Visual Global Context: Word Image Matching in a Methodology for Degraded Text Recognition

Jonathan J. Hull, Siamak Khoubyari, and Tin Kam Ho
Center of Excellence for Document Analysis and Recognition
Department of Computer Science
State University of New York at Buffalo
Buffalo, New York 14260-0001
hull@cs.buffalo.edu

Abstract

A technique for the use of global context in text recognition is presented that determines equivalences between word images in a passage of text. Initial hypotheses for the identities of words are then generated by matching the word groups to language statistics that predict the frequency at which certain words will occur. This is followed by a recognition step and a relaxation-based control structure that resolves inconsistencies between several knowledge sources.

This paper concentrates on the equivalence determination algorithm. A word matching technique is presented and its performance on a running text of about 1000 word images is determined. Several levels of noise are introduced to simulate different amounts of degradation introduced by fax machines or photocopiers. It is shown that the word matching algorithm maintains its ability to locate small groups of equivalent word images with high reliability even in the presence of noise.

1. Introduction

The recognition of digital images of text is typically performed by segmentation and recognition of isolated characters. This is often followed by a contextual postprocessing step in which word-level context is used to correct for errors in individual character recognition by matching character decisions to words in a dictionary. An alternative approach to word recognition based on human performance bypasses the character recognition phase and matches the features of whole words to entries in a dictionary [6]. A recent version of this method uses multiple classifiers and outputs a ranking of the dictionary [5]. This technique has proven to be tolerant to a wide range of image noise and is thus ideally suited to the recognition of degraded word images.

The use of global context above the level of individual words to improve text recognition performance has also been considered. For example, the semantics of a

constrained domain (chess games) has been used to correct for character recognition errors [2]. Language-level syntax has also been employed to improve word recognition by reducing the number of alternatives for a word's identity based on the hypothesized syntactic categories for two adjacent words [7].

This paper proposes to use an additional global contextual knowledge source that can be derived from passages of running text, namely, the repetitions of words. It is known from the analysis of language statistics that certain words occur more frequently than others. For example, as shown in Table 1, in a typical English language document seven percent of the words are "the" and three percent are "of". Furthermore, the ten most frequent words in English make up more than 23 percent of the word tokens and the top twenty words comprise about 29 percent of the sample. Thus, nearly one third of the words in a passage of text could be recognized by a method that could distinguish only twenty word images.

We propose to utilize such knowledge by matching word images to one another and determining groups of equivalences. The matching process is followed by a clustering step that determines equivalence classes. These equivalence classes are then matched to language statistics to derive word identifications. The advantages of this approach include its tolerance to image degradation. By matching whole word images, the internal featural context of words is used to compensate for features that are missing or distorted. This is an improvement on a previous technique that matched only isolated character images to a-priori statistics [3].

word	freq	word	freq	word	freq	word	freq
THE	0.071	IN	0.022	ON	0.008	AS	0.006
OF	0.032	FOR	0.010	HE	0.007	BY	0.006
AND	0.024	THAT	0.009	AT	0.007	IT	0.005
A	0.024	IS	0.008	WITH	0.006	HIS	0.005
TO	0.023	WAS	0.008	BE	0.006	SAID	0.004

Table 1. The top twenty most frequent words in a sample of over 90,000 words of newspaper reportage.

A further advantage to word image matching is that it could be used to improve the performance of word recognition. If such an approach was given that N word images were equivalent, each of those words could be individually input to a recognition routine that matches it to a dictionary. Since different features would be present or missing in each image, the recognition results would reinforce one another. An improved result for all the equivalent images could then be derived.

Word matching also integrates naturally with other global contextual knowledge sources. These include probabilities that individual words will re-occur after a certain number of words have appeared. For example, it is known that consecutive instances of "the" will occur within three to six words with probability 0.291 and within seven to ten words with probability 0.198. This information could be used to help refine the initial identification of those words. Another example of the integration of word image matching with global context is in syntactic analysis. If a word in one sentence is known to be the same as a word in another sentence, then the simultaneous syntactic analysis of these sentences must produce consistent results for those words.

The remainder of this paper presents a method for word image matching that is based on an initial clustering of word-level feature descriptions and assignment of possible word identities. A relaxation control structure is then proposed to refine the identifications. Experimental results are presented that validate the word matching approach with several noise models.

2. Overall Algorithm Description

The algorithmic framework in which the word matching process is intended to operate is illustrated in Figure 1. In this approach, an image of a passage of text is segmented into words and the word matching and clustering processes are applied to find equivalence classes among the word images. A-priori probabilities are then used to assign tentative word identifications to the word groups. This is followed by a separate word recognition step that is applied to each word image. This calculates a group of words from a lexicon (called a neighborhood) that are visually similar to each input word image. The groups of equivalent word images that were located by word matching as well as the neighborhoods for each word are then passed to a relaxation control structure that uses the probabilities of word re-occurrences as well as the results of a syntactic analysis to derive recognition decisions for the words that are consistent with these information sources.

The rest of this paper explores the word matching process in detail. An experimental investigation of various aspects of word matching are discussed. This is followed by a brief outline of other aspects of the overall algorithm and their relationship to word matching.

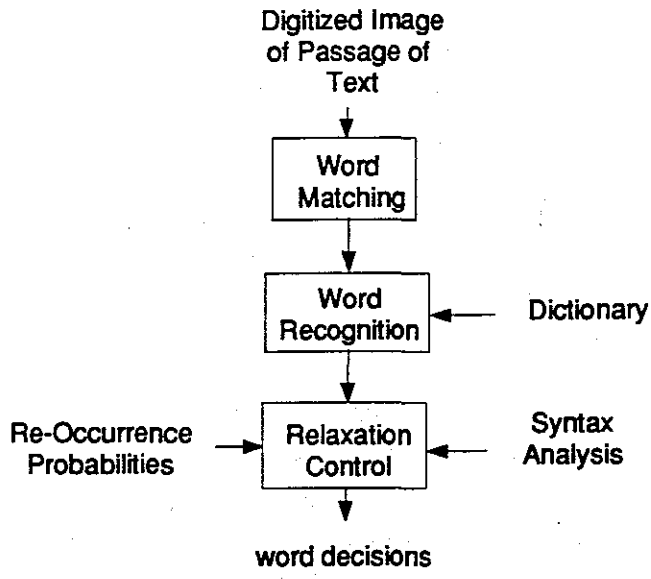


Figure 1. Overall algorithm design.

3. Word Image Matching

The word matching process first calculates a feature description for every word in a passage of text. The feature descriptions are then compared and a distance is calculated between them. Images with small distances should be equivalent to one another.

Because this approach should be tolerant to noise such as broken, smeared, touching, or degraded characters, a robust method for matching was chosen that is based on the analysis of the whole shape of a word. A set of features that is referred to as the *stroke direction distribution* is used to describe the shape of a word. It captures the spatial distribution of black pixels belonging to strokes of various directions.

The features are extracted using the local direction contribution method suggested for use with isolated Chinese characters in [9]. At each black pixel in the image, the longest continuous run of black pixels in each of the four directions east-west, northeast-southwest, north-south, and northwest-southeast is computed. The pixel is labeled with the direction in which the run length is a maximum. That is, each black pixel is labeled as part of a stroke of one of the four directions. Figure 2 shows an example of such pixel labeling.

The word image is first divided into a four-by-ten grid and the number of labeled black pixels of each type in each grid cell are counted. The counts are then normalized

by the total number of black pixels in the image. The stroke direction distribution is represented by a 160-dimensional feature vector, which stores the normalized counts of black pixels of each of the four types in the 40 cells.

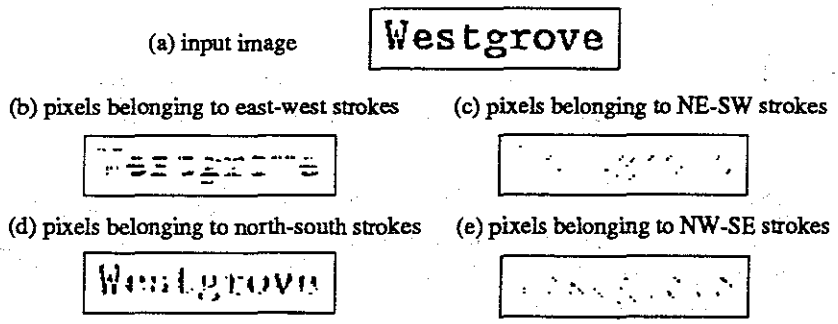


Figure 2. An example of the stroke direction distribution

A city-block distance metric is used to compare the feature vectors of two word images. The distance metric is the sum of the absolute differences of corresponding feature components [4]. That is, if $\mathbf{x} = < x_1, x_2, ..., x_n >$ and $\mathbf{y} = < y_1, y_2, ..., y_n >$ are two feature vectors, then the distance

$$d = \sum_{i=1}^{n} | x_i - y_i |$$

where $n = 160$ for the stroke direction distribution vector.

A similar word shape comparison technique can be developed using other feature sets. One example of an alternative feature set are the 32 seven-by-seven feature templates proposed in [1]. The features defined by these templates are detected by convolution and thresholding. Each non-zero response represents that a feature of a particular type is detected at that pixel position. The outputs are described by a 1280-dimensional feature vector, which stores counts of the 32 features detected in the 40 cells. The same distance metric can be applied to these feature vectors.

4. Experimental Environment

A series of experiments were conducted to determine the applicability of word image matching in a realistic environment of running text. Also, the tolerance of word

matching to noise was tested. The objective of these experiments was to show that the basic approach is a reliable generator of word match hypotheses and is thus a useful front-end in the proposed relaxation-based text recognition system.

The word images used to test the procedure were generated from a soft copy (ASCII) text sample. The test images were formatted to approximate the appearance of an original text sample and printed out on a laser printer. The resultant page images were then scanned on a desktop digitizer and stored on disk. An algorithm is applied to the page images to automatically segment them into isolated words. It was possible to introduce noise either in the page image, to test word segmentation as well as word matching, or in the word images, to specifically test the robustness of the word matching procedure.

The soft copy text from which the images were generated was the Brown Corpus [8]. This is a sample of over one million words of running text that was constructed to be representative of modern, edited American English. The corpus is divided into 500 samples of about 2000 words each. The samples were chosen from 15 different subject categories that span a range from Newspaper Reportage to Military Science. The corpus contains graphic codes that allow images to be generated that are a reasonable approximation of the appearance of the text.

One sample was chosen from the corpus for experimentation because it was a continuous article written by a single author. Many other samples had been split into several disjoint pieces. The subject category was "Belles Lettres" and the specific sample was G02. This is a portion of an article by Arthur S. Miller entitled "Toward a Concept of National Responsibility" that appeared in *The Yale Review* in December, 1961.

Approximately the first 1000 words of the sample were printed in an 11 point Times Roman font on plain white paper by a laser printer. The resultant pages were then scanned at 200 pixels per inch in 8 bit grayscale on a desktop digitizer. The conversion from grayscale to binary was performed automatically by a method that has proven suitable for similar applications [10].

Three levels of noise were introduced into the segmented word images by an iterative Gaussian procedure. The objective was to model the degradation caused by a facsimile machine or by repeated photocopying. At each iteration, every pixel of the image was processed sequentially. When a black pixel on a boundary was encountered, a random number was drawn from a normal distribution with mean zero and a given standard deviation. If the absolute value of that random number was greater than a threshold, the boundary pixel was changed to white. Multiple iterations were provided to control the maximum depth from which pixels were removed. A single iteration with a threshold of zero would thus remove all the boundary pixels.

Examples of images generated by this procedure are shown in Figures 3(a) to (d). Figure 3(a) shows the original "noise-free" text. Even though no noise has been applied, the individual words still contain enough touching characters to make the application of isolated character recognition difficult. Figure 3(b) shows the result of

two iterations of noise both of which used standard deviations of 20 and thresholds of 20. (The same threshold was used in all cases.) Even with such a relatively low level of noise, a significant number of boundary pixels were removed. The effect of two iterations of noise with standard deviations of 30 is shown in Figure 3(c). A relatively large amount of noise is shown in Figure 3(d) where two iterations with standard deviations of 50 and 30 were applied. At this level even some strokes are broken.

5. Performance Analysis

The ability of the word matching procedure to reliably locate groups of equivalent images within the same passage of text was analyzed. This was done by measuring the distance between the stroke direction vectors of each word image in the test sample of text (G02) and all the other word images in the sample. The results were then sorted in increasing order by distance.

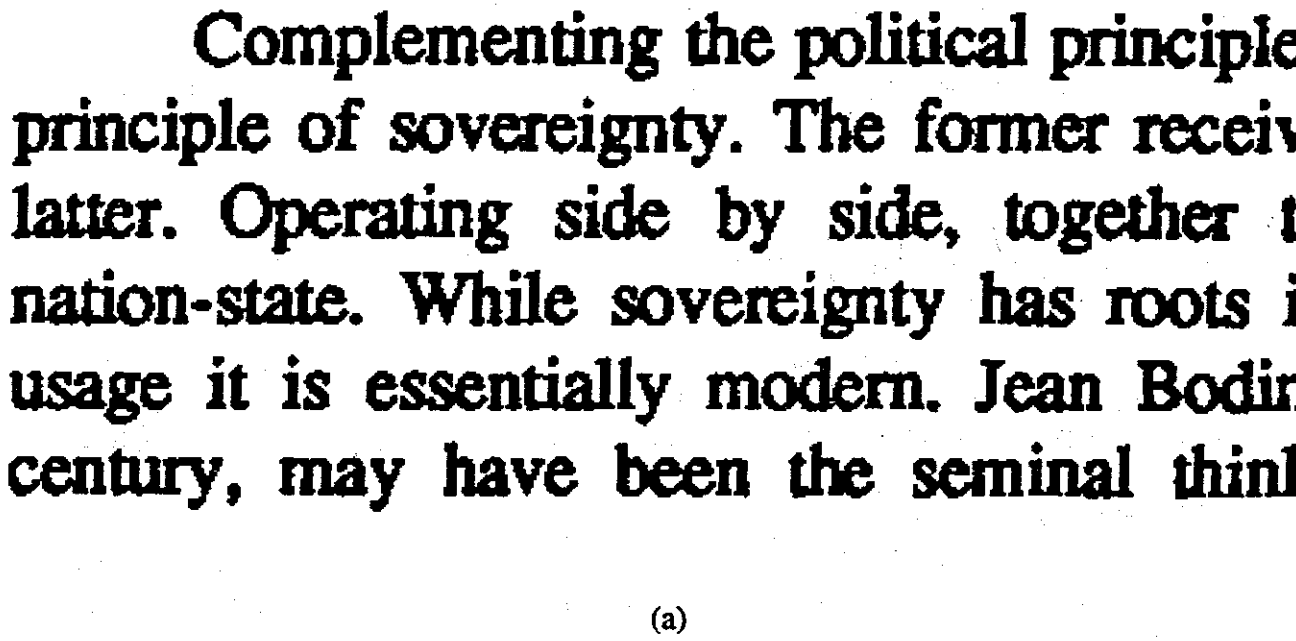
If a threshold on the distance could be found that would separate many of the correct matches from the incorrect ones, then these "equivalent" images above the threshold could be used as "seeds" for the subsequent clustering procedure. The number of correct matches within such a group of "equivalent" images could be small relative to the total number of possible correct matches since clustering is expected to compensate for this. Also, a small number of errors could be tolerated in such groups for a similar reason.

Performance in word matching was measured to reflect these characteristics. The sorted lists of matches for each word were separated at four different thresholds. The average number of correct and incorrect (errors) matches within each group were chosen as performance measurements.

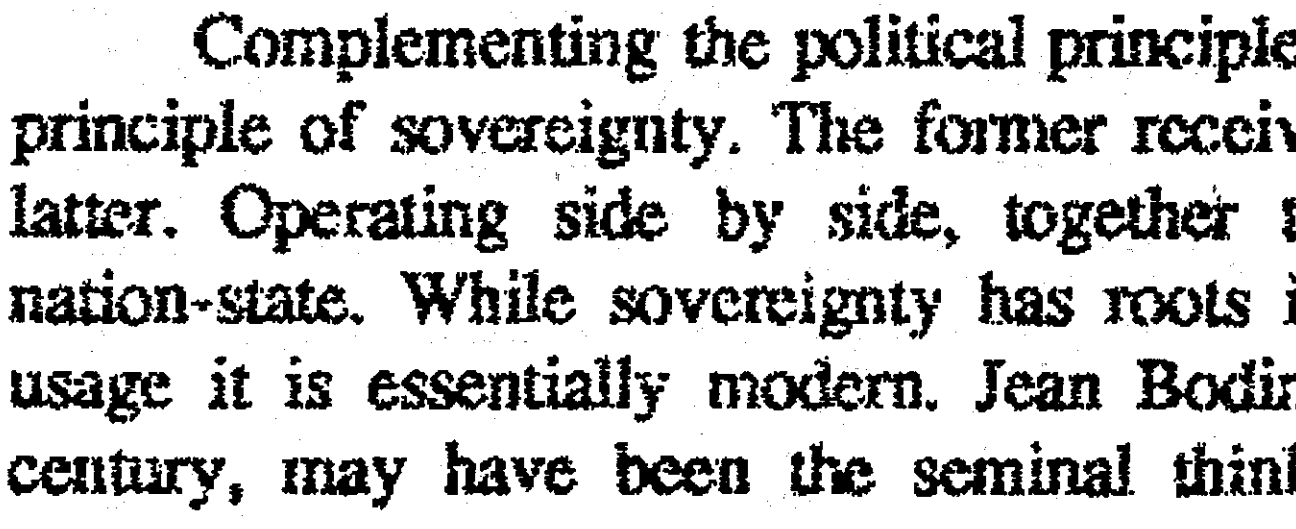
The minimum threshold that included all the correct matches was first determined. In this case, the average number of correct matches is an upper bound on the correct rate that can be used for comparison to performance with the other thresholds.

Since a small number of errors can be tolerated within a group of equivalent words, it is interesting to determine how close the average number of correct matches can get to the upper bound if a small number of errors are allowed within each group. This was measured by applying three additional thresholds that included at most 2, 4, or 6 errors within each group of equivalent words. The average number of correct and incorrect matches per word within these groups were also determined. The average number of incorrect matches may not equal the allowed number of errors because in some cases it may be possible to find a threshold that includes all the correct choices with fewer than the allowed number of errors.

An example of measuring performance by these criteria is discussed below for a text sample that contains seven instances of the word "the". The hypothetical results of matching the third instance of "the" (word number 12 in the sample) to all the other words are shown in Figure 4. This is the group of images that are "equivalent" to

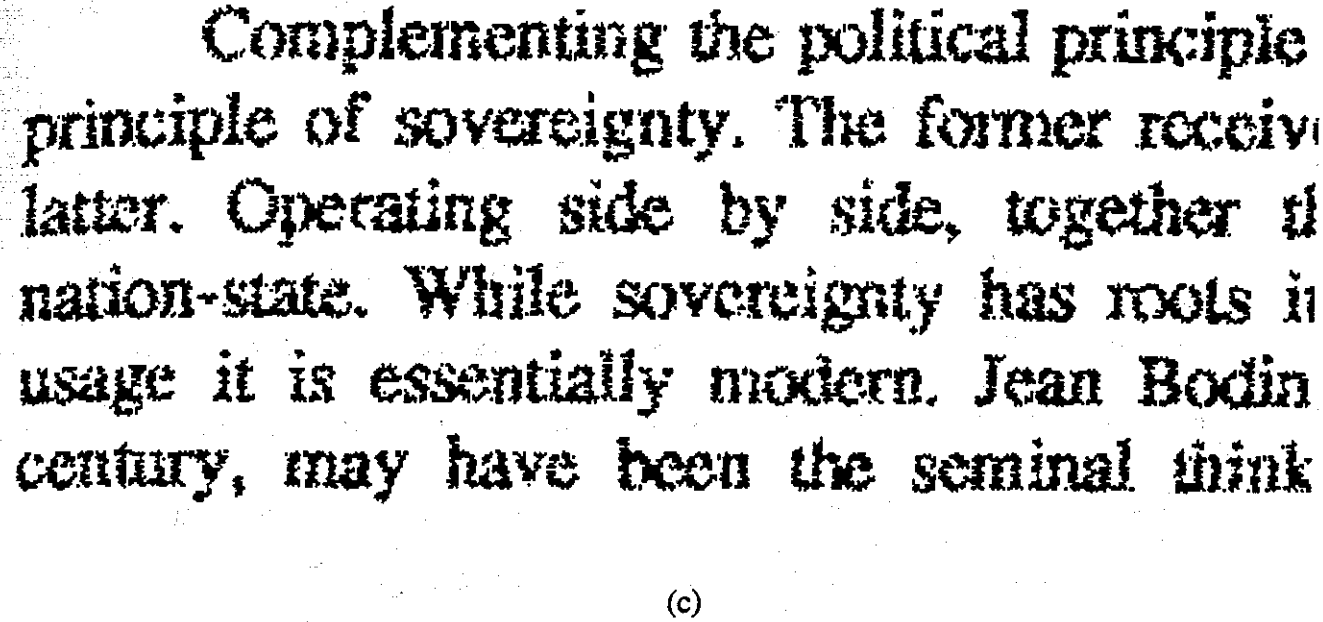


(a)

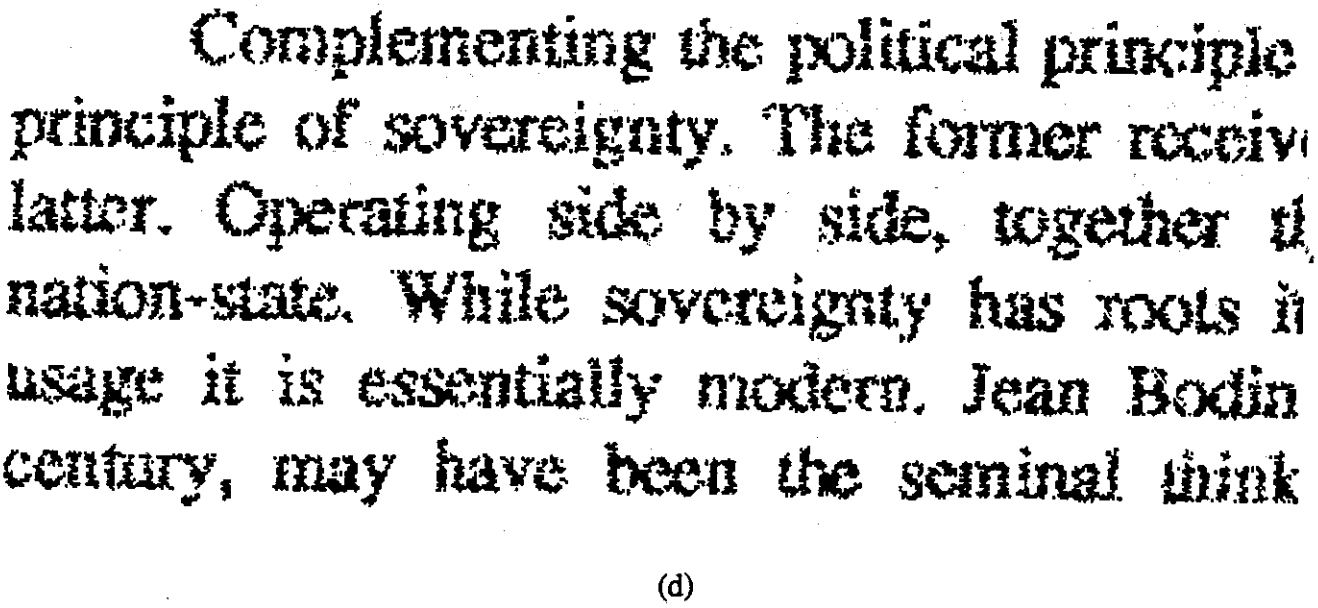


(b)

Figure 3. (a) original noise-free image, (b) after two iterations with s.d.= 20.



(c)



(d)

Figure 3. (cont.) (c) two iterations of noise, both with s.d. = 30; (d) two iterations of noise with s.d. = 50 and 30.

image number 12. If a threshold is applied that allows two errors within a group, then five of the six possible correct matches would be found. If four errors are permitted, all six correct matches would be located but only three actual erroneous matches.

6. Experimental Performance Measurement

The word matching procedure was applied over the isolated words segmented from the page images for sample G02. Noise at the levels illustrated in Figure 3 was introduced in the individual words after segmentation. Each word image was then compared to all the other word images by the distance between their stroke direction feature vectors. The distance measures for each image were then sorted and the average number of correct and incorrect matches per word with four thresholds were determined.

The results of these experiments are shown in Tables 2 to 5. The stroke direction features described earlier were used here. Performance is broken down by word length (number of characters). The number of words of each length are shown in the second column of the tables. The third column shows the number of words of each length that are equivalent to at least one other image. The performance with a threshold that included all the correct matches for each word was determined. In this case, the sorted list for each word was searched until all the correct matches were found. The number of errors that were present in the list before the last correct match were also recorded. The sum of these figures across all the words were divided by the total number of words to generate the average number of correct and erroneous matches per word.

word no.	identity	distance
28	the	76
14	the	80
8	the	95
46	tab	106
45	the	107
32	tot	110
75	the	126
2	top	129
20	the	134

Figure 4. Example of word matching results

(Only the word images that match at least one other image were used in this calculation.) A similar procedure was used to calculate performance with 2, 4, and 6 errors except that the sorted list was only searched until at most 2, 4, or 6 errors were located before the averages were calculated.

The results show that with two iterations of noise at a standard deviation of 20, the matching performance on two and three letter words (these lengths occur often in the top twenty most frequent words) yielded an average number of correct matches per word of 22 and 39 if up to two errors were allowed. This is in comparison the upper bounds of 30 and 43 respectively. If up to six errors are allowed, near perfect performance of about 28 and 41 are obtained with an average of only about 3 and 2 errors per word. Slightly worse performance is obtained with the next higher noise level (two iterations of noise with a standard deviation of 30) where the average number of correct matches for the two and three letter words was 24 and 38 with about 4 and 3 errors per word at a tolerance of at most six errors per word. Performance degrades slightly more when the highest noise level is considered (standard deviations of 50 and 30) where the average numbers of correct are about 21 and 32 per word and the average numbers of errors are 3 and 3.

word	no. words	no. words	all correct inc.		2 errors		4 errors		6 errors	
length	total	gt.1 match	avg.no. corr/word	avg.no. err/word	avg.no. corr/word	avg.no. err/word	avg.no. corr/word	avg.no. err/word	avg.no. corr/word	avg.no. err/word
1	25	25	16.00	0.04	16.00	0.04	16.00	0.04	16.00	0.04
2	195	187	30.13	1.10	28.88	0.34	29.87	0.93	30.11	1.03
3	200	184	42.80	0.33	42.76	0.20	42.77	0.22	42.79	0.28
4	121	78	2.68	0.24	2.64	0.06	2.67	0.12	2.67	0.12
5	85	33	2.70	0.61	2.42	0.26	2.70	0.61	2.70	0.61
6	84	26	1.54	0.12	1.54	0.12	1.54	0.12	1.54	0.12
7	81	27	1.33	0.26	1.31	0.08	1.33	0.26	1.33	0.26
8	87	39	3.54	4.72	3.49	0.05	3.38	0.26	3.38	0.26
9	56	9	1.33	0.11	1.33	0.11	1.33	0.11	1.33	0.11
10	50	11	1.27	0.00	1.27	0.00	1.27	0.00	1.27	0.00
11	32	7	1.43	0.00	1.43	0.00	1.43	0.00	1.43	0.00
12	16	3	2.00	0.00	2.00	0.00	2.00	0.00	2.00	0.00
13	8	2	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
14	16	11	4.18	1.18	3.36	0.64	4.18	1.18	4.18	1.18
15	4	2	1.00	5.00	0.00	0.00	0.00	0.00	1.00	5.00

Table 2. Word matching performance with no noise.

word length	no. words total	no. words gt.1 match	all correct inc. avg.no. corr/word	all correct inc. avg.no. err/word	2 errors avg.no. corr/word	2 errors avg.no. err/word	4 errors avg.no. corr/word	4 errors avg.no. err/word	6 errors avg.no. corr/word	6 errors avg.no. err/word
1	25	25	16.00	46.92	11.68	0.52	12.84	1.68	13.84	2.80
2	195	187	30.13	47.96	22.03	0.70	26.06	2.02	27.93	2.94
3	200	184	42.80	26.20	38.50	0.46	40.52	1.38	41.14	2.12
4	121	78	2.68	5.42	2.31	0.17	2.43	0.39	2.53	0.71
5	85	33	2.70	1.55	2.06	0.36	2.58	0.97	2.67	1.21
6	84	26	1.54	0.73	1.64	0.32	1.56	0.56	1.54	0.73
7	81	27	1.33	0.63	1.27	0.12	1.35	0.19	1.35	0.19
8	87	39	3.54	5.59	2.95	0.35	3.08	0.63	3.24	0.95
9	56	9	1.33	1.33	1.43	0.00	1.38	0.25	1.38	0.25
10	50	11	1.27	0.00	1.27	0.00	1.27	0.00	1.27	0.00
11	32	7	1.43	0.29	1.43	0.29	1.43	0.29	1.43	0.29
12	16	3	2.00	7.33	1.50	0.00	1.33	0.67	1.33	0.67
13	8	2	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
14	16	11	4.18	1.18	2.60	0.30	4.18	1.18	4.18	1.18
15	4	2	1.00	2.50	0.00	0.00	1.00	2.50	1.00	2.50

Table 3. Word matching performance with two iterations of noise, s.d.=20.

word length	no. words total	no. words gt.1 match	all correct inc. avg.no. corr/word	all correct inc. avg.no. err/word	2 errors avg.no. corr/word	2 errors avg.no. err/word	4 errors avg.no. corr/word	4 errors avg.no. err/word	6 errors avg.no. corr/word	6 errors avg.no. err/word
1	25	25	16.00	350.20	7.84	0.48	9.48	1.84	10.20	2.76
2	195	187	30.13	444.99	15.14	0.71	20.47	2.09	23.80	3.59
3	200	184	42.80	169.79	33.78	0.63	36.42	1.87	37.98	2.97
4	121	78	2.68	44.49	2.10	0.28	2.20	0.75	2.24	1.09
5	85	33	2.70	11.12	2.00	0.34	2.32	1.19	2.44	1.59
6	84	26	1.54	1.46	1.52	0.33	1.52	0.52	1.61	0.83
7	81	27	1.33	2.70	1.41	0.23	1.41	0.23	1.39	0.39
8	87	39	3.54	12.21	2.41	0.41	2.68	0.82	2.89	1.24
9	56	9	1.33	1.33	1.38	0.00	1.38	0.00	1.38	0.00
10	50	11	1.27	8.36	1.00	0.22	1.00	0.22	1.00	0.22
11	32	7	1.43	3.71	1.14	0.14	1.14	0.14	1.14	0.14
12	16	3	2.00	0.33	2.00	0.33	2.00	0.33	2.00	0.33
13	8	2	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
14	16	11	4.18	1.27	2.82	0.55	4.18	1.27	4.18	1.27
15	4	2	1.00	5.50	0.00	0.00	0.00	0.00	1.00	5.00

Table 4. Word matching performance with two noise iterations, s.d.=30.

word length	no. words total	no. words gt.1 match	all correct inc. avg.no. corr/word	all correct inc. avg.no. err/word	2 errors avg.no. corr/word	2 errors avg.no. err/word	4 errors avg.no. corr/word	4 errors avg.no. err/word	6 errors avg.no. corr/word	6 errors avg.no. err/word
1	25	25	16.00	486.24	7.08	0.72	8.12	1.84	8.80	3.16
2	195	187	30.13	504.20	15.40	0.66	19.95	2.03	21.30	3.37
3	200	184	42.80	242.20	27.14	0.63	30.41	2.01	31.90	3.12
4	121	78	2.68	51.74	1.80	0.24	1.95	0.92	2.02	1.43
5	85	33	2.70	22.45	1.76	0.32	2.15	0.96	2.27	1.31
6	84	26	1.54	8.19	1.57	0.38	1.57	0.65	1.57	0.65
7	81	27	1.33	12.15	1.24	0.18	1.19	0.57	1.18	0.77
8	87	39	3.54	32.74	2.29	0.26	2.38	0.62	2.50	1.21
9	56	9	1.33	6.67	1.43	0.00	1.38	0.25	1.38	0.25
10	50	11	1.27	4.09	1.25	0.38	1.38	0.62	1.33	1.00
11	32	7	1.43	0.29	1.50	0.00	1.43	0.29	1.43	0.29
12	16	3	2.00	2.67	1.33	0.00	1.67	1.00	2.00	2.67
13	8	2	1.00	0.00	1.00	0.00	1.00	0.00	1.00	0.00
14	16	11	4.18	0.73	4.09	0.64	4.18	0.73	4.18	0.73
15	4	2	1.00	6.50	0.00	0.00	0.00	0.00	0.00	0.00

Table 5. Word matching performance with two noise iterations, s.d.=50 and 30.

7. Discussion and Conclusions

A word matching procedure that locates equivalences between groups of word images in a passage of text was presented. This method is more tolerant to image noise than a traditional word recognition approach since determining whether two words are the same can be much easier than determining their classification.

An algorithm for word image matching was presented that calculates a feature description for all the words in a passage of text. The feature description is based on an analysis of the shape of a whole word and is thus tolerant to noise that causes broken or touching characters to occur. A ranked list of possible word matches was generated by comparing the feature description of an unknown word to those of all the other images. Words near the top of the ranking were more likely to be the same as the unknown.

Experimentation on images of text in which noise had been introduced showed that the word matching procedure was robust. For example, at the first noise level (s.d. 20) the average correct rate was 95 percent for the two and three letter words (27.93+41.14)/(30.13+42.8) at an average cost of between two and three errors per word. At the second and third noise levels, the average correct rates were about 85 and 73 percent with costs of about three to four errors per word.

Even more interesting is the performance when only two errors are allowed. In this case, a correct rate of about 58 percent was obtained with an average cost of less than one error per word. It is important to note that even in the presence of high levels of noise (these figures are for the s.d. equal 50 and 30 case), it is possible to locate groups of equivalent word images with high accuracy. This is especially useful since such high confidence "seeds" could be used in a subsequent clustering procedure.

Future work on this technique will include investigation of multiple feature sets and the incorporation of a clustering algorithm to produce better groupings of equivalent words in a passage of text. The use of word re-occurence probabilities and syntactic analysis as additional knowledge sources for a relaxation based control structure for word recognition will also be explored.

Acknowledgments

Chris Crowner contributed to the original development of this approach.

References

1. H. S. Baird, H. P. Graf, L. D. Jackel and W. E. Hubbard, "A VLSI Architecture for Binary Image Classification," in *From Pixels to Features*, J. C. Simon (editor), North Holland, 1989, 275-286.
2. H. S. Baird and K. Thompson, "Reading Chess," *IEEE Transactions on Pattern Analysis and Machine Intelligence 12* (1990), 552-559.
3. R. G. Casey and G. Nagy, "An autonomous reading machine," *IEEE Transactions on Computers C-17*, 4 (May, 1968).
4. R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, Addison-Wesley, New York, 1973.
5. T. K. Ho, J. J. Hull and S. N. Srihari, "Combination of Structural Classifiers," *IAPR Workshop on Syntactic and Structural Pattern Recognition*, Murray Hill, New Jersey, June 13-15, 1990, 123-136.
6. J. J. Hull, "Hypothesis generation in a computational model for visual word recognition," *IEEE Expert 1*, 3 (Fall, 1986), 63-70.
7. J. J. Hull, "Feature selection and language syntax in text recognition," in *From Pixels to Features*, J. C. Simon (editor), North Holland, 1989, 249-260.
8. H. Kucera and W. N. Francis, *Computational analysis of present-day American English*, Brown University Press, Providence, Rhode Island, 1967.
9. S. Mori, K. Yamamoto and M. Yasuda, "Research on Machine Recognition of Handprinted Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-6*, 4 (July 1984), 386-405.
10. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics SMC-9*, 1 (January, 1979), 63-66.