At the Frontiers of OCR

GEORGE NAGY, SENIOR MEMBER, IEEE

Invited Paper

It is time for a major change of approach to character recognition research. The traditional approach, focusing on the the correct classification of isolated characters, has been exhausted. The demonstration of the superiority of a new classification method under operational conditions requires large experimental facilities and data bases beyond the resources of most researchers. In any case, even perfect classification of individual characters is insufficient for the conversion of complex archival documents to a useful computer-readable form. Many practical OCR tasks require integrated treatment of entire documents and well-organized typographic and domain-specific knowledge. New OCR systems should take advantage of the typographic uniformity of paragraphs or other layout components. They should also exploit the unavoidable interaction with human operators to improve themselves without explicit "training."

Keywords*—*Pattern recognition, optical character recognition; character classification; digitized document analysis; document conversion; scan digitization; learning.

I. Introduction

The thesis of this article is that much of the spectacular progress of OCR since its commercial debut in 1955 has been due to advances in processor and digitizer technology rather than to improved classification techniques for isolated patterns. Therefore further progress may depend more on new methods for "recognizing" entire documents than on better character-by-character feature extraction and classification algorithms. Possibilities for exploiting the relationships between the patterns in a document, some new and some not so new, are suggested. The emphasis is on *printed* characters, and the presentation is biased toward techniques in the realm of the author's personal experience.

During the 1960's and 1970's tens of thousands of high-performance OCR systems with fast document transports and hard-wired recognition logic were sold in the United States. At one time, over 50 manufacturers were in the market (Schantz [1]). They targeted large-volume applications with repetitive layouts and controlled typefaces such as preprinted order forms or invoices, and typed documents. During a three-month period in 1967, for instance, one machine scanned 33.1 million lines of typescript, of which 16.3 million lines were positively recognized, 3.6 million lines contained one or more unrecognizable characters, and 13.2 million lines appeared on pages that were rejected by the system (Hennis *et al.* [2]). Often documents were retyped specifically for OCR (using a fixed-pitch typeface and double spacing) as an alternative to keypunching or key-to-tape. A single system typically performed the work of ten to 20 key-entry operators. Postal address readers constituted a profitable niche for a few specialized companies, but they achieved their greatest success in Japan (Genchi *et al.* [3]).

Some systems could be customized to a specific set of formats and typefaces, but this was generally done by the vendor before delivery. No commercial systems were available for free-form business letters; typeset English text; complex documents containing illustrations, tables, and formulas; engineering drawings; or utility maps. For such applications, the relatively narrow-gauge classifiers in use would have required in-house training, which at the time involved at least burning in PROM's. Furthermore, the poor geometric linearity and resolution of flying-spot scanners inhibited character location and isolation on densely set, full-sized pages (Nagy [4]).

Each character image was processed *sequentially* and *independently*. Only in *preprocessing* (gray-scale normalization, noise elimination, line finding, and alpha/numeric field definition) and *postprocessing* (spelling verification or correction, sometimes incorporating customized lexicons) were larger entities taken into consideration. The legacy of this period, the paradigm of *character location–character isolation–character classification*, is still with us.

The paradigm was dictated by the universal use of fast but relatively inflexible hard-wired classification engines. Although programmable processors were already available—and were sometimes used in postprocessing—their recognition throughput did not match the document digitization hardware, which required expensive lasers, mechanical scanners, or cathode-ray tubes. In other words, the optical scanner and the page transport dominated the design. Inexpensive and accurate (but relatively slow) optical scanners surfaced only in the early 1980's with the advent of solid-state electro-optical arrays.

Manuscript received February 4, 1991; revised November 21, 1991. This work was supported by the U.S. Department of Education College Library Technology Grant Program.

The author is with the Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute,Troy, NY 12180.

IEEE Log Number 9202480.

Academic research mirrored the paradigm. Hundreds of papers were published on alternative line-thinning, contour-following, feature-extraction, and classification algorithms. Almost all of these methods were restricted to isolated characters. Not infrequently, the sample patterns were manually isolated or printed specifically for the classification experiments. With a few notable exceptions the samples were too small to yield significant results. To the extent that the parameters of the classifier were adjustable, this was done on a separate "training set" of data, mimicking factory customization. The test set was often implicitly reused in design, and therefore lost its predictive value. Once the classifier was designed, it could not change, and would make the same mistakes forever (Nagy [5]).

In the last 20 years, research and commercial development of printed document processing systems has been vigorously pursued in Europe and Japan. However, the attention of university researchers turned to handprinted rather than machine-printed characters, with a smattering of experiments on cursive script. Large-scale experiments on realistic data were conducted on postal addresses and on Japanese characters, where sizable standard data sets were available to the research community (Casey and Nagy [6]). Incremental training modes were extensively investigated in on-line recognition of characters entered on a graphic tablet with a stylus.

The 1980's saw the emergence of OCR systems intended for use with personal computers. The prices of page-digitizers tumbled. Image compression standards evolved, partly as a result of the requirements of digital facsimile (Netravali [7]). Some current OCR systems require an additional processor board, but many carry out the entire process in software.The results of a prescan of the page can be displayed on the screen at reduced resolution, and the operator uses a mouse to outline the extent and order of the fields to be processed. Some systems can handle fairly complex multicolumn formats and, importantly, retain some of the formatting information such as indentation, centering, and italicization. The results of the recognition process can be streamed directly into a standard word processor, allowing its built-in functions to be used for spell-checking and postediting. Line art, photographs, and even halftones interspersed with the text can be digitized and saved as a TIFF (tagged image file format) file for desktop publishing applications.

Even as PC-based OCR became practical for the occasional user, much more powerful systems, with fast scanners, special processors, a workstation host, and multiple editing terminals, were also developed. As a result of improvements arising from multiple parallel microprocessors and solid-state scanners, the old line of stand-alone, fixed-format machines were extended to handprinted amount fields on credit-card slips and giro forms. Generally, however, very little information has been published on the principles of operation of commercial systems. Some performance figures are available from comparisons conducted by the computer magazines on a few sample documents. For a thoughtful appraisal of the current status of document analysis and OCR systems, see the work by Casey and Wong [8] in addition to the articles in this issue.

In spite of recent progress, we have not nearly reached the stage where a printed page can be inserted in an OCR system and a coded file comparable to a keyed-in version emerges. Unusual typefaces may baffle the system, but most of the classification errors will be due to missegmented characters (particularly on poor copies, small point sizes, and kerned, bold, or italicized typefaces). Formulas and equations are mangled. Lines with drop-caps, large mathematical symbols, or superscripts may be missed altogether. Omitted characters must be keyed in and misidentified ones corrected. A major problem is the reading order: figure captions are likely to show up unexpectedly in the text, paragraphs in multicolumn documents appear in the wrong order, headings are not recognized, and tables are garbled. Even systems with sophisticated format analysis depend on the user for some information about the layout, and others expect each field to be demarcated prior to the OCR stage (*zoning*). In trainable systems, the user must spend a considerable amount of time identifying characters. Many of the mistakes seem irrational to the user and, still, the same ones are repeated again and again.

The following observations are based on the conversion of 250 000 pages containing typed and typeset text, graphics, multiple columns, blowbacks from microfilm, and copies of copies. The documents were drawn from the U.S. Department of Energy's Licensing Support System for nuclear plants (LSS). The specifications called for a final accuracy of 99.8% at the character level (four errors on an average 1800-character LSS page). The best of several commercial hardware and software OCR systems tested achieved 98.6% correct recognition on a prototype data base. It proved cheaper to rekey pages with many errors than to correct the errors. (Pages with excessive skew, curl, speckle, blur, low contrast, or a profusion of technical symbols or special characters should not even be submitted to the OCR system.) Of the average conversion cost of $3.79 per page, two thirds was due to editing, and about one fifth to manual zoning (Bradford [9], Dickey [10]).

In the following sections, we discuss possible means of alleviating these problems.

II. Document Analysis

Current standards for document interchange differentiate between the *logical structure* and the *layout structure* of the document. Correspondingly, there are two approaches for converting a page-image to a computer-readable file. One approach ignores the logical structure and uses knowledge of *generic typesetting practices* only. The other approach is restricted to specific families of documents, for which *publication-specific* information is available to label logical components.

Methods based only on typesetting conventions generally embed the information in the preprocessing routines. Even when the parameters are explicit, it is difficult to relate them to document characteristics (Wahl *et al.* [11], Wong *et al.* [12]). It is therefore impossible to take an existing program and form a clear conception of its capabilities, short of extensive experimentation. Further progress will depend on the development of more consistent and comprehensive knowledge bases through explicit codification of such information. Some examples of generic typesetting knowledge for text set in derivatives of the Latin alphabet are as follows:

- Printed lines are parallel and roughly horizontal.
- The baselines of characters are aligned.
- Each line of text is set in a single point size.
- Ascenders, descenders, and capitals have consistent heights. Serifs are aligned.
- Typefaces (including variants such as italic or bold) do not change within a word.
- Within a line of text, word spaces are larger than character spaces.
- The baselines of text in a paragraph are spaced uniformly.
- Each paragraph is left-justified or right-justified (or both), with special provisions for the first and last line of a paragraph.
- Paragraphs are separated by wider spaces than lines within a paragraph, or by indentation.
- Illustrations are confined to rectangular frames.
- In multicolumn formats, the columns are of the same width.

The role of typesetting cues to aid document understanding is discussed cogently by Holstege and Tokuda [13]. It is shown that visual parsing can locate important document components such as titles and subtitles unambiguously and determine the normal reading order for multicolumn pages.

Typesetting rules also help in discriminations such as c versus C, s versus 5, and g versus 9, which in a multifont environment simply cannot be made reliably on a single character (Kahan *et al.* [14]). They can also be used to reduce the search space in the classification process by dividing the characters into groups according to their height and position with respect to the baseline (Luca and Gisotti [15]). Classification of the characters into half a dozen classes according to position relative to the baseline and size also allows accurate baseline location even for tightly set and vertically overlapping lines of text (Kanai [16]).

The prevalence of segmentation errors suggests combining character location with character classification. Character segmention boundaries can be located according to acceptable identification of each of the fragments (Kovalevsky [17], Casey *et al.* [18]). When several characters are connected, a number of segmentation hypotheses must be tested, but since usually less than 10%–20% of the characters on a page touch, the overhead of more sophisticated segmentation is acceptable. A more efficient alternative on natural language text is to draw word hypotheses from a lexicon, derive the approximate segmentation boundaries from the character identities, and then verify whether the segmented patterns match (Paoli and Pareschi [19]). It now is clear that on closely spaced print (and, *a fortiori*, on handprinted and cursive writing), adequate segmentation cannot be accomplished without recognition, and vice versa.

In publication-specific systems, the knowledge base is usually more explicit. It may be in the form of if-then rules (Kubota *et al.* [20]), geometric tree (Dengel and Barth [21]), form-definition language (Fujisawa *et al.* [22]), expert system (Thomas [23]), or page grammar (Viswanathan [24]). Some examples of rules for articles in the IEEE TRANSACTIONS ON PATTERN RECOGNITION AND MACHINE INTELLIGENCE are:

- Title-lines are set in 21/23-point roman bold.
- There are at most four lines in the title.
- Bylines follow the title and are separated by 17-point leading.
- Bylines are set in 10/12-point roman caps.
- Paragraphs are indented, except the first, which begins with a 26-point drop-cap.
- The page numbers are set flush with the margin and alternate from left to right.
- Footnotes are set 6/7 point, numbered with leading superscripts, and separated from the narrative by at least four-point leading.

Publication-specific systems have been successfully demonstrated on newspaper pages (Toyoda *et al.* [25], Lam *et al.* [26]), business letters (Dengel [27]), printed tables (Watanabe *et al.* [28]), résumés (Holstege and Tokuda [13]), technical journal articles (Nagy [29], Chevenoy and Belaid [30]), technical reviews (Fujisawa *et al.* [22]), official gazettes (Boccignone *et al.* [31]), trademarks (Kato *et al.* [32]), patent applications (Yamashita *et al.* [33]), typed forms with a prespecified layout (Casey and Ferguson [34]), and, strikingly, the periodical *Chess Informant* (Baird and Thompson [35]). A complex knowledge-based system that demonstrates full interaction between different components was developed for postal address location and interpretation (Srihari *et al.* [36], Lam and Srihari [37]).

The general problem with this approach, however, is the enormous effort required to develop each application-specific layout knowledge base. Among the possible sources of information are direct measurements on the hardcopy, computer-aided measurements on scanned pages, publishers' style manuals, and, for the material prepared on a computerized system, the source code from the system itself. The most challenging approach and the one with potentially the highest yield is nonsupervised estimation to adjust the parameters on the basis of statistical observations in an operational system (Spencer [38]). In a simpler context, the concept was admirably demonstrated by Robert Lucky's adaptive equalizer [39]. Such incremental learning may well be among the more appropriate applications of neural networks.

Many documents contain components that cannot be processed by OCR, such as graphics and halftones. Such fields must be stored as separate elements, usually in some compressed format. It may also be desirable to store the text itself in image form in addition to the coded

(ASCII) representation because of the inevitable loss of some format information, including that of mathematical formulas, during OCR (Nagy and Viswanathan [40]). For browsing and automated retrieval, the graphics and image fields must be accurately delineated and linked to the text itself in some type of structured or hypertext representation (Ingold and Armangil [41]). Automatic insertion of the appropriate links remains a challenge (Fujisawa *et al.* [22], Yashiro *et al.* [42]).

III. Adaptive Classification

The essence of adaptive OCR is to reduce multifont classification to single-font classification by taking advantage of the normal occurrence of long strings of characters in the same typeface.

Most trainable character-recognition algorithms tolerate a reasonable number of mislabeled samples in the training set. Now consider a trainable character recognition system that achieves, say, 95% correct recognition for characters of each class in a new typeface. Five errors per hundred is clearly unacceptable. So let the recognized characters constitute the training set, and redesign the classifier using this training set. Since even with the mislabeled samples (the result of misclassification on the previous round), this training set may well be more representative of the new typeface than the one on which the machine was originally trained, we may expect that if we run the system again on the same character set, we will get a lower error rate. Such is indeed the case (Nagy and Shelton [43], Nagy and Tuong [44]). Furthermore, now that we have a less contaminated set of characters, we can iterate the design-and-classification cycle. Of course, this bootstrapping method works only on homogeneous data and, if the initial performance is very poor, it may get even worse (Nagy [45]). It is expensive in terms of computing resources, but in today's OCR environment saving the user's time at the expense of the machine's may well be acceptable. Neural networks offer a potential alternative (Wilson *et al.* [46]), but their classification performance has not yet been adequately compared with that of the best of the statistical and syntactic methods.

Typeface homogeneity may also help increase throughput. Some commercial systems are loaded initially with a large set of type tables, each of which contains the parameters necessary for classification of a set of similar typefaces. Initially, every type table is applied to each character. However, the system keeps track of which type table is used most successfully, and eventually draws only on that table for classification. A similar idea based on implicit font identification is described by Paoli and Pareschi [19]. For even greater throughput, the complex feature-based recognizer may be replaced by a single-font template matcher (Casey and Jih [47]) or a pixel-based decision tree (Casey and Nagy [48], Boccignone *et al.*[31]). If the confidence level decreases or the reject rate increases beyond a preset threshold, the system reverts to its multifont configuration.

Decision trees are not restricted to single-pixel features, and classification trees of more complex features may be used in less restrictive environments (Pavlidis [49]). Because the extraction of features is itself time-consuming, incomplete classification information from adjacent characters should be used to restrict the search space, and an entire group of characters should be classified by a single process (for instance, size, stroke width, slant, serif indicators, and letter context can propagate). A general method of exploiting the hierarchical structure of both linguistic and graphic document constructs is constraint satisfaction (Mulder *et al.* [50]).

A more radical approach is to let the system start without any preconception whatever of character shapes. Apply a clustering algorithm, such as *k-means*, to either the video or to derived features (moments, projections, lakes-and-bays, lids, stroke orientations and junctions, *n*-tuples, Fourier coefficients, Haar transforms, line-adjacency graphs, etc.) to partition all the samples into roughly the number of classes in the alphabet. Then identify each class either automatically (see below) or by displaying a typical member of the class and letting the user key in its identity. Finally, relabel every sample according to the alphabetic label assigned to its cluster. A fast, single-pass alternative to clustering is a decision-tree generated on the fly (Casey and Wong [8]).

Labeling the clusters after classification takes less time for the user than culling and labeling training samples beforehand or during classification. It is certainly the most economical way of determining just which patterns need to be labeled (Ascher *et al.* [51], Casey and Wong [8]). Clustering has also led to high-performance document-image compression schemes (Ascher and Nagy [52]).

Understandably, commercial OCR systems do not start with *tabula rasa*, but at least one offers something similar in spirit: when a misidentified character is corrected after classification, the user may request the system to relabel every character that was assigned the same label (Meng [53]). This scheme is most useful for relatively rare patterns that vary greatly from typeface to typeface, such as certain punctuation symbols. An efficient way to train a system to a specific typeface is to display word hypotheses generated by an omnifont classifier and a lexicon. The user may confirm the top hypothesis, select one of the other candidate words, or type in the correct labels. This allows adaptation to unusual symbols and ligatures (Paoli and Pareschi [19]).

To label the clusters *automatically*, one must draw on some external source of knowledge. If the material is natural language text, then one can rely either on the expected n-gram frequencies of the language (Casey and Nagy [54], [55]), or on a lexicon (Casey [56]). Letter frequencies are very reliable for identifying the most common letters, but they are vulnerable to anomalies in the clustering process. Short lexicons (500 most common words) are appropriate for identification of all but the most infrequent letters in passages ranging from 300 to 1000 words (Nagy *et al.* [57]). A complete spelling lexicon is generally needed to label q's, j's, and z's. Such a lexicon can also be used at the same time to identify words containing missegmented and misclustered characters. The organization of a lexicon suitable for matching words with spurious insertions and deletions (caused by missegmentation), substitutions (classification errors), and wild cards (unidentified letters) is an open problem (Lettera *et al.* [58], Hoch and Dengel [59]).

Of course, letter context has long been used to correct classification errors (Riseman and Hanson [60], Toussaint [61], Anigbou and Belaid [62]). Specialized lexicons are applied to sorting outgoing mail (country, state, and city names and common abbreviations). But in the method above, the lexicon is used as an integral part of the recognition process, rather than as a postprocessing step. In practice, a method based primarily on letter context would have to be combined with a shape-based method for punctuation symbols and numerals. Even though lexicons of over 110 000 words for English and over 360 000 words for Italian are already available (Paolo and Pareschi [19]), automatic lexicon-compilation techniques will undoubtedly be necessary for specialized domains. In restricted-vocabulary applications, whole-word recognition by shape may be advantageous (Ho *et al.* [63], O'Hair and Kabrisky [64]).

IV. Systems Approaches to OCR

Since different OCR systems do not necessarily make the same mistakes, it is appealing to use several machines and take a majority vote. In order to merge the recognition results, the outputs must be synchronized with respect to the input documents. This is a nontrivial task because of line-finding and character-segmentation errors. In the experiments reported so far, the text was scanned on different scanners, and it was impossible to separate the effects of multiple scans from that of multiple recognition. It appears possible to achieve a decrease in the error rate of a factor of 2 or 3, using three to five commercial OCR systems (Handley and Hickey [65], Bradford and Nartker [66]). Possibly the greatest benefit is that multiple systems are less likely to miss an entire line of print. The results suggest combining different classification algorithms and scanning regimes within the same system.

In some applications one can draw on several sources of information simultaneously. This can be done, for instance, in processing the amount fields on checks. The first bank to handle a check normally receives a deposit slip from the payee. For individual nonbusiness customers, each deposit slip lists (without any specific order) the amounts of several checks, and a total. Usually both the deposit slip and the courtesy field of the check are handprinted, but the writing on the deposit slip may be presumed to be that of the owner of the account. Under some fairly realistic assumptions, one can assign a consistent interpretation to all the amount fields simultaneously (Chang and Nagy [67]). Even imperfect recognition of the (usually cursive) legal amount field would further increase accuracy. Redundancy is built into, and can be exploited in, many financial documents, such as income-tax returns or balance sheets.

So far, we have discussed only the constraints induced by linguistic or arithmetic context. In most documents, however, there is also considerable graphic context. For instance, even complex engineering drawings and maps are usually the work of only a few draftspersons. This permits reject recovery by comparing characters recognized with a low degree of confidence (i.e., patterns that give rise to several almost equally rated candidate identities) to confidently recognized examplars of the top candidates elsewhere in the document. The metric used for comparing two different characters need not be very complex since the primary assumption is that characters of the same class produced by the same source do not vary greatly.

Variations of this method can be applied to printed characters where part of a particular pattern may be too mutilated for initial classification. Confident classification of another instance of the same class may be the result of a more perfect specimen, or of a character occurring in a lexicon word (Casey and Wong [8]).

We are not familiar with any report on the use of external knowledge gained from editing the OCR output to subsequently alter the parameters of the classifier. Yet after contextual or manual postprocessing, a complete, labeled training set is available, without any additional cost. In many applications, such a training set is far more representative of new material to be submitted to the system than the original design set. This information should therefore be used to automatically redesign the preprocessor, the classification algorithms, and the automated part of the postprocessor. The necessary computation can be performed when the system is idle, in a manner completely transparent to the operator (who would notice only the improved performance).

Finally, the notion of *inverse formatting* makes it possible to test the ultimate limits of format analysis and OCR (Okamoto and Miyazawa [68]). Inverse formatting is applicable only to computer-produced documents. Its goal is to reproduce the document-preparation code that produced the original document. Although the mapping between page bit maps and source code is one-to-many, in principle one can always test two source codes for equivalence by comparing the bit maps they generate. Analogously, one may attempt to recover the source code for a drawing produced by drafting software.

Many papers on character recognition are intended primarily to illustrate new, "universal" pattern classification methods. Examples of serious efforts to address the specific problems of converting documents to computer-readable form include those by Kahan *et al.* [14], Paoli and Pareschi [19], Shlien [69], Boccignone *et al.* [31], Bradford and Nartker [66], and Sabourin and Mitiche [70]).

V. Data Sets for OCR Research

The recognition rate, however we choose to measure it, depends primarily on the differences in shape among the most similar characters, and on the variability, layout, context, and quality of the material. We do not have a predictive model for OCR because we do not yet have a mathematically tractable model for the universe of digitized printed documents. We also do not know how to measure quality accurately enough to predict performance on new data. Therefore repeatable, statistically valid comparison experiments on both isolated characters and complete documents are essential for further progress (Nagy [71]).

The IEEE Computer Society's pattern recognition data base was established in 1966. Highleyman's 500-sample data set of hand-digitized characters on punched cards proved immensely popular. Over the years, the U.S. Postal Research Office has released increasingly larger character sets lifted from envelope address blocks. The National Institute of Standards and Technology advertises a CD-ROM sample set with over a million alphanumeric characters by 2100 writers. An even larger volume of scanned text is under preparation at the UNLV Information Science Research Institute. Also useful are the CD-ROM disks of 300-dpi scan-digitized technical journals and conference proceedings released by the IEEE and the ACM, some of which contain ASCII versions of the text. In Japan, the ETL-8 (160 sets of 950 handprinted educational characters) and ETL-9 (200 sets of 3036 handprinted characters by 4000 authors) tapes, prepared by the Electrotechnical Laboratory, have fostered comparative classification experiments.

It is now possible to produce large perfect sets of bit maps of characters using document preparation facilities (Pavlidis [72], Kahan *et al.* [14], Baird and Fossey [73]). The advantage is that one can separate the difficulties caused by the similar shape of characters from those arising from noise introduced in the printing and scanning process. This is done by running the OCR system on the bit map of the page that is sent to the laser printer or phototypesetter, and comparing results with those obtained on printed and scanned versions of the same page.

One can also use document preparation facilities to generate reproducibly noisy characters. By now most researchers understand that noise produced by random bit-switching in binary character images or additive white noise in gray-scale characters does not emulate any significant imperfections found in practice. Realistic noise generators, which incorporate really significant phenomena such as skew, stroke-width variation, and quantization noise, have been developed (Pavlidis [72], Baird [74]). Synthetic character generation offers small research groups the possibility of relatively painless experimentation on huge character samples.

VI. Summary

We have attempted to show that digitized document analysis forms an integral part of OCR. Character classification by itself cannot be expected to produce usable results on complex documents without a correct interpretation of the relationship between larger textual and graphic units. The common thread that runs through this paper is the desirablity of recognizing groups of similar characters rather than individual specimens. In order to extract groups of similar characters, some type of document analysis is necessary.

We have given examples of the varied uses of linguistic context. Although we have not discussed grammatical constructs and semantics, the analysis must be extended to these domains as well (as it already has been in the recognition of continuous speech). Likewise, we need to take advantage of the graphic context arising from the uniformity and homogeneity of pixel patterns of identical symbols within the same document.

We have urged the development of systems that learn. We should aim at learning not through explicit training, but through normal operation. The feedback necessary for learning may be realized by unsupervised parameter estimation rendered possible by the high initial performance of the system, the availability of additional sources of information, or the routine editing of imperfect output.

It would appear that in real-world OCR many different techniques must be combined to yield high recognition rates. Since the performance of individual modules can only be judged in the context of a complete system, academic researchers without access to a collaborative organization are at a disadvantage and cannot hope to match the performance of the best of the current commercial OCR sytems. Widely available modular software for routine subtasks would accelerate progress significantly.

Finally, we listed some new tools for OCR research that may facilitate the tasks ahead. OCR research offers many greener pastures than the overgrazed paddock of isolated character classification.

Acknowledgment

The authors gratefully acknowledges helpful suggestions by R. Bradford (SAIC) and Prof. T. Nartker (UNLV).

References

[1] H. R. Schantz, "The history of OCR," Recognition Technologies Users Association, 1972.

[2] R. B. Hennis, M. R. Bartz, D. R. Andrews, A. J. Atrubin, and K. C. Hu, "The IBM 1975 optical page reader," *IBM J. Res. Develop.* vol. 12, pp. 345–371, 1968.

[3] H. Genchi, K. Mori, S. Watanabe, and S. Katsuragi, "Recognition of handwritten numeral characters for automatic letter sorting," *Proc. IEEE* vol. 56, pp. 1292–1301, 1968.

[4] G. Nagy, "Optical scanning digitizers," *IEEE Computer*, vol. 16, pp. 13–25, May 1983.

[5] G. Nagy, "Optical character recognition—Theory and practice," in *Handbook of Statistics II*, L. N. Kanal and P. R. Krisnaiah, Eds. Amsterdam: North Holland, 1982, pp. 621–649.

[6] R. G. Casey and G. Nagy, "Chinese character recognition: A twenty-five-year retrospective," in *Proc. Int. Conf. Pattern Recognition—ICPR 9* (Rome, Italy), Oct. 1988, pp. 1023–1026.

[7] A. N. Netravali (guest editor), Special Issue on Digital Encoding of Graphics, *Proc. IEEE* vol. 68, July 1980.

[8] R. G. Casey and K. Y. Wong, "Document-analysis systems and techniques," in *Image Analysis Applications*, R. Kasturi and M. M. Trivedi, Eds. New York: Marcel Dekker, 1990, pp. 1–36.

[9] R. Bradford, "Technical factors in the creation of large full-text databases," in *Proc. DOE Infotech Conf.* (Washington, DC), May 1991.

[10] L. A. Dickey, "Operational factors in the creation of large full-text databases," in *Proc. DOE Infotech Conf.* (Washington, DC), May 1991.

[11] F. M. Wahl, K. Y. Wong, and R. G. Casey, "Block segmentation and text extraction in mixed text/image documents," *Computer Graphics and Image Processing*, vol. 2, pp. 375–390, 1982.

[12] K. Y. Wong, R. G. Casey, and F. M. Wahl, "Document analysis system," *IBM J. Res. Develop.* vol. 26, no. 6, pp. 647–656, 1982.

[13] M. Holstege and L. Tokuda, “Visual parsing: An aid to text understanding,” in *Intelligent Text and Image Handling, RIAO Conf. Proc.* (Universitat Autonoma de Barcelona), Apr. 1991, pp. 175–193.

[14] S. Kahan, T. Pavlidis, and H. S. Baird, “On the recognition of printed characters of any font and size,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, no. 2, pp. 274–288, 1987.

[15] P. G. Luca and A. Gisotti, “How to take advantage of word structure in printed character recognition,” in *Intelligent Text and Image Handling, RIAO Conf. Proc.* (Universitat Autonoma de Barcelona), Apr. 1991, pp. 148–159.

[16] J. Kanai, “Text-line extraction using character prototypes,” in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 182–191.

[17] V. A. Kovalevsky, *Image Pattern Recognition*. New York: Springer-Verlag, 1980, ch. VIII..

[18] R. G. Casey, and G. Nagy, “Recursive segmentation and classification of composite character patterns,” in *Proc. Sixth Int. Conf. Pattern Recognition* (Munich), Oct. 1982, pp. 1023–1026.

[19] C. Paoli and M. T. Pareschi, “A system for the automatic reading of printed documents,” in *Office Information Systems: The Design Process*, B. Pernici and A. A. Verrijn-Stuart, Eds. New York: Elsevier, 1989, pp. 311–322.

[20] K. Kubota, O. Iwaki, and H. Arakawa, “Document understanding system,” in *Procs. ICPR-7* (Montreal), 1984, pp. 612–614.

[21] A. Dengel and G. Barth, “Document description and analysis by cuts,” in *Proc. R. I. A. O.* (MIT), Mar. 1988, pp. 940–952.

[22] H. Fujisawa *et al.*, “Document analysis and decomposition method for multimedia contents retrieval,” in *Proc. Second Int. Symp. Interoperable Systems* (Japan), 1988, pp. 231–238.

[23] M. Thomas, “Knowledge representation schemes for document analysis system,” Master's thesis, Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Mar. 1988.

[24] M. Viswanathan, “Analysis of scanned documents—A syntactic approach,” in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 450–459.

[25] T. Toyoda, Y. Noguchi, and Y. Nishimura, “Study of extracting Japanese newspaper article,” in *Procs. ICPR-6* (Munich), 1982, pp. 1113–1115.

[26] S. Lam, D. Wang, and S. N. Srihari, “Reading newspaper text,” in *Proc. ICPR-10* (Atlantic City), June 1990.

[27] A. Dengel, “Document image analysis—Expectation-driven text recognition,” in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 78–87.

[28] T. Watanabe, H. Naruse, and N. Sugie, “Structure analysis of table-form documents on the basis of recognition of horizontal and vertical segments,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 638–646.

[29] G. Nagy, “Toward a structured-document facility,” in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 293–309.

[30] Y. Chevenoy and A. Belaid, “Hypothesis management for structured image recognition,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 121–129.

[31] G. Boccignone, L. Freina, S. Mogliotti, and M. R. Spada, “Toward an evaluation of an experimental OCR system by means of a complex document,” in *Proc. 5th Int. Conf. Image Analysis and Processing* (Positano), pp. 543–550.

[32] T. Kato, H. Shimogaki, and K. Fujimura, “Architecture and user interface of intelligent multimedia database system TRADEMARK,” *Bull. Electrotech. Lab.*, vol. 52, no. 7, pp. 1119–1038, 1988.

[33] A. Yamashita, T. Amano, H. Takahashi, and K. Toyokawa, “A model-based layout understanding method for the document recognition system,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 130–138.

[34] R. G. Casey and D. R. Ferguson, “Intelligent forms processing,” *IBM J. Res. Develop.* vol. 29, no. 3, pp. 435–450, 1990.

[35] H. S. Baird and K. Thompson, “Reading chess,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, pp. 552–559, June 1990.

[36] S. N. Srihari, C. H. Wang, J. J. Hull, and P. Palumbo, “Address-block location: specialized tools and problem-solving architectures,” *AI Magazine*, pp. 25–40, winter 1987.

[37] S. W. Lam, S. N. Srihari, “Multi-domain document layout understanding,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 112–120.

[38] T. Spencer, “Automating the transition from paper to hypertext,” in *Proc. Fourth Annual Rocky Mountain Conf. Artificial Intelligence* (Denver), June 1989, pp. 33–36.

[39] R. W. Lucky, “Techniques for adaptive equalization of digital communication systems,” *Bell Syst. Tech. J.* vol. 45, pp. 255–286, 1966.

[40] G. Nagy and M. Viswanathan, “Dual representation of scanned technical documents,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 141–151.

[41] R. Ingold and D. Armangil, “A top-down document analysis method for logical structure recognition,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 41–49.

[42] H. Yashiro, T. Murakami, Y. Shima, Y. Nakano, and H. Fujisawa, “A new method of document structure extraction using generic layout knowledge,” in *Proc. Int. Workshop Industrial Applications of Machine Intelligence and Vision (MIV-89)* (Tokyo), 1988, pp. 282–287.

[43] G. Nagy and G. L. Shelton, “Self-corrective character recognition system,” *IEEE Trans. Inform. Theory* vol. 12, pp. 215–222, Apr. 1966.

[44] G. Nagy and N. G. Tuong, “On a theorectical pattern recognition model of Ho and Agrawala,” *Proc. IEEE*, vol. 56, pp. 1108–1109, June 1968.

[45] G. Nagy, “The application of nonsupervised learning to character recognition,” in *Pattern Recognition*, L. Kanal, Ed. Washington: Thompson, 1968, pp. 391–398.

[46] C. L. Wilson, R. A. Wilkinson, and M. D. Garris, “Self-organizing neural network character recognition using adaptive filtering and feature extraction,” *Neural Networks*, vol. 3, 1991.

[47] R. G. Casey and C. R. Jih, “A processor-based OCR system,” *IBM J. Res. Develop.*, vol. 27, no. 4, pp. 386–399, 1983.

[48] R. G. Casey and G. Nagy, “Decision tree design using a probabilistic model,” *IEEE Trans. Inform. Theory*, vol. 30, pp. 91–99, Jan. 1984.

[49] T. Pavlidis, “Computer recognition of handwritten numerals by polygonal approximations,” *IEEE Trans. Syst., Man, Cybern.*, vol. 5, no. 6, pp. 610–614, 1975.

[50] J. A. Mulder, A. K. Mackworth, and W. S. Havens, “Knowledge structuring and constraint satisfaction: The Mapsee approach,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, pp. 866–879, 1988.

[51] R. N. Ascher, G. Koppelman, M. J. Miller, G. Nagy, and G. L. Shelton, “An interactive system for reading unformatted printed text,” *IEEE Trans. Comp.*, vol. 20, pp. 1527–1543, Dec. 1971.

[52] R. N. Ascher and G. Nagy, “A means for achieving a high degree of compaction on scan-digitized printed text,” *IEEE Trans. Comp.*, vol. 23, pp. 1174–1179, Nov. 1974.

[53] B. Meng, “Text without typing,” *Macworld*, , pp. 177–184, Oct. 1990.

[54] R. G. Casey and G. Nagy, “Autonomous reading machine,” *IEEE Trans. Comput.* vol. 17, pp. 492–503, May 1968.

[55] R. G. Casey and G. Nagy, “Advances in pattern recognition,” *Scientific American*, vol. 224, no. 4, pp. 56–71, Apr. 1971.

[56] R. G. Casey, “Text recognition by solving a cryptogram,” in *Procs. ICPR-8* (Paris), 1986, pp. 349–351.

[57] G. Nagy, S. Seth, and K. Einspahr, “Decoding substitution ciphers by means of word matching with application to OCR,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 9, pp. 710–715, Sept. 1987.

[58] C. Lettera, M. Masera, C. Paoli, and R. Porinelli, “Use of a dictionary in conjunction with a handwritten text recognizer,” in *Proc. ICPR-8* (Paris), 1986, pp. 699–701.

[59] R. Hoch and A. Dengel, “Fragmentary string matching by selective access to hybrid tries,” in *Proc. ICPR-11* (The Hague), Aug. 1992.

[60] E. M. Riseman and A. R. Hanson, “A contextual postprocessing system for error correction using binary N-grams,” *IEEE Trans. Comput.*, vol. 23, pp. 397–403, 1971.

[61] G. T. Toussaint, “The use of context in pattern recognition,” *Pattern Recognition*, vol. 10, pp. 189–204, 1978.

[62] J. C. Anigbogu and A. Belaid, “Application of hidden Markov models to multifont text recognition,” in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 785–793.

[63] T. K. Ho, J. J. Hull, and S. N. Srihari, “Word recognition

with multi-level contextual knowledge," in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 905–916.

[64] M. A. O'Hair and M. Kabrisky, "Recognizing whole words as single symbols," in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 350–358.

[65] J. C. Handley and T. B. Hickey, "Merging optical character recognition outputs for improved accuracy," in *Intelligent Text and Image Handling, RIAO Conf. Proc.* (Universitat Autonoma de Barcelona), Apr. 1991, pp. 160–173.

[66] R. Bradford and T. Nartker, "Error correlation in contemporary OCR systems," in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 516–523.

[67] S. K. Chang and G. Nagy, "Deposit-slip-first check reading," *IEEE Trans. Syst., Man, Cybern.*, vol. 7, pp. 64–68, Jan. 1977.

[68] M. Okamoto and A. Miyazawa, "An experimental implementation of document recognition system for papers containing mathematical expressions," in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 335–350.

[69] S. Shlien, "Multifont character recognition for typeset documents," *Int. J. Pattern Recognition and Artificial Intelligence* vol. 2, no. 4, pp. 603–620, 1988.

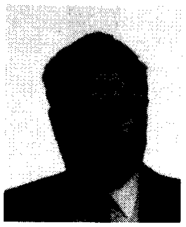
[70] M. Sabourin and A. Mitiche, "Optical character recognition by a neural network," *J. Neural Networks*, in press.

[71] G. Nagy, "Candide's practical principles of experimental pattern recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 5, pp. 199–200, Mar. 1983.

[72] T. Pavlidis, "Effects of distortion on the recognition rate of a structural OCR system," *Procs CVPR-83* (Washington), June 1983, pp. 303–309.

[73] H. S. Baird R. Fossey, "A 100-font recognizer," in *Proc. Int. Conf. Document Analysis and Recognition* (St. Malo, France), Sept. 1991, pp. 332–340.

[74] H. S. Baird, "Document image defect models," in *Pre-proc. IAPR Workshop on Syntactic and Structural Pattern Recognition* (Murray Hill, NJ), June 1990, pp. 78–87.



George Nagy (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees from McGill University and the Ph.D. in electrical engineering from Cornell University in 1962 (on neural networks).

For the next ten years he conducted research on various aspects of pattern recognition and OCR at the IBM T. J. Watson Research Center, in Yorktown Heights, NY. From 1972 to 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 visiting appointments at the Stanford Research Institute, Cornell, the University of Montreal, the National Scientific Research Institute of Quebec, the University of Genoa and the Italian National Research Council in Naples and Genoa, AT&T Bell Laboratories, IBM Almaden, and McGill University. In addition to digitized document analysis and character recognition, his interests include solid modeling, finite-precision spatial computation, and computer vision.