Image-based Word Recognition in Oriental Language Document Images

Jason Zhu and Jonathan J. Hull
Center of Excellence for Document Analysis and Recognition
Department of Computer Science
State University of New York at Buffalo
Buffalo, New York 14260 USA
hull@cs.buffalo.edu

Abstract

An algorithm for word recognition in Oriental languages such as Chinese, Japanese, and Korean is presented. The objective is to recognize words, that are composed of a number of consecutive characters, in document images where there are no explicit visually defined word boundaries. The technique exploits the redundancy in these languages that is expressed by the difference between the number of possible character strings of a fixed length and the number of legal words of that length.

Sequences of character images are matched simultaneously to lists of legal words and illegal strings that are likely to occur. A word is located if its image is more likely to occur in the current context than any of the illegal strings that are visually similar to it. No intermediate character recognition step is used. The application of contextual information directly to the interpretation of features extracted from the image overcomes noise that could have made isolated character recognition impossible and the location of words with conventional postprocessing algorithms difficult. Experimental results are presented that show the ability of this algorithm to correctly recognize text in the presence of noise.

1. Introduction

The recognition of words in Oriental languages such as Chinese, Japanese, and Korean is an important part of extracting information about the content of the text. The identities of words provide clues about the topic of the text that are useful in translating from the source language to a target language such as English. The ability to recognize words in degraded images such as facsimile transmissions or poor photocopies will also provide an improved transcription of those image to an ASCII-like representation such as BIG-5 or JIS and will also improve the ability to automatically translate those documents.

The recognition of words in Oriental languages is a difficult problem since words are composed of one or more characters and a running text contains no visual indication of word boundaries. Word recognition in Oriental languages has traditionally been solved by *post-processing* the results of isolated character recognition [4, 6]. Typically, the top N best decisions about the identity of each character in a running text are matched to a list of legal words. A word decision is output if it has a high confidence in comparison to other legal words.

Word recognition techniques take advantage of the redundancy present in Oriental languages. For example, in Chinese there are approximately 3500 commonly used characters. However, out of the $3500^2 \approx 12$ million possible strings of two-character words, only about 25,000 are legal words. Since legal words make up such a small percentage of all possible character strings, their occurrence in the output of a recognizer is a strong indication that they are correct.

A problem with postprocessing methods is that they require most if not all of the characters in a word to be recognized correctly in the top-N decisions output by the character recognizer. This assumption can be difficult to guarantee when the input image is subject to degradations that occur in facsimile transmissions or photocopies. This problem has been addressed in English word recognition by matching dictionary entries directly to image data [3].

An algorithm is proposed in this paper for Oriental language word recognition that overcomes the limitations of postprocessing techniques. The feature vectors extracted from sequences of individual characters are matched directly to the sequences of features for words

1051-4651/94 $04.00 © 1994 IEEE

in lists of legal and illegal words. Each entry in the legal word list points to entries in the list of legal words that are visually similar. A word decision is output if the legal word is significantly more similar to the input than to any of the associated illegal words.

The advantages of this technique include its application of word-level contextual information directly to the interpretation of the feature vectors. The direct use of image data allows for the recognition of words in context even though their individual characters may be unrecognizable in isolation. This overcomes a problem in traditional postprocessing techniques that require reliable character recognition results.

The rest of this paper presents the algorithm in more detail. The model of legal and illegal words is explained and an example is given of how it is used in practice. Experimental results are presented on a database of Chinese text that illustrates the ability of the algorithm to overcome noise that would be difficult to compensate for in a postprocessing approach.

2. Algorithm Description

The implementation of the proposed algorithm for word recognition is described in Figure 1. The input is composed of a stream of characters as they would appear, for example, in a Chinese language book. At each character position, all the words of each length are matched to the image data. Words that contain two to eight characters are used here.

Features extracted from consecutive characters in the image are matched directly to the feature vectors for the characters that compose words in the dictionary. In Figure 1 the dictionary words are the centers of the clusters. The other entries in the clusters are the *confusion set* for that word.

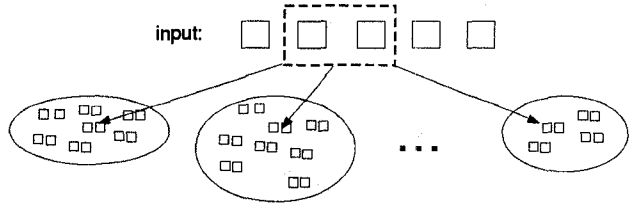

Figure 1. Implementation of word recognition.

If the similarity between the consecutive characters in the input image and the characters that compose any word is above a threshold, the input characters are compared to the other entries in the confusion set for that word. A word decision is a plausible output only if the similarity between it and the input is less than the similarity between the input and any other word in the confusion set. This process is repeated for all the words in the dictionary and a decision is output only if a single word is plausible at a given location.

The confusion set is compiled from a large training corpus of running text. Each consecutive string of characters in the corpus is compared to every dictionary word. If a string of characters contains at least one character in common with a dictionary word and those characters occur consecutively more than a fixed number of times in the corpus, they are added to the confusion set. The reasoning behind using the confusion set is that false positives are most likely to happen when the input is visually similar to a dictionary word. The confusion set models the most likely false positives and provides a method to suppress their identification.

A precise statement of the algorithm is given below. A training phase shown in Figure 2 is used in which the confusion sets for each dictionary word are calculated. In step 1 all the character strings of each length from two to six are extracted from a training corpus along with their frequencies. In step 2 the character strings are determined that are the same length as each dictionary word and have one character in common with it. These strings are added to the confusion list for the dictionary word if the similarity between the character string and the dictionary word is greater than δ_0 or it is greater than δ_1 and the character string occurs more than t_{conf} times in the corpus.

In the testing phase shown in Figure 3 each character position p is inspected. Each dictionary word dw of a given length l is compared sequentially to the input stream starting at position p. Longer words are compared before shorter words. If the similarity between the next l characters from the input stream starting at position p and dw is greater than a threshold $t_{recog,l}$, each word cw in the confusion set of dw is compared to the input. If none of the members of the confusion set match the input better than dw (i.e., the similarity between the input characters and every cw is less than the similarity between the input and dw), then dw is added to the output set. If after all the dictionary words have been considered, the output set contains a single word, that word is written out and the next l characters are skipped in the input stream.

```
1.   for each length l = 2,3,4,5,6,7,8
       - extract all the character strings of length l and
         their frequencies from a training corpus;
2.   for each dictionary word dw
       - determine the character strings cw found in step 1 that
         are the same length as dw and have at least one
         character in common;
       - calculate the similarity between each cw and dw;
       - add those strings to the confusion list for dw
         with a similarity greater than δ0(cw) or
         with a frequency greater than t_conf and
         a similarity greater than δ1(cw);
```

Figure 2. Training phase.

```
for each character position p in the input {
  output_set = NULL;
  for each dictionary word dw of length l {
    if (Similarity(l input chars, dw) > t_recog,l) {
        output_word = dw;
        for each word cw in the confusion set of dw
          if (Similarity(l input chars, cw) < Similarity(l input chars, dw)
                output_word = cw;
        if (output_word == dw) output_set = output_set + dw;
      }
  } /* end for each dw */
  if (|output_set| == 1) {
    write(output_set);
    p = p + l;
    break;
  }
}
```

Figure 3. Testing phase

The similarity between two strings of character images is calculated as the average of the distance between their feature vectors. The local stroke direction (LSD) vector is extracted from the input characters as well as the characters that compose the dictionary words [2,5]. The LSD divides each character image into a 6x6 grid and assigns each pixel the direction (vertical, horizontal, diagonal right, and diagonal left) of the vector that covers the maximum number of consecutive black pixels. The number of pixels in each grid cell that are assigned each direction are output. This provides a feature vector of 144 elements for each isolated character.

The distance calculation assumes that there exists an image of each character in the training corpus and the dictionary. This image data is extracted from digitized fonts that contain labeled images of isolated characters.

3. Experimental Results

Experiments were conducted to investigate several aspects of the word recognition algorithm. An implementation of the algorithm was constructed that located

keywords in a Chinese language document. This image is the introduction from a book on Chinese OCR techniques that was scanned at 300 ppi in binary mode [7]. The document image was degraded to simulate the noise present in poor quality facsimiles or photocopies. The recognition performance of the proposed algorithm was compared to that of a conventional character recognition algorithm and a traditional postprocessing technique.

The training data for building the confusion tables was composed of two million characters of running text from the Pin-Yin Hanzi (PH) corpus [1] plus three million characters of running text from the China News Digest. The dictionary was composed of the twenty keywords shown in Table 1. Fourteen of those words occur 48 times in the test image and the other six words do not occur. The length of each keyword is given as well as its English translation. The confusion lists on average contained 16.9 character strings for each word.

The distances between the characters in the test image and the characters in the dictionary words and the confusion sets were calculated as described above. A Kunlun 96x96 pixel Sung font was used as the font training data.

Noise was introduced into the test image by downsampling it to 200 ppi and 100 ppi and corrupting the 300 ppi and 200 ppi versions with a Gaussian bit-flip model. Each pixel in the test image was processed sequentially. If a random number drawn from a Gaussian distribution was greater than one standard deviation from the mean, the color of that pixel was changed from white to black or from black to white. Three levels of noise were used. At the first level, one iteration of noise was added. At second level two iterations of noise were added and at the third level three iterations were applied. At each iteration a unique seed was supplied to the random number generator.

Length of Chinese word	English translation	Length of Chinese word	English translation
2	recognition	4	formal language
2	Chinese character	6	automaton
6	computer	4	system
4	information	7	electron microscope
4	pattern	7	operating system
4	technique	7	programming design
4	Chinese characters	6	percentage
4	mathematics	2	workstation
7	artificial intelligence	2	chinese language
8	language and character study	2	language character

Table 1. Keywords in the dictionary.

The experimental results are reported in Table 2. The proposed algorithm for word recognition is compared to a character recognition technique and a postprocessing algorithm. The character recognition algorithm compared the LSD feature vector extracted from characters in the test image to characters in the font training data. The percentage of characters from the test image that are correctly located as the first choice of the character recognition technique are reported. The percentage correct in the five and ten best choices found by the LSD recognizer are also given.

The postprocessing algorithm was given the N best decisions from the character recognition algorithm. It was assumed that a keyword could be located if all its characters (excluding at most one) were found in the N best decisions. The percentage of keywords correctly located by this criterion are reported. This is a generous over-estimate of the expected performance of a well engineered postprocessing technique. However, it provides a reasonable upper bound to compare against the word recognition method.

The performance of the keyword recognition algorithm is reported as the percentage of keywords in the test image that are correctly located as well as the number of errors that occurred. That is, the number of keywords that were incorrectly recognized. The percentage of keywords in the test image that were not located by this technique are also reported.

The experimental results show that the proposed algorithm for word recognition outperforms the postprocessing algorithm in all cases. Correct rates above 94 percent are achieved in the 300 ppi and 200 ppi images. When the 100 ppi image was processed, the postprocessing algorithm needed ten choices to achieve an 88 percent correct rate while the keyword recognition method found 96.5 percent of the target keywords correctly with a zero percent error rate.

4. Discussion and Conclusions

An algorithm for word recognition in Oriental language documents was presented. This technique compared the feature vectors extracted from sequences of characters directly to the feature vectors for words. A simultaneous comparison to character strings that are likely to be confused with each dictionary word was used to suppress false positives. This provides a method that effectively uses word level contextual information directly at the image level. The bypassing of any intermediate character recognition step allows for the

	ppi	300 ppi				200 ppi				100 ppi
	noise	clear	1	2	3	clear	1	2	3	clear
char.	top1	96.7	94.2	91.6	86.8	94.8	85.0	77.1	64.3	67.1
recog	top5	99.5	99.0	97.7	96.5	99.2	96.5	93.6	83.9	88.9
perf.	top10	99.7	99.3	98.7	97.5	99.7	98.3	95.4	89.3	93.4
post-	top1	89.5	84.9	87.2	75.6	91.9	66.3	60.5	38.4	45.4
proc.	top5	100.0	98.8	93.0	93.0	100.0	91.9	87.2	74.4	83.7
perf.	top10	100.0	98.8	95.3	93.0	100.0	94.2	94.2	82.6	88.4
word	correct	100.0	100.0	100.0	100.0	100.0	98.8	96.5	94.2	96.5
recog	error	0.0	0.0	0.0	0.0	0.0	0.0	2.3	2.3	0.0
perf.	missing	0.0	0.0	0.0	0.0	0.0	1.2	3.5	5.2	3.5

Table 2. Recognition performance comparison.

recognition of words in the presence of noise that would make successful postprocessing of those decisions versus a dictionary nearly impossible.

Current work on this method includes the development of a preprocessing step that reduces the time required for dictionary matching. The word recognition algorithm is also being applied to Japanese language documents using a dictionary of several thousand words and a training corpus of over 30 million characters.

Acknowledgment

This work was partially supported by a grant from the California Research Center of the Ricoh Corporation.

References

1. J. Guo and H. C. Liu, "PH - A Chinese corpus for Pinyin-Hanzi Transcription," ISS Technical Report, National University of Singapore, Singapore, 1992.
2. T. K. Ho, J. J. Hull and S. N. Srihari, "A word shape analysis approach to lexicon based word recognition," *Pattern Recognition Letters 13* (November, 1992), 821-826.
3. T. K. Ho, J. J. Hull and S. N. Srihari, "A computational model for recognition of multifont word images," *Machine Vision and Applications, special issue on Document Image Analysis*, Summer, 1992, 157-168.
4. S. Mori, C. Y. Suen and K. Yamamoto, "Historical review of OCR research and development," *Proceedings of the IEEE 80*, 7 (July, 1992), 1029-1058.
5. S. Naito, K. Komori and S. Mori, "Stroke density feature for handprinted Chinese character recognition," *Journal of IECE of Japan PRL 81-32* (1981).
6. K. Seino, Y. Tanabe and K. Sakai, "A linguistic post-processing based on word occurrence probability," in *From Pixels to Features III: Frontiers in Handwriting Recognition*, S. Impedovo and J. C. Simon (editor), Elsevier Science Publishers, Amsterdam, 1992, 191-199.
7. X. Z. Zhang, *Chinese recognition techniques*, QingHua University, Bejing, China, 1992. (in Chinese).