Visual Inter-Word Relations and their Use in OCR Postprocessing

Tao Hong and Jonathan J. Hull*
Center of Excellence for Document Analysis and Recognition (CEDAR)
State University of New York at Buffalo
Buffalo, New York 14228-2567
taohong@cs.buffalo.edu

RICOH California Research Center*
2882 Sand Hill Road, Suite 115
Menlo Park, CA 94025
hull@crc.ricoh.com

Abstract

A technique is presented that uses visual relationships between word images in a document to improve the recognition of the text it contains. This technique takes advantage of the visual relationships between *word* images *that are usually lost in most conventional optical character recognition (OCR) techniques. The visual relations are defined to be the equivalence that exists between images of the same word or portions of word images. An algorithm is presented that calculates these relationships in a document. The resulting clusters are integrated with the recognition results provided by an OCR system. Inconsistencies in OCR results between equivalent images are identified and used to improve recognition performance. Experimental results are presented in which the input is provided directly from a commercial OCR system.*

1 Introduction

Conventional OCR systems process a digital image of text by segmenting it into isolated characters, recognizing the character images, and postprocessing the decisions versus a dictionary of "legal" words [2]. This strategy can provide good performance if the image is relatively noise-free and the characters can be located with high reliability. However, this approach uses only limited information above the level of individual words and ignores the relationships that exist *between* the *images* of words in a document. That is, once the individual characters are recognized, the image of the text is not processed further.

A recent approach locates groups of equivalent word images in a document and uses information about the spatial arrangement of those words to identify the predominant font in which the document is printed as well as to recognize function words (such as *the, of, and, a, to,*, etc.) in the text [7]. This technique uses the contextual information in word images to overcome uniform noise in the document and to achieve high accuracy. This is possible since it can be determined that two words are equivalent even though they are degraded by noise. Information about equivalent word images has also been used to improve the performance of a word-based postprocessing algorithm [4] as well as to increase the accuracy of a commercial OCR system [5].

This paper proposes an approach that uses a number of visual relations between words in a document to improve the recognition of the text it contains. The basic idea of detecting equivalent word images is extended to detecting equivalences between portions of words. For example, the first three characters in *theater* are the same as the first three characters in *thesis.* This information is then used to constrain the decisions of a recognizer so that they are the same in both cases. Also, the network of such relations in a document are used together with the results from a conventional OCR system to detect typographic characteristics and to build an image representation for each character. This information is used to train an algorithm that recognizes the rest of the text in the document.

The approach proposed here is an extension of the character-based clustering and deciphering algorithms used previously [3], [8]. The concept of a self-teaching

0-8186-7128-9/95 $4.00 © 1995 IEEE

	Possible Relations between W_1 and W_2	
	at image level	at symbolic level
1	$W_1 \approx W_2$	$W_1 = W_2$
2	$W_1 \approx subimage\_of(W_2)$	$W_2 = X \bullet W_1 \bullet Y$
3	$left\_part\_of(W_1) \approx left\_part\_of(W_2)$	$prefix\_of(W_1) = prefix\_of(W_2)$
4	$right\_part\_of(W_1) \approx right\_part\_of(W_2)$	$suffix\_of(W_1) = suffix\_of(W_2)$
5	$right\_part\_of(W_1) \approx left\_part\_of(W_2)$	$suffix\_of(W_1) = prefix\_of(W_2)$
6	$subimage\_of(W_1) \approx subimage\_of(W_2)$	$W_1 = X_1 \bullet Y \bullet Z_1$ and $W_1 = X_1 \bullet Y \bullet Z_1$

Note: "≈" means approximately match at image level; "•" means concatenation.

Figure 1: Word relations at the image and symbolic levels.

OCR system has also been used in a character classifier that automatically adapts itself to a single font [1]. The underlying assumption that a given document is printed primarily in a small number of fonts is utilized also used in the algorithm proposed here.

The rest of this paper discusses the proposed algorithm. The procedure for computing visual inter-word relations is discussed. Experimental results that operate directly on the output of a commercial OCR system are presented. The bounding boxes of words provided by that approach are used in the calculation of the visual inter-word relations and the character recognition results are used to derive the font representation. Future extensions of the approach are discussed that will improve its accuracy.

2 Visual Inter-Word Relations

Word images from a page of text are related to each other by the six relations defined in Figure 1.

Relation number one describes two images that are equivalent. This relation occurs often in normal English text where the same word is used many times in a single passage. In fact, it has been observed that the ten most frequent function words account for about 20% of a normal English language text [7]. Relation number two defines the occurrence of a *subimage*. That is, one word is entirely contained in another. Relations three through five define the *left_part_of* and *right_part_of* relations. These occur often because of the use of common prefixes and suffixes. Relation number six defines the occurrence of a subimage from one word as the subimage of another.

The primary characteristic used in the algorithm proposed in this paper is that the existence of a *visual* inter-word relation implies the existence of a *symbolic* equivalence. That is, if a portion of one word image is equivalent to a portion of another word image, the recognition results for the corresponding portions of those words *must* be the same.

The algorithm described in the next section of the paper takes advantage of this characteristic to improve OCR performance. An algorithm for visual relation analysis determines the occurrence of the six visual relations in a document image. A postprocessing algorithm then uses the visual relations within the document to correct for recognition errors in a portion of one word that are impossible given that it has a certain visual relation with another word that has been recognized differently.

3 Algorithm for Detecting Visual Inter-Word Relations

The algorithm that detects the six visual inter-word relations is composed of six separate steps. First, the whole-word equivalence relation is detected by an image clustering algorithm. The image prototypes from each cluster (i.e., the average of the word images in the cluster) are then compared in five separate steps that determine the other five relations (including the *subimage*, as well as the various *left_part_of* and *right_part_of* relations). The prototypes are used since the averaging step is effective at removing uniform noise and generating a better quality image than any of the individual words [7].

4 OCR Postprocessing With Visual Inter-Word Relations

A four-step algorithm is proposed in this section that postprocesses OCR results using visual inter-word relations. The objective of the first three steps is to locate word decisions that are correct with high

page id.	# of words	# of clusters	# of large clusters	# of words in large clusters	# of visual inter-word relations btw clusters (*type-2,3,4*)
P_1	827	520	96	403	4269
P_2	1129	690	139	578	7917
P_3	826	494	90	422	17406
P_4	535	389	45	191	8784
P_5	686	467	78	297	14358
P_6	1019	607	113	525	25745

Table 1: Results of visual inter-word relation analysis.

confidence. The first step uses the equivalence relation between words in a cluster to do this and the third uses the sub-image relation between words in different clusters. In the course of locating such high confidence decisions, some OCR errors are corrected. These high confidence word decisions are then used to learn images that correspond to individual characters and character sequences. These images are then used to decompose the remaining word images and generate new recognition results for them. Details of the four steps are presented below.

In the first step, a *voting* procedure is used on the whole-word clusters. The word decisions from clusters that contain two or more words are inspected and if a majority of them agree, that decision is output for the words in that cluster.

In the second step, a *font learning* method is performed in which the visual interword relations are used to decompose the prototypes for the clusters that voting produced decisions for. This results in image prototypes for many individual characters.

In the third step, a *verification* algorithm is executed on the word images that voting was unable to make a decision on. Visual inter-word relations are calculated between each image and the prototypes for the clusters output by voting. A word image is "verified" if its decomposition into sub-patterns is mapped onto ASCII decisions that agree with the original OCR result. An OCR error can also be corrected in this step if there are high confidence visual inter-word relations between the input image and portions of the cluster prototypes found during voting. The verification step processes each word in a cluster sequentially and generates a list of alternatives for all the words in the cluster. This is done by appending the verified results for each word.

In the fourth step, a *re-recognition* procedure is executed on all the remaining word images. Every such image is decomposed into sub-parts using visual relations calculated from the images output by voting, font learning, and verification. This produces a lattice of possibly overlapping sub-images along with their OCR results. Then all the paths through this lattice are traced and a score is calculated that measures the degree to which each sub-image in the path matches the original word image. All the complete paths that also occur in a dictionary are placed in the candidate list for the word and the complete path with the best cost is output. Appropriate thresholds are incorporated in the algorithm so that character strings not in the dictionary may also be output. This approach is similar to some methods used in cursive script recognition. The primary difference is that the algorithm proposed here learns the character image information it uses from the input page rather than from a previous training step.

5 Experimental Results

An experimental system was developed to test the postprocessing algorithm discussed above. The input to this system is the output from a commercial OCR (i.e., Caere's AnyFont package) as well as the page images that were provided to the OCR. The commercial device provides at least a single decision for each word and in cases where it is unsure, several alternatives are produced. Also, the bounding box coordinates for each word are output.

Six page images were used to test the system. These were scanned at 300 ppi and the binary image produced by the scanning hardware was used. Uniform noise was added to each image using the documentation degradation model (DDM) package from the University of Washington [6]. The parameter set for

		OCR			Postprocessing		
word set	# of words	decision corr. rate	corr. rate of candidate list	avg. # of candidate	decision corr. rate	corr. rate of candidate list	avg. # of candidates
voting	1403	1293 92.2%	1298 92.5%	1.6	1375 98.0%	1375 98.0%	1.0
verification	2160	1752 81.1%	1799 83.3%	2.6	1912 88.5%	2016 93.3%	3.2
rerecognition	1459	644 44.1%	675 46.3%	3.1	644 44.1%	768% 52.6%	2.4
voting+ *verification*	3563	3050 85.6%	3097 86.9%	2.5	3287 92.3%	3391% 95.2%	2.4
voting+ *verif.* + *rerecog.*	5022	3694 73.5%	3772 75.1%	2.6	3931 78.3%	4159 82.8%	2.3

Table 2: Results of postprocessing.

DDM was (820, 0.0, 1.0, 1.0, 1.0, 1.0, 3).

The accuracy of Caere's AnyFont OCR package on original pages is very high, more than 98% correct at the word level. After adding uniform noise with DDM, the word correct rate dropped to 73.5%. It was observed that the word alternatives produced by the OCR do not improve performance significantly.

Word clustering was then computed using the bounding boxes output by the OCR and inter-word relations were calculated between pairs of clusters. In the present implementation, only the first four visual relations in Figure 1 were used.

Table 1 shows the result of visual inter-word relation analysis. On average, about half of words are in large clusters (containing two or more word images). The number of visual inter-word relations is large and varies from page to page.

After applying the proposed postprocessing system, the word images are divided into three sets: *voting*, *verification* and *rerecognition*. The system generates one decision for each word in the *voting* set and there are no other candidates for each word. The results given in Table 2) show that the accuracy of the words in the *voting* set was improved from 92.2% to 98.0%. The accuracy of the words in the *verification* set was improved from 83.1% to 88.5% and the correct rate of the word alternatives was improved from 83.3% to 93.3%.

The correct rate of the words in the combination of the *voting* and *verification* sets was improved from 85.6% to 92.3% and the accuracy of their alternative lists was improved from 86.9% to 95.2%. It is important to note that the images in these sets account for about 71% of the words in the original text pages.

6 Conclusions

In this paper an approach was proposed that used visual relations between word images to improve the performance of an OCR system. The proposed algorithm first calculates clusters of equivalent word images and then determines which sub-parts of the prototypes for the clusters are equivalent. This information is then used in a four-step method for postprocessing the OCR results. Experimental results showed the effectiveness of the approach on input images that were degraded by a uniform noise model.

References

[1] H. S. Baird and G. Nagy, "A Self-Correcting 100-Font Classifier," in Proceedings of the Conference on Document Recognition of 1994 IS&T/SPIE Symposium, San Jose, CA, February 6-10, 1994.

[2] M. Bokser, "Omnidocument Technologies," in Proceedings of the IEEE, Vol. 80, No. 7, pp. 1066-1078, 1992.

[3] R. G. Casey, "Text OCR by Solving a Cryptogram," in IEEE Proceedings, pp. 349-351", 1986.

[4] T. Hong, "Integration Of Visual Inter-Word Constraints And Linguistic Knowledge In Degraded Text Recognition", in Proceedings of 32nd Annual Meeting of Association for Computational Linguistics(student sessions), Las Cruces, New Mexico, 27-30 June, 1994, 328-330.

[5] T. Hong and J.J. Hull, "Improving OCR Performance With Word Image Equivalence," Fourth Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, April 24-26, 1995.

[6] T. Kanungo, R. M. Haralick and I. Phillips, "Global and Local Document Degradation Models", Proceedings of the Second International Conference on Document Analysis and Recognition ICDAR-93, Tsukuba,Japan, 730-734, October 20-22, 1993.

[7] S. Khoubyari and J.J. Hull, "Font and Function Word Identification in Document Recognition," Computer Vision, Graphics, and Image Processing: Image Understanding, accepted to appear, 1995.

[8] G. Nagy, S. Seth and K. Einspahr, "Decoding Substitution Ciphers by Means of Word Matching with Application to OCR", PAMI, No. 9, pp. 710-715, 1987.

0-8186-7128-9/95 $10.00 © 1995 IEEE