

# WORD SHAPE RECOGNITION FOR IMAGE-BASED DOCUMENT RETRIEVAL

*Weihua Huang, Chew Lim Tan, Sam Yuan Sung and Yi Xu*

School of Computing, National University of Singapore  
Kent Ridge, Singapore 117543

## ABSTRACT

In this paper, we propose a word shape recognition method for retrieving image-based documents. Document images are segmented at the word level first. Then the proposed method detects local extrema points in word segments to form so-called vertical bar patterns. These vertical bar patterns form the feature vector of a document. Scalar product of two document feature vectors is calculated to measure the pair-wise similarity of document images. The proposed method is robust to changing fonts and styles, and is less affected by degradation of document qualities. Three groups of words in different fonts and image qualities were used to test the validity of our method. Real-life document images were also used to test the method's ability of retrieving relevant documents.

## 1. INTRODUCTION

The National University of Singapore Library has set up a digital library containing digitized images of antiquated Chinese newspapers and past student thesis. An original plan was to do OCR on these images so as to index the textual contents for easy retrieval. However, the proposed OCR work proved to be too costly. Thus an alternative approach is proposed now to do text retrieval based on word shape recognition without resorting to full OCR.

Traditional word shape recognition techniques make use of various features of a word image including histograms, strokes and curves, etc. Histogram-based methods have the advantage of simplicity, but they are generally font-dependent. Methods based on strokes or curves analysis are computationally complex and are not suitable for processing large amount of data. This paper selects a simple feature called the vertical bar pattern that is constructed through local extrema points detection. This feature is easy to construct and can be used for effective document image retrieval.

The remainder of this paper will discuss the proposed method in details. Section 2 surveys some related works in word shape recognition techniques. Section 3 describes the processes of extracting vertical bar patterns and forming feature vectors from the document images. Section 4 presents the methods for measuring similarities between words and between documents. Section 5 discusses experimental results that confirm the validity of the proposed model. Finally conclusions and future work are given in Section 6.

## 2. RELATED WORKS

Many methods were proposed to extract word zoning information [1-2] as the features for recognition. Recently, fuzzy boundary line detection based on histograms of traverse density and pixel density was proposed [3]. Image version N-gram methods extract statistical characterization of documents for similarity measure, making use of histogram features from characters [4]. These methods are generally font-dependent and sensitive to the degradation of image qualities, especially due to touching characters and broken characters. Word segmentation methods normally require database support or user-adaptive processes [5-6], which are not feasible when dealing with a large amount of data.

The local extrema point detection method is initially suggested for handwriting characters recognition [7]. This feature can reflect the property of a word, and it is relatively invariant to touching and broken characters. Extending the original idea, we apply the local extrema points detection and form the so-called vertical bar patterns from these points. Experimental results show that this method is also relatively invariant to changing of fonts and styles, and to the degradation of document qualities.

## 3. FEATURE EXTRACTION

Figure 1 describes the overall structure of the proposed image-based document similarity measure system. Basically the system can be divided into three stages: preprocessing, feature extraction and similarity measure.

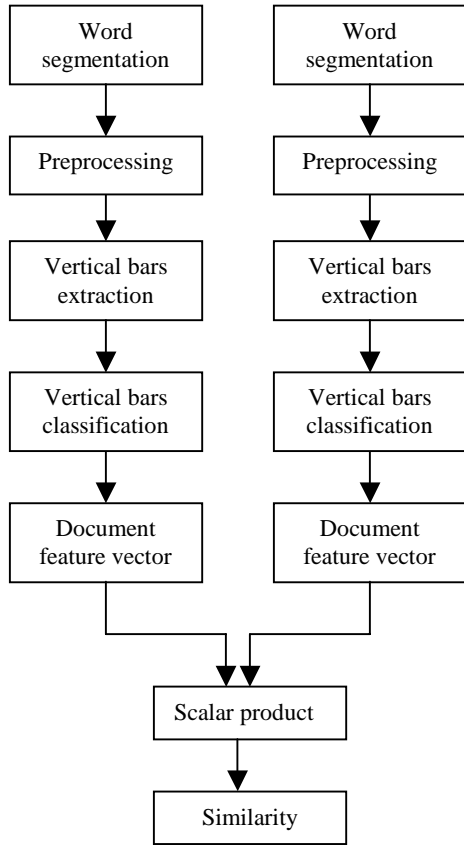


Figure 1 Similarity measure of document images

In preprocessing stage, document images are segmented into word units and filters are applied to remove punctuation and small noises. In feature extraction stage, local extrema points are detected and vertical bar patterns are constructed. All words in a document are converted into such patterns and form a feature vector of that document. In the similarity measure stage, feature vectors of two documents are compared using their scalar product.

### 3.1. The vertical bar pattern

Each alphabetic character has a number of local maxima as well as a number of local minima. If we pair up these local maxima and minima, they form vertical bars whose top is a local maximum and bottom is a local minimum. Since a word is made up of characters, it can then be represented by a list of vertical bars contributed by characters.

Vertical scan lines are traversed across the word from left to right, top-most and bottom-most black pixels are

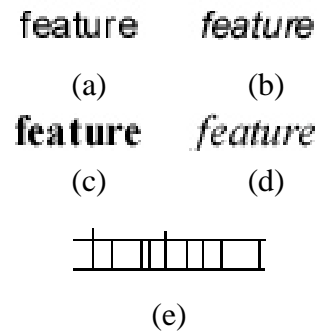


Figure 2 Sample word segments and corresponding vertical bar pattern. (a) Arial font 12 point type. (b) Arial font 12 point type with italic style. (c) Times New Roman font 14 point type with bold style. (d) Times New Roman font 14-point type with italic style. (e) Same set of vertical bars extracted after short bars removal and classification.

recorded as the maximum and minimum points for visited lines. An algorithm is designed to detect local extrema points by keep tracking of the increasing and decreasing trends of the maximum and minimum points between neighboring lines. Some glitches may exist along a horizontal image edge, they are detected and ignored to ensure the validity of the local extrema points detected.

We are concentrating on the effective vertical bars only, and so short vertical bars are treated as noises. A length filter is applied to remove these noise bars. Figure 2 illustrates how vertical bar extraction works on words of different fonts, sizes and styles. Note that the same set of vertical bars is extracted even though the words have different fonts and sizes.

### 3.2. Classification of vertical bars

To make the vertical bars comparable between documents regardless of their horizontal position, classification need to be done according to the vertical bars' length and vertical position. By locating the upper boundary and lower boundary lines, we can divide the whole word into three vertical zones. Then vertical bars that protrude into the upper zone are in class "d", vertical bars that protrude into the lower zone are in class "q", and vertical bars that are only inside the middle zone are in class "m". Figure 2 (e) indicates the two boundary lines detected. In this way, a word can be represented as a list of symbols consisting of "d", "m" and "q", where each symbol indicates a vertical bar from the corresponding class. For example the word "feature" is converted into vertical bar pattern "dmmmdmmmm".

### 3.3. Document feature vector

After all word units in the document image are converted into their vertical bar patterns, a hash table is created to keep track of the frequency of occurrence of a vertical bar pattern. Each hash table can be treated as a vector, so called the document vector. Each entry in the vector records a specific vertical bar pattern and its number of occurrence. When inserting a vertical bar pattern from a document into the hash table, the number of occurrence of the corresponding entry is increased by one.

The occurrence frequency of each vertical bar pattern is normalized by dividing the occurrence by the total number of occurrences of all vertical bar patterns. This means that the absolute number of occurrence will be replaced with the relative frequencies of corresponding vertical bar patterns. The reason for doing so is that similar texts of different lengths after this normalization will have similar document vectors.

## 4. SIMILARITY MEASURE

To measure similarity between single words, we convert each word segment into a vertical bar pattern. Dynamic programming is used to look for the minimum difference between two vertical bar patterns. The similarity is then calculated by subtracting this difference from the length of the vertical bar pattern. The minimum difference between vertical bar patterns  $q$  and  $d$  is stored in element  $[0, 0]$  of a matrix  $f$ , and each element of  $f$  is calculated as:

$$f[i, j] = \begin{cases} Lq - j, & \text{if } i = Ld \\ Ld - i, & \text{if } j = Lq \\ \text{Min} \left[ \begin{matrix} \text{Diff}(i, j) + f[i+1, j+1], \\ f[i+1, j] \end{matrix} \right], & \text{otherwise} \end{cases}$$

where  $Lq$  and  $Ld$  are the lengths of the first pattern and the second pattern respectively;  $i$  and  $j$  are indices; the function  $\text{Diff}(i, j)$  accepts the  $i$ th character from the first pattern and the  $j$ th character from the second pattern and returns 0 if two characters are the same, 1 otherwise.

To measure similarity between documents, we know that in a vector space, document vectors for similar documents generally point in the same direction. The similarity score between two document vectors is defined as their scalar product divided by their lengths. A scalar product is calculated through summing up the products of the corresponding elements. This is equivalent to the cosine of the angle between two document vectors seen from the origin. So, the similarity between document images  $m$  and  $n$  will be:

$$S(X_m, X_n) = \frac{\sum_{j=1}^J x_{mj} x_{nj}}{\left( \sum_{j=1}^J x_{mj}^2 \sum_{j=1}^J x_{nj}^2 \right)^{1/2}}$$

where  $X_m$  and  $X_n$  are the document vectors of image  $m$  and  $n$  respectively,  $J$  is the dimension number of document vector, and  $X_i = x_{i1} x_{i2} \cdots x_{iJ}$ .

## 5. EXPERIMENTAL RESULTS AND DISCUSSION

To test the validity and effectiveness of the proposed method, two types of experiments were conducted. The first experiment applied the proposed method to measure similarities between single word images. The second experiment used the proposed method to retrieve image-based documents.

### 5.1. Testing with single words

There are three groups of word images. The first group contains 20 word images cut from documents with good scan quality. The second group contains 20 word images cut from documents with poor quality. And the last group contains machine generated word images with different fonts, styles and sizes. The testing is done by converting target words from keyboard inputs into vertical bar patterns through manual assignments and comparing them with the vertical bar patterns extracted from the corresponding word images. The similarity values suggest the ability of the proposed method to correctly recall words. Testing results are listed in table 1.

### 5.2. Testing with real-life documents

We generate four groups of testing document images generated from one set of documents containing 26 articles that address four different kinds of topic. Each of the first three groups is generated with different font sizes and scan qualities. The last testing group contains a mixture of 24 documents from the first three groups. There are altogether 102 testing document images. In each group, the first article from each topic is used as a query document and the similarities between the query document and all other documents in the group are calculated. Figure 3 illustrates one example of such comparisons using the first document in testing group 1 to compare with all documents in that group.

From figure 3 we can see that there is a clear gap between relevant documents and irrelevant documents. Thus we can set a threshold to decide whether a document

Testing group	Recall %
Poor quality	75.86
Good quality	98.71
Auto-Generated	95.45

Table 1 Recall by similarities for the three groups of words on the average.

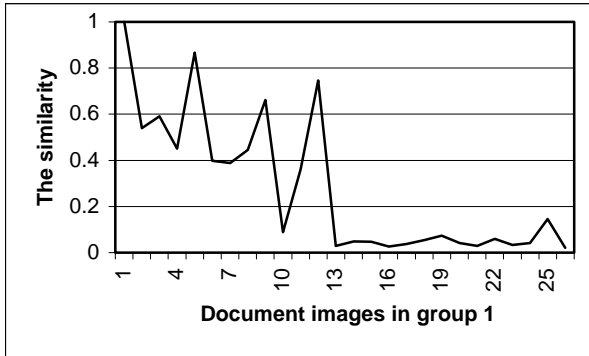


Figure 3 Similarity values between the first document and all documents in the first testing group.

Article	Threshold = 0.2		Threshold = 0.15		Threshold = 0.1	
	R %	P %	R %	P %	R %	P %
Group1	69.17	95.83	75.42	92.86	81.67	90.78
Group2	47.08	100	64.17	100	75.42	93.33
Group3	35	100	40	100	54.17	97.92
Group4	61.11	100	88.89	100	94.44	100
Avg.	52.56	98.89	65.67	97.54	75.22	95.21

Table 2 Recall and precision of testing results using different threshold values, R = Recall, P = Precision

is similar to the query document. Choosing different threshold values affects the documents retrieved. After comparing the retrieval results with the results of retrieval done manually, the recalls and precisions are calculated. Table 2 records recall and precision of document retrieval by the proposed method using threshold values 0.2, 0.15 and 0.1.

### 5.3. Discussions

From the testing results obtained, we can see that the proposed method works well comparing single words even when the words have different fonts and sizes and when the image quality is poor. When threshold is set to 1.0, both recall and precision of the proposed method are satisfying. Testing results for group 4 documents show that most relevant documents can be retrieved even when documents have different font sizes and qualities.

The results in table 2 also suggest that lower threshold value generates higher recall but lower precision as a trade-off. Thus appropriate threshold values can be chosen depending on whether the user wants high recall or high precision.

## 6. CONCLUSION AND FUTURE WORK

This paper proposed a word shape recognition method for image-based document retrieval based on local extrema points detection. Features called the vertical bar patterns are extracted from the local extrema points detected, and form the document vector. Document similarity is calculated by finding the scalar product between two vectors. Experimental results suggest that the method is relatively robust to declining image qualities, and is suitable for retrieving document images with different font sizes and styles.

However, the method has the limitation that it can only handle documents written in Latin languages. So one of the future research directions is to design a language independent technique to handle this problem. Some parts of the method also need to be refined to enhance the consistency of the vertical bar patterns extracted.

## 7. REFERENCES

- [1] S. Madhvanath and V. Govindaraju, "Local reference lines for handwritten phrase recognition," *Pattern Recognition*, Vol. 32, pp. 2021-2028, 1999.
- [2] J. Wang, K.H. Leung and S.C. Hui, "Cursive Word Reference Line Detection," *Pattern Recognition*, Vol. 30, No. 3, pp. 503-511, 1997.
- [3] C.L. Tan, S.Y. Sung, D. Shi and Y. Xu, "News Article Retrieval from Microfilms," *Text Mining Workshop, International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, pp. 110-116, 1999.
- [4] C.L. Tan, S.Y. Sung, Z. Yu and Y. Xu, "Text Retrieval from Document Images based on N-Gram Algorithm," *Text and Web Mining Workshop, 6th Pacific Rim International Conference on Artificial Intelligence*, Publisher, Melbourne, Australia, 2000.
- [5] T. Bayler, U. Krebel and M. Hammelsbeck, "Segmenting Merged Characters," *Pattern Recognition*, Vol. 10, No. 6, pp. 346-349, 1992.
- [6] L. Duneau and B. Dorizzi, "Online Cursive Script Recognition: A User-adaptive System for World Identification," *Pattern Recognition*, Vol. 29, No. 12, pp. 1981-1994, 1996.
- [7] R.K. Powalka, N. Sherkat and R.J. Whitrow, "Word Shape Analysis for a Hybrid Recognition System," *Pattern Recognition*, Vol. 30, No. 3, pp. 421-445, 1997.