Twenty Years of Document Image Analysis in PAMI

George Nagy, *Senior Member, IEEE*

Abstract—The contributions to document image analysis of 99 papers published in the *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)* are clustered, summarized, interpolated, interpreted, and tactfully evaluated.

Index Terms—Document image analysis, image processing, OCR, character recognition, forms processing, graphics recognition.

1 PAMI AND DIA

INSTEAD of attempting to survey the entire field of document image analysis (DIA), we review only results reported in *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Parochial as this may seem, it gives a sharp, well-defined cross-section of the evolution of DIA research. The 99 relevant papers that were found (less than five percent of all articles) are contemplated from a perspective bolstered by lively students, eclectic reading, participation in conferences, and discussions with knowledgeable and opinionated colleagues.

This section considers the role played by *PAMI* in relation to other sources of published information and current commercial practice, differentiates document image analysis from allied disciplines, and describes its major constituents. It should be skipped by old hands with their own cognitive map of the field. The next five sections summarize the DIA tasks addressed in *PAMI* in the last two decades. Only *PAMI* papers are cited, but a short bibliography provides additional entry points to the literature. The conclusion is the author's classification of the domain into *problems solved* and *problems remaining.*

1.1 PAMI vs. Other Sources

It would appear that 99 articles among the several thousand published about DIA in the past 20 years can represent at best a fraction of the state of the art. However, *PAMI* covers much more ground than this ratio would indicate. Before *PAMI*'s birth in 1979, character recognition research appeared mainly in *IEEE Transactions on Computers (TC)* (*IEEE Transcaction on Electronic Computers (EC)* until 1968) and in the *Proceedings of the IEEE,* in addition to occasional specialized conferences and workshops. (One of the earliest, the 1966 IEEE Pattern Recognition Workshop in Puerto Rico, resulted in the *IEEE Computer Group's*[1] pattern recognition database.) The *Journal of Pattern Recognition* has regularly published articles on character recognition since its inception in 1971, as has *Pattern Recognition Letters.* Relevant articles appear occasionally in the *IEEE Transactions on Information Theory, Systems, Man, and Cybernetics, Neural Networks,* and *Image Processing,* as well as in a dozen commercially published journals of artificial intelligence, pattern recognition, computer vision, and image processing. The best source of current trade news is the monthly, *Imaging and Document Solutions.* In 1998, Elsevier launched the *International Journal of Document Analysis and Recognition* with the goal of capturing the fractionated DIA and OCR literature. One indication of the dispersal of this literature is that few of the citations in *PAMI* articles reference *PAMI.*

Since 1973, the biennial International Conference on Pattern Recognition (ICPR) has been a steady source of ideas. It has been supplemented since 1991 by the International Conference on Document Analysis and Recognition (ICDAR). Worthwhile contributions have appeared at the annual SPIE Document Recognition and Retrieval (DR&R) symposia in San Jose, and at the peregrinating biennial Document Analysis Systems (DAS) and Structural and Syntactic Pattern Recognition (SSPR) workshops, each of which attracts about 100 participants. During its five-year life span, the Symposium on Document Analysis and Information Retrieval (SDAIR) fostered interaction between DIA and IR specialists. It also featured the results of a large-scale in-house evaluation of commercial OCR technology. Several countries have instituted national conferences on OCR or DIA. The articles found in *PAMI* reflect the international constituency of document analysis. We are often reminded that English is blessed with one of the simplest scripts in the world.

1.2 PAMI vs. Current Practice

We consider DIA and OCR as essentially engineering disciplines, although a case can be made for a more fundamental role. Published work (not only in *PAMI*) has been moderately successful in anticipating emerging applications. The many papers on hand-printed and handwritten character recognition (cf. article by Plamondon and Srihari in this issue) probably did contribute to current products, but some of the print recognition methods explored by researchers risk lagging the capabilities of give-away shrink-wrapped page readers. The research emphasis in

[1] Predecessor of the Computer Society.

• *The author is with the Rensselaer Polytechnic Institute, Troy, NY 12180. E-mail: nagy@ecse.rpi.edu.*

Manuscript received 10 Aug. 1999; accepted 12 Oct. 1999.
Recommended for acceptance by K. Bowyer.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 110406.

TABLE 1
The Growth of DIA in PAMI

Period	DIA Papers	All papers
1979-83	12	381
1984-88	14	407
1989-93	22	584
1994-98	47	667

character recognition has long favored "universal" algorithms, whereas the commercial emphasis is now on tool kits that can be customized for large, homogenous applications. There was little in *PAMI* that foreshadowed the rapid conversion of large archives of printed material for digital libraries.

Published research on forms processing lags behind the many challenging applications and the need for further improvement. Graphics recognition should connect with current CAD practices to improve the widely-used interactive techniques for engineering drawing and map conversion. We must be more alert to guide the extraction of hidden structure from indigenous electronic documents images (which we hereby define as *computer-created documents that have never been scanned but appear in some unstructured format like GIFF or TIFF*). Research on document image security, authentication ("digital watermarking") and privacy should not fall behind commercial products. Some of these issues were the topics of conference presentations, but never made it to *PAMI*. Finally, we hope that our survey for the 40th anniversary issue will include some critical overviews of test protocols, benchmarking, evaluation, validation, and standard data sets that are essential for the health of a mature discipline.

On the positive side, we found the source of many ideas that are now part of the infrastructure. Character recognition has long been a favorite test vehicle for general-purpose classification algorithms and stimulated the examination of important issues in feature selection, clustering, and small-sample estimation of classifier parameters and of error rates (cf. Jain et al., in this issue). Graphics recognition is a wonderful application of syntactic and structural methods. Document layout analysis *is* image processing, while document content analysis draws on machine learning. DIA benefits from and contributes to artificial intelligence, especially in natural language understanding, knowledge representation, and neural networks. Symbiosis aside, the number of specifically DIA-oriented papers published in *PAMI* has grown significantly in the last five years (Table 1). The articles show increased sophistication and appreciation of engineering aspects. As we shall see, three editors-in-chief, Theo Pavlidis, Anil Jain, and Rangachar Kasturi made, recent, influential contributions to the discipline.

1.3 DIA in Context

Document image analysis is the subfield of digital image processing that aims at converting document images to symbolic form for modification, storage, retrieval, reuse, and transmission. It helps the transition from bookshelves and filing cabinets to the paperless (and perhaps even wireless) world. Although there is no evidence yet of less paper, electronic documents already abound.

Regardless of how images are obtained, we may classify them according to their content. For our purposes, it is convenient to divide them into two categories: *natural* and *symbolic*. Portraits, fingerprints, aerial photographs, satellite images, and X-rays depict natural scenes or objects. On the other hand, postal addresses, printed articles, bureaucratic forms, sheet music, engineering drawings, and topographic maps represent symbolic objects. We therefore propose the following working definition:

> Document image analysis (DIA) is the theory and practice of recovering the symbol structure of digital images scanned from paper or produced by computer.

For ease of reading, pictures of symbols tend to be produced with high contrast: most text and line art is essentially black-on-white. Color is applied where necessary, without fine tonal gradations (though trendy magazines often splash color without regard for legibility). Photographs are reproduced as halftones. Accordingly, linear signal-analysis techniques based on frequency transforms are less prevalent in DIA than in computer vision and in natural picture processing.

There is considerable current interest in recognizing incidental text (license plates, billboards, and subtitles) in photographs and video. It may therefore be convenient to extend the definition of DIA to any multimedia recording. Here, however, we will limit *document* to objects created expressly to convey information encoded as iconic symbols.

1.4 A Short Tour of DIA

The field arose well before digital computers could represent information that traditionally appeared on paper. Patents on optical character recognition, for reading aids for the blind and for input to the telegraph, were filed in the 19th century, and working models were demonstrated by 1916, but DIA as we know it is only about 40 years old. OCR on specially designed printed digits (*OCR fonts*) was first used in business in the 1950s. The first postal address readers and the Social Security Administration machine to read typewritten earnings reports were installed in 1965. Devices to read typeset material and simple hand-printed forms came into their own in the 1980s, when the prices of OCR systems dropped by a factor of ten due to the advent of microprocessors, bit-mapped displays, and solid-state scanners.

Within another decade, optical disks and tape cartridges had sufficient capacity, at a reasonable price, for storing thousands of document images in the form of compressed bitmaps. Interest grew quickly in converting them to computer-searchable form. In 1999, *document imaging*, i.e., electronic document storage without sophisticated image manipulation, is a billion-dollar business, but DIA, when considered as document image *interpretation*, is still only a small part of it. It is fortunate that such a practical field of study has so many intellectually stimulating aspects.

The only principles of physics that impact DIA are those that govern the production of documents through printing, copying, and digitization. However, the quality of these technology-sensitive steps has enormous effect on the complexity of extracting information. The combined cost of conversion and processing determines where DIA can displace human data entry and correction.

The digitized images can be stored in a simple array format. Once the symbolic information is extracted and interpreted, the best format depends on the application at hand. Some formats are rooted in word processors (e.g., RTF), page composition software (LATEX, PDF), web browsers (HTML), or drafting programs (DXF). Others are constructed specifically for the manipulation of scanned documents (DAFS, XDOC, and ODIL). The many formats, and the methods of converting one to another, seldom form the topic of research articles (but we include a glossary of acronyms).

Some of the early stages of processing scanned documents are independent of the type of document. Many noise filtering, binarization, edge extraction, and segmentation methods can be applied equally well to printed or handwritten text, line drawings or maps. Half-tones require specialized treatment. (Binarization does, of course, segment gray-scale pictures. But we prefer to reserve *segmentation* for methods that may be applicable even to already binarized pictures. Connected components analysis is often the starting point here, but its development has not been chronicled in *PAMI*.)

Once a document is segmented into its constituent components, more specific techniques are needed. Traditionally the field has differentiated between *mostly text* and *mostly graphics* documents. Mostly-text pages are separated into columns, paragraph-blocks, text-lines, words, and characters. OCR converts the individual word or character images into a character code like ASCII or Unicode. But, there is much more to a text document than a string of symbols. Additional, and sometimes essential, information is conveyed by long-established conventions of layout and format, choice of type size and typeface, italics and boldface, and the two-dimensional arrangement of tables and formulas. To capture the whole meaning of a document, DIA must extract and interpret all of this subtle encoding. (Nowadays, this information about information is called *metadata*, but the term has different meanings in library science, web searches, programming, and scripting languages.) Specialized techniques are appropriate and affordable for high-volume text applications like envelopes, business letters, bank-checks, and invoices.

Engineering drawings, maps, music scores, schematic diagrams, and organization charts are examples of mostly-graphics documents. Line drawings are decomposed into straight-line and curve segments, junctions and crossings before higher-level components, such as dimensions, center-lines, and cross-hatching can be interpreted. Maps may require color separation, and the association of text (labels) with map symbols. Line drawings typically contain a great deal of lettering that must be located, perhaps isolated, and recognized. The extraction of information from digitized photographs remains a thriving topic of research, but since photographs seldom contain symbolic objects, our main concern in DIA is mistaking them for text or line drawings. At every stage of research, it is essential to be able to display the results of processing in a form suitable for human judgment. Accurate rendering of digitized pictures at various scales requires some care.

Table 2 is our attempt to fit the major concepts of DIA into a coherent schema. We divide the analysis into five steps according to the granularity of the process. Here we have opted to group documents into only two types. The major decision is top-down vs. bottom-up analysis. In mostly text processing, top-down analysis attempts to find the larger components, like columns and paragraph blocks, before proceeding to the text-line and word levels. Bottom-up analysis forms words into text-lines, lines into paragraphs, and so on [63]. Because black pixels are usually a much smaller fraction of a line drawing than of text, most techniques for graphics are bottom-up approaches. The distinction is elusive, because both methods must access individual pixels, runs, or connected components. In our survey, we have chosen to proceed from small to large, and to summarize work on mostly text separately from research on mostly graphics.

Document image understanding (or *interpretation*) is the formal representation of the abstract relationships indicated by the two-dimensional arrangement of the symbols. (In contrast, *document understanding* refers to the natural language aspects of one-dimensional text flow.) Domain-specific knowledge appears essential for document interpretation. To the best of our knowledge, no one has ever attempted to develop a system to interpret arbitrary documents (consider the knowledge required to understand the periodic table, a foreign railroad schedule, an IRS form, a chartered accountant's report, or an abstruse formula).

Both model-driven and data-driven approaches have been investigated. Models have been developed for formulas, equations, business forms, tables, flowcharts, mechanical drawings, circuit schematics, music, and chess notation. Some of these models reflect the properties of natural language, while others have domain-specific constraints, like the correspondence between a dimension statement and the radius of an arc, the rules that determine whether an equation is well formed, and the relationships between the fields of an invoice. The models and underlying assumptions are not always explicitly stated in descriptions of new approaches.

Table 3 is a document taxonomy that emphasizes the DIA requirements related to the symbolic structure. It also gives examples of ancillary data that helps to extract it. The variety of documents that appear in Table 3 is illustrated in Fig. 1. As a minimum, it is desirable to extract automatically sufficient information for indexing every document. Usually, additional text will have to be extracted, though not necessarily in the appropriate reading order; in some applications, word occurrence may suffice. If the physical layout and the typographic information are recovered, then at least the document can be converted to some editable format.

The recovery of the symbolic structure can be loosely divided into logical and functional analysis. The former is the type of information that can be captured in a wordprocessor or HTML style-sheet, while the latter would

TABLE 2
Schema for Document Image Analysis

LEVEL OF PROCESSING	DOCUMENT TYPE	
(low to high)	MOSTLY-TEXT	MOSTLY-GRAPHICS
Pixels	**Preprocessing**	**Preprocessing**
	Representation	Representation
	Noise reduction	Noise reduction
	Binarization	Binarization
	Skew detection	Thinning
	Zoning	Vectorization
	Character segmentation	
	Script, language, & font recognition	
	Character scaling	
Primitives	**Glyph recognition**	**Primitive recognition**
	Connected components	Straight-lines & curve segments
	Strokes	Junctions and nodes
	Characters, diacritics, punctuation	Loops
	Words	Characters
Structures	**Text recognition**	**Structure recognition**
	Word segmentation	Text fields
	Text line reconstruction	Legends
	Table analysis	Label attribution
	Morphological context	Dimensions
	Lexical context	Graphics symbols
	Syntax, semantics	Aerial and texture features
		Beautification (constraints)
Documents	**Page layout analysis**	**Interpretation**
	Text versus non-text	Component recognition
	Physical component analysis	Connectivity analysis
	Logical component analysis	CAD/GIS layer separation
	Functional components (content tags)	Database attribute extraction
	Compression	Compression
Corpus	**Information retrieval**	**Database, CAD, GIS interface**
	Document classification and indexing	Validation
	Search	Search
	Security, authentication, privacy	Update

be tagged in XML (Extensible Markup Language), or in the data structure of a computer-aided drafting system vs. that of a design and manufacturing system. We can argue that the first corresponds to syntax and the second to semantics, though these words may carry different meanings for different researchers. *Title* and *author* may be obvious semantic categories, but what about *subheading* and *footnote*? Content-oriented tagging may well be the next plateau for DIA.

2 Preprocessing

Preprocessing generally consists of a series of image-to-image transformations. It does not increase our knowledge of the contents of the document, but may help to extract it. We also discuss here the identification of script, language, and font, which we regard as metadata that assists information recovery.

2.1 Compressed Representation

Until the advent of massive random access memories, there was considerable interest in character-level data

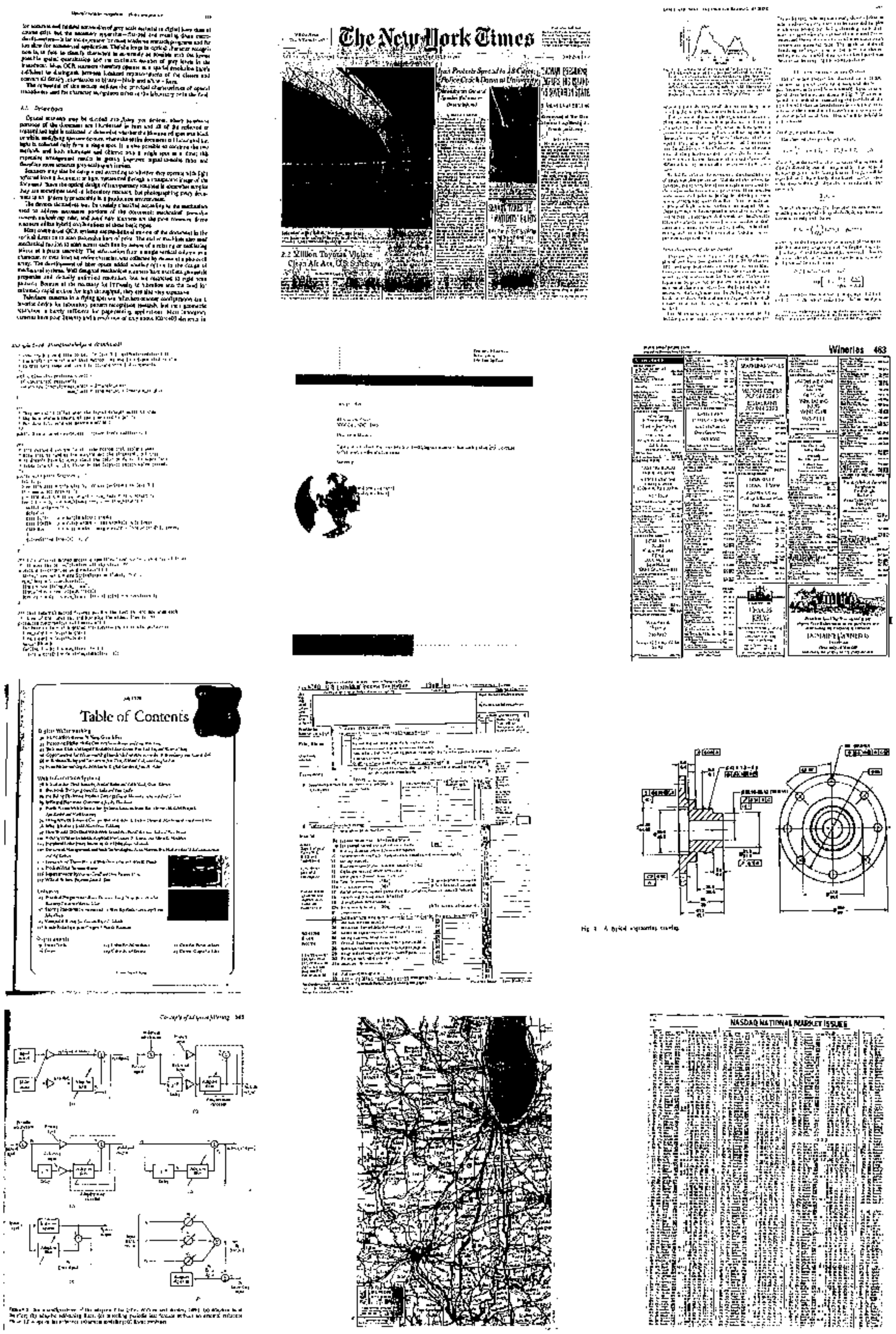

The New York Times
2.2 Million Toyotas Violate Clean Air Act, U.S. Says
Wineries 463
Table of Contents
Digital Watermarking
NASDAQ NATIONAL MARKET ISSUES

Fig.1. Illustration of document types for the taxonomy of Table 3.

TABLE 3
A Document Taxonomy

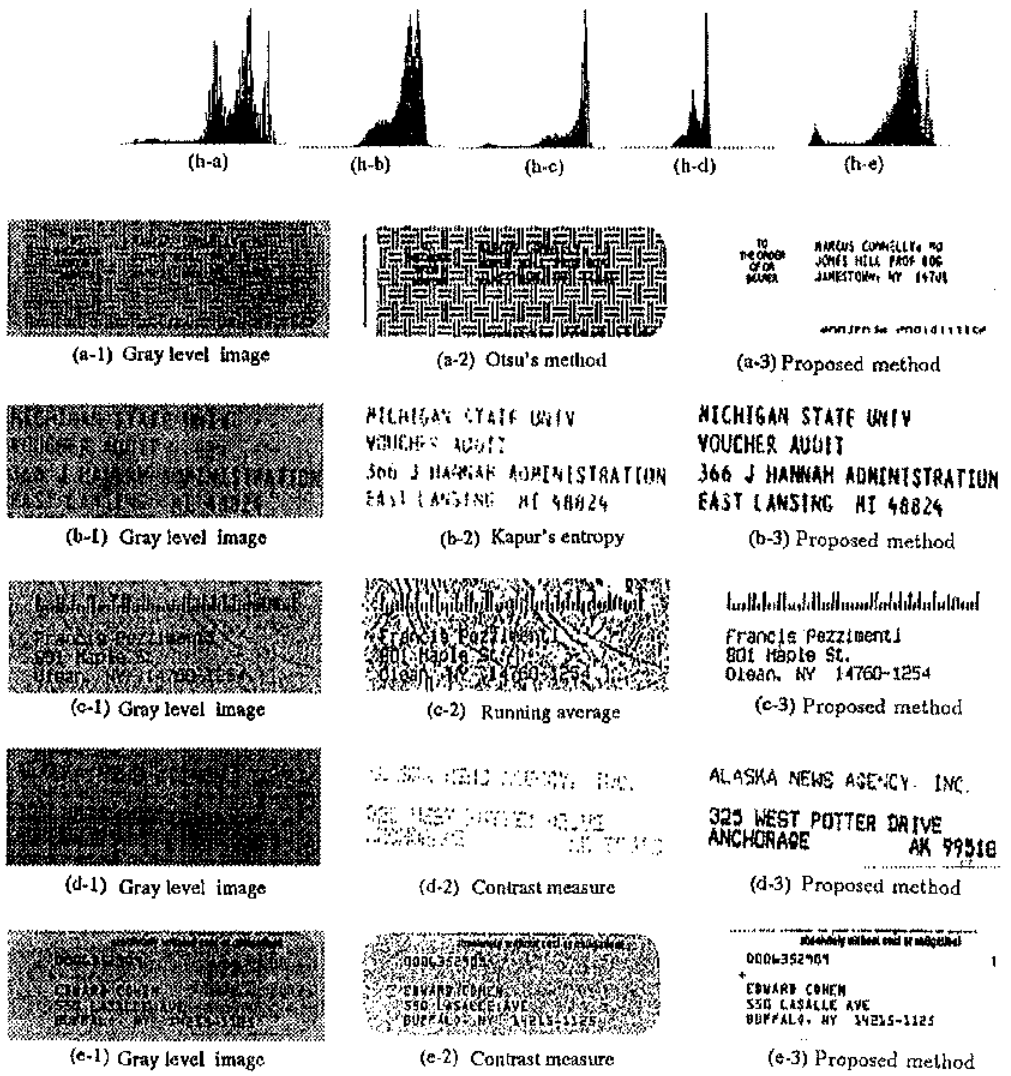
Type	Example	DIA Task	Ancillary data
plain text (narrative or descriptive)	Moby Dick, Gettysburg Address	extract correct word order	English lexicon
newspaper, magazine	NY Times, Vogue	separate and reassemble articles; pointers to illustrations	publication-specific format
scholarly & technical text	IEEE-PAMI, Dr. Dobbs Journal	index: author, title, page; pointers to refs, figs, tables, footnotes, equations	abbreviations, acronyms, units
formal text	program listing, chess, bridge, recipe	extract executable, or compilable, form	program, chess, bridge syntax
letter, envelope	information request, complaint, recommendation	extract routing info; index: sender, date, subject	directories
directory	telephone directory, street index	extract name-attribute pairs	previous edition
structured list	organization chart, table of contents, catalog	recover hierarchy; cross-references	previous edition
business form	order, invoice, subscription, survey, IRS-1040	link field content to dbms; convert to SGML or XML format;	formatted data, dbms, workflow system, lexicons
engineering drawing	assembly or part drawing; isometric view	convert to CAD format	part lists, drawing standards
schematic diagram	circuits, utility maps	extract net list or convert to CAD format	P-SPICE, manhole inventory
map	topographic quad, street map, road map	convert to GIS format	gazetteer, other maps, GIS
music score	Moonlight Sonata	recover MIDI representation	music syntax
table	airline schedules, stock quotes	construct formal model: headers ↔entries	airline and stock abbreviations, previous edition

compression methods simply to avoid disk access during page analysis. Run-length coding (RLC) and Freeman chain codes were used early on. Methods that came along later include reduced terminal sequences of context-free grammars [43], coding on hexagonal meshes [94], production rules for subblocks [58], and filtered contours [10]. The July 1980 special edition of the *Proceedings of the IEEE* on digital encoding of graphics contains many excellent surveys, mostly targeted at facsimile. For lossless, bilevel page compression, JBIG is gradually replacing CCITT-G3 and G4. The major remaining application of character encoding is font libraries.

2.2 Binarization

Most early document scanners had hardware reflectance thresholds, but current scanners typically produce 8-bit gray-scale (or color) output. Researchers from the University of Oslo and Michigan State University conducted a sustained, thorough comparison and evaluation of published adaptive binarization methods (including their own) on hydrographic charts [83], [84], [85], [86]. Niblack's method, based on a threshold set below the mean gray-level of a 15×15 window by a fixed fraction (0.2) of the standard deviation of the gray-levels, gave the best results on their maps. (A small modification is necessary when it is evident that the entire window is covered by a large foreground blot). They recommended postprocessing with the method of Yanowitz and Bruckstein, which iteratively creates a threshold surface that is essentially a low-pass-filtered version of the reflectance map. They also reported that character segmentation and recognition did not necessarily benefit from direct gray-scale processing as opposed to adaptive binarization [86].

Textured backgrounds are particularly difficult to handle. Liu and Srihari [53] provide a solution for postal address readers. It requires: 1) preliminary binarization



(h-a) (h-b) (h-c) (h-d) (h-e)

(a-1) Gray level image (a-2) Otsu's method (a-3) Proposed method

(b-1) Gray level image (b-2) Kapur's entropy (b-3) Proposed method

(c-1) Gray level image (c-2) Running average (c-3) Proposed method

(d-1) Gray level image (d-2) Contrast measure (d-3) Proposed method

(e-1) Gray level image (e-2) Contrast measure (e-3) Proposed method

Fig. 2. Comparison of binarization results [53].

based on a multimodal mixture distribution, 2) texture analysis with run-length histograms, and 3) selection of the threshold (using a small decision tree) that yields the stroke widths and lengths expected in a printed address. Fig. 2 compares their results with those of some alternative methods.

Sawaki and Hagita [71] recently demonstrated another specialized binarization method for textured and reverse-video (white-on-black) Japanese headlines. Their method is based on the complementary relationship between characters and their backgrounds as indicated by a similarity measure for black and white runs.

In general, the appropriate binarization threshold is a sensitive function of the local reflectance map, but for high-contrast printed matter, it is difficult to improve on a fixed threshold centered between the extreme observed values. In adopting published binarization algorithms for applications, in which the gray-level distribution is not clearly bimodal, it is important to take into consideration the amplitude transfer function of the specific scanner, as well as the spatial and gray-scale characteristics of the image. We echo a statement from [83] that still rings true today: "However, no single thresholding scheme gives satisfactory segmentation results on a variety of images".

Motivated by the observation that voting OCR labels obtained from multiple scans of the same page resulted in a substantial decrease in error rate, Sarkar et al. [70] examined the interplay between sampling and thresholding and counted the number of different bitmaps that can be produced for the same pattern by random displacement of the sampling grid. As shown earlier, the resulting uncertainty limits the precision with which small patterns can be located [24], [25] but the effect is less significant with gray-scale scans. It also affects electronically-produced documents converted to bitmaps for display, and impedes character recognition in GIFF pictures.

2.3 Skew Detection

Almost as many algorithms have been developed for skew detection as for binarization. All of them are accurate on full pages of uniformly aligned printed text. The better algorithms are less affected by the presence of graphics, paragraphs with different skew, curvilinear distortion arising from photocopying books, large areas of dark pixels near the margin, and few, short text lines.

A novel method is the Subspaced-Based Line Detection based on an analogy between text-lines on a page and a linear antenna array emitting a planar propagating wavefront [1]. The distance from a reference line of each foreground pixel is converted into the phase of a complex sine wave. The detection algorithm, based on radar signal processing, determines the spatial coherence between the contributions from different rows in the image. Aghajan and Kailath [1] claimed that their method was more efficient and more accurate for text skew than the Hough algorithm (which is, however, seldom used for documents without extensive modification). It is not clear from the paper whether any parameter requires adjustment for type size or line spacing.

One method was designed specifically for Indian scripts like Devanagari and Bangla [13] that have a head line (*shirorekha* or *matra*). It was developed as part of a complete Bangla OCR system. The results of skew detection proved comparable to those from the Hough transform, but require less computation.

After skew detection, the page image is often rotated to a reference direction to facilitate further format analysis and OCR. On binarized pages the required resampling tends to distort the character patterns. Instead, it may be possible to modify the processing algorithms to take into account the skew [34]. Alternatively, either the document can be rotated before binarization, or the rotation can be approximated by small, distortion-free translations of entire word blocks.

2.4 Character Segmentation

In 1996, Casey and Lecolinet [11] surveyed the many approaches that have been proposed since 1959 to segment touching or fragmented character patterns. Missegmentation of characters is responsible for many OCR errors (e.g., r n → m or m → r n). It is the consensus that *light* patterns are more difficult than *heavy* patterns, perhaps because of the greater import of missing and already scarce foreground pixels. The degree of difficulty depends on the typeface and print-source (smudged italics are difficult to segment) as well as on the ratio of font size to scanner resolution (point-spread function and spatial sampling rate).

Casey and Lecolinet [11] defined *dissection* as the attempt to divide the image into classifiable units, whereas *recognition-based segmentation* either classifies a multicharacter block at once, or segments the image according to features extracted from the entire block. *Hybrid* classification is a kind of soft segmentation, where the choice between multiple segmentation candidates is based on recognition. A fine example of the latter appeared in *PAMI* soon after the survey [51]. Here the winners among the jagged boundaries, imposed on the gray-scale patterns by pre-segmentation, are determined by the optimal path through a word-scale lattice (cf. [68]).

This comprehensive and scholarly survey concentrated on the underlying principles and did not attempt to evaluate the effectiveness of the various approaches. (By happenstance, a lively explanation of the difficulties of benchmarking segmentation, in a different venue, appeared in the preceding article of the same issue [29]). If there is a general conclusion, it is this: where dissection does not cut the mustard, the required gestalt techniques need to be so thoroughly integrated with recognition and context that character-level segmentation will soon disappear as a distinct area of research.

2.5 Character Scaling

In OCR, very small and very large word or character images are often scaled to a standard size, even though the outlines of characters of different sizes in the same typeface are not congruent. Resampling on gray-level arrays is relatively easy, but bilinear or bicubic interpolation distorts bilevel characters. The standard alternative is a two-stage process. The original smooth contour of the sampled character is first approximated using a weighted convolution filter and bilevel amplitude quantization. This stage is followed by resampling.

Scaling for OCR is not the purpose of the following methods, but they might find application in simulating or modeling OCR. Ulichney and Troxel [87], writing at a time when hardware cycles were more scarce, developed "telescoping templates" for high-fidelity scaling with only logical operations. Namane and Sid-Ahmed [62] designed their algorithm for characters captured by a camera. After detecting the borders of the character, the contour is scaled, then interpolated (for magnification) with cubic splines. A 5×5 template is used to construct a bilevel image. Their results appear smoother than those obtained by replication or by telescoping templates. Also applicable here is the sophisticated contour construction of [10].

2.6 Script, Language, and Font Recognition

Script recognition reduces the number of different symbol classes that must be considered during classification. Language recognition is necessary for use of the appropriate context models. Font classification reduces the number of alternative shapes for each class, leading to essentially single-font character recognition. Script, language, and font classification are also desirable for document indexing and interpretation. Four recent papers offer methods for routing documents to the appropriate recognition software.

Hochberg et al. [28] classified scripts using templates for connected components ("textual symbols") that occur frequently in each script. The templates, size normalized to 30×30 pixels, are obtained by clustering components in training documents, and selected according to their power of script discrimination. Classifying 100 components proved sufficient to identify 71 of 73 samples from 12 scripts and three dozen languages. (They suggested identifying the language by dictionary lookup after the script and characters are recognized).

Spitz [76], a pioneer in foreign language processing, classified both script and language as indicated in Fig. 3. He first differentiated between Latin and Han scripts according to the standard deviation of the vertical location of upward concavities with respect to the base line. (In Latin print,

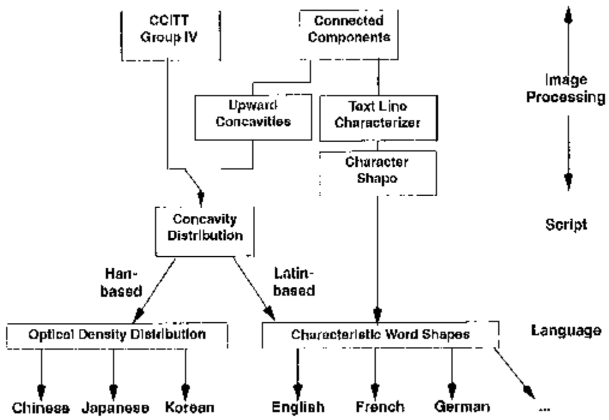


Fig. 3. Spitz's script and language classification scheme [76].

these are mostly at the baseline or the x-line, whereas in complex Chinese, Japanese, and Korean characters they are more uniformly distributed). The three Oriental scripts are then recognized according to histograms of foreground pixel densities. The languages in Latin scripts are identified by the frequency of characteristic Word Shape Tokens (such as high-high-low for "the" in English, high-low for "le" and "la" in French, and high-low-low for "der" and "das" in German). Spitz uses run-length coding, bounding boxes, and the pass codes of CCITT-G4 to speed up processing. The Han/Latin dichotomy appears infallible. The three Oriental scripts are recognized when more than a couple of text lines are available. Some European languages are recognized with only 90 percent accuracy. Spitz co-edited *Document Analysis Systems*.

Tan [80], classified 128 × 128 pixel samples into six script classes with 97 percent accuracy using rotation-invariant Gabor function coefficients. The presence of new typefaces affected, surprisingly, only the recognition of Chinese samples.

The three methods operate at very different levels. Tan was interested in the power of texture-based classification; therefore, his method is "global" and requires no preprocessing whatsoever. The basic unit for Spitz is the printed line. Hochberg et al. [28] advocated a bottom-up approach that requires *only* the connected components. It is possible that either of the last two methods can be modified to determine the script—and perhaps language—of individual words or phrases in multilingual documents. Comparing the three error rates obtained on different data is, of course, meaningless.

In practice, font recognition is likely to be faced with more classes than script or even language recognition. Fonts are classified according to typeface, weight, slope, width, and size. ApOFIS (*A priori Optical Font Identification System*) has a second-generation font model base for 280 fonts (10 typefaces, seven sizes, and four styles). Each model has statistics for six features estimated from 100 short text lines scanned at 300 dpi. Using this database and a Bayesian classifier, Zramdini and Ingold [99] classified fonts with 97 percent accuracy, and typeface, size, weight, and slope with 97.5-99.9 percent. The accuracy increases rapidly with the size of the test sample, which may mean that short inserts of italics or boldface may be missed, and technical manuals with a variety of typefaces will be a challenge. Nevertheless, this is a fine instance of computer classification that will outperform all but the most skilled typographer.

3 Character Recognition

The character recognition phase may precede, follow, or run concurrently with layout analysis. In the past, page decomposition analysis was usually applied without OCR because researchers did not have access to multifont OCR. Similarly, the early character recognition work was carried out on specially formatted pages because researchers did not have access to page decomposition software. We discuss character recognition before page decomposition partly to follow our bottom-up agenda, and partly to honor historical development. We consider it under four headings: 1) shape-based analysis of Latin (Roman) print, 2) shape analysis of other scripts, 3) context, and 4) global classification.

3.1 Shape Analysis of Latin Print

Elastic pixel and curve matching is often attributed to Burr [9], who traces it to Widrow's elastic templates. Burr gave an iterative algorithm to warp one contour towards another. The stiffness of the warped image is decreased gradually to take advantage of increasingly reliable local matches and correspondences. Elastic matching has been widely used for structural shape analysis of bilevel and gray-scale characters and line drawings.

Another generally useful method is orthogonal Zernike moments, which are translation, scale, and rotation invariant. Khotanzad and Hong [44] examined the interplay of the number of coefficients and noise using a small set of upper case letters, and both nearest-neighbors and minimum-distance classifiers. They show that Zernike features compare favorably with regular moments and with Hu moment-invariants.

N-tuples (judiciously selected configurations of N pixels) are attractive features for statistical classification of arbitrary, but rigid, character shapes. Among the billions of possible candidates, one can always find some N-tuple that fits one shape and does not fit another. Stentiford found 316 such features to classify digits and upper-case letters of postal addresses. His search algorithm, operating on about 20,000 training characters, attempts to maximize class-conditional independence [77]. Jung et al. [37] also tried to preserve independence, but focused on methods for finding N-tuples that fit the positive class and miss the negative class with some margin of safety. The complexity of the resulting search was mitigated by using very small training sets for document-specific classification.

Stringa [78], an early developer of postal readers, advocated syntactic methods. His bottom-up parsing procedure is based on phrase-structure grammars, where each transformation of a 2×2 primitive reduces the character size by a linear factor of two. Once the pattern is reduced sufficiently, it is classified by comparison to reference patterns. The method was designed for both print and block lettering and tested on 20,000 ZIP codes.

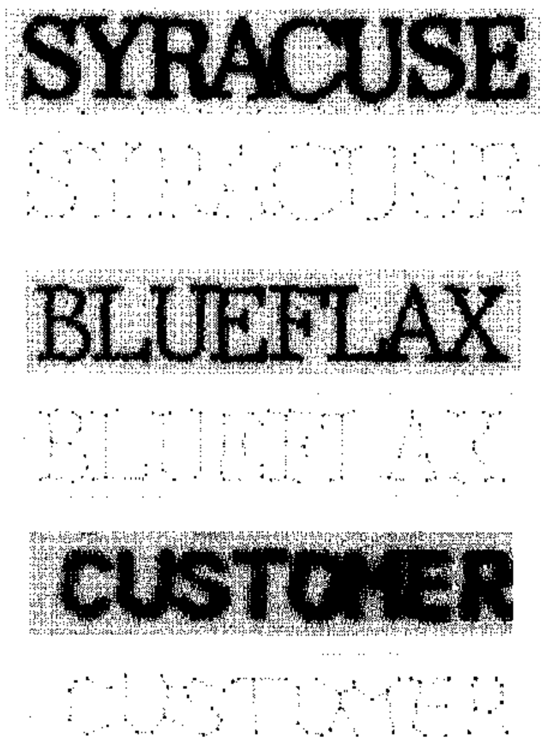


Fig. 4. Ridge graphs for segmentation of gray-scanned characters from a postal database [68].

In 1987, Kahan et al. [38] at Bell Laboratories published a landmark paper that combined many techniques for multifont page recognition that had been investigated hitherto only in isolation. Although they did not attain their goal of 99.9 percent accuracy with no rejects, they did reach 99 percent on single fonts, and also on top-three recognition for up to six mixed fonts of 12 point or larger type, scanned at 200 dpi. Their structural character representation, the Line Adjacency Graph (LAG), was used for thinning, traversing contours, and clustering to extract some 300 binary "stroke" features. A trainable Bayes classifier produced an ordered list of recognition candidates. Conjoined patterns with low posterior probabilities were segmented by vertical projection. Finally, the output text was assembled according to the most likely configuration suggested by the location of bounding boxes, plausible trigrams, Unix *spell*, and a few heuristic rules. In addition to their magnum opus' impact on the research community and perhaps also on emerging commercial multifont OCR, various components of the system continued to serve Bell Labs researchers and developers for many years. Baird convened and meticulously edited the proceedings of the 1990 Workshop on Syntactic and Structural Pattern Recognition that was devoted entirely to document image analysis.

During the current decade, neural networks continued to gain in popularity over parametric classifiers. Avi-Itzhak et al. [3] reported their results on characters gray-scanned at 400 dpi. We cite from their noteworthy conclusion. On 12-point Courier: *The neural network was trained with a database of 94 character images. The neural network was tested on a database of 1,072,000 character images and achieved perfect recognition.* On 12 common fonts in 8 to 12 point sizes: *The latter neural network was trained with 1,128 character images, and it achieved perfect recognition on a testing database of 347,712 multisize and multifont characters.* Section 3 explains that corrupted, unrecognizable character images, and indistinguishable pairs like I, l, and 1, were excluded from the error count.

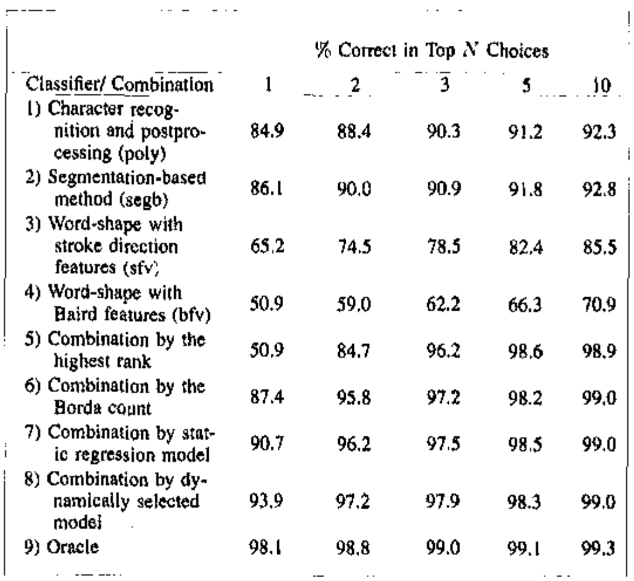
In spite of this unprecedented success with neural networks, in a follow-up paper, Avi-Itzhak and other colleagues [4] developed a template-construction scheme that produces an optimal template that maximizes the worst case (i.e., minimum) correlation with its subclass templates. If a single template is insufficient for perfect recognition, the class is split and additional templates are constructed. We cite again: *Next, the aggregate templates were used to recognize approximately 300,000 isolated characters having point sizes of 10, 12, and 14 and a uniform mixture of 12 commonly used fonts. The data was generated on one QMS-810 PostScript laser printer. No errors were incurred.* These papers may lower the curtain on research on isolated greenhouse characters printed specifically for testing recognition algorithms.

By 1993, Pavlidis turned his attention to structural, gray-scale, postal OCR. His team published three papers on the subject in *PAMI*. In the first paper [91], they developed a method for topographic analysis of the reflectance surface and for extracting features that approximate the ridge lines. These are assembled in a graph that can be matched to reference graphs for the recognition of degraded printed and hand-printed ZIP codes.

The next paper improved the comparison of the extracted shape features with the stored prototypes. *The major new contribution of this paper is the use of abstract definitions of characters for prototypes and reliance on conceptual models for variations and distortions to cover the large number of forms in which a character may appear,* [67]. Approximate graph matching measures were developed to minimize the cost of the transformations required to map a candidate graph into the corresponding prototype graph. The match must, of course, take into account geometry as well as topology. Testing the system on a 24,000 digit database of the USPS printed addresses yielded a correct recognition rate of over 98 percent. The resulting errors were classified to guide further research.

The last paper of this series extended the above method to *unsegmented* strings of gray-scanned characters [68]. The most reliably recognized characters are detected first, but overlapping, underlined, or broken characters can be recognized as imbedded subgraphs anywhere in the word. In the final phase, a search of the candidates' trellis finds the most consistent succession of characters. The system does as well without segmentation as the earlier system did with presegmented characters. With multiple prototypes for only a few classes, it also segments correctly over 90 percent of 12,000 upper-case words, and recognizes 94 percent of the characters, from another USPS database (Fig. 4). This is a remarkable result without exploiting any linguistic context.

Like the above, the research described in the next paper also benefits both character and word recognition, but its



	% Correct in Top N Choices				
Classifier/ Combination	1	2	3	5	10
1) Character recognition and postprocessing (poly)	84.9	88.4	90.3	91.2	92.3
2) Segmentation-based method (segb)	86.1	90.0	90.9	91.8	92.8
3) Word-shape with stroke direction features (sfv)	65.2	74.5	78.5	82.4	85.5
4) Word-shape with Baird features (bfv)	50.9	59.0	62.2	66.3	70.9
5) Combination by the highest rank	50.9	84.7	96.2	98.6	98.9
6) Combination by the Borda count	87.4	95.8	97.2	98.2	99.0
7) Combination by static regression model	90.7	96.2	97.5	98.5	99.0
8) Combination by dynamically selected model	93.9	97.2	97.9	98.3	99.0
9) Oracle	98.1	98.8	99.0	99.1	99.3

Fig. 5. Comparison of results by individual classifier and their combinations [26].

emphasis is on classifier combination. From the point of view of classifier combination, the only difference between character and word recognition is the number of classes. The classifiers can be based on the same or on different features. It is convenient to use only the ranked lists of candidates produced by the classifiers because their confidence and similarity measures may be incommensurable. The errors produced by different classifiers do, of course, tend to be correlated (if they were independent, three cooperating classifiers would solve most recognition problems). Nevertheless, classifier combination can produce a substantial reduction even in already low error rates.

Ho et al. [26] examined prior proposals for classifier combination and developed methods based on the Borda Count, Logistic Regression, and Dynamic Selection. They also explored the identification of redundant classifiers. Their results in assigning 1,384 word images to one of 1,365 lexicon entries, using up to four classifiers, are shown in Fig. 5. For top choice, the 13.9 percent error rate of the best single classifier is reduced to 6.1 percent for the combo. The improvement at the lower error rates reported for top-N is even more dramatic, approaching that of an *oracle* that could select the best classifier for each sample.

The conclusion of this paper anticipates Ho's thought-provoking research on combinations of weak classifiers and on decision forests, but we now turn to her attempt with Baird [27] to numerically evaluate the asymptotic classification error on characters generated by Baird's ten-parameter pseudorandom defect model. This character generator exhibits a strong central tendency, i.e., a relatively small fraction of the bitmaps accounts for a large fraction of the samples. In their c/e discrimination task, the Bayes error was due exclusively to bitmaps that sometimes carry the label "c," sometimes "e."

Ho and Baird observed that the pessimistic Holdout Estimate and the optimistic Resubstitution Estimate converged very slowly even after hundreds of thousands of training samples of c's and e's, and even for small point-sizes with fewer possible bitmaps. For small patterns, the Bayes error is larger because there are more identical bitmaps from the two classes. This may also explain why e's are more often mistaken for c's than vice versa. (Sparse bitmaps are generated from either class, but more often from "c," and are therefore labeled "c" in the training set. Therefore, every similar "e" in the test set will be misclassified).

Ho and Baird compared three classifiers (nearest neighbors, decision trees, and distribution maps) trained on up to 300,000 patterns. They found no statistically significant difference between the error rates when the training sets are larger than 100,000 samples. Their unoptimized nearest neighbors classifier was, of course, hideously slow with so many prototypes. Aside from the fascination of statistically-homogenous megasample training and test sets, their most interesting conclusion was that the defect parameters that affect classification most are blurring and thresholding.

Baird's and others' pseudorandom defect models continued to attract interest but it proved difficult to set their parameters to emulate real data. Li et al. [52] focused on the difference between the patterns of classifier errors on real and synthetic data. They proposed four measures of the error patterns that are sensitive to differences between data sets. Here, as in all other OCR research, it is essential to obtain accurate *ground truth* (the ugly, but accepted term for the true labels). Kanungo and Haralick [41] proposed a geometric registration method for aligning the OCR labels on a page printed from a file of text. The method produces exact character locations and allows for degradation through copying and scanning the printout, but unlike string-matching techniques, it applies only to synthetically generated (computer-set) pages.

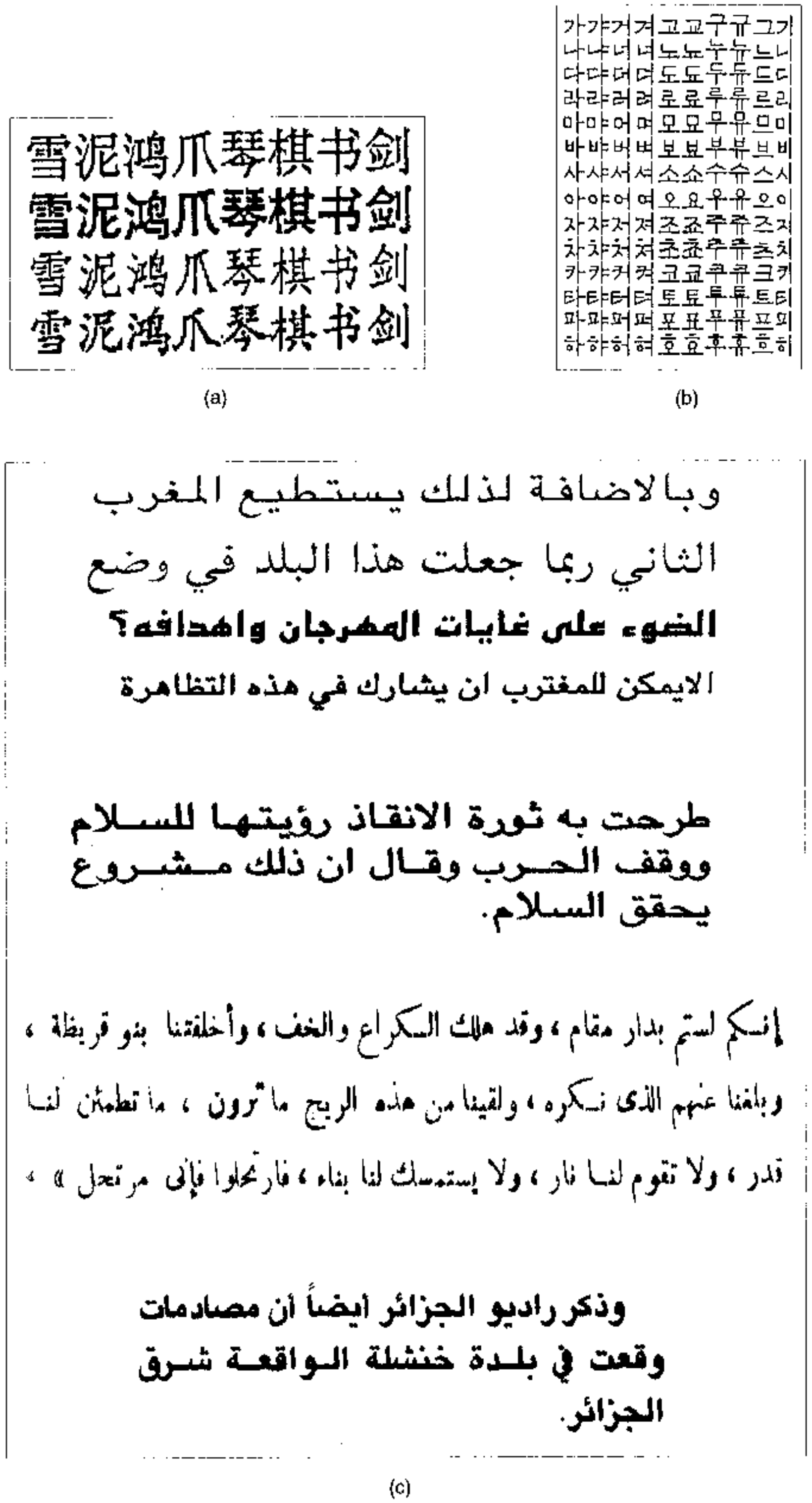

雪泥鸿爪琴棋书剑
雪泥鸿爪琴棋书剑
雪泥鸿爪琴棋书剑
雪泥鸿爪琴棋书剑
(a)
가갸거겨고교구규그기
나냐너녀노뇨누뉴느니
다댜더뎌도됴두듀드디
라랴러려로료루류르리
마먀머며모묘무뮤므미
바뱌버벼보뵤부뷰브비
사샤서셔소쇼수슈스시
아야어여오요우유으이
자쟈저져조죠주쥬즈지
차챠처쳐초쵸추츄츠치
카캬커켜코쿄쿠큐크키
타탸터텨토툐투튜트티
파퍄퍼펴포표푸퓨프피
하햐허혀호효후휴흐히
(b)
وبالاضافة لذلك يستطيع المغرب
الثاني ربما جعلت هذا البلد في وضع
الضوء على غايات المهرجان واهدافه؟
الايمكن للمغترب ان يشارك في هذه التظاهرة
طرحت به ثورة الانقاذ رؤيتها للسلام
ووقف الحرب وقال ان ذلك مشروع
يحقق السلام.
إنكم لستم بدار مقام ، وقد هلك الكراع والخف ، وأخلفتنا بنو قريظة ،
وبلغنا عنهم الذى نكره ، ولقينا من هذه الريح ما ترون ، ما تطمئن لنا
قدر ، ولا تقوم لنا نار ، ولا يستمسك لنا بناء ، فارتحلوا فإني مرتحل »
وذكر راديو الجزائر أيضاً أن مصادمات
وقعت في بلدة خنشلة الواقعة شرق
الجزائر.
(c)

Fig. 6. Oriental fonts (a) Chinese [30]. (b) Korean [58]. (c) Arabic [6].

3.2 Shape Analysis of Oriental Characters

The classification of Han characters differs from that of Latin characters by the enormous number and complex structure of ideographs. Mercifully, fewer fonts are used commonly than in Western languages (Fig. 6). Hangul has a syllabic script. As many as seven of its 24 phonetic symbols can be squeezed into a square character frame in thousands of ways. Arabic script is alphabetic, but its recognition is complicated by the changes in character shape depending on position within a word, and by the presence of secondaries (dots and zigzags). Partly because Arabic script is used in so many countries, there are many, many fonts. Aside from its intrinsic difficulty, for many years research on all these scripts was hampered by the lack of standardization of the corresponding character codes.

Structural approaches are appealing given the hierarchical structure of Chinese characters. This structure can be represented by constraint graphs [cf. 56] that link each of the thousands of characters with the 400 basic components and 20 essential strokes. Huang et al. [30] performed the graph matching for classification with a cooperative relaxation algorithm that simulates force-driven elastic matching. On "numerous large sets of Chinese characters" of over 100,000 samples in nine fonts and four sizes, scanned at 300 dpi, they achieved 95-99 percent recognition on nearly 4,000 classes (including 96 non-Chinese symbols).

Building on another useful approach to many-category classification, Suen's team progressively improved decision tree design [23], [88], [89]. The first version selects the best Walsh coefficient at each node; the second one concentrates on analysis of tree classifiers using progressive entropy reduction; the third applies clustering to minimize the class overlap between nodes and fuzzy membership to limit the accumulation of error on each path. Once the tree is designed for a training set, global training is used to select features and set thresholds at each terminal node. The classification was tested on samples produced from 3,200 characters using a noise model.

The Korean script was designed on royal order by a scholar and so might be expected to be particularly amenable to a syntactic approach. It can also be argued that Korean characters exhibit so little redundancy that syntactic analysis is more robust than feature-based statistical methods. Lee et al. [50] staked their method on this claim, but see also Nagahashi and Nakatsuyama [58]. The syntactic primitives are thinned line segments. A set of 377 production rules were derived by inspection and trial-and-error from a training set of over 4,000 characters! The programmed grammar of stroke sequences has attributes of stroke angle, length, and connectivity, and includes switches to the next production. On markedly different data, the recognition accuracy was 95-98 percent. The success of this method is reminiscent of that of the hand-crafted rules in the early commercial OCR devices. This is a very readable paper, with a good description of the script and an excellent simplified example.

To recognize isolated Arabic print, Al-Yousefi and Udpa [2] separated the external secondaries (dots and zigzags) by a structural approach based on projections, then applied a quadratic Bayes classifier to moment features derived from the same projections. They call attention to the aspects of Arabic scripts that preclude emulating methods devised for Latin characters. (The essentially connected nature of Arabic print and the position dependence of the symbols also undermine the value of research on isolated specimens.)

3.3 Linguistic Context

Character N-grams have been used for contextual OCR error correction since the mid-sixties, when large lexicons could not yet be stored, but the first large-scale computational study of their relevant characteristics was Suen's [79]. Suen tabulated word-lengths, the growth in the number of distinct n-grams as a function of vocabulary size, their word-positional dependence, and the influence of the selected corpus on n-gram statistics. He advocated taking into consideration the "context" of the text, such as "novel, proper names, news items, computer programs, etc." The entropy of N-grams ($N \leq 5$) of English dictionary words (without regard to word frequency) was later studied from an information-theoretic point of view [96].

Shinghal and Toussaint [72], [73] conducted extensive studies on the application of Markovian assumptions to Bayesian classification. They popularized the Viterbi algorithm for finding the highest-probability path through the word-trellis of recognition candidates and introduced modifications to accelerate the computation. They also investigated the sensitivity of the results to the source statistics, and the appropriate application of the confusion probabilities of a given classifier. Hull and Srihari [31] bridge lexical and morphological (N-gram) error correction by quantizing N-gram frequencies to 1 or 0 depending on whether the N-gram occurs in a lexicon of valid words. Alternatives for combining the two approaches were compared in [32]. Here the Viterbi algorithm was used to produce the most likely outcome, constrained by the presence of the output word in a lexicon. Srihari's 1985 Computer Society tutorial, *Computer Text Recognition and Error Correction*, remains a valuable reference on this topic. Hull was co-editor of *Document Analysis Systems II*.

A serious effort to extend hybrid contextual correction to real-life wide-vocabulary documents that may contain abbreviations and proper nouns that do not appear in the lexicon, and to missegmented classifier output, was presented in [75]. The most likely N-gram based word is augmented by candidates (*word-hypotheses*) within a small string-edit distance, thus providing additional lexical candidates and allowing for the possibility of missegmentation. Special handlers are developed for numerals, punctuation, all-cap words, etc. The final hypothesis selection is based on a cost model that takes into account the confusion matrix, the lexicon, trigram frequencies, as well as transient information collected from words reliably recognized elsewhere in the same document.

A word-recognition method based on Left-Right-Top-Bottom (almost 2D) Hidden Markov Model (HMM) was proposed for spotting keywords on poorly scanned pages [48]. On a test of 26,000 words, 99 percent accuracy was achieved on the same font size, but the recognition deteriorated on variable-sized fonts.

Striving to eliminate completely character-level segmentation errors, Hull [33] recognized entire sentences using a probabilistic parts-of-speech grammar. The recognition primitives here are clustered word shapes (a group of 10 visually similar words is called a *neighborhood*.) The procedure is similar to that of a HMM that links word candidates with the part-of-speech transition probabilities.

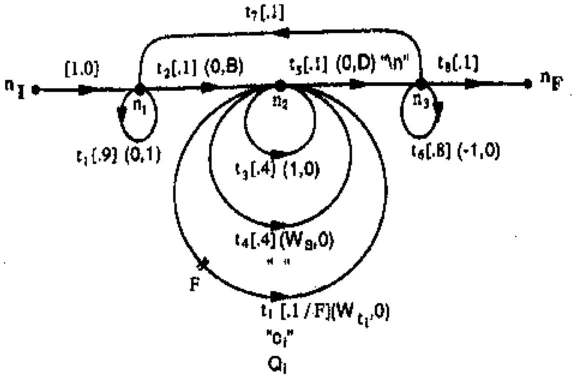


Fig. 7. Document Image Decoding [45]. In this simplified model of a text column, only the "F" transitions correspond to inked characters. The other transitions are horizontal and mertical moves. The transition probabilities are in brackets.

The experimental results on simulated data generated from a coded corpus of text support the inclusion of probabilistic grammars in the contextual OCR repertory. Such grammars can be generated for any language and sphere of discourse with a computer-readable corpus of text.

In a refreshing departure from natural language context, Baird and Thompson [5], taking advantage of the tight constraints on chess text, built a system that transcribed 142 complete games from the poorly-printed *Chess Informant*. They interpreted all but three of the games correctly, corresponding to 99.99 percent accuracy on moves and to 99.995 percent on characters. The base shape analysis was *only* 99.5 percent correct. The context here includes the permissible combinations of symbols (*syntax*) and the permissible sequences of moves (*semantics*). By luck, the semantic analyzer was already available from Belle, the computer chess champion. This paper represents a wonderful example of the power of context in the right context.

The recognition methods described so far are all based on some preconception of the character shapes. It is, however, possible to cluster well-formed character bitmaps without any such preconception or training. The resulting 26 cluster labels can then be replaced with the corresponding alphabetic labels by solving the substitution cipher generated by the reading order of the cluster labels [59]. The Achilles' heel of this scheme is "imperfect" clustering, which may separate equivalent forms of the same class (E & e), merge different classes (c & e), and create nonconforming clusters *(iconoclasts)* of touching and broken characters, digits, and punctuation.

Recently, BBN researchers drew on their HMM tools for speech and handwritten-character recognition to develop a multifont reader with language-independent algorithms and shape features, and language-dependent orthographic rules, character models, lexicons, and grammars [6]. The orthographic rules specify the orientation and reading order of the script. The character models are 14-state transition graphs of feature vectors extracted from narrow overlapping bars *(frames)* orthogonal to a text line. The word models are concatenated character models. The language model *(grammar)* consists of word-bigram frequencies derived independently from the scanned training set.

Recognition is based on multimodal Gaussian distributions for the feature vectors. It is assumed that different modes correspond to different *styles* of print. However, the formulation appears to estimate the style of each frame separately instead of conditioning the entire character on the style. This results in an unexpected exponential in the estimated style probabilities, which surfaces in experiments on italic and plain fonts. It is compensated by Bazzi et al. [6] by manipulating the proportions of different styles in the training data.

The system was tested on large test sets from the University of Washington, English Document Image Database and from the DARPA Arabic OCR Corpus. The results indicated that using a 30,000-word English lexicon and word-bigram frequencies reduces the error rate by about a factor of three over use of character bigrams and trigrams alone (called *open vocabulary recognition)*. Combining them gives almost another factor of two. The error rate on English was about three times lower than on Arabic. The authors speculate that this was caused by the intrinsic similarity of some Arabic characters, their connectedness and ligatures, wider font diversity, and lower print quality.

Now, we mention three papers that developed techniques for other purposes but may also have applications in contextual document processing. Tanaka and Kojima [81] presented a fast, clever, multistage string-correction algorithm based on hierarchical files. Oommen [65] improved the computational time and space requirements of constrained edit distances for recognizing noisy subsequences. The constraints favor particular edit operations. He demonstrated the algorithm on long, corrupted strings from Duda and Hart's immortal *Pattern Classification and Scene Analysis*. Wang and Pavlidis [90] found the optimal correspondences to code words of string subsequences. They formulated the string-to-regular expression comparison with a large alphabet and applied their algorithm to the continuously-variable widths of individual bars to decode scanned Universal Product Codes (UPC).

3.4 Document Image Decoding

A radical paradigm shift that unifies some aspects of preprocessing (e.g., line finding), layout analysis, and character recognition, was Kopec and Chou's [45] *Document Image Decoding. DID* draws, on one hand, on communications theory and on successful applications of HMM to speech recognition and, on the other, on the side-bearing model of character placement and on simple models of noise for printed pages. They model each column of text as a Markov source (Fig. 7) whose transitions generate character placements, white spaces, line feeds, carriage returns, and character bitmaps that are degraded by printing and scanning ("channel noise"). The decoding process, based on dynamic programming, attempts to identify the most likely sequence of transitions from the observed pixels. The required input consists of character templates that produce nonoverlapping character bitmaps, transition probabilities of character classes, and the frequencies of transitions that govern the layout.

A model with 1,700 nodes and over 6,000 branches was run on 48 columns from ten pages from the Yellow Pages in about 36 hours (on a SPARCstation10). The error rate on the

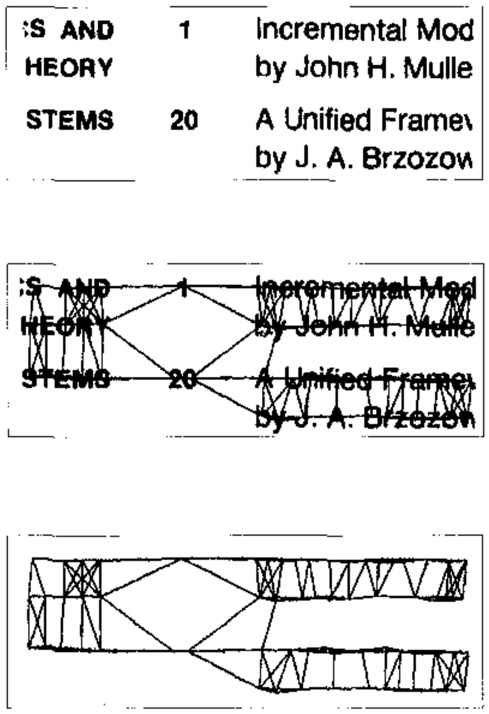


Fig. 8. The elements of the Document Spectrum are the five vectors that link each connected component to its nearest neighbor. The Docstrum is the two-dimensional distribution of the vectors' length and orientation [63].

listings was less than 2 percent for names, and less than 0.5 percent for telephone numbers. For this task, the font metrics and sample characters were extracted from training pages with a bitmap editor (which was a noteworthy contribution in its own right). The character templates were obtained by manual labeling of bitmap clusters. The asymmetric bit-flip channel parameter was estimated from the variation among the glyphs in a cluster.

Within a year or so, Kam and Kopec [39] speeded up their reader by up to 25 times, depending on the regularity of the layout. The key was factoring the model into separate Markov processes for horizontal and vertical traversals. Heuristically bounding the contribution of unlikely paths in the Viterbi trellis allows concentrating the processing for each line of text along its baseline and, therefore, on a single chain of character-to-character transitions. The heuristics that allow truncating unpromising paths exploit the consistency of black pixel distributions at different heights above the baseline, the strong vertical stroke structure, and the correlation between neighboring pixels at 300 dpi.

The next contribution was an improved method of template generation from labeled samples of text [46]. It is an iterative waltz: 1) estimate glyph alignment with the current parameters; 2) refine the template estimates by aligning and thresholding the glyphs of each class; 3) refine the channel parameter estimates. The initial template estimates are composite bitmaps generated from four PostScript fonts. The results compared favorably with those of TextBridge in a test of over a million characters of body-text, especially when a few of the templates were manually edited. Kopec and Lomelin [46] found that using incorrect TextBridge labels, instead of the ground truth, for supervised template estimation required additional template editing. This is the only comparison that we found in *PAMI* between a research classifier and a commercial (albeit in-house) classifier.

4 Page Decomposition

As mentioned in Section 1, page decomposition can be divided according to the nature of the extracted structure. To facilitate OCR, it is necessary only to separate text from nontext. Further processing is facilitated by partitioning text into columns and paragraphs, nontext into line art and halftones, and demarcating tabular regions. Beyond this point, there is a choice between detailed modeling for identifying logical and functional components through typographic attributes, and delaying such analysis until the OCR results are available.

All of the work on physical decomposition that we found, and report in Section 4.1, was published since 1993 and addresses technical journals. Two papers in Section 4.2 are about logical decomposition, i.e., identifying document components for content-based processing. They analyze technical articles and mail sorting. Section 4.3 covers tables and forms (and so does the first paragraph of Section 4.1). We consider *tables* as a means of presenting information, and *forms* as a means of collecting information [21]. They can be distinguished according to whether the cells are printed at the same time as, or before, the cell contents. We group them together because forms often have a tabular layout; therefore, many processing steps are similar.

4.1 Physical or Geometric Layout

The table of contents of technical journals exhibit a pleasing variety of layouts and challenging configurations of title-author-page associations. A converted machine-vision buff developed a robust, rotation-invariant, data-driven method for extracting complex page layouts including tables of content (TOCs) [63]. Document spectrum *(docstrum)* analysis clusters the connected components by a nearest-neighbors algorithm with an orientation-sensitive metric (Fig. 8). Most of the required parameters are estimated adaptively from size histograms on the page. Docstrum was successfully applied to thousands of TOCs at the Bell Laboratories Libraries' *RightPages* information diffusion system. Subsequently, O'Gorman and Kasturi co-authored the influential and reference-studded *Document Image Analysis*.

Multiscale segmentation methods that are popular in machine-vision can also be used to partition a gray-scanned page into *text, image,* and *graphics* regions [19]. Rotation-invariant features at different scales are derived from wavelet packets. Neural networks are used to differentiate the textures. Neighboring blocks are fuzzily combined by weighted voting within and across scales. Because texture or frequency based layout analysis often results in nested labels, model-based postprocessing was required to recover the rectilinear layout of the page.

An intuitive and quite general model of the rectilinear physical layout of technical journals led to efficient bottom-up extraction of the text structure [74]. The reading order is found by traversing a minimum spanning tree where the edge lengths are the larger of the horizontal and vertical distances between the bounding boxes of connected components. The subtrees of the tree correspond to physical layout units. Although the model assumes horizontal text lines, it tolerates up to 5 degrees of skew. With the addition of a few heuristics, a test on 98 pages from three journals of chemistry required manual correction of only seven pages.

After extensive experimentation with Gabor texture filters implemented as convolution masks on gray-scale document images, Jain and Yu [34] developed a much faster bottom-up method to partition a page into columns of *text, drawings, images, table regions,* and *rulers.* Following the application of a hierarchical Hough transform to connected components, the estimated skew is accommodated by introducing generalized text lines. Foreground pixels are grouped into rectangular blocks with adjacent same-length horizontal runs preserved as nodes in a Block Adjacency Graph (BAG); see [38] for a similar idea. The BAG nodes are successively grouped into connected components, text lines, and region blocks, and simultaneously inserted into a novel page model. The segmented regions are classified using empirical rules and thresholds. The model is a tree where the branches of the root correspond to the five types of entities, and the leaves are the BAG nodes. Selected results from performance tests on 150 varied page images are illustrated in color. The paper contains an informed analysis and a thorough literature review of geometric and logical page decomposition.

All commercial OCR systems provide the option of manual or automatic zoning (the decomposition of the page into text and nontext regions). The worst error made by automatic zoning is *decolumnization,* which fuses adjacent columns and requires very unpleasant postediting. In the context of the OCR evaluation conducted at the Information Science Research Institute, Kanai et al. [40] proposed a method for evaluating zoning accuracy according to a string-edit metric between the OCR outputs on automatically and manually zoned pages. Other methods for zoning evaluation are based on geometric comparison of the zone coordinates or of the assignment of individual pixels to the two classes.

4.2 Logical or Functional Layout

Krishnamoorthy et al. [47] combined the nested X-Y tree decomposition of rectangles into rectangles with recursive horizontal and vertical application of a publication-specific "block grammar" to determine the major logical components of technical articles. This model-driven approach isolates specific document components for selective OCR. The effort required to construct the grammars in lex/yacc notation is a drawback, as is skew sensitivity.

Although Cohen et al. [14] developed the following process for handwritten addresses, it appears equally applicable to print. Each part of the process is splendidly illustrated in the paper. The system (euphoniously called *HWAIS*) segments the address into lines and words (a nontrivial task) and parses the gray-scale, 300 dpi address, according to the permissible sequences of *P.O. Box, street number, street, city, state,* and *5 or 9 digit zip-code.* Abbreviations are expanded. The confidently recognized fields are used to assist the recognition of the remaining fields through postal zip-code, street, and street number directories. The final objective is to assign the eleven-digit delivery point code (DPC). Postal recognition systems are set to run at a low error rate, and route rejects to a data entry terminal.

In some applications, document classification is a necessary precursor of interpretation. A distance measure based on a machine-learning approach was thoroughly analyzed in [18]. It measures the difference between two structural descriptions (expressed as well-formed formulas in Michalski's extension of the predicate calculus for inductive inference) in terms of substitutions leading to the most general unifier. Esposito et al. [18] mention, and describe in detail elsewhere, the application of this recondite formulation to the classification of 35 document images into seven classes using primitives like width, height, and relative position.

4.3 Tables and Forms

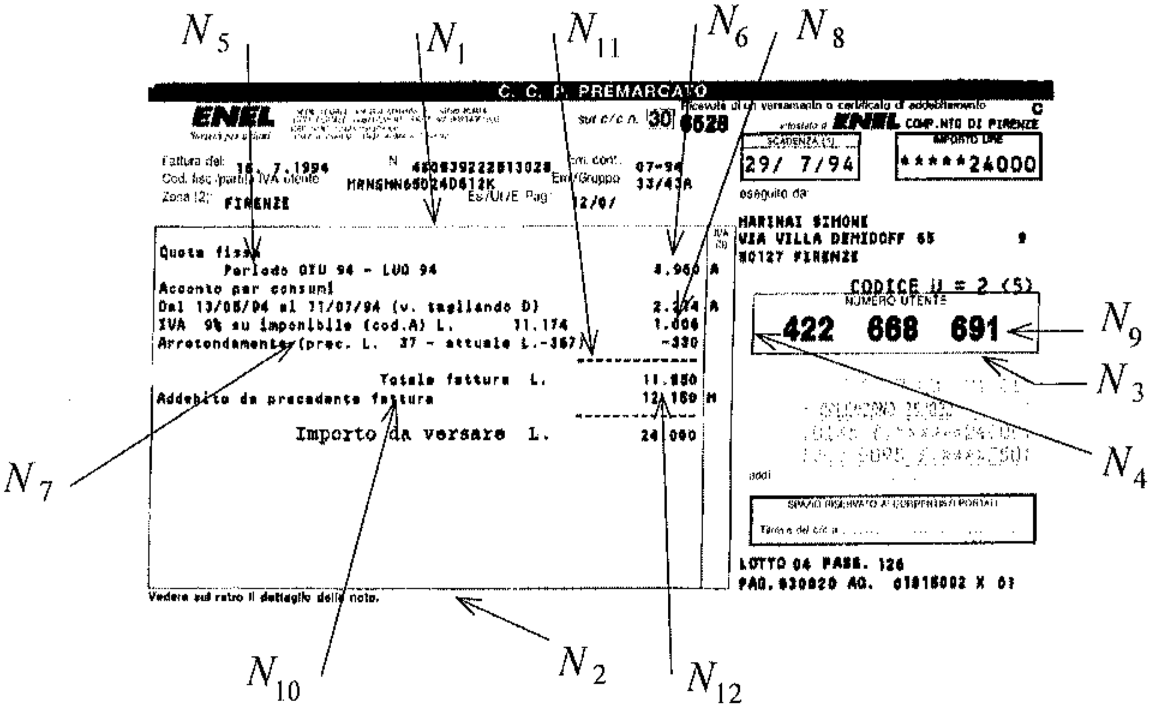
Ruled forms and tables lie in some sense between our categories of text and graphic documents. They are a latecomer to *PAMI* (in fact, none of the papers below address *tables*), but over one hundred papers on the subject have been published.

Starting out before 1990, Watanabe and his colleagues [92] developed powerful methods for analyzing and classifying many different ruled layouts (primarily Japanese business forms). Their representation is a classification tree, each leaf of which points to table-specific global and local structure trees. The classification tree is used to determine whether an incoming form is one of the current set; if not, a new leaf is sprouted. The nodes of the structure trees are single and repeating blocks. There are very few restrictions on accepted layouts. The most exciting aspect of this formulation is dynamic table-knowledge acquisition.

Yu and Jain [97] applied BAGs (see above) to the extraction of form frames and preprinted data. Templates were constructed from empty forms and correlated with filled-in forms. Once the form was registered, they first extracted the prevalent horizontal rulings, then traced them to the vertical rulings. The strokes of characters that overlap the frame are reconstructed after removal of the line.

A data rather than model driven method to remove rulings was presented in [82]. The novelty of the paper was not so much the removal of long horizontal and vertical lines, which is done routinely by a variety of methods in commercial forms processing, but that it was done with two-dimensional multiresolution wavelet analysis (MRA). Experimental results were demonstrated on three financial forms. According to the authors, "They indicate superb performance of the proposed new approach for document processing."

Informys *(Invoice-like form-reader system)* is a high-level approach to form data extraction [12]. The system processes rulings, logos, instruction fields, and information fields. These are represented by nodes of attributed graphs whose arcs correspond to the relative positions of linked items. Therefore an information field can be identified not only if it is in a fixed location, but also by its position relative to a


N_5
N_1
N_{11}
N_6
N_8
N_9
N_3
N_4
N_7
N_{10}
N_2
N_{12}

C. C. P. PREMARCATO

ENEL

Fattura del: 16. 7.1994 N 46083922281302B Cod. fisc./partita IVA utente MRNSMN65D24D612K Zona (2): FIRENZE Em. cont. 07-94 Em/Gruppo 33/43A Es/Ut/E Pag: 12/01

sul c/c n. 30 6528

Quota fissa
Periodo GIU 94 - LUG 94 4.960 A
Acconto per consumi
Dal 13/05/94 al 11/07/94 (v. tagliando D) 2.214 A
IVA 9% su imponibile (cod.A) L. 11.174 1.006
Arrotondamento (prec. L. 37 - attuale L.-367) -330

Totale fattura L. 11.850
Addebito da precedente fattura 12.150 M

Importo da versare L. 24.000

Vedere sul retro il dettaglio delle note.

Ricevuta di un versamento o certificato di addebitamento C
intestato a ENEL COMP.NTO DI FIRENZE

SCADENZA (1) 29/ 7/94
IMPORTO LIRE *****24000

eseguito da:
MARINAI SIMONE
VIA VILLA DEMIDOFF 65
50127 FIRENZE

CODICE U = 2 (5)
NUMERO UTENTE
422 668 691

addi

SPAZIO RISERVATO AI CORRENTISTI POSTALI
Timbro del c/c n. ...

LOTTO 04 PASS. 126
PAG.030020 AG. 01815002 X 01

Fig. 9. Invoice form showing features used as fiducial marks [12].

fixed reference object, or by its relative position with respect to *two* unfixed reference objects. The reference objects, like logos and instruction fields, are recognized with neural networks. The text is recognized by whole-word recognition and by string-matching against the model text (after character recognition). The model for each form is constructed with an interactive system. The system was tested on almost three hundred invoices (Fig. 9) with few errors. The article contains a short analysis of alternative approaches to forms processing.

5 Graphics Recognition

The papers in the first subsection address operations on pixels, runs, and line segments. The second subsection pans a wider range of abstraction from graphic symbols to netlists and three-dimensional interpretation.

5.1 Primitives Extraction

Di Zenzo et al. [15] presented their formal analysis of the GR *(Graph Representation)* of a bilevel image long after applying it to the large-scale conversion of land register maps. The GR is a general-purpose structure of adjacent foreground runs similar to the LAG and the BAG. It can be used for extracting connected components, Euler numbers, diameters, convex hulls, concavities, and convexities. It is also an efficient data structure for vectorization.

Because of the large size of maps, there is a premium on speed. Yamada et al. [95] developed a set of multiangled parallel (MAP) anisotropic neighborhood operations, that they called directional morphology, for cartographic features. They extracted linear features and hatched areas using a VITec image processor attached to a SUN. The main contribution of this paper was the extension of the multiplane computational framework to symbol matching.

The fundamental problem in line art analysis is accurate, fast vectorization. Since most drawings consist of an overwhelming majority of background pixels and long chains of connected foreground pixels, the trick is to avoid processing all the background pixels while recovering the continuity of the foreground pixels. Most vectorization algorithms approximate curved lines by sequences of short straight-line segments, but that only postpones interpretation. Dori [16] applied to arc segmentation the orthogonal zig-zag vectorization (OZZ) algorithm that he had developed a few years earlier. The basic zig-zag algorithm finds a chain of "bars." Arc centers are hypothesized at the intersections of the perpendicular bisectors of straight-line segments with monotonically changing slopes. More complex pixel-level processing is required to detect circles, locate accurately the center and end points of circular arcs, maintain continuity with tangent lines, and alleviate the effects of small holes and islands.

Further experimentation evidently convinced Dori's group to base arc segmentation on fully vectorized images. The resulting incremental method [93], that progressively checks the cocircularity of clustered arc fragments, is faster than the above hybrid method. It is also very accurate, as demonstrated both on a multitude of synthetically generated bitmaps and on a "host" of real-life drawings including dashed arcs. The algorithm was reported to work well on arc segments greater than 10 pixels in radius, $\pi/4$ in angle, and one pixel in width.

Dori also revisited OZZ and turned it into the Sparse Pixel Vectorization (SPV) algorithm [17]. SPV approximates arc fragments more accurately and visits only a subset of points on the medial axis of a segment. The tracing bounces from the edge of a strip to its center and, hence, to an edge further along the curve. Points between the detected center points are interpolated. Line width is preserved. Special provisions are, of course, necessary for line endings, junctions, and intersections. Several quality measures were developed and applied to vectorization of complex drawings from different sources. As expected, the algorithms performed well. It takes 3 to 5 seconds for a $1,000 \times 1,000$ pixel drawing with moderate line densities. The code for Sun Solaris is available by ftp. This article contains an informative discussion of vectorization and of the pros and cons of alternative approaches.

The separation of text from graphics has been challenging researchers for many years. A recent communication in *PAMI* [54] proposed extracting the text by erasing all the pixels that are part of long linear components, are in regions of low stroke density, belong to large dense configurations, or are not adjacent to a string of similar components. This is accomplished in a series of steps that draw on the recognition of already identified components and gradually isolate the text. The method was demonstrated on many examples, some with mixed Chinese/Latin/Greek/Arabic lettering or digits. Although most of the text was extracted successfully, it is marred by graphics residuals that would impede recognition.

5.2 Drawing Interpretation

In 1983, Bunke [8] extended the notion of attributed programmed string grammars to graph grammars. Programmed grammars control the order in which productions are applied. An attribute can be the *location* of a vertex or the *length* of a line segment. The graph grammar acts as a generator to transform the graph representation of an input drawing to the graph representation of its description. The method was applied to schematic diagrams. Extensions were presented, without any direct application to graphic documents, in [55]. In addition to his influential work in DIA, Bunke hosted and edited the 1992 *SSPR*, and was coeditor of the *Handbook of Character Recognition and Document Image Analysis.*

Sometime before 1983, researchers at the Toshiba Research and Development Center reached the remarkable conclusion that drawing logic circuit schematics carefully on paper, scanning them on a drum scanner, recognizing them using a special-purpose computer, and correcting the residual errors with an interactive computer-aided drafting/design system, would be about twice as fast as entering and correcting them with CAD tools alone. Ozaki and his colleagues [66] developed the required automatic drawing interpretation system. Their article, which is one of the few detailed descriptions of an operational graphics-recognition system, has been widely cited. Even now, it represents an endeavor far beyond the scope of most academic researchers.

Ozaki et al. [66] focused on the recognition of loop symbols that occur by the hundreds in logic diagrams. Their decision-tree-controlled approach combined connected component analysis, thinning, filtering, line tracing, multiple clipping-windows, 2×2 masks, geometric and topological features, and digital templates derived from drafting templates. They distinguished 93 classes of symbols divided into 14 subcategories. The image processor was implemented with special-purpose, pipelined, multiprocessor hardware (this was probably initiated about 1980; the same issue of *PAMI* contains two articles about parallel image processing computers of the same vintage). An interactive decision-tree editor provided the required support for adding new symbols and characters to the recognition logic. Character recognition was accomplished with attributed string matching developed earlier.

The system was tested in the development of 79 gate-array circuits on over 850 drawings of up to JIS A1 size (594 mm x 841 mm), gray-scanned at 250 dpi. The 10 participating draftsmen were held to strict drawing standards. The symbols were recognized with less than 1 percent error and less than 3 percent reject. Processing A1 drawings took 30 minutes, of which half was taken for symbol and character recognition. One cannot help wondering how much the trade-off between interactive graphics entry and automated drawing recognition has been affected by their relative progress in the intervening years.

At about the same time, Kasturi, at Pennsylvania State University, began to systematically tackle the various subtasks of graphics conversion. With Fletcher, he extracted strings of text from line drawings [20]. Their method is based on connected component analysis followed by the application of the Hough transform to find isolated collinear blobs of the size predicted by area-histogram analysis of all the components. The paper contains a listing of the requirements on the size and spacing of the characters for robust text extraction. (This paper really belongs in Section 5.1, but is included here for continuity.)

In 1990, with a large group of MS students, Kasturi [42] developed the tools necessary to identify outline and solid polygonal objects and their spatial containment hierarchy, circular arc segments, hatched areas, dashed lines, connectors between objects, and text strings (including phrases and words touching or overlapping the line art). They exploited many of the same graphics techniques as Ozaki et al., including connected components, collinear component grouping, thinning, boundary tracking, loop analysis. The image-specific parameters and thresholds that would have to be modified for different applications are tabulated. A test image developed for tuning some of the algorithms (Fig. 10) found application as a benchmark.

The next correspondence, with Lai, addressed an important component of mechanical drawings, the dimension sets. This was also the target of Dori's early research, published elsewhere. ANSI dimensions have a complex syntax for arrowheads and tails, leaders, datum markers and feature control frames. They are normally either horizontal or vertical, and although the leader pairs may cross other graphic components, the lettering is isolated. To avoid confusion with dashed lines and to take advantage of the isothetic orientation, Lai and Kasturi [49] abandoned the Hough transform and resorted to tracing the text strings from small connected components suspected of characterhood.

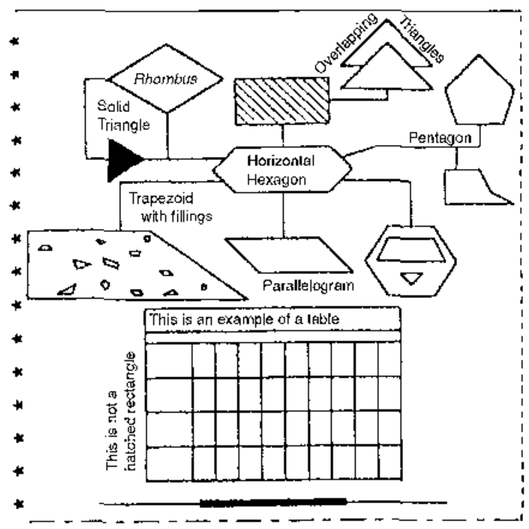


Fig. 10. Test Image 1 for graphics recognition [42].

A large (20,000 lines of C code) and principled experimental system modeled on the human perception cycle is presented in [35]. The incremental analysis strategy is imbedded in 191 yacc rules that generate a 313-state parser that controls a set of low-level image processing routines. A set of schema classes describes solid, dashed, and chained lines, solid and dashed curves, cross hatching, physical outlines, text, witness and leader lines, and certain forms of dimensioning. Each schema maintains geometric and structural descriptions of the entity it represents. Tracing the drawing can be initiated automatically and refined with interactively specified start points and radii. The article is concerned more with methodology than performance. Like Kasturi's, the system had no provision for text recognition, but it could reconstruct (render) the recognized data structure for comparison with the original. Joseph and Pridmore [35] pointed out that evaluating the system by comparing its output with the "correct" CAD file would be difficult because of the many equivalent CAD models of a drawing.

The focus of MARCO is map retrieval by content using database techniques [69]. MAGELLAN (Map Acquisition of Geographic Labels by Legend Analysis) is the front-end image processing system for MARCO. Samet and Soffer used Khoros (an image-processing software development system) to extract symbols from 280 256 × 256 pixel bilevel tiles of the red symbol layer of a large-scale map of part of Finland, scanned at 240 bpi. They also scanned, at lower resolution, the corresponding colored composite map. Twenty-two classes of symbols were identified by a modified nearest-neighbors comparison of feature vectors (moments, circularity, eccentricity, intersections, etc.) with prototypes extracted from the Legend. The classification was conducted in a user-verified mode on the first 50 tiles only. Misrecognized symbols were added to the prototype library. The remaining tiles were processed automatically. The label, certainty (confidence value), location and semantic category of the 1,093 extracted symbols were stored in a relational database. In addition to the usual SQL interface, an easy-to-use GUI was developed. Most of the extensive testing addressed the image indexing and spatial information retrieval aspects of the system, using queries like-*display all layer tiles that contain a beach* and *display all composite and layer tiles that contain an airfield north of a beach.*

Interpretation of complete schematic diagrams that consist of symbols linked by connection lines is the topic of [98]. The test set of over 100 drawings included electrical circuit diagrams, logic circuits, chemical plant schematics, and flowcharts. The output was a graph or a textual netlist indicating the location and type of symbols and their connectivity. After out-sourced thinning and vectorization, and in-house manual text removal (required, unlike in [35], where the text is boxed without recognition), the graphic symbols are extracted using generic segmentation rules and matched to a domain-specific library that contains hierarchical symbol descriptors. The drawing is then traversed to identify the connectors and any missed symbols. The system contains an interactive error-correction phase that automatically logs every user intervention, but many of the drawings were processed perfectly in a few seconds. Contributions were coupling domain-independent algorithms with domain-specific knowledge bases, and evaluation by interactive correction. The article ends with a cogent literature review.

The remaining articles assume that the necessary low-level processing has already been accomplished. They signal that graphics *interpretation* is waiting in the wings.

An influential paper that traces the evolution of constraint satisfaction algorithms over more than a decade is Mulder et al. [56]. The main theme is the integration of constraint satisfaction with a hierarchical schema-based representation. Examples are given of the applications of three generations of Mackworth's Mapsee to sketch maps. An example of a constraint is: *If a closed curve in a land-sea map contains only one other closed curve, they cannot both represent the external shoreline of an island.* Subsequently, constraint satisfaction proved useful for "beautifying" vectorization.

Bergevin and Levine [7] also analyze idealized sketch maps using constraints, but their goal is the characterization (using *geons)* and identification of a three-dimensional object (pot, flashlight, ironing board, stool, wineglass) from a single two-dimensional view. Because the emphasis is on high-level integration, some of the low-level operations are carried out by hand. The main themes are part segmentation, part labeling and object model matching using Biederman's recognition-by-components (RBC) notions. The conclusion of the article addresses weak points of the approach that require further research.

Finally, two articles by Nalwa [60], [61] examine the constraints on orthogonal projections of straight lines and conic curves, and those induced by bilateral (reflective) symmetry as seen in two-dimensional sketches of three-dimensional objects.

6 Outliers

Any clustering algorithm must deal with outliers, and ours is no exception. We summarize here, some contributions that are not near neighbors of any of the ones above.

An expanded index generated by garbling the stored index terms according to the confusion matrix of an OCR device can improve precision and recall [57].

You *can* tell a book by its cover. That is what Gotoh et al. [22] set out to prove. They scanned three sets of 750 magazine covers in a wide range of orientations. One set was used for training and two for recognition. Different issues of the same magazine were considered separate classes. With some tuning, they reached a recognition rate of 94 percent using decision-tree-based clustering.

Peak locations in the waveforms of transversal scans of alternating black and white bars are less sensitive to blur than edges. Joseph and Pavlidis [36] demonstrated that a decoder based on careful analysis of the peaks compared favorably with commercial bar-code recognition rates of the order of one error per million. Bar codes are often used in document indexing and some scanners have a bar-code printer.

Using the word *document,* in the sense of personal document, or photo identification card, the irrepressible O'Gorman proposed with Rabinovich, a method of secure document verification [64]. His objective was to determine whether a scanned photo is, or is not, the same as the originally scanned photo. Differences in the photo-signature between the two digitized images may arise due to scanning and to physical deterioration of the printed photo between the two scans. The method encodes several multiresolution versions of the photo (constructed from the highest resolution image) into a few dozen bytes according to the relative intensities of adjacent pixels. The photo-signature (a well-chosen term) yielded error rates of under 1 percent on a few hundred small mug shots (from the NIST database) gray-scanned at 300 dpi. O'Gorman suggested that the signatures be stored in a central database, or combined with some textual information, encrypted with the ANSI standard Digital Signature Algorithm, and stored on the identification card itself. Alternative identification methods based on signatures, fingerprints, and face recognition are discussed elsewhere in this issue.

7 Problems Solved and Problems Remaining

We are now ready to look back at what we have wrought in the last 20 years. Mulling over what we have seen in *PAMI* and elsewhere, we shall try to assess how far we have come and where we are heading.

7.1 Problems Solved

Our work is getting easier! The major boon to DIA has been the advent of CCD scanners that are far superior to the CRT and drum scanners of the pioneers. Consistent, high-volume, high-resolution, gray-scale, and color digitization is no longer a bottleneck in research. The capabilities of desktop workstations keep overtaking those of special-purpose image-processing architectures. Lossless bilevel page compression has become mainly an issue of standardization. JBIG provides even higher compression than CCITT-G4 and JBIG-2 will offer the long-sought means of information extraction from compressed documents.

Preprocessing has come a long way. Adequate techniques are available for binarizing high-contrast documents. On the other hand, it may be time to concede that many document images cannot be binarized without extensive gray-level analysis (that undermines the economies of binarization). Additive noise and isolated specks have been filtered and averaged out and are now on the endangered species list. Many good techniques have been developed for precise global skew estimation (but this is now less important than accurate word baseline determination). Excellent methods are available for locating all the text on high-contrast printed pages down to the word level with an accuracy that exceeds that of subsequent steps.

Techniques for recognizing isolated document-objects may already have reached a plateau. The error rate on isolated characters, OCR fonts, clean print, and one-dimensional bar codes is so low that it is difficult to conduct significant experiments in a research setting. That is also the case for segmented graphics primitives and for vectorization of high-quality line drawings and map separates. String matching turned out to provide a sound foundation for accurate OCR evaluation and benchmarking, and informative error-reporting formats have emerged. Equally reliable comparison methods have been proposed so far only for computer-generated graphics.

Current model-driven methods can home in on the framework of most forms and tables. Adequate interactive methods are available for model specification starting with either blank or filled-in forms. It appears that robust validation of cell content in known forms can be accomplished with cross-checks and ad hoc methods that access databases, directories, and dynamic work-flow systems, but no supporting theory has emerged so far. It is time to move beyond table geometry.

Among minor achievements, DIA-oriented interpretations have been discovered for almost all of the 26^3 TLAs (three-letter acronyms). Will FLAs be next?

7.2 Problems Remaining

There are still a few aspects of preprocessing that might benefit from increased attention. Exploiting compressed formats for improving the speed of subsequent processing remains appealing. We need to find better techniques for color separation of few-color documents and halftone backgrounds or exploit color as we do gray scale. The reconstruction of overlapping rulings and text (often the result of printing from multiple source layers) needs building on some good beginnings.

We are far from being able to convert complex engineering drawings and maps to any CAD format. Instead of attempting to build graphics interpretation from the ground up, perhaps we should begin where the commercial systems leave off. To the extent that human intervention remains necessary, we might give more thought to where it is best applied and how the automated parts of the system can profit from it.

OCR research must aim to decrease error rates by more than an order of magnitude on poor images from old fax machines, copiers, or high-speed printing on cheap paper. It is vexing that language context helps least where accuracy is most important, as on proper nouns and numeric fields! Several directions can be explored:

- Feature-based multifont document image decoding;
- Adaptation of more direct HMM methods to two-dimensional document pixel maps (instead of considering text lines as a form of speech);
- Large-vocabulary word or partial-word recognition combined with word-transition probabilities;
- Style-based recognition that exploits consistency of typeface, layout, and document quality; and
- Application-oriented benchmarking and classification of techniques.

We can never resist proselytizing for learning and adaptation. Perhaps static recognition is now good enough to take a fresh look at dynamic systems. Confidently recognized patterns can help identify their more ambiguous brethren. (The speech community is certainly moving in that direction. Bootstrap training is widely used for HMM). Exploiting feedback from human editing and downstream validation software that reflects the application-specific error patterns should lead to systems that improve with use. Recent theoretical results that explore the combination of supervised and unsupervised learning might apply. Perhaps the SSS (syntactic, structural, and statistical) federation should seek a rapprochement with the machine-learning community.

Document content tagging is just around the corner. XML will play an important role as a target representation in application-specific DIA. Extracting XML tags will require a principled approach to the critical interface between DIA front-ends and back-ends that incorporate database management systems, work-flow, or formal table-models. Similar considerations apply to drawings and high-level CAD/CAM data interchange standards, and to maps and GIS. DIA products cannot be expected to function in isolation.

Computer-generated documents in impoverished digital formats that preclude full access to content are proliferating. Aspects that differentiate such indigenous document images from scanned documents provide fresh opportunities for DIA, and help to legitimize claims to multimodality.

Better evaluation remains, as always, on the to-do list: metadata for mostly-text, live documents for mostly-graphics, and content-based evaluation for everything. What are *representative* documents? We still have no idea how to conduct a document census, especially on the web. Nor do we know how to systematically select and combine techniques (i.e., design systems) for specific document types and application requirements.

We hope that we have inspired you to browse some of the back issues of *PAMI* (at least those that are already posted on the web—at the time of writing the rest are still waiting for improvements in DIA). That is all, at least for now.

Appendix A

Glossary

ANSI: American National Standards Institute

ApOFIS: A priori Optical Font Identification System

ASCII: American Standard Code for Information Interchange

BAG: Block Adjacency Graph

CAD: Computer Aided Design (or Drafting)

CAM: Computer Aided Manufacturing

CCD: Charge-coupled device

CCITT: Comite Consultatif de Telephonie et de Telegraphie

CRT: Cathode Ray Tube

DAFS: Document Attribute Format Specification

DARPA: Defense Advanced Research Projects Agency

DAS: Workshop on Document Analysis Systems

DBMS: Database Management System

DIA: Document Image Analysis

DID: Document Image Decoding

DIMUND: Document Image Understanding (DARPA project)

DOCSTRUM: Document Spectrum

DPC: Delivery Point Code

DPI: Dots per inch

DR&R: SPIE Symposium on Document Recognitionand Retrieva

DXF: Document eXchange Format

GIFF: Graphics Image File Format

GIS: Geographic Information System

GR: Graph Representation

GUI: Graphical User Interface

HMM: Hidden Markov Model

HTML: Hypertext Markup Language

HWAIS: Handwritten Address Interpretation System

ICDAR: International Conference on Document Analysis and Recognition

ICPR: International Conference on Pattern Recognition

IJDAR: International Journal of Document Analysis and Recognition

Infomys: Invoice-like form reader system

IR: Information Retrieval

IRS: Internal Revenue Service

ISRI: Information Science Research Institute (Las Vegas)

JBIG: Joint Bilevel Image Experts Group (encoding standards)

JIS: Japanese Information Society

LAG: Line Adjacency Graph

MAGELLAN: Map Acquisition of Geographic Labels by Legend Analysis

MAP: Multiangled Parallel (more often, Maximum A Posteriori)

MARCO: Map Retrieval by Content

ML: Machine learning

MRA: Multiresolution Wavelet Analysis

NIST: National Institute of Standards and Technology

OCR: Optical Character Recognition

ODIL: Office Document Image description Language (based on SGML)

OZZ: Orthogonal Zig-Zag

PDF: Portable Document Format (Adobe)

P-SPICE: Personal Simulation Program with Integrated Circuit Emphasis

RBC: Recognition by Components

RLC: Run Length Coding

RTF: Rich Text Format

SDAIR: Symposium on Document Analysis and Information Retrieval

SGML: Standard Generalized Markup Language

SPARC: Scaleable Processor Architecture

SPV: Sparse Pixel Vectorization

SQL: Simple Query Language

SSPR: Workshop on Structural and Syntactic Pattern

Recognition
TOC: Table of Contents
TIFF: Tagged Image File Format
UPC: Universal Product Code
USPS: United States Postal Service
XDOC: Xerox TextBridge rich document format
XML: Extensible Markup Language (based on SGML)
ZIP: Zone Improvement Program

Appendix B

Additional References

Bibliographies

1. F.J. Thomas and L.P. Horwitz, "Character Recognition Bibliography and Classification," IBM research report RC-1088, T.J. Watson Research Center, Yorktown Heights, New York, 1964.
2. R. Shillman, C. Cox, T. Kuklinsky, J. Ventura, M. Eden, and B. Blesser, "A Bibliography in Character Recognition Techniques for Describing Characters," *Visible Language*, vol. 7, pp. 151-166, The Cleveland Museum of Art, Cleveland, Ohio, 1974.
3. F. Jenkins and J. Kanai, "A Keyword-Indexed Bibliography of Character Recognition and Document Analysis," revision 2.0, Information Science Research Inst., Univ. of Nevada, Las Vegas, 1993.
4. C.F. Sabourin, "Optical Character Recognition and Document Segmentation: Bibliography," Infolingua Inc., Montreal, Canada, 1994.

Monographs and Reports

1. M.E. Stevens, "Automatic Character Recognition—State-of-the-Art Report," Nat'l Bureau of Standards & Technology, Technical Note 112, Washington, 1961.
2. British Computer Society Character Recognition, British Computer Society, London, 1967.
3. V.A. Kovalevsky, *Character Readers and Pattern Recognition.* Washington, D.C.: Spartan Books, 1968.
4. J. Ullman, *Pattern Recognition Techniques.* New York: Crane, Russak, and Co., 1973.
5. R. McLean, *Thames and Hudson Manual of Typography.* London: Thames and Hudson, 1980.
6. H.F. Schantz, *The History of OCR.* Recognition Technologies Users Assoc., Boston, 1982.
7. H.S. Hou, *Digital Document Processing.* New York: Wiley, 1983.
8. K.R. McConnell, D. Bodson, and R. Schaphorst, *FAX: Digital Facsimile Technology and Applications.* Norwood, Mass.: Artech House, 1989.
9. M. Nadler and E.J. Smith, *Pattern Recognition Engineering.* New York: Wiley, 1993.
10. Annual Report, Information Science Research Inst. Univ. of Nevada, Las Vegas, 1991-1996.
11. J. Schurmann, *Pattern Classification.* New York: Wiley, 1996.
12. S.V. Rice, G. Nagy, and T.A. Nartker, *Optical Character Recognition: An Illustrated Guide to the State of the Art,* Kluwer, 1999.

Edited Collections

1. G.L. Fisher, *Optical Character Recognition.* Washington, D.C.: Spartan Books, 1960.
2. L.K. Kanal, *Pattern Recognition.* Washington, D.C.: Thompson Book Co., 1968.
3. S.N. Srihari, *Computer Text Recognition and Error Correction.* IEEE CS Press, Los Alamitos, California, 1985.
4. J.C. Van Vliet, ed., *Document Manipulation and Typography.* Cambridge UK, Cambridge Univ. Press, 1988.
5. H.S. Baird, H. Bunke, and K. Yamamoto, *Structured Document Image Analysis.* Berlin: Springer-Verlag, 1992.
6. H. Bunke, *Advances in Structural and Syntactic Pattern Recognition.* Singapore, World Scientific, 1992.
7. A.L. Spitz and A. Dengel, *Document Analysis Systems.* Singapore, World Scientific, 1995.
8. L. O'Gorman and R. Kasturi, *Document Image Analysis.* Los Alamitos, California, IEEE CS Press, 1995.
9. R. Kasturi and K. Tombre, *Graphics Recognition: Methods and Applications.* Lecture Notes in Computer Science 1,072, Springer, 1996.
10. N.A. Murshed and F. Bortolozzi, *Advances in Document Analysis.* Lecture Notes in Computer Science 1,339, Springer, 1997.
11. H. Bunke and P.S.P. Want, *Handbook of Character Recognition and Document Image Analysis.* World Scientific, 1997.
12. K. Tombre and A.K. Chhabra, *Graphics Recognition: Algorithms and Systems.* Lecture Notes in Computer Science 1,389, Springer, 1998.
13. J.J. Hull and S.L. Taylor, *Document Analysis Systems II.* World Scientific, 1998.

Special Issues

1. *Proc. IEEE Special 80th Anniversary Issue Optical Character Recognition,* T. Pavlidis and S. Mori, eds., vol. 80, no. 7, IEEE, New York, 1992.
2. *Machine Vision and Applications.* L. O'Gorman and R. Kasturi, eds., Special Issue document image analysis techniques, vol. 5, no. 3, Springer, Int'l 1992.
3. *IEEE Computer Special Issue Document Image Analysis Systems,* L. O'Gorman and R. Kasturi, eds., vol. 25 no. 7, IEEE CS, 1992.

Conference Proceedings

1. *Procs. Int'l Conf. Pattern Recognition (ICPR),* IEEE CS, biennial since 1973.
2. *Procs. Int'l Conf. Document Analysis and Recognition (ICDAR),* IEEE CS, biennial since 1991.
3. *Procs. Symp. Document Analysis and Information Retrieval (SDAIR),* Information Science Research Inst., University of Nevada, Las Vegas, 1991-1996.

4. *Procs. Symp. Document Recognition and Retrieval (DR&R), Int'l Soc. for Optical Eng. (SPIE)*, annual since 1993.

Scholarly Journals

1. *Computer Vision, Graphics, and Image Processing,* Academic Press, since 1975.
2. *Electronic Imaging,* SPIE, since 1992.
3. *Image and Vision Computing,* Elsevier, since 1982.
4. *Int'l J. Digital Libraries,* Springer, since 1999.
5. *Int'l J. Document Analysis and Recognition,* Springer, since 1998.
6. *Machine Vision and Applications,* Springer, since 1988.
7. *Pattern Analysis and Applications,* Springer, since 1998.
8. *Pattern Recognition,* Pergamon, since 1969.
9. *Pattern Recognition and Artificial Intelligence,* World Scientific, since 1987.
10. *Pattern Recognition Letters,* Elsevier, since 1979.
11. *Visible Language,* Cleveland Museum of Art, Cleveland, since 1967.

Magazines

1. *Imaging and Document Solutions,* San Francisco: Miller Freeman.
2. *Advanced Imaging,* Melville, New York: Cygnus Publishing.

ACKNOWLEDGMENTS

The author gratefully acknowledges the suggestions and encouragement of the *PAMI* referees and of Tin Ho, Dan Lopresti, Jill Nagy, Tom Nartker, Prateek Sarkar, Larry Spitz, and Claudia Wenzel. Yihong Xu contributed Fig. 1.

REFERENCES

[1] H.K. Aghajan and T. Kailath, "SLIDE: Subspace-Based Line Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 11, pp. 1,057-1,073, Nov. 1994.

[2] H. Al-Yousefi and S.S. Udpa, "Recognition of Arabic Characters" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 8, pp. 853-857, Aug. 1992.

[3] H.I. Avi-Itzhak, T.A. Diep, and H. Garland, "High Accuracy Optical Character Recognition Using Neural Networks with Centroid Dithering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 2, pp. 218-223, Feb. 1995.

[4] H.I. Avi-Itzhak, J.A. Van Mieghem, and L. Rub, "Multiple Subclass Pattern Recognition: A Maximin Correlation Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 4, pp. 418-431, Apr. 1995.

[5] H.S. Baird and K. Thompson, "Reading Chess," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 6, pp. 552-559, June 1990.

[6] I. Bazzi, R. Schwartz, and J. Makhoul, "An Omnifont Open-Vocabulary OCR System for English and Arabic," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 6, pp. 495-504, June 1999.

[7] R. Bergevin and M.D. Levine, "Generic Object Recognition: Building and Matching Coarse Descriptions from Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 1, pp. 19-36, Jan. 1993.

[8] H. Bunke, "Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 4, no. 6, pp. 574-582, June 1983.

[9] D.J. Burr, "Elastic Matching of Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, no. 6, pp. 708-712, June 1981.

[10] C.A. Cabrelli and U.M. Molter, "Automatic Representation of Binary Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 12, pp. 1,190-1,195, Dec. 1990.

[11] R.G. Casey and E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 690-706, July 1996.

[12] F. Cesarini, M. Gori, S. Marinai, and G. Soda, "INFORMys: A Flexible Invoice-Like Form-Reader System," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 7, pp. 730-745, July 1998.

[13] B.B. Chaudhuri and U. Pal, "Skew Angle Detection of Digitized Script Documents" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 182-186, Feb. 1997.

[14] E. Cohen, J.J. Hull, and S.N. Srihari, "Control Structure for Interpreting Handwritten Addresses," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 10, pp. 1,049-1,055, Oct. 1994.

[15] S. Di Zenzo, L. Cinque, and S. Levialdi, "Run-Based Algorithms for Binary Image Analysis and Processing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 1, pp. 83-88, Jan. 1996.

[16] D. Dori, "Vector-Based Arc Segmentation in the Machine Drawing Understanding System Environment," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 11, pp. 1,057-1,068, Nov. 1995.

[17] D. Dori and W. Liu, Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 3, pp. 202-215, Mar. 1999.

[18] F. Esposito, D. Malerba, and G. Semeraro, "Classification in Noisy Environments Using a Distance Measure Between Structural Symbolic Descriptions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 3, pp. 390-402, Mar. 1992.

[19] K. Etemad, D. Doerman, and R. Chellappa, "Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 92-96, Jan. 1997.

[20] L.A. Fletcher and R. Kasturi, "Robust Algorithm for Text String Separation from Mixed Text/Graphics Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 6, pp. 910-918, June 1988.

[21] M.D. Garris and D.L. Dimmick, "Form Design for High Accuracy Optical Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 6, pp. 653-656, June 1996.

[22] T. Gotoh, T. Toriu, S. Sasaki, and M. Yoshida, "A Flexible Vision-Based Algorithm for a Book Sorting System," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 3, pp. 393-399, Mar. 1988.

[23] Y.X. Gu, Q.R. Wang, and C.Y. Suen, "Application of a Multilayer Decision Tree in Computer Recognition of Chinese Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 1, pp. 83-88, Jan. 1983.

[24] D.I. Havelock, "Geometric Precision in Noise-Free Digital Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 10, pp. 1,065-1,075, Oct. 1989.

[25] D.I. Havelock, "The Topology of Locales and Its Effect on Position Uncertainty," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 4, pp. 380-386, Apr. 1991.

[26] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision Combination in Multiple Classifier Systems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 1, pp. 66-75, Jan. 1994.

[27] T.K. Ho and H.S. Baird, "Large-Scale Simulation Studies in Image Pattern Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 10, pp. 1,067-1,079, Oct. 1997.

[28] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic Script Identification from Document Images Using Cluster-Based Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 2, pp. 176-181, Feb. 1997.

[29] A. Hoover et al., "An Experimental Comparison of Range Image Segmentation Algorithms" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 7, pp. 673-689, July 1996.

[30] X. Huang, J. Gu, and Y. Wu, "A Constrained Approach to Multifont Chinese Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 8, pp. 838-843, Aug. 1993.

[31] J.J. Hull and S.N. Srihari, "Experiments in Text Recognition with Binary n-Gram and Viterbi Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 520-529, May 1982.

[32] J.J. Hull, S.N. Srihari, and R. Choudhari, "An Integrated Algorithm for Text Recognition: Comparison with a Cascaded Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, no. 4, pp. 384-395, Apr. 1983.

[33] J.J. Hull, "Incorporating Language Syntax in Visual Text Recognition with a Statistical Mode," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1,251-1,256, Dec. 1996.

[34] A.K. Jain and B. Yu, "Document Representation and Its Application to Page Decomposition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 294-308, Mar. 1998.

[35] S.H. Joseph and T.P. Pridmore, "Knowledge-Directed Interpretation of Mechanical Engineering Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 9, pp. 928-940, Sept. 1992.

[36] E. Joseph and T. Pavlidis, "Bar Code Waveform Recognition Using Peak Locations" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 630-640, June 1994.

[37] D.-M. Jung, G. Nagy, and A. Shapira, "N-tuple Features for OCR Revisited," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 734-745, July 1996.

[38] S. Kahan, T. Pavlidis, and H.S. Baird, "On the Recognition of Printed Characters of Any Font and Size," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 2, pp. 274-288, Feb. 1987.

[39] A.C. Kam and G.E. Kopec, "Document Image Decoding by Heuristic Search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 9, pp. 945-950, Sept. 1996.

[40] J. Kanai, S.V. Rice, T.A. Nartker, and G. Nagy, "Automated Evaluation of OCR Zoning" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 86-89, Jan. 1995.

[41] T. Kanungo and R.M. Haralick, "An Automatic Closed-Loop Methodology for Generating Character Groundtruth for Scanned Documents," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 179-183, Feb. 1999.

[42] R. Kasturi, S.T. Bow, W. El-Masri, Y. Shah, J.R. Gattiker, and U.B. Mokate, "A System for Interpretation of Line Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 10, pp. 978-992, Oct. 1990.

[43] E.T. Endo, "On a Method of Binary-Picture Representation and Its Application to Data Compression," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 1, pp. 27-35, Jan. 1980.

[44] A. Khotanzad and Y.H. Hong, "Invariant Image Recognition by Zernike Moments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 5, pp. 489-497, May 1990.

[45] G.E. Kopec and P.A. Chou, "Document Image Decoding Using Markov Source Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 602-617, June 1994.

[46] G.E. Kopec and M. Lomelin, "Supervised Template Estimation for Document Image Decoding," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, pp. 1,313-1,324, Dec. 1997.

[47] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, "Syntactic Segmentation and Labeling of Digitized Pages from Technical Journals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 7, pp. 737-747, July 1993.

[48] S. Kuo and O.E. Agazzi, "Keyword Spotting in Poorly Printed Documents," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 842-847, Aug. 1994.

[49] C.P. Lai and R. Kasturi, "Detection of Dimension Sets in Engineering Drawings" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 8, pp. 848-854, 1994.

[50] K.H. Lee, K.-B. Eom, and R.L. Kashyap, "Character Recognition Based on Programmed Attribute-Dependent Grammar," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1,122-1,128, Nov. 1992.

[51] S.-W. Lee, D.-J. Lee, and H.-S. Park, "A New Methodology for Gray-Scale Character Segmentation and Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 1,045-1,050, Oct. 1996.

[52] Y. Li, D. Lopresti, G. Nagy, and A. Tomkins, "Validation of Image Defect Models for Optical Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 99-108, Feb. 1996.

[53] Y. Liu and S. Srihari, "Document Image Binarization Based on Texture Features," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 540-544, May 1997.

[54] Z. Lu, "Detection of Text Regions from Digital Engineering Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 431-439, Apr. 1998.

[55] B.T. Messmer and H. Bunke, "A New Algorithm for Error-Tolerant Subgraph Isomorphims Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, May 1998.

[56] J.A. Mulder, A.K. Mackworth, and W.S. Havens, "Knowledge Structuring and Constraint Satisfaction: The Mapsee Approach" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 866-879, June 1988.

[57] J.K. Mullin, "Reliable Indexing Using Unreliable Recognition Devices," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 3, no. 3, pp. 347-350, Mar. 1981.

[58] H. Nagahashi and M. Nakatsuyama, "A Pattern Description and Generation Method of Structural Characters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 1, pp. 112-117, Jan. 1986.

[59] G. Nagy, S. Seth, and K. Einspahr, "Decoding Substitution Ciphers by Means of Word Matching with Application to OCR," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 710-714, May 1987.

[60] V.S. Nalwa, "Line-Drawing Interpretation: Straight Lines and Conic Sections," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 4, pp. 514-529, Apr. 1988.

[61] V.S. Nalwa, "Line-Drawing Interpretation: Bilateral Symmetry," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 10, pp. 1,117-1,120, Oct. 1989.

[62] A. Namane and M.A. Sid-Ahmed, "Character Scaling by Contour Method," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 600-606, June 1990.

[63] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1,162-1,173, Nov. 1993.

[64] L. O'Gorman and I. Rabinovich, "Secure Identification Documents via Pattern Recognition and Public-Key Cryptography," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1,097-1,102, Oct. 1998.

[65] B.J. Oommen, "Recognition of Noisy Subsequences Using Constrained Edit Distances," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, no. 5, pp. 675-685, May 1987.

[66] A. Ozaki, T. Kondo, K. Mori, S. Tsunekawa, and E. Kawamoto, "An Automatic Circuit Diagram Reader with Loop-Structure-Based Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 331-341, Mar. 1988.

[67] J. Rocha and T. Pavlidis, "A Shape Analysis Model with Applications to a Character Recognition System," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 4, pp. 393-404, Apr. 1994.

[68] J. Rocha and T. Pavlidis, "Character Recognition without Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 903-909, Sept. 1995.

[69] H. Samet and A. Soffer, "MARCO: MAp Retrieval by COntent," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 783-798, Aug. 1996.

[70] P. Sarkar, G. Nagy, J. Zhou, and D. Lopresti, "Spatial Sampling of Printed Patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 344-350, Mar. 1998.

[71] M. Sawaki and N. Hagita, "Text-Line Extraction and Character Recognition of Document Headlines with Graphical Designs Using Complementary Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1,103-1,108, Oct. 1998.

[72] R. Shinghal and G.T. Toussaint, "Experiments in Text Recognition with the Modified Viterbi Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 184-192, Feb. 1979.

[73] R. Shinghal and G.T. Toussaint, "The Sensitivity of the Modified Viterbi Algorithm to the Source Statistics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 2, pp. 1,181-1,184, Feb. 1980.

[74] A. Simon, J.-C. Pret, and A.P. Johnson, "A Fast Algorithm for Bottom-Up Layout Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 273-277, Mar. 1997.

[75] R.M.K. Sinha, B. Prasada, G.H. Houle, and M. Sabourin, "Hybrid Contextual Text Recognition with String Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 915-923, Sept. 1993.

[76] A.L. Spitz, "Determination of the Script and Language Content of Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 3, pp. 235-245, Mar. 1997.

[77] F.W.M. Stentiford, "Automatic Feature Design for Optical Character Recognition Using an Evolutionary Search Procedure," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, no. 3, pp. 349-354, Mar. 1985.

[78] L. Stringa, "A New Set of Constraint-Free Character Recognition Grammars," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 12, pp. 1,210-1,216, Dec. 1990.

[79] C.Y. Suen, "N-Gram Statistics for Natural Language Under-Standing and Text Processing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 164-172, Feb. 1979.

[80] T.N. Tan, "Rotation Invariant Texture Features and Their Use in Automatic Script Identification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 7, pp. 751-756, July 1998.

[81] E. Tanaka and Y. Kojima, "A High Speed String Correction Method Using a Hierarchical File," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 6, pp. 806-815, June 1987.

[82] Y.T Tang, H. Ma, Y. Liu, B.F. Li, and D. Xi, "Multiresolution Analysis in Extraction of Reference Lines from Documents with Gray Level Background," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 921-925, Aug. 1997.

[83] T. Taxt, P.J. Flynn, and A.K. Jain, "Segmentation of Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 12, pp. 1,322-1,329, Dec. 1989.

[84] O.D. Trier and T. Taxt, "Evaluation of Binarization Methods for Document Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 3, pp. 312-314, Mar. 1995.

[85] O.D. Trier and A.K. Jain, "Goal-Directed Evaluation of Binarization Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 12, pp. 1,191-1,201, Dec. 1995.

[86] O.D. Trier, T. Taxt, and A.K. Jain, "Recognition of Digits in Hydrographic Maps: Binary Versus Topographic Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 4, pp. 399-404, Apr. 1997.

[87] R.J. Ulichney and D.T. Troxel, "Scaling Binary Images with a Telescoping Template" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 4, no. 3, pp. 331-335, Mar. 1982.

[88] Q.R. Wang and C.Y. Suen, "Analysis and Design of a Decision Tree Based on Entropy Reduction and Its Application to Large Character set Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 4, pp. 406-417, Apr. 1986.

[89] Q.R. Wang and C.Y. Suen, "Large Tree Classifier with Heuristic Search and Global Training," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, no. 1, pp. 91-102, Jan. 1987.

[90] Y.P. Wang and T. Pavlidis, "Optimal Correspondence of String Subsequences," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 11, pp. 1,080-1,087, Nov. 1990.

[91] T. Pavlidis and L. Wang, "Direct Gray-Scale Extraction of Features for Character Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 10, pp. 1,053-1,067, Oct. 1993.

[92] T. Watanabe, Q. Luo, and N. Sugie, "Layout Recognition of Multi-Kinds of Table-Form Documents," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 17, no. 4, pp. 432-445, Apr. 1995.

[93] L. Wenyin and D. Dori, "Incremental Arc Segmentation Algorithm and Its Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 4, pp. 424-430, Apr. 1998.

[94] S. Yajima, J.L. Goodsell, T. Ichida, and H. Hirasishi, "Data Compression of Kanji Character Patterns Digitized on a Hexagonal Mesh," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, no. 2, pp. 121-229, Feb. 1981.

[95] H. Yamada, K. Yamamoto, and K. Hosokawa, "Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 4, pp. 380-387, Apr. 1993.

[96] E.J. Yannakoudakis and G. Angelidakis, "An Insight Into the Entropy and Redundancy of the English Dictionary," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 10, no. 6, pp. 960-969, June 1988.

[97] B. Yu and A.K. Jain, "A Generic System for Form Dropout," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 11, pp. 1,127-1,134, Nov. 1996.

[98] Y. Yu, A. Samal, and S.C. Seth, "A System for Recognizing a Large Class of Engineering Drawings," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 868-890, Aug. 1997.

[99] A. Zramdini and R. Ingold, "Optical Font Identification Using Typographic Features," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 877-882, Aug. 1998.

George Nagy received the BEng and MEng degrees from McGill University, and the PhD degree in electrical engineering from Cornell University in 1962 (on neural networks). For the next 10 years, he conducted research on pattern recognition and OCR at the IBM T.J. Watson Research Center in Yorktown Heights, New York. From 1972-1985, he was professor of computer science at the University of Nebraska, Lincoln, and worked on remote sensing applications, geographic information systems, computational geometry, and human-computer interfaces. Since 1985, he has been professor of computer engineering at Rensselaer Polytechnic Institute. He has held many visiting appointments in the United States and abroad. In addition to document image analysis and character recognition, his research interests include geometric modeling, finite-precision spatial computation, and computer vision. He is co-author of the 1999 monograph, *Optical Character Recognition: An Illustrated Guide to the Frontier.* He is a senior member of the IEEE.