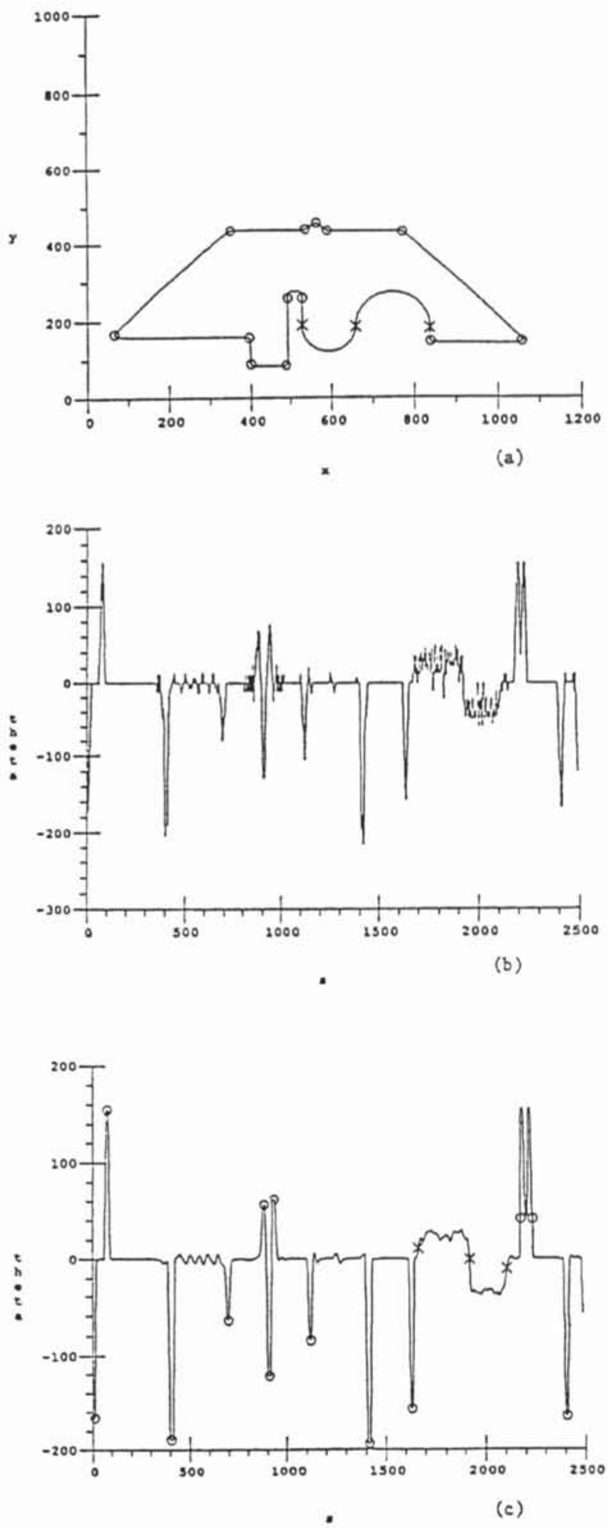

Figure 5. (a) A closed curve for evaluating adaptive smoothing algorithm (O's and X's indicate critical points). (b) Plot of curvature as a function of position. (c) Curvature after adaptive smoothing.