

# Vertical Bar Detection for Gauging Text Similarity of Document Images

Weihua Huang, Chew Lim Tan, Sam Yuan Sung and Yi Xu  
School of Computing, National University of Singapore  
Kent Ridge, Singapore 117543

## Abstract

*A new method for gauging text similarity of image-based document using word shape recognition is proposed in this paper. Image features are directly extracted instead of using OCR (optical character recognition). The proposed method forms so-called vertical bar patterns by detecting local extrema points in word units extracted by segmenting the document images. These vertical bar patterns form the feature vector of a document. The pair-wise similarity of document images is measured by calculating the scalar product of two document feature vectors. The proposed method is robust to changing fonts and styles, and is less affected by degradation of document qualities. To test the validity of the method, four corpora of document images were used and the ability of the method to retrieve relevant documents is reported.*

**Keywords:** Local Extrema, Vertical Bar Pattern, Document Image, Similarity Measure

## 1. Introduction

A digital library is set up by the National University of Singapore Library to store digitized images of various kinds of reading materials including antiquated Chinese newspapers and past student theses. 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 to do text retrieval by comparing text similarity of document images directly.

Traditional document similarity measure techniques can either analyze the similarity of the documents' content based on semantics or use statistical methods to gauge the text similarity directly without understanding the semantics. The former approach requires an in-depth linguistic knowledge while the latter is easy to implement. In both cases, machine readable text (e.g. ASCII text) must be available for the system to find text similarity. This paper adopts the second approach for text similarity measure but without the availability of ASCII text data. Instead, a simple image feature called the vertical bar pattern is used to facilitate statistical similarity measure.

The remainder of this paper will discuss the proposed method in details. Section 2 surveys some related works in text processing in both the text as well as the image domains. Section 3 describes

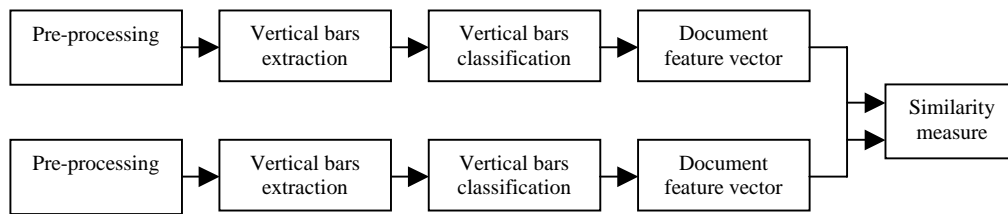
the processes of extracting vertical bar patterns and forming feature vectors from the document images. Section 4 presents the methods for measuring similarities between documents. Section 5 discusses experimental results that confirm the validity of the proposed model. Finally conclusions and future work are given in Section 6.

## **2. Related Works**

Over the past few decades, many methods of categorisation and retrieval of machine-readable texts had been proposed. Among them, the purely statistical characterisation of text in terms of word frequency or N-Grams (sequences of N consecutive characters) [1] has been widely applied to text analysis and document processing, including text compression [2], text search and retrieval [3-4], etc. Basing on this statistical characterisation, M. Damashek [5] has proposed a simple but novel vector-space technique that makes sorting, clustering and retrieval feasible in a large multilingual collection of documents. His method collects the frequency of each N-Gram to build a vector for each document and the processes of sorting, clustering and retrieval can be implemented by measuring the similarity of the document vectors. This method provides a high degree of robustness when handling ASCII text documents.

Text in document images is a more complicated matter for text processing. Recently, several researchers have made such an attempt in a number of applications. For example, Hull and Cullen [6] have proposed a method to detect equivalent document images by matching the pass codes of document. They create a feature vector that counts the numbers of pass codes in each cell of a fixed grid in the image and equivalent images are located by applying the Hausdorff distance to the feature vectors. Tan and Yu [7] have proposed an image version of Damashek's N-Gram method to retrieve news articles in multiple languages. They make use of image features named Vertical Traverse Density (VTD) and Horizontal Traverse Density (HTD) to form the document vector for similarity measure between document images.

The local extrema point detection method was initially suggested for handwriting characters recognition [8]. 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 the changing of fonts and styles, and to the degradation of document qualities.



**Figure 1. Gauging the similarity of document images based on content**

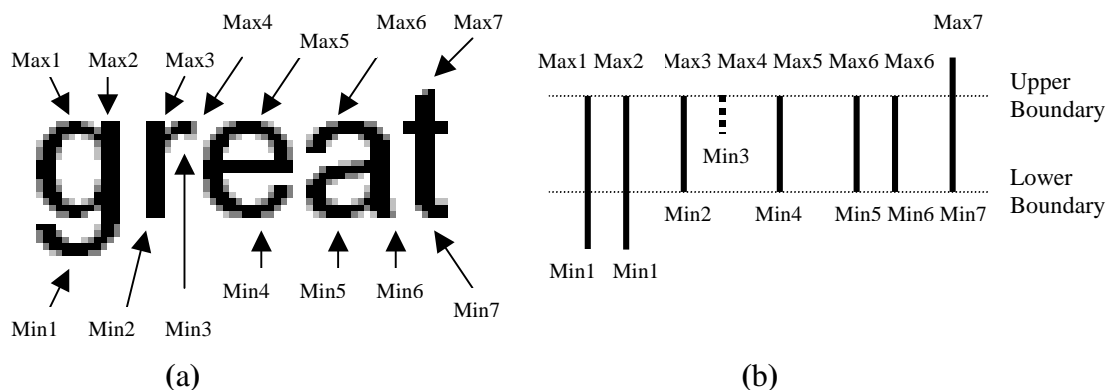
### 3. Feature Extraction

Figure 1 outlines the steps in gauging the similarity of document images based on contents. In the pre-processing stage, document images are segmented into word units through layout analysis and filters are applied to remove punctuation and small noises. In the feature extraction stage, local maximum points and local minimum points within these word units are identified. After these local maximum and minimum points are paired up, a list of vertical bars is obtained. After classification of these vertical bars (to be explained in section 3.2), each word unit is converted into a list of symbols consisting "d", "m" and "q", or what we call the vertical bar pattern. A feature vector can be obtained from each document image based on the frequency of occurrences of the vertical bar patterns. Feature vectors are used to calculate the similarity of document images by calculating 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 the characters in the word.

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



**Figure 2. Vertical bar detection. (a) Finding local extrema points, (b) Vertical bars extracted with boundary lines indicated**

We are concentrating on the effective vertical bars only, and so short vertical bars are treated as noise. A length filter is applied to remove these noise bars. Figure 2 (a) illustrates the local extrema points detected for sample word “great”. Figure 2 (b) shows the list of vertical bars extracted by pairing up local maxima points and local minima points. Note that the number of local maxima points and the number of local minima points may not be equal, thus some of the local extrema points are shared among vertical bars within the same character object. The dashed bar in figure 2 (b) is an example of short bars that are removed by the filter.

### **3.2. Classification of vertical bars**

To make the vertical bars comparable between documents regardless of their horizontal positions, classification needs 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 (b) 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 "great" is converted into vertical bar pattern "qqmmmmmd".

### **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 vertical bar patterns being studied. Each hash table can be treated as a vector to represent the particular document. Each entry in the document vector records a specific vertical bar pattern and its number of occurrences. Every time a vertical bar pattern is picked, the number of occurrences of the corresponding entry is increased by one.

The occurrence frequency of each vertical bar pattern is normalized by dividing the number of occurrences by the total number of occurrences of all vertical bar patterns. This means that the absolute number of occurrences 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**

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

$$\text{Similarity} (X_m, X_n) = \frac{\sum_{j=1}^J x_{mj} x_{nj}}{\sqrt{\sum_{j=1}^J x_{mj}^2 \sum_{j=1}^J x_{nj}^2}}$$

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

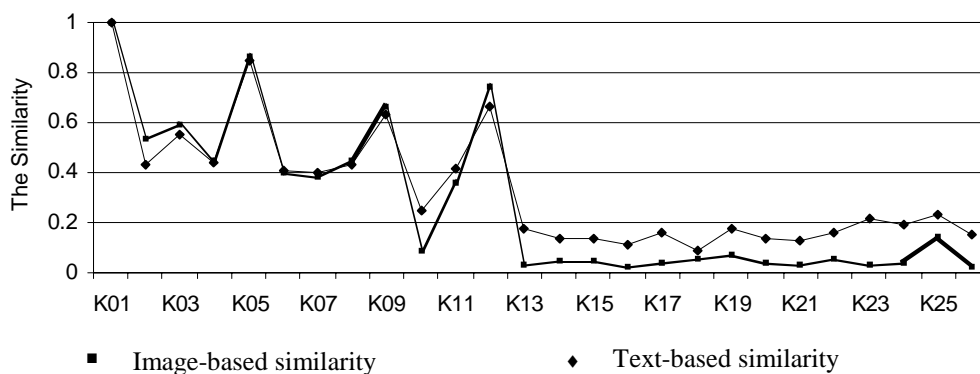
For each vertical bar pattern in the feature vector of the query document image, we look for the corresponding entry in the feature vector of the database document image. If the pattern is found, then their product is obtained by multiplying of their normalized frequencies, otherwise their product is simply zero since one of the feature vectors does not contain such a pattern. Because for all document images, the vertical bar patterns in the feature vectors are always in the form of strings containing “d”, “m” and “q”, no unification is needed and comparisons between these strings are straightforward to ensure the computational efficiency of the proposed method.

## 5. Experimental Results

Experiments were carried out to test the effectiveness of our image-based similarity measure in comparison with the traditional text-based similarity N-Gram algorithm. Four different corpora of document images were used in the following tests. To create the ASCII versions of these documents as a means of benchmarking, an OCR system was used to extract the text from the images. The extracted texts were corrected by hand for any error from the OCR.

Corpus One (K01 - K26) is made up of articles that were extracted from the Internet and were already electronically available. The news articles were printed using MS-Word in 10-point Times New Roman font. The printed documents were then scanned as images. These articles address four different kinds of topic, respectively. K01-K12 talk about economic crises in Brazil, K13-K17 refer to personal computers, K18-K21 tell of scholarship and K22-K26 describe the news of a nuclear

spy in US. For each topic, we picked the first one of each group as the query article and thus K01, K13, K18 and K22 were selected. Similarity measures of all the articles in this corpus with the respective four query articles were made using the image-based and text-based methods. Corpus Two (N01- N26) and Corpus Three (T01 – T26) are generated using the same set of documents as in Corpus One. But Corpus Two images are scanned in with varying degrees of poorer scan quality to simulate the degradation of image quality and Corpus Three images are scanned in from documents printed in larger font sizes. Experiments were carried out in a similar way as in Corpus One. Corpus Four (M1 – M24) contains documents selected from the previous three corpora, eight images from each corpus within which two images are selected from each of the four topics. Thus this corpus contains a mixture of documents having different scan qualities and different font sizes. We pick the first article in each topic as the query article, and measure similarities between all the articles in this group with the query articles. Figure 3 shows the curve plotted from the similarity values obtained for document K1 comparing with other documents in Corpus One.



**Figure 3. Comparison of Image-based and Text-based Similarity between K1 and other documents in Corpus One**

From the results obtained from experiments above, we can see that the result of the text version of documents provides more distinguishable similarity measures. This is because the vertical bar patterns extracted from the document images are not as distinguishable as words themselves in text documents since some words unfortunately have the same vertical bar pattern although this is not common. Also the decreased similarity values for Corpus Two and Three suggest that the

degradation of image qualities affects the consistency of the vertical bar patterns extracted to some extent. Nevertheless, the image-based similarity remains as an adequate means to retrieve similar news articles with respect to a query article. Furthermore, the results from Corpus Four show the encouraging ability of the proposed method to handle a heterogeneous set of documents in different font sizes and different image qualities, although the inconsistency of vertical bar patterns causes the results to be somewhat less favorable.

From the testing with the four corpora, it can be seen that a threshold may be set to decide whether a text is similar to a query article. The threshold lies somewhere in the region of 0.1 to 0.2. To further evaluate the performance of the proposed model, the *accuracy* and the precision (percentage of the number of correctly retrieved articles over the number of all retrieved articles) and recall (percentage of the number of correctly retrieved articles over the number of articles in the category) of the testing results are measured and presented. Knowing the number of articles in topic  $i$  (let it be  $n_i$ ), we first allowed the system to retrieve  $n_i$  topmost similar articles and determined how many of these  $n_i$  articles are about topic  $i$ . Let this number of correctly retrieved articles be  $m_i$ . We define *accuracy* of this retrieval process as  $m_i/n_i$ . We next retrieved articles based on the threshold instead of a pre-determined number of articles. We set threshold at 0.1, 0.15 and 0.2 in the next three experiments respectively, and find the values of precision and recall. We carried out the above experiments for all articles in the four corpora, and taking one query article at a time. The average precision and recalls were then obtained for each choice of threshold value. All accuracies, precisions and recalls are tabulated in Table 1. It can be seen that if the number of relevant articles are known beforehand, then retrieving that number of articles for a topic in question can achieve an average accuracy of 88%. Using a threshold as a basis of retrieval, one can see a trade-off between precision and recall. At the threshold of 0.2, the average precision and recall are 98.89% and 52.56%, respectively, whereas choosing 0.1 as the threshold will give an average precision and recall of 95.21% and 75.22%, respectively. Thus, setting a higher threshold gives a better precision

but poorer recall, and the reverse is true for a lower threshold. If the emphasis is on retrieving only relevant articles, then a 0.2 threshold should be used. On the other hand, if the intent is to retrieve as many as possible news articles, then a threshold of 0.1 may be adopted. Overall speaking, the 0.15 threshold appears to be a good compromise.

Query article	NR	Accuracy %	Threshold = 0.2		Threshold = 0.15		Threshold = 0.10	
			P %	R %	P %	R %	P %	R %
K1	12	91.67	100	91.67	100	91.67	91.67	91.67
K13	5	100	83.33	100	71.43	100	71.43	100
K18	4	100	100	25	100	50	100	75
K22	5	100	100	60	100	60	100	60
N1	12	91.67	100	83.33	91.67	91.67	73.33	91.67
N13	5	100	100	40	100	100	100	100
N18	4	75	100	25	100	25	100	50
N22	5	80	100	40	100	40	100	60
T1	12	91.67	100	75	100	75	91.67	91.67
T13	5	100	100	20	100	20	100	60
T18	4	50	100	25	100	25	100	25
T22	5	40	100	20	100	40	100	40
M1	8	100	100	50	100	83.33	100	100
M9	8	100	100	66.67	100	100	100	100
M17	8	100	100	66.67	100	83.33	100	83.33
Average		88.00	98.89	52.56	97.54	65.67	95.21	75.22

**Table 1. Overall performance evaluation of image-based document text retrieval (NR = Number of relevant documents in the group, P = Precision, R = Recall)**

## 6. Conclusion and Future Work

We propose an approach to measure document text similarity without the use of any text data but instead by relying on image features in the document. Features called the vertical bar patterns are extracted from word units in a document image through local extrema points detection to form the document vector. Document similarity is calculated by finding the scalar product between two vectors. Experiments using four corpora of news articles have confirmed the validity of the model with an average of precision ranging from 95.21% to 98.89%, an average recall ranging from 52% to 75.22%, depending on the similarity threshold.

The method is suitable for gauging the similarity of document images written in Latin languages. One of our future research directions is to extend the current method to handle

documents in other languages like Chinese and Tamil (a dominant Indian language used in Singapore). We can convert ASCII text documents into vertical bar patterns by a proper lookup table. In this way, the present text retrieval method based on text similarity can be applied to a mixture of imaged documents and ASCII text documents, as well as to a range of different languages.

## References

- [1] C. E. Shannon, "The Mathematical Theory of Communication," University of Illinois Press, Urbana, 1949.
- [2] E. J. Yannakoudakis, P. Goyal, and J. A. Huggill, "The Generation and Use of Text Fragments for Data Compression," *Inf. Proc. Mgt.* 18, 15, 1982.
- [3] P. Willett, "Document Retrieval Experiments Using Indexing Vocabularies of Varying Size. II. Hashing, Truncation. Digram and Trigram Encoding of Index Terms." *J. Doc.* 35, 296, 1979.
- [4] W. B. Cavnar, "N-Gram-based Text Filtering for TREC-2," The Second Text Retrieval Conference (TREC-2), NIST Special Publication 500-215, National Institute of Standards and Technology, Gaithersburg, Maryland, 1994.
- [5] Marc Damashek, "Gauging Similarity via N-Grams: Language-independent Sorting, Categorization, and Retrieval of Text," *Science*, 267, pp.843-848, 1995.
- [6] J. J. Hull; J. F. Cullen, "Document Image Similarity and Equivalence Detection," Proceedings of the Fourth International Conference on Document Analysis and Recognition (ICDAR'97), Volume 1, pp. 308-312, 1997.
- [7] 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.
- [8] 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.