Automatic Prototype Extraction for Adaptive OCR

George Nagy Yihong Xu

Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180 USA

Abstract

A Bayesian method of isolating character bitmaps from paragraph-length samples of heavily degraded text images is demonstrated. The method requires a transcript of the text, but it is sufficiently robust to tolerate errors in transcripts obtained from multifont commercial OCR software. The resulting prototypes (labeled character images) are used to recognize additional text in the same document.

1 Introduction

The research reported here is motivated by (1) the growing consensus that a representative (but not necessarily large) training sample is more important for classification accuracy than the specific choice of features or classifiers [HB94], and (2) the accumulation of evidence that character-level segmentation followed by recognition is inadequate for many printed pages submitted to OCR [Bokser92, CL96].

The most representative training sample for a given document is a set of character samples drawn from the same document. However, trainable OCR systems are unpopular because they impose a burden on the operator. We are therefore concentrating on minimizing the human intervention necessary to extract character samples from a scanned page image.

The recognition process itself can then be based on a few training samples extracted from the same document. This reduces the OCR task to an essentially single-font recognition problem. Most shape and size variations due to the choice of typeface, and to gross printing, copying and scanning effects, are eliminated. Under these circumstances, we believe that judicious use of template matching techniques, which have long proved their worth in single-font applications [Ullman73], are adequate. A major benefit of template matching schemes is that, unlike many feature-based methods, they do not require prior segmentation of the printed text into isolated character blocks.

Our prototype extraction method is based on comparing the bitmaps of pairs of words that contain the same character. Extending the recognition process from characters to words and using document-specific methods has been long advocated by Jonathan Hull and by Larry Spitz, and their colleagues [HH95, Spitz95]. Rolf Ingold used character identities derived from the text under consideration [Ingold90]. Among the most exciting current methods for document-specific OCR are the sophisticated dynamic programming approaches developed by Gary Kopec and his colleagues [KC94, KL96]. Kopec also derived least-squares estimation of sidebearing parameters from pairs of character images [Kopec93].

The contribution presented here is a sound Bayesian method for determining matching columns of pixels that correspond to the same character, in selected pairs of labeled word blocks. An earlier version of prototype extraction, based on recursive segmentation of word blocks, was presented at DAS96 [NX96].

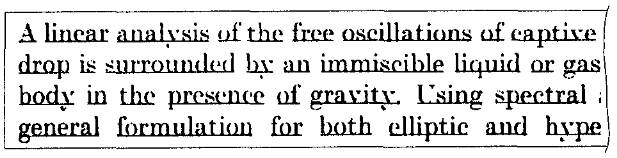
Our improved method is based on the observation that when two bitmaps of word blocks are shifted over each other and correlated pixel column by column, high similarity values between columns occur in windows that contain the same letter in the two words. The known sequence of symbols in the two words can be used to extract the bitmaps of the pair of matching characters. This process can be performed efficiently for every pair of matching characters. The robustness of the algorithm derives from the redundancy of matching each character to several other samples of the same character.

A transcript may be obtained by key entry. Keying in a few lines of text takes a good typist less than a minute. Alternatively, we can use the imperfect output of a commercial OCR package on the same image. The transcript is used to find every pair of words that contain matching characters and to estimate the position of each character within each word.

In a typical paragraph of printed text, all but the least frequent lower-case letters (x, q, z, j) tend to appear more than once. Common letters such as a or s may appear in dozens of words. The frequent letters, for which prototypes can be extracted from a paragraph or two of labeled text, statistically account for over 95% of all text in the entire document. Improved recognition performance on these common letters therefore substantially reduces the overall error rate.

In a complete system, the remaining text - including, for instance, occasional italics - would be recognized by a static multifont classifier with a higher error rate.

Preprocessing consists of line finding, word segmentation, and word baseline location. Figure 1 shows the result of preprocessing four lines of text from Document A065 of the University of Washington database [PCHH95]. The image was obtained at 300 dpi with a bi-level scanner.



A linear analysis of the free oscillations of captive drop is surrounded by an immiscible liquid or gas body in the presence of gravity. Using spectral general formulation for both elliptic and hype

Figure 1: Broken text scanned at 300 dpi.

The prototypes are extracted by the word-shift algorithm. After a set of prototypes has been obtained for every identifiable character in the labeled text image, the class-conditional pixel probabilities are estimated for each class of symbols. Each probability array is called a *template.* To recognize new text, the algorithm chooses the combination of templates that produces the highest probability for the pixel array of each word. The result is a string of symbol identities for each word in the new text.

2 Word Shift Algorithm and Prototype Extraction

The core of the method is the word shift algorithm. Its input is a set of scores S that reflect the similarity between every pair of columns in each pair of words that contain at least one instance of the same letter. The similarity scores are converted to probabilities. Next, the *a posteriori* probabilities of candidate windows are computed for every probable location and width of the target character. The pair of character bitmaps with largest *a posteriori* probability is saved.

The word shift algorithm makes use of prior probabilities for width and location. These are obtained as follows.

The width of each class of characters is estimated by solving multi-variable regression equations where the variables are the widths of the characters. The sums of the character widths are equal to the known word lengths, and constitute an overdetermined linear system which is solved for the mean width of each class of characters, and for the inter-character spaces.

The location of each character with respect to the beginning of the word is the sum of the widths of the preceding characters and inter-character spaces. (For characters nearer to the end of the word, we compute the position from the right.) We consider the width of each character as a random variable, and obtain the probability distribution of the sum of the widths of a string of characters by convolving the individual width distributions. The estimated locations of the characters in two words are shown in Figure 2. These estimated widths and character location probabilities are used in the Bayesian matching algorithm as priors.

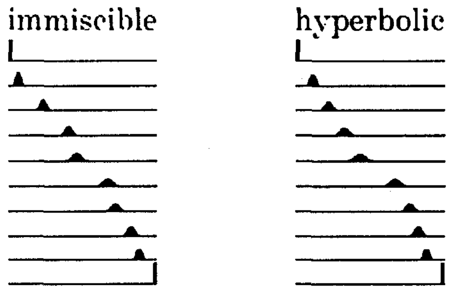
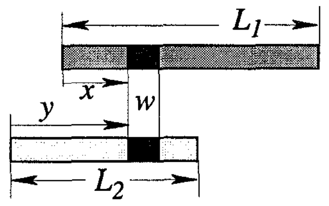


Figure 2: Probability distribution of character location for two words, estimated from both ends.

The similarity scores are converted to match and non-match probabilities using a generic table based on histograms of similarities for columns from matching and non-matching characters. The same table is used in every example. When the location and width of the window corresponds to a pair of matching characters, then high similarity scores are probable for corresponding pairs of columns within the window, and low similarity scores are probable for all other column pairs. Therefore the window that contains the target character in both words will have the highest *a posteriori* probability.

The pseudocode of the word shift algorithm is shown in Figure 3. The match probability conditioned on the similarity score is $P(x, y, w|S)$. The range $W^+_{x,y}$ of column scores consists of every pair of columns within the window w located at x in word 1 and y in word 2. $W^-_{x,y}$ consists of every other column pair in the two words, at every (x, y). $P(x), P(y), P(w)$ are the previously estimated *a priori* probabilities for

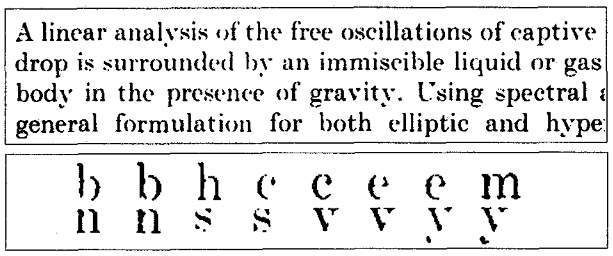


```
// x, y: location of window in each word
// $P_+(\cdot)$: match probability
// $P_-(\cdot)$: no-match probability
// P(x), P(y): location probability
// P(w): width probability
// $L_1$: width of word 1 ;  $L_2$ : width of word 2

PROCEDURE word-shift( )
  FOR $\forall x, \forall y, \forall w$ DO
    $P(x, y, w \mid S) \propto$
        $P(x)\,P(y)\,P(w) \prod_{i \in W^+_{x,y}} P_+(S_i) \prod_{i \in W^-_{x,y}} P_-(S_i)$
        $= Q(x, y, w);$
  END
  $(x^*, y^*, w^*) = \text{argmax}\, Q(x, y, w);$
END

Output:
Extract $w^*$ columns from each word at shift
position $x^*$ and $y^*$ respectively.
```

Figure 3: Pseudocode for word shift and character bitmap extraction.

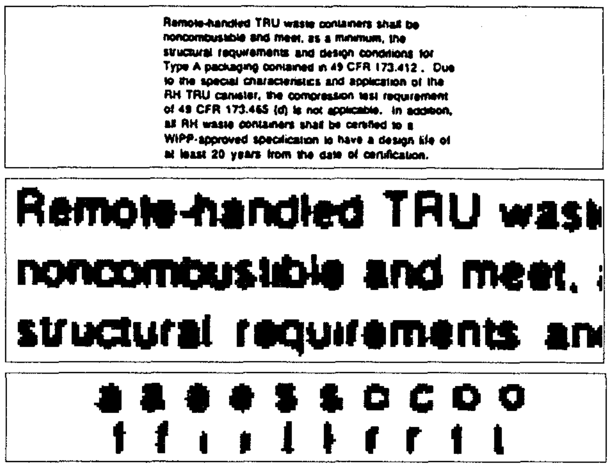

A linear analysis of the free oscillations of captive
drop is surrounded by an immiscible liquid or gas
body in the presence of gravity. Using spectral
general formulation for both elliptic and hype

b b h c c e e m
n n s s v v y y

Figure 4: Top-left part of text, above, reproduced full scale. Prototypes, below, enlarged 2X. [b,b,h,c,c,e,e,m,n,n,s,s,v,v,y,y].

location and width (x, y, and w are assumed to be independent of one another).

Prototype extraction is illustrated in Figures 4-6 with three examples from the UW database. Below each paragraph from which the prototypes were extracted are representative examples of the prototypes of several classes. We selected typical prototypes from several confusion pairs in order to emphasize the poor quality of the print in the chosen examples.


Remote-handled TRU waste containers shall be
noncombustible and meet, as a minimum, the
structural requirements and design conditions for
Type A packaging contained in 49 CFR 173.412 . Due
to the special characteristics and application of the
RH TRU canister, the compression test requirement
of 49 CFR 173.465 (d) is not applicable. In addition,
all RH waste containers shall be certified to a
WIPP-approved specification to have a design life of
at least 20 years from the date of certification.

Remote-handled TRU wast
noncombustible and meet,
structural requirements an

Figure 5: Text, above, reproduced full scale. Beginning of the text, middle, enlarged 4X. Prototypes, below, enlarged 4X. [a,a,e,e,s,s,c,c,o,o,f,f, i,i,l,l,r,r,t,t].

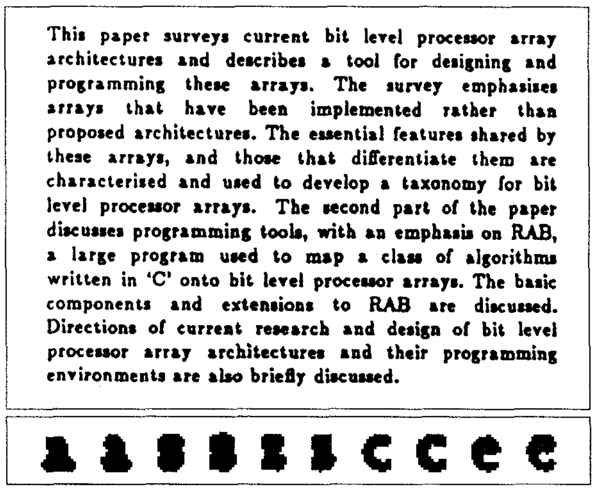

This paper surveys current bit level processor array architectures and describes a tool for designing and programming these arrays. The survey emphasises arrays that have been implemented rather than proposed architectures. The essential features shared by these arrays, and those that differentiate them are characterised and used to develop a taxonomy for bit level processor arrays. The second part of the paper discusses programming tools, with an emphasis on RAB, a large program used to map a class of algorithms written in 'C' onto bit level processor arrays. The basic components and extensions to RAB are discussed. Directions of current research and design of bit level processor array architectures and their programming environments are also briefly discussed.

Figure 6: Text, above, reproduced full scale. Prototypes, below, enlarged 4X. [a,a,s,s,z,z,c,c,e,e].

The first example (A065 in the UW database) has uneven weight and some broken characters. The second example (J04E) is in a very small and badly blurred type. In the third example (I000), some classes, especially s and z, are virtually undistinguishable to the eye.

The extracted prototypes are combined in a probability array to form a template for each class. The templates for the first example are shown in the form of gray-scale maps in Figure 7. In the next section, we describe the recognition of new text from the same pages, using the templates constructed from the extracted prototypes.

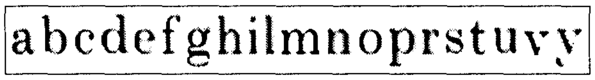


Figure 7: Template probabilities from the prototypes of Figure 4, shown in gray scale.

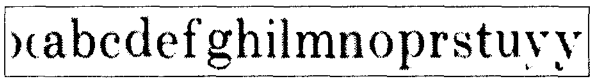


Figure 8: Templates obtained with an imperfect transcript.

To demonstrate the robustness of the algorithm to labeling errors, instead of the correct transcript we use the transcript generated by a commercial OCR software. We also use the error-prone word-segmentation boundaries determined by the OCR software. The OCR output has 37 errors on the 280 character training set of A065, as counted by a longest-common-substring algorithm. Note that using the training feature of the OCR software does not improve its results much because of the many segmentation errors and because the package fails to pinpoint the error-prone classes. Nevertheless, the class-conditional pixel distributions produced by our algorithm (Figure 8), are similar to those obtained from the correct transcript (Figure 7). The extra template classes ')' and '(' are due to broken-character recognition errors.

3 Recognition

The recognition of new text without character segmentation requires estimation of the best match among the prototypes in each column position. The best sequence of prototypes is selected by *level-building*, a dynamic programming algorithm used in speech recognition with Hidden Markov methods [RL85]. The templates used for recognition include a single-column "blank", one or more instances of which may be inserted between character templates to improve the overall match.

The results of the prototype extraction and recognition experiments are shown in Table 1 for the three examples. The Table shows the number of characters in the training set (row 1) and the test set (row 2), the number of prototypes that were extracted from the training set (row 3), the number of characters in the test set from the classes represented by these prototypes (row 4), the classification error rate on these characters (row 5), and the error rate of a commercial OCR device on the entire test set (row 6).

Table 1: Recognition results.

UW filename	A065	J04E	I000
No. of characters (training set)	280	418	656
No. of characters (test set)	274	437	4094
No. classes (with prototype)	21	39	29
No. of admissible characters	265	339	3936
Error on admissible text	2/265 0.75%	77/339 19.7%	170/3936 4.32%
Commercial OCR error on test set	25/274 9.12%	262/437 60%	869/4094 21%

We have not yet included any provision to recognize the characters for which no prototypes are available. If the *a posteriori* match probability of the template responsible for any symbol is too low, a reject symbol is inserted. Having a reject option is mandatory in our method because the training set does not generally contain samples of every symbol class (e.g. q, W in Figure 9). In principle, the symbol string produced by our method could be aligned with the symbol string produced by the omnifont OCR, and most reject symbols replaced by the omnifont OCR output.

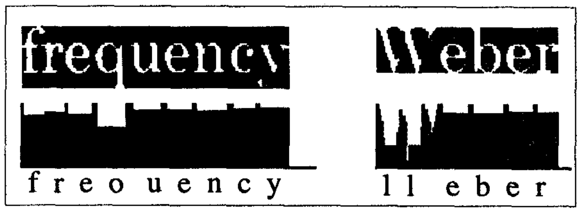


Figure 9: Low scores due to missing q, W templates.

As indicated in Table 1, the material in all three examples is well beyond the capabilities of current omnifont OCR devices, even with their extensive letter n-gram tables and lexicons. For instance, the error rate of a commercial OCR on the second example is 60% (yes, really.) However, we understand that the sample sizes are far too small to draw significant conclusions, and present these examples for illustrative purposes only.

4 Conclusions

The availability of high-speed processors with large internal memories offers an opportunity to revisit adaptive learning methods for OCR. Some commercial systems already make use of limited forms of adaptation, but their methods are seldom revealed to the research community.

The method presented here is not comparable with current commercial OCR systems with their extensive routines for special bitmap configurations, punctuation, capitalization, and morphological and lexical context. It is, nevertheless, a demonstration of the power of adapting the classifier to each document. We are now working on improving the following aspects of our system:

- Modification of the match function and level-building programs for effective template matching on kerned text.
- Development of a robust method for combining the outputs of our classifier and of the multifont OCR device. This requires an OCR device that provides word-block coordinates.
- Reduction of the effect of word-segmentation errors, and extending the method to unsegmented Chinese and Japanese text.
- Comparison of the Bayesian word shift prototype extraction with HMM for printed text [BK94, EI95], and with the elaborate methods of [KC94, KL96], to determine their relative advantages and limitations.

Acknowledgments

The authors gratefully acknowledge the support of the Central Research Laboratory, Hitachi, Ltd. and of the Nortel Educational Research Network. This work has been conducted in the New York State Center for Advanced Technology (CAT) in Automation, Robotics and Manufacturing at Rensselaer, which is partially funded by a block grant from the New York State Science and Technology Foundation. We thank the ICDAR referees for their thorough reviews and sound suggestions.

References

[BK94] C. B. Bose and S. S. Kuo, Connected and Degraded Text Recognition Using Hidden Markov Model, *Pattern Recognition*, vol. 27, no. 10, pp. 1345-1363, 1994.

[Bokser92] M. Bokser, Omnidocument Technologies, *Proceedings of the IEEE*, 80, 7, pp. 1066-1078, July 1992.

[CL96] R. G. Casey and E. Lecolinet, A survey of methods and strategies in character segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-18, 7, pp. 690-706, July 1996.

[EI95] A. J. Elms and J. Illingworth, The Recognition of Noisy Polyfont Printed Text Using Combined HMMs, *Procs. SDAIR-95*, pp. 203-216, Las Vegas, April 1995.

[HB94] T. K. Ho and H. S. Baird, Asymptotic Accuracy of Two-class Discrimination, *Procs. SDAIR-94*, pp. 275-288, Las Vegas, 1994.

[HH95] T. Hong and J. J. Hull, Character Segmentation Using Visual Inter-word Constraints in a Text Page, *Procs. SPIE vol.2422*, pp. 15-25, San Jose, 1995.

[Ingold90] R. Ingold, *Structure de documents et lecture optique: une nouvelle approche*, Doctoral dissertation, Ecole Polytechnique Federale de Lausanne, Presses Polytechniques Romandes, Lausanne, Switzerland 1990.

[Kopec93] G. Kopec, Least-squares font metric estimation from images, *IEEE Trans. on Image Processing*, vol.2, no.7, pp. 510-519, Oct. 1993.

[KC94] G. Kopec and P. Chou, Document Image Decoding Using Markov Source Models, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-16, 7, pp. 602-617, July 1994.

[KL96] G. Kopec and M. Lomelin, Document-specific character template estimation, *Proc. SPIE vol. 2660*, pp. 14-26, San Jose, 1996.

[NX96] G. Nagy and Y. Xu, Priming the Recognizer, *Procs. DAS-96*, Malvern, PA, pp. 263-281, 1996.

[PCHH95] I. T. Phillips, S. Chen, J. Ha, R. M. Haralick, English document database design and implementation methodology, *Procs. SDAIR-95*, pp. 65-104, Las Vegas, 1995.

[RL85] L. R. Rabiner and Bing-Hwang Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.

[Spitz95] A. L. Spitz, An OCR Based on Character Shape Codes and Lexical Information, *Procs. ICDAR-95* , Montreal, pp. 723-728, 1995.

[Ullman73] J. Ullman, *Pattern Recognition Techniques*, Butterworth 1973.