OPTICAL CHARACTER RECOGNITION OF TOUCHING CHARACTERS

S. Shlien and K. Kubota[1]

Communications Research Centre
3701 Carling Avenue
Ottawa, Ontario, Canada
K2H 8S2

[1]Visiting scientist from Nippon Telegraph and Telephone
Yokosuka Electrical Communication Laboratory,
Yokosuka-shi, 238 Japan

ABSTRACT

A method for isolating and recognizing characters in typeset text containing touching or overlapping characters is described. The method relies on vector quantization techniques to represent the text by an ordered sequence of codes. Characters or character fragments are extracted from this code using a form of string matching and network searches. Compatibility relationships are applied to these fragments to obtain a list of possible input strings. Our simulations on the Symbolics 3640 workstation do not indicate that this method would replace conventional template matching schemes.

KEYWORDS: optical character recognition, template matching, vector quantization, consistent labeling problem.

INTRODUCTION

With the continuous drop in the price of computer memory and VLSI components, it is easy to envision new and more powerful OCR devices becoming available. Such devices would utilize several different recognition strategies depending upon the quality of the input; they would be capable of learning a new font without human supervision; and they would contain large spelling dictionaries for applying contextual postprocessing [1-4]. Software development costs rather than the cost of hardware components would be the limiting factor to building such a system.

Automatic reading machines can process good quality typescript input at error rates approaching those of the average typist, however, only a few machines can handle typeset material containing proportionally-spaced characters [5]. A recurrent problem is the presence of touching characters which invariably results in segmentation and misclassification errors (eg. rn confused with m). Without gaps separating characters, the character extraction and classification process can no longer be performed independently, and this increases the complexity of the recognition problem considerably.

There have been several studies [6-8] on the segmentation problem of touching characters. Some excellent results have been obtained using a combination of dynamic programming and template matching

[8]. A new and promising approach uses word shape matching [9]. Our interest in this problem has been motivated primarily by its similarity to the connected speech recognition problem.

In fluent spoken speech, there are no silence gaps between words [10]. The computational requirements for segmenting speech into words presently precludes the operation of any real time speech recognizer on a vocabulary of 5000 words [11].

In both speech recognition and optical character recognition, template matching techniques have been the preferred approach in commercial devices [12 and 13]. Though it is recognized that feature extraction techniques are more robust in handling multifonts, template matching techniques are fast and easy to implement and can handle poor quality input containing noise, voids and touching characters [13]. However, template matching schemes have little tolerance to minor type-face variations, of which there are more than 3000 in common usage [5]. Some of the newer schemes [14] attempt to apply combinations of these two techniques and hopefully reap the benefits of both methods.

Preprocessing techniques in speech recognition rely on data compression techniques (eg. linear predictive coding) in order to extract the essential information from the input data. Some of the more recent recognition methods also apply the rediscovered vector quantization coding schemes [15 and 16] to reduce the data to an input stream of symbols. The Hidden Markov Model [17-21] is then used to interpret this stream of symbols. Though the accuracy of the vector-quantization based algorithms is lower than that of the conventional dynamic time warping techniques, the method requires an order of magnitude less processing time [20]. A further advantage of the Hidden Markov Model is that it avoids the need to explicitly code subjective rules about the interpretation of the patterns [22].

In our investigation, we exploited the vector quantization coder to extract the structural features of the characters. String matching techniques were then used to extract possible characters or character fragments from the text and network constraints were used to eliminate some of the interpretations.

Our study has so far been limited to the computer-generated fonts on a Symbolics Model/3640 workstation. This approach is effective for prototyping our design since it expedites the training process and factors out other processes in an OCR such as skew detection, correction for uneven illumination and distortions introduced by the optics system. Our simulations have provided us with useful information.

PREPROCESSING AND DATA REDUCTION

In the first step the image representation of a line of text is converted into an ordered list of discrete symbols to permit the application of one of many contextual pattern recognition schemes [1 and 23]. For example, the stream of symbols could be viewed as a grammar containing sufficient information to separate and identify the characters using syntactical pattern recognition schemes [24].

A short character string was displayed in a window of the Symbolics workstation using the TIMESROMAN12 font (one of a hundred fonts available in this workstation). Character spacing was adjusted so that adjacent characters either touched or overlapped in the horizontal direction. A low pass spatial gaussian filter [14] was applied to this window to simulate the optics of an OCR scanner and the resulting image was resampled to a lower resolution along a regular rectangular grid. The resampled image was thresholded to form a binary representation.

The sample column vectors extracted from this grid consist of sequences of ones and zeros which were then run length encoded. Using an appropriate distance measure and a training set of 5074 sample column vectors which were derived from the characters in the TIMESROMAN12 font, the sample column vectors were grouped into 79 categories. The distance measure was based on the number of runs of black samples (ones), their position and their length. The categories were chosen so as to ensure that no particular training vector was further than a certain distance from the category template.

Using this library of category templates which we call the prototype list, the sample vectors in the text window were converted into a list of prototype codes which referenced the position of the closest matching prototype in the prototype list. This encoding scheme is similar to the vector quantization compression scheme applied to speech. The text data can be reconstructed from the code list with a nominal amount of distortion (Figure 1).

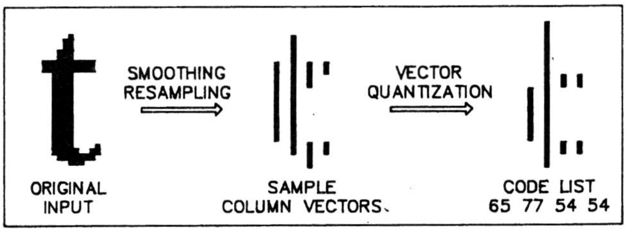


Figure 1. The characters in the TIMESROMAN12 font (39 points high including descenders) were smoothed and resampled with a unit variance two dimensional Gaussian filter and thresholded at a level of 0.3. The resampling density was chosen to yield 16 points per column. The sample column vectors were quantized to one of 79 producing a code-list as shown for the letter t.

SEGMENTATION AND CHARACTER RECOGNITION

Given the stream of prototype codes from a text window, the labeling of the codes as elements of characters is a form of the consistent labeling problem [25]. Each column vector may be a part of many possible characters however local contextual constraints limit the solution. The consistent labeling problem is an NP-problem and includes the graph homomorphism problem, the graph colouring problem, the packing problem, the scene labeling problem, the shape matching problem, the constraint satisfaction problem and theorem proving [25]. For this reason, the Symbolics work station was considered as a suitable tool for performing this study.

A related problem is the restoration of spaces between words in the following text string
'heworkswithcomputers'.
Using a large spelling dictionary, the computer can find nine possible words

embedded in the string:

computer he or with works
computers it put work.

However, assuming that every letter must belong to exactly one word in the dictionary, the computer will arrive at the correct solution

'he works with computers'.

The dictionary search is the most computationally intensive component of this problem. Special string comparison VLSIs [26] allow rapid solution of this problem on a PC with a large spelling dictionary.

In the OCR problem, two additional levels of complexity are introduced. Since the position of the scanning grid varies randomly with respect to any character [27], the same character may have many possible prototype code sequences (Figure 2). Other factors increasing this variability are the evenness of the lighting and the ink absorbing characteristics of the paper. Secondly, when adjacent characters are in contact, the initial or final sequence of codes of a character may be distorted or lost. The OCR must rely on the information in the central portion of the character. For some of the narrow characters (such as 1,1,r,t and the punctuation marks), this poses a major difficulty.

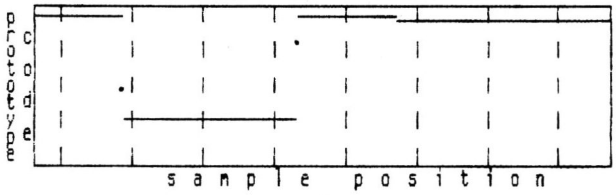


Figure 2. The prototype code (a number between 1 and 79) versus horizontal sampling position is shown for the upper case character L. The vertical dashed lines indicate sampling positions for a sampling rate equal to the vertical sampling rate. Note some prototype codes represented by the two dots in the plot have a very narrow range of existence. Such codes are easily missed during the training process.

To allow for this variability, the string matching procedure was implemented as a network search. A cross reference table was associated with every prototype code. This table lists all the characters that reference this code and the relative position in the character. In the previous example, the cross reference table associated with the letter o would include the following possible interpretations:

(or 1) (work 2) (works 2) (computer 2) (computer 2) (program 3) ...

The number associated with each word indicates the position of the character o in the word.

Given an unknown string, the algorithm attempts to create an interpretation list which satisfies the following constraint where c_i represents the letter at column i and (w_i s_i) is one of the interpretations corresponding to c_i. Two interpretations (w_i s_i) and (w_{i+1} s_{i+1}) corresponding to c_i and c_{i+1} are compatible if one of the following conditions are satisfied.

1 $w_{i+1} = w_i$ and $s_{i+1} = s_i$
or
2 $w_{i+1} = w_i$ and s_i is the last character position w_i and s_{i+1} is the first character position of w_{i+1}.

In our OCR application the two conditions were relaxed to allow the additional variability.

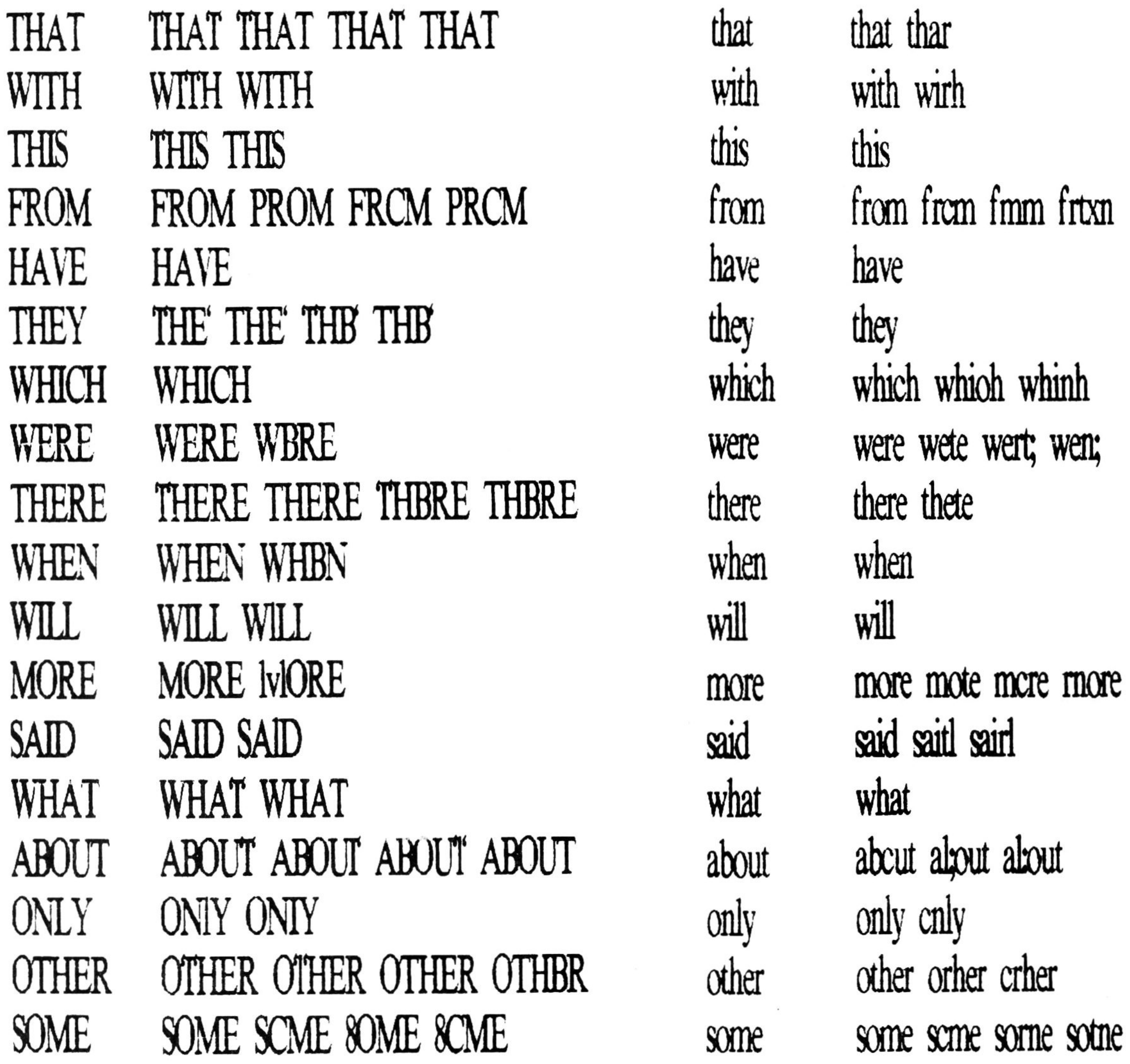
IMPLEMENTATION AND RESULTS

A cross reference table was created by resampling and vector quantizing the characters at a high sampling rate. In order to limit the table to a workable size, the positional information, s_i, was stored at lower resolution. On average, there were 27 interpretations associated with each of the 79 prototype codes.

The recognition software was divided into two sections. In the first pass, character fragments were extracted from the code list by searching for the longer chains of compatible interpretations. In the next pass, the algorithm attempted to patch the character fragments together in order to obtain a consistent interpretation. Frequently, the algorithm would return several consistent interpretations; they were ranked by the average character visibility in the text window.

The algorithm was implemented in compiled LISP on our Symbolics 3640 workstation with 512K words memory and without a floating point accelerator. A string of 7 touching characters was processed in approximately one minute. Eighty percent of this time was spent preprocessing the data and vector quantizing the sampled column vectors. The use of a VLSI pattern matching chip for performing the vector quantization [28] could reduce this time.

The algorithm had almost no difficulty identifying the isolated characters. However, it was necessary to use the maximum horizontal resolution (double the vertical resolution). The algorithm did not always distinguish some of the punctuation marks (:;) and confused the T



THAT	THAT THAT THAT THAT	that	that thar
WITH	WITH WITH	with	with wirh
THIS	THIS THIS	this	this
FROM	FROM PROM FRCM PRCM	from	from frcm fmm frtxn
HAVE	HAVE	have	have
THEY	THE' THE' THB' THB'	they	they
WHICH	WHICH	which	which whioh whinh
WERE	WERE WBRE	were	were wete wert; wen;
THERE	THERE THERE THBRE THBRE	there	there thete
WHEN	WHEN WHBN	when	when
WILL	WILL WILL	will	will
MORE	MORE lvlORE	more	more mote mcre more
SAID	SAID SAID	said	said saitl sairl
WHAT	WHAT WHAT	what	what
ABOUT	ABOUT ABOUT ABOUT' ABOUT	about	abcut al;out alout
ONLY	ONIY ONIY	only	only cnly
OTHER	OTHER OTHER OTHER OTHBR	other	other orher crher
SOME	SOME SCME 8OME 8CME	some	some scme sorne sotne

Figure 3. Results of applying the recognition algorithm on some common 4 and 5 letter English words displayed in TIMESROMAN12 font in both upper case and lower case characters. The second and fourth columns contain the list of possible character strings which match the input in decreasing order of average character visibility.

with the three character sequence `I`.

The algorithm was next tested on a set of the twenty most common 4 and 5 letter words which were displayed in both upper and lower case touching characters. The algorithm usually returned several consistent interpretations (eg. more, mote, mcre, rnore), however the first choice was almost always the correct choice (Fig. 3). Occasionally, the algorithm failed to recognize a particular letter (eg. Y in THEY). The failure was traced to a missing entry in the cross reference table which resulted in a broken chain of interpretations for this letter.

The next test consisted in applying the OCR to the TIMESROMAN15 font. Though the resampling process automatically normalizes the character dimensions, the smoothing filter was fixed and resulted in some small differences for the larger font. The OCR initially failed to recognize the new characters in this font, but when the cross reference table was extended to include these characters the algorithm was able to handle both fonts.

DISCUSSION OF RESULTS

The purpose of this study was to examine the feasibility of using string matching techniques to isolate and recognize proportionally-spaced characters which touched or overlapped. We represented the unknown string of characters by a sequence of equally spaced sample column vectors which were extracted from the text. The column vectors were categorized into about eighty types and their codes were sent to the classifier which applied string matching techniques to extract and identify the characters.

Our simulations so far were restricted to a computer generated font, TIMESROMAN12 on a Symbolics workstation. The algorithm succeeded in identifying most of the characters but encountered difficulties distinguishing some pairs such as (r,t), (o,c), (l,1) and (T,`I`) when the beginning or end of the characters were obscured by neighbouring characters.

Our preliminary investigations do not demonstrate any clear advantage of the string matching approach over the conventional template matching schemes (eg. [8]). Despite the use of a cross reference table to accelerate the matching process, the recognition process was still complex. A sequence of filtering processes were required to remove many of the character fragments which tended to impede the recognizer.

The accuracy measurements indicate that a higher vertical sampling rate and a larger prototype list would be needed to preserve the details of the characters. Furthermore, in order to handle italics and Greek mathematical symbols, the prototype list would need to double in size. Increasing the prototype list would increase the cross reference table, the network seach time and the sensitivity of the recognizer to any image noise.

REFERENCES

1. Godfried T. Toussaint, The use of context in pattern recognition, Pattern Recognition, vol. 10, 189-204, 1978.

2. Ching Y. Suen, n-Gram statistics for natural language understanding and text processing, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2, 164-172, 1979.

3. A.R. Hanson, E.M. Riseman and E. Fisher, Context in word recognition, Pattern Recognition, vol. 8, 35-44, 1976.

4. Rajjan Shinghal and Godfried T. Toussaint, Experiments in text recognition with the modified Viterbi algorithm; IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-1, no. 2, 184-193, 1979.

5. George Nagy, Optical character recognition - theory and practice, Handbook of Statistics, (ed. Krishnaiah and Kanal), North Holland Publishing, 621-649, 1982.

6. R.L. Hoffman and J.W. McCullough, Segmentation methods for recognition of machine-printed characters, IBM Journal of Research and Development, vol. 15, 153-165, 1971.

7. V.A. Kovalevsky, Image Pattern Recognition, Springer-Verlag, chapter VIII, 1980.

8. R.G. Casey and G. Nagy, Recursive segmentation and classification of composite character patterns, 6 th International Conference on Pattern Recognition, 1023-1026, 1982.

9. Jonathan J. Hull, Word shape analysis in a knowledge-based system for reading text, 2 nd Conference on Artificial Intelligence Applications, 114-119, 1985.

10. Ronald A. Cole (ed.) Perception and Production of Fluent Speech, Lawrence Erlbaum Associates, 1980.

11. Frederick Jelinek, The development of an experimental discrete dictation recognizer, Proc. of IEEE, vol. 73, 1616-1624, 1985.

12. Victor W. Zue, The use of speech knowledge in automatic speech recognition, Proc. of IEEE, vol. 73, 1602-1614, 1985.

13. Arthur W. Holt, The impact of new hardware on OCR designs, Pattern Recognition, vol. 8, 99-105, 1976.

14. A. Lashas, R. Shurna, A Verikas and A. Dosinas, Optical character recognition based on analog preprocessing and automatic feature extraction, Computer Vision, Graphics and Image Processing, vol. 32, 191-207, 1985.

15. R.M. Gray, Vector Quantization, IEEE Acoustics, Speech and Signal Processing Magazine, vol. 1, 4-29, 1984.

16. John Makhoul, Salim Roucos and Herbert Gish, Vector quantization in speech coding, Proc. of IEEE, vol. 73, 1551-1588, 1985.

17. J.K. Baker, The Dragon System - an overview, IEEE Trans. on Acoustics, Speech and Signal Processing, vol.

ASSP-23, 24-29, 1975.

18. L.E. Baum, T. Petrie, G. Soules and N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functio.ı of Markov chains, Annals of Mathematical Statistics, vol. 41, 164-171, 1970.

19 S.E. Levinson, Structural methods in automatic speech recogniton, Proceedings of the IEEE, vol. 73, 1625-1649, 1985.

20. L.R. Rabiner, S.E. Levinson and M.M. Sondhi, On the application of vector quantization and Hidden Markov Models to speaker-independent, isolated word recognition, The Bell System Technical Journal, vol. 63, 627-642, 1984.

21. Frederick Jelinek, Continuous speech recognition by statistical methods, Proc. of IEEE, vol. 64, 532-556, 1976.

22. R. Nag, R.H. Wong and F. Fallside, Script recognition using Hidden Markov Models, International Conference on Acoustics, Speech and Signal Processing, 1986.

23. Robert M. Haralick, Decision making in context, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-5, 417-428, 1983.

24. K.S. Fu, Syntactical Pattern Recognition and Applications, Prentice-Hall, Englewood Cliffs, N.J., 1982.

25. Robert M. Haralick and Linda G. Shapiro, The consistent labeling problem: part I, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. PAMI-1, 173-184, 1979.

26. Steve Rosenthal, The PF474, BYTE, vol. 9, no. 12, November 1984.

27. W.R. Throssell and P.R. Fryer, The measurement of print quality for optical character recognition systems, Pattern Recognition, vol. 6, 141-147, 1974.

28. Grant Davidson, Terry Stanhope, R. Aravind and Allen Gersho, Real-time speech compression with a VLSI vector quantization processor, Proc. of International Conference on Acoustics, Speech and Signal Processing, 1985.