Model-Directed Document Image Analysis

Henry S. Baird

Xerox Palo Alto Research Center,
3333 Coyote Hill Road, Palo Alto, CA 94304 USA

E-mail: baird@parc.xerox.com

Abstract

If current OCR engineering trends continue, then, we believe, "general–purpose" systems — that is, fully automatic and nonretargetable systems — will leave many potential users unsatisfied, and lucrative application niches unfilled, for years to come. However, for users who care enough to volunteer some manual effort — to help customize the system to their document(s) — significantly higher accuracy may be achievable, without delay. We discuss in detail two state–of–the–art document recognition systems — Lucent Technologies' Table Reader System (TRS) and Xerox's "document image decoding" (DID) research prototype — which yield high accuracy by reliance on explicitly stated models of properties of the target document, whether iconic (known typefaces and image degradations), geometric (restricted classes of layouts), or symbolic (linguistic and pragmatic contextual constraints). How great are the performance advantages that can be realized by sacrificing automation in these ways? To what extent can the necessary customizations be (semi–)automated? We outline recent and planned research at Xerox PARC motivated by these questions.

1 Performance of Current OCR Systems

The dominant type of present–day commercial OCR system, whether on the desktop or in service–bureau settings, is designed to operate fully automatically, refusing to accept guidance from the user. The majority of desk–top users welcome this since they are untrained and impatient with inconvenience. There is a similar reliance on more or less completely automatic operation in almost all of the highly specialized OCR application niches such as postal–code and financial–document processing, even though their costly equipment is tended by trained staff in controlled service–bureau settings. In this case, it is largely the daunting throughput requirements that dictate fully automatic operation.

Both of these user communities — the casual SOHO users and the sophisticated special-document users — tolerate surprisingly low performance. The latest competitive studies, at UNLV in 1996 [1], showed, for example, that desk-top OCR packages misrecognize 3–15% of characters — an intolerably high error rate, most users would agree — in over 40% of magazine pages: for other document categories, performance was far worse. The best current systems for reading hand–written courtesy amounts on checks [2] are tuned to reject 33–55% of the input in order to hold substitution errors below 1%. Similarly, the best handwritten postal-address readers fail to "finalize" 35% of the input [3].

All of these technologies are improvable, of course, and are improving: but slowly and at a high cost. The UNLV data suggested that the best desk–top OCR machines have been cutting character error rates by about 15–20% per year [1]. Every sanguine person hopes for sudden breakthroughs in performance – and individual researchers characteristically hope that these will result from isolated technical innovations – but the record of the last ten years does not en-

[0] Invited unrefereed talk presented at the DOD-sponsored Symposium on Document Image Understanding Technology, Annapolis, MD, April 14–16, 1999.

courage such hopes.

Instead, the pattern I see is that overall performance of these increasingly complex systems does not dramatically improve as a result of any single *localized* improvement: say, a more accurate character classification algorithm, or a more refined linguistic model, or a more robust layout segmentor. On the contrary, as year after year the weakest components have been most improved, we are entering a regime where the origins of errors are more evenly distributed among the components of the system. The principal driver of improvements is large-scale empirical testing by ever-growing test data bases, followed by tedious manual analysis of failure cases. These training and testing databases are certainly large and growing larger, but it is not feasible to collect them systematically enough to guarantee coverage of the full cross-product of ranges of typefaces, type sizes, image degradations, layouts styles, scripts, languages, etc etc that occur in practice. Even worse from an engineering pont of view, it is becoming increasingly problematic to isolate one cause, or even the dominant cause, of specific failures. The cause is more and more often a subtle 'conspiracy' among the components of the system which is hard to understand. 'Fixing' one problem breaks another. So for multiple reasons it is often not clear even to the researchers and engineers most familiar with the internals of the machine where they should apply their next year's efforts to achieve the largest gain. Too often, all that can be found to work is a specific manual patch for that particular case. The systems are growing monotonically in the number of lines of code and the number of modules with specialized functions.

The perceptions I list above are not mine alone. At the IAPR DAS'98 Workshop in Nagano, Japan, I took the opportunity to ask three engineering managers of world-class OCR systems about their rate of progress, and the most serious obstacles to progress that they face. Most of them agreed with most of the points above.

I do not mean to paint the bleakest possible picture of the future. Ingenious researchers and engineers continue to solve hard problems. If systems complexity bogs us down, certainly Moore's law buoys us up.

But, overall I feel that OCR engineering current trends support these conclusions:

- the search for more strongly general-purpose, higher-performance document recognition systems will continue to absorb large engineering resources and continue to yield only incremental overall performance improvements;
- since no one system, in markets with many players, is able to sprint ahead of the others, competition on technical grounds will not slacken; and
- most players will have no choice but to continue incremental refinements within their idiosyncratic, slowly evolving, and increasingly complex system architectures.

This is bad news for the many users whose particular documents are poorly served by current machines. They may have to wait years for technology that performs adequately on their class of documents. Potentially lucrative application niches will remain unfilled.

2 The Case for Model-Directed Recognition

One way to summarize the state of the art of OCR systems is that we cannot now, and will not for many years, simultaneously achieve these three desirable properties:

- *high accuracy*, i.e. near-perfect character-by-character transcription;
- *versatility*: applicability to many types of documents, image qualities, etc; and
- *full automation*, requiring no assistance from the user.

How then can research help these many underserved users in the near term?

What if we relax one or more of these goals? If, for example, we attack problems that do not require high accuracy, can we achieve versatility and automation? Yes, clearly: one example is the use of OCR as a front end for word-token-based information retrieval. It has been amply documented that recall and precision are little affected by OCR error rates [4].

What if we sacrifice versatility? There are hosts of successful examples of this approach, from the adoption of the OCR-A font standard to special-forms readers.

What if we sacrifice automation, and so ask the user to intervene manually for each document (or document class)? This is the alternative research direction discussed in this paper. It is, of course, not new: in fact, it was already reasonably well articulated in May of 1992 by a few DIA researchers attending the first DARPA-funded Document Understanding Workshop, held at Xerox PARC. The "Model-Based OCR" panel of the workshop included Phil Chou, Andrew Gilles, Dan Huttenlocher, Tapas Kanungo, Gary Kopec, Prasana Mulgaonkar, Theo Pavlidis, Azriel Rosenfeld, Sargur Srihari, Steve Munt, Steve Dennis, and the present author. We were excited by the potential of a research program that somehow would exploit explicitly specified and often detailed models of the input document in hopes of achieving far higher performance (accuracy and speed) and versatility (range of documents handled) than any of the then–existing systems or their likely successors.

This panel concluded by recommending that DARPA encourage the development of:

(a) "a core technology in which all the assumptions about the writing system, language constraints, context, are explicit such that they can be replaced by new modules [...];"

(b) "alternative architectures and algorithms including promising novel approaches whose initial performance is inferior [...];"

(c) "uniform technology which is transportable across a variety of writing systems [...];" and

(d) "a core technology for developing and using explicit, quantitative, parameterized models of [image] distortion [...]".

It is remarkable to look back, six years later, and see with what tenacity a few of us — Gary and Phil at PARC; myself, David Ittner, and Tin Kam Ho at Bell Labs; and Tapas and Bob Haralick at Univ. Washington — struggled to realize these dreams. Gary and Phil seemed to me to be most committed to goals (a) and (b), while Tapas, Bob, and the Bell Labs folks focussed on (c) with a low-level but persistent pursuit of (d).

All four of these goals were felt to be dauntingly ambitious at the time. They were crafted in conscious contrast to the engineering — and research — methodologies dominant at that time. They are, in fact, continuing today. At considerable risk of oversimplification, and with no desire to understate the creativity, skill, and energy with which they have been pursued, I may characterize them as follows. The emphasis is on modularization of OCR systems into (typically) a pipeline of specialized components performing physical layout analysis and interpretation, isolated-character classification, hypothesize-and-test word segmentation, and contextual analysis. Each of these components is developed to a large degree in isolation from the others. With the exception of image classification and some aspects of contextual analysis, they are not trainable by example but must be substantially hand crafted and manually tuned for good results. They are rarely based on an explicit model of the class of documents to be read, so there is no escape from large-scale (but still unsystematic) empirical testing regimes which inevitably escalate to the limits of affordability. No matter how well the components perform in isolation, their integration is an unpredictable and often frustratingly unstable engineering exercise.

The end result of these dominant methodologies, for most leading OCR technology developers, has been a large and steadily growing software suite which is difficult to improve systematically and which therefore drains larger and larger engineering resources in return for chronically incremental performance improvements. As tempting as it must often be to restart from scratch and rearchitect more rationally, their large investment in code and the uncertainties of the OCR state of the art argue against radical course corrections. It was this morass of individually plausible but collectively *ad hoc* methods that the panel foresaw and were trying to circumvent.

What progress has been made towards these four "Model–Based OCR" goals, and what should be attempted next? The rest of this paper gives a partial answer to these questions: partial in that it emphasizes work in which the author has been, and remains, personally involved.

The next two sections describe two model–directed OCR systems which embody many of these principles. The first is a retargetable table–reader product developed by a team in Bell Laboratories (including the present author), first used on a large scale within AT&T, and now offered for sale by Lucent Technologies. The second is an experimental prototype within Xerox PARC,

whose development was led by Gary Kopec and Phil Chou, and which has been successfully applied to a variety of uniquely challenging documents, especially in the context of the UC Berkeley Digital Library Initiative project. Although Phil has left Xerox and Gary died in December 1998, extensions and refinements of the DID system remain active topics of research at PARC by a team that includes the present author. We list a number of open research problems, engineering challenges, and opportunities for feasibility trials and joint work.

3 A Retargetable Table Reader

At least one model–directed, manually retargetable document image analysis system exists and is heavily used today. It is a system for reading machine–printed documents in known predefined tabular–data layout styles [5] (telephone bills, to be precise). In these tables, textual data are presented in 'record' lines made up of fixed–width fields. Tables often do not rely on line–art (ruled lines) to delimit fields, and in this way differ crucially from fixed forms. This table–reader system performs these steps: identifies multiple tables per page; identifies records within tables (ignoring non–record text); segments records into fields; and recognizes characters within fields, constrained by field–specific contextual knowledge.

Obstacles to good performance on these tables included small print, tight line–spacing, poor–quality text (such as photocopies), and line–art or background patterns that touch the text. Precise skew–correction and pitch–estimation, and high–performance OCR using neural nets proved crucial in overcoming these obstacles. However, the principal obstacle to building a system of this sort was the wide variability of layouts among the hundreds of table form types encountered. The variability would overwhelm any fixed, fully automatic system; if each distinct "form model" had to be manually specified, then the retargeting effort must be small and "deskilled." Therefore the most significant technical advances in this work appear to be algorithms for identifying and segmenting records with *known layout*, together with the integration of these algorithms with an efficient graphical user interface (GUI) for *defining new layouts*.

Unlike most prior work on forms and table analysis, the system does not depend on guidance from line–art or fiducial marks. The operator describes a new layout model by annotating images of a sample page (noting the location of fields, and whether certain characters are required or optional, etc). This example is thus abstracted into "record–line template" which is matched (using simple convolution–based methods) to every text–line in the image, to distinguish record lines from non–record text and to splt each record line into fields. The model–specification GUI has been ergonomically designed to make efficient and intuitive use of exemplary images, so that the skill and manual effort required to retarget the system to new table layouts are held to a minimum. In fact, each tabular layout model can typically be specified in less than 15 minutes by a clerk with data–entry skills.

In short, the system succeeds because a user can quickly specify a layout model which can then be effectively and fully automatically applied to every page of tables of the same layout. The system has been applied in this way to more than 400 distinct tabular layouts. Over a period of three years the system read over fifty million records with high accuracy. Large scale tests have shown that the system fully automatically achieves 97% to 99.98% characters correct. The GUI also supports manual correction, which typically yields a semi-automatic accuracy of greater than 99.99%.

This performance is so much higher than any previously published on tables, and the range of table–types handled is so much greater than any previous commercial table–reader system, that it is tempting to assert that the key determinants of success were (a) restriction to known predefined layouts and (b) exploitation of field–specific context. That is, manual specification and automatic exploitation of detailed models.

Thus, this table reader system (now offered for sale by Lucent Technologies) is an example of a model–directed OCR system of the type we envisaged. It has successfully colonized a previously underserved application niche.

It is significant that this application niche is a service-bureau operation, where the operating staff (however non-technical their entry skills) can be trained and managed, and where engineers are available to back them up in the occasional difficult case. This is a far cry from desk-top casual-use OCR.

4 The Document Image Decoding Prototype

As early as 1990 Gary Kopec and Phil Chou of Xerox PARC were consciously adapting to OCR the paradigms characteristic of the early days of signal processing research, especially the communications–theory framework [6]: applied to document images, this views any observed document image as a signal which has been synthesized through several distinct stages: the underlying message (e.g. the ASCII text) is first "encoded" as an ideal image by choices of typefaces and page layout, and this ideal image is, in turn, "degraded" by noise introduced during printing and scanning, yielding the observed image. Recognition is then viewed, in this framework, as an attempt to "decode" the observed signal by estimating the most probable transmitted message, among all messages implied by the models, that may have led to it. The models of encoding that Gary and Phil used usually involved probabilistic finite–state machines and rigid character template images. The typical model of degradation was probabilistic asymmetric bit-flip.

Gary and Phil's collaboration was, it seems to me, distinguished from the work of their peers most clearly by two principles:

- every stage of the system is explicitly modeled; and
- the system, as a whole, is simultaneously optimized by minimizing the expected "loss" between the message sent and the message decoded.

Everyone else in the DIA field — including myself — backed away from one or both of these principles, at times, in the face of theoretical difficulties or from a desire to exhibit a near-term practical success.

In the face of many technical difficulties Gary, Phil, and their collaborators managed to illustrate many strengths of this approach [7,8,9,10,11]. They showed that their family of encoding models – probabilistic regular grammars, sometimes attributed — was rich enough to capture not only plain text but textual markup, logical layout labeling, highly structured technical text and tables, and mathematical expressions — even music notation. By insisting that the system be optimized simultaneously as a whole, not a single component at a time, they obviated several artificial distinctions — notably between recognition and segmentation of characters — which trigger complexity, confusion, and errors in other systems. They showed that the optimal decoding (for a 0–1 loss function) could be approximately found by a segmental Viterbi search through the 2-D trellis implied by the composition of the synthesis models. The models were formally and practically separate from the recognition (search) engine, and as a result many ways were found to improve (e.g. speed up) the search engine independent of any model. They found ways to infer some aspects of the models — e.g. character bi–gram probabilities and character templates — automatically from ground–truthed training data (using maximum–likelihood estimation), thus reducing the effort to retarget the system to particular documents.

Perhaps most impressively, from the point of view of potential users of the system, they showed repeatedly that it could *drop the character error rate, by up to an order of magnitude* in many cases, compared to commercial OCR systems. There are well-understood technical reasons for this extraordinary advantage. Our decoding algorithm gives, by rigorous probabilistic search, the best possible result given the model and the scanned image: the result is exactly that data which is most likely to give rise to the printed and scanned image. Thus although our results can be improved using a better model — a more complex, more specific model that fits the document better — nevertheless whenever we use a specific model we do as well as possible consistent with it.

Further, by judicious use of attributed grammars in modeling the encoding stage, the logical structure of text — e.g. the functional parts of a dictionary entry — can be captured and preserved, as a beneficial side–effect of recognition. Few if any commercial OCR systems offer such a feature; the manual effort to add the structural tags to the plain ASCII that they produce is usually prohibitive.

As of a year ago, certain weaknesses were nevertheless still apparent. The asymmetric bit-flip model of degradation had proven brittle in practice; later extensions to "multi-level templates" allowed close approximation of arbitrary blur and additive noise, but not to other common degradations such as affine distortions. First attempts to incorporate language models richer than uni-

gram character probabilities caused an explosion in time complexity. In spite of the fact that, given a modest amount of ground-truthed training data, character templates could be learned almost fully automatically, it was still the case that the manual effort and technical skill required to use the system was often excessive. Compounding this was the fact that the system was composed from routines in several languages (both C and LISP).

But perhaps the most serious deficiency of the system was its low speed: it often ran two orders of magnitude or more slower than competing commercially available systems.

Happily, within the past year, significant progress has been made on some of these fronts. Algorithmic improvements to the search — not yet published — have yielded an *order of magnitude speed-up*, with no loss of accuracy or generality, over a large test set. All of the system components needed for ordinary use (on, e.g., English text) is now written entirely in Python and C, is readily portable to several computing platforms, and is thus able to be shared with collaborators.

This system is now ready for further feasibility trials. Some trials will be carried out this summer, in close association with the UC Berkeley Digital Library Initiative project. We are selecting one or more botanical reference books which are effectively illegible by commercial OCR systems for various reasons (uncommon typefaces, low image quality, or highly structured text), and whose contents are not yet on–line and would complement the already large and useful data base assembled in the UCB 'CalFlora' website (cf. http://elib.cs.berkeley.edu/calflora/botanical.html). We intend to retarget the DID system to each of these books, and thus provide, through the UC Berkeley Digital Library, unique scholarly resources to the botanical research community, years earlier than existing commercially available OCR systems could make possible.

So, in summary, our present technology offers a tradeoff: far higher accuracy and (uniquely) preservation of structure *versus* some manual start-up effort and significantly longer runtimes. This contrasts with current commercial OCR packages, which require no manual effort and are much faster, but which are oblivious to the document's structure and whose accuracy is fixed and unimprovable. If their error rate happens to be too high on your document, you have no way, short of manually correcting the output, to improve it. The actual trade-offs that are achievable in practice with the DID system appear to depend strongly on details of each document and the workflow surrounding it.

We understand in general terms how to pick different operating points on the DID trade-off curves: for example, how to reduce error by using more complex, and therefore more restrictive, grammars. More complex grammars not only often reduce error, but they allow more refined tagging of the output. Generally the more complex the grammar and the more symbols and typefaces that are expected, the slower and more expensive the decoding: but we are exploring new heuristics that promise speed-ups with no sacrifice of accuracy or tagging.

The most promising immediate future directions for DID research, it seems to us in the DID area at PARC, include:

- incorporating language models inferable from corpora, without large speed penalties;
- incorporating more realistic image degradation models (e.g. [12] or [13]); and
- further 'deskilling' of the retargeting task to bring it within the reach of non-expert users.

We believe the time has come to look outside PARC for commercially attractive applications where these trade-offs can be concretely explored. Here is a sketch of a possible field trial of the decoder software, as part of a semi-automatic workflow requiring the conversion of a sequence of documents to text with an accuracy far higher than commercially available OCR systems can uniformly provide.

> The engineer in the field, at first working closely with PARC, will:
>
> 1. select, from the set of documents to be converted, those which are most likely to benefit from decoding: these will typically be relatively long (tens or hundreds of pages) and possess uniform printing characteristics (e.g. only a few fonts and type sizes, and similar image 'quality');
> 2. manually transcribe – or merely correct the commercial-OCR output of

– a subset of each document (a few pages at most);

3. run our automatic typeface-inference tool;
4. specify the document layout grammar and design the output encoding, by editing a special file (for many documents, a good model may already exist, and can be merely taken 'off the shelf'); and
5. run the decoder on the complete documents, for far higher accuracy and detailed structural tagging.

We would be happy to discuss joint feasibility trials or collaborative research with interested parties.

5 Conclusions

We have argued that present OCR engineering practice will leave many potential users underserved for years to come. In the meantime, motivated users who are willing to invest some effort in manually customizing a retargetable OCR system to their (class of) document(s) may succeed. We have shown that model-directed, manually retargetable OCR systems have made substantial progress since their inception almost a decade ago. Successful applications have been built: at least one is heavily used. Laboratory prototypes are making steady progress, and are ready for extended feasibility trials.

Acknowledgements

The Table Reader System was developed by John Shamilian, Tom Wood, and the present author, with significant early help from Michele Battista (all then at Bell Labs). The DID system was the creation primarily of Gary Kopec and Phil Chou (both then at Xerox PARC). The DID area at PARC in the past year included Gary Kopec, Dan Bloomberg, Les Niles, Kris Popat, Tom Minka, and the present author. Kris Halvorsen, their Lab manager, supported and encouraged the research direction described here. We are indebted also to Profs. Robert Wilensky and Richard Fateman, and to Taku Tokuyasu, of the UC Berkeley Digital Library project.

References

[1] S. V. Rice, F. R. Jenkins, & T. Nartker, "The Fifth Annual Text of OCR Accuracy", *UNLV Information Science Research Institute Annual Report*, Las Vegas, NV, April 1996.

[2] C. Y. Suen, K. Liu, & N. W. Strathy, "Sorting and Recognizing Cheques and Financial Documents," *Proc., IAPR 1998 Workshop on Document Analysis Systems (DAS'98)*, Nagano, Japan, pp. 1-18, November 1998.

[3] A. Filatov, N. Nikitin, A. Volgunin, & P. Zelinsky, "The AddressScript Recognition System for Handwritten Envelopes," *Proc., IAPR 1998 Workshop on Document Analysis Systems (DAS'98)*, Nagano, Japan, pp. 222-236, November 1998.

[4] J. Shamilian, H. S. Baird, & T. Wood, "A Retargetable Table Reader," *Proc., IAPR 1997 Int'l Conf. on Document Analysis and Recognition*, Ulm, Germany, August 18-20, 1997.

[5] K. Taghva, J. Borsack, A. Condit, S. Erva, "The Effects of Noisy Data on Text Retrieval", *Journal of the American Society for Information Science*, pg. 50-58, vol. 45, 1994.

[6] G.E. Kopec and P.A. Chou, "Document image decoding using Markov source models," Asilomar Conf. On Signal, Systems, and Computers, October 1992. Expanded for publication in *IEEE Trans. Pattern Analysis and Machine Intelligence*, June 1994.

[7] G.E. Kopec, P.A. Chou, and D. Maltz, "Markov source models for printed music decoding," *J. Electronic Imaging*, January 1996.

[8] A. Kam and G.E. Kopec, "Document image decoding by heuristic search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, September 1996.

[9] G.E. Kopec and M. Lomelin, "Supervised template estimation for document image decoding," *IEEE Trans. Pattern Analysis and Machine Intelligence*, December 1997.

[10] G.E. Kopec, "An EM algorithm for character template estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, submitted for possible publication.

[11] G.E. Kopec, "Multilevel character templates for document image decoding," *Proc. SPIE Document Recognition IV*, San Jose, CA, February 1997.

[12] H. S. Baird, "Document Image Defect Models," in H. S. Baird, H. Bunke, and K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag: New York, 1992, pp. 546-556.

[13] T. Kanungo, R. M. Haralick, and I. Phillips, "Global and Local Document Degradation Models," *Proceedings, IAPR 2nd ICDAR*, Tsukuba, Japan, October 20-22, 1993.