Automatic Document Orientation Detection and Categorization through Document Vectorization

Shijian Lu, Chew Lim Tan
Department of Computer Science, School of Computing
National University of Singapore, Singapore
lusj@comp.nus.edu.sg, tancl@comp.nus.edu.sg

ABSTRACT

This paper presents an automatic orientation detection and categorization technique that is capable of detecting the orientation of multilingual documents with arbitrary skew and categorizing document images according to the underlying languages. We carry out orientation detection and categorization through document vectorization, which encodes document orientation and language information and converts each document image into an electronic document vector through the exploitation of the density and distribution of vertical component runs. For each language of interest, a pair of vector templates is first constructed through a training process. Orientation and category of the query image are then determined based on distances between the query document vector and the constructed vector templates. Experiments over 492 testing document images show that the average orientation detection and categorization rates reach up to 97.56% and 99.59%, respectively.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*shape*; I.7.5 [**Document and Text Processing**]: Document Capture—*document analysis, scanning*

General Terms

Algorithms, Experimentation

Keywords

Document Orientation Detection, Document Image Categorization

1. INTRODUCTION

Thanks to the advances of document capturing capabilities, more and more document images of different languages are being produced by scanners, digital cameras, and fax machines. Without human checking, physical documents may be improperly positioned as illustrated in Figure 1 during manual or machine feeding process. In addition, in a multilingual environment, visual inspection is required to categorize document images according to the underlying languages before the ensuing text recognition or retrieval. In some situation such as digital libraries, the manual aligning and checking processes are time consuming and prohibitively expensive. Automatic document orientation detection and document categorization according to the underlying languages are in demand to reduce the human involvements during document digitalization process.

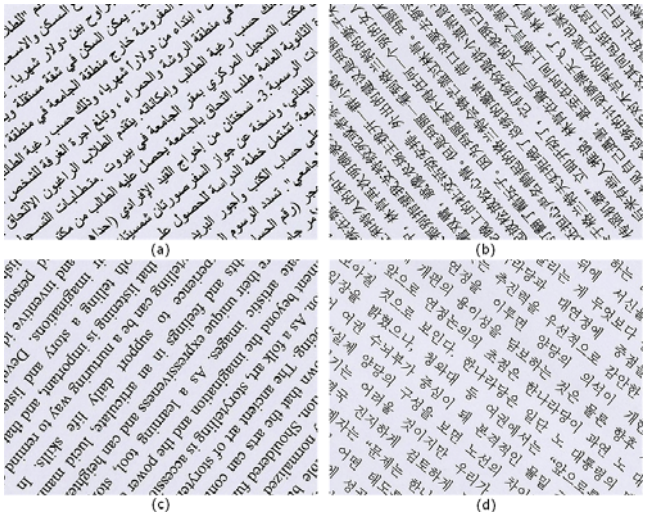

Figure 1: Sample documents of different languages with skew angle arbitrarily lying within (a) [0° 90°]; (b) [90° 180°]; (c) [180° 270°]; (d) [270° 360°], respectively.

Some image orientation detection works have been reported in the literature and they can be classified into three categories. The first two categories focus on document images, which deal with landscape & portrait detection [1, 2] and up-down orientation detection [3, 4, 5]. For landscape & portrait detection, global [1] and local [2] projection profiles are exploited and the largest variation gives the document orientation. For up-down detection, document orientation is uniformly determined based on the fact that the number of character ascenders is normally much larger than that of character descenders for Latin based text. The third category [6, 7] instead detects the orientation of natural scene images rotated by 90°, 180°, or 270°, respectively. Besides, some works have also been reported to identify languages from imaged documents using specific text tokens [10], statistics of shape features [9], or document texture [11].

In this paper, we propose a document orientation detection and categorization algorithm. The proposed algorithm

MM'06, October 23–27, 2006, Santa Barbara, California, USA.
ACM 1-59593-447-2/06/0010.

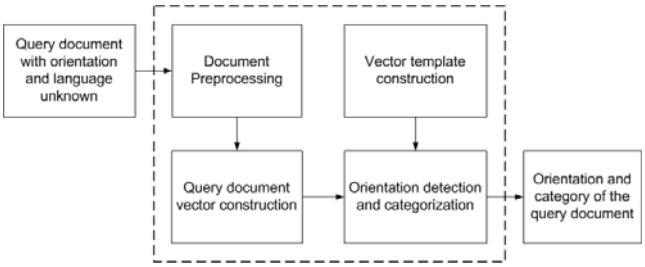

Figure 2: The framework of the proposed system.

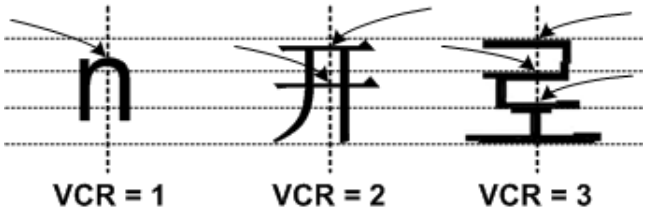

Figure 3: Definition of the vertical component run

is able to detect the orientation of document images of four different languages (Arabic, Chinese, Korean, and Roman) with arbitrary skew. At the same time, it is capable of classifying document images into categories according to the underlying languages. Compared with Roman letters, characters of some other languages such as Chinese have no ascenders or descenders. As a result, the up-down orientation cannot be detected like in [3, 4, 5]. Besides, the language identification works reported in [9, 10, 11] cannot detect document orientation either. We carry out orientation detection and categorization through document vectorization, which encodes document orientation and language information and converts each document image into a document vector through the exploitation of the density and distribution of vertical component run (*VCR*) [8].

Figure 2 gives the flowchart of the proposed system. Given a query document image with both underlying language and text orientation unknown, preprocessing including segmentation, noise removal, and skew correction is first carried out and a clean binary image is produced. The corresponding query document vector is then constructed based on the proposed document vectorization technique. Lastly, the orientation and language of the query document image are determined based on the distances between the query document vector and multiple document vector templates, which are pre-constructed using a set of training images of different languages at the correct and upside down orientation.

2. METHODS

2.1 Document Preprocessing

Preprocessing is required before document orientation detection and categorization. Firstly, document text must be segmented from the background and we adopt Otsu's method [12] for document binarization. Noise is then removed through the two rounds of size filtering described in [8]. Lastly, document preprocessing is finished through skew detection and correction, which have been studied extensively in the literature [13]. In this paper, we directly adopt our earlier work [14] for skew correction. It should be clarified that skew angle at this stage is assume to lie within the range $[-90^\circ\ +90^\circ]$. Therefore, text within the preprocessed document images may be either correctly oriented (Figure 1(a, d)) or upside down (Figure 1(b, c)).

2.2 Document Vectorization

We exploit the number and position of *VCR* for document orientation detection and categorization. Scanning from the top to the bottom, a *VCR* is detected when a vertical scan line passing through the component centroid enters the component regions from the background. The run position is located at the topmost pixel of the document component that meets the vertical scan line. To study *VCR* distribution, we divide text lines into three equidistant zones, namely, the upper zone around the x line, the middle zone around the middle line, and the base zone around the base line, respectively. Detected *VCR* can be classified to three text zones accordingly as illustrated in Figure 3.

We characterize the number and position of *VCR* using a document vector of dimension 32. The first 8 elements of the document vector characterize the density of *VCR* along text lines in the vertical direction. We set the upper limit of *VCR* number at 8 as the number of *VCR* in each scanning round is normally no bigger than 8 for most languages currently under study. Considering the fact that all *VCR* may occur within a single text zone, the upper limit of *VCR* number in three text zones is set at 8 as well. Each document can thus be transformed into a document vector of dimension 32 where the first 8 elements record the number of document components with *VCR* number equal to their indices and the following 24 elements record the position of *VCR* within the top, middle, and base text zones, respectively.

$$VRV = \left[N_1 \cdots N_8 \;\; U_1 \cdots U_8 \;\; M_1 \cdots M_8 \;\; L_1 \cdots L_8\right] \quad (1)$$

where N_i, i = $1 \cdots 8$, gives the numbers of document components with *VCR* number equal to i. U_i, M_i, and L_i, i = $1 \cdots 8$, define *VCR* positions within the three text zones.

Document vector encodes the underlying language information through the exploitation of *VCR* density. Take English and Chinese as examples. Most English characters hold at most three *VCR*, but a large portion of Chinese characters hold more than 4 *VCR*. On the other hand, document vector incorporates the underlying text orientation information based on the *VCR* distribution within the three text zones. For the English characters "n" in Figure 3, there is just one *VCR* occurring within the top text zone. As a result, N_1 and U_1 are set to 1 within the corresponding document vector. However, if character "n" is captured upside down, N_1 and L_1 are set to 1 instead because the only *VCR* occurs within the base zone in the upside down situation.

Since a document image will contain multiple characters, the corresponding document vector can be simply determined as the sum of the vertical run vectors of all preprocessed document components. Before orientation detection and categorization, document vector must be normalized to cancel the effects of document length difference:

$$\overline{VRV_i} = \begin{cases} \frac{VRV_i}{\sum_{j=1}^{8} VRV_j} & \text{if } i \leq 8 \\ \frac{VRV_i}{\sum_{j=9}^{32} VRV_j} & \text{if } i > 8 \end{cases} \quad (2)$$

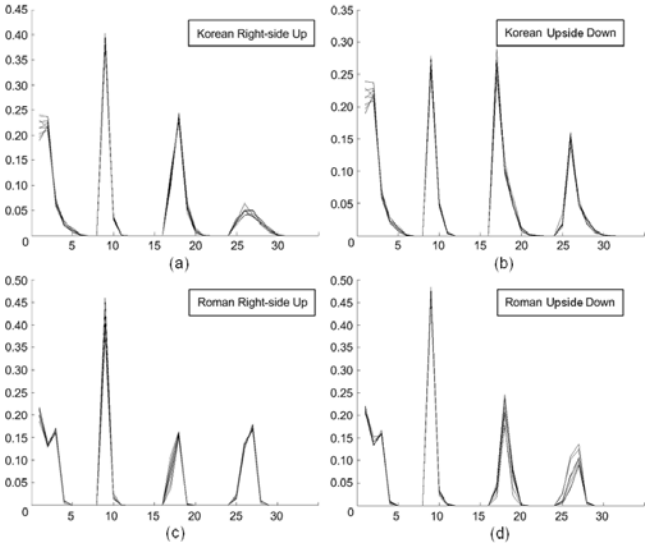

Figure 4: Documents vectors of Korean and Roman where every 8 document vectors are given for each language at the correct and upside down orientation.

where the normalization factors at $i \leq 8$ and $i > 8$ are equal to the number of the preprocessed document components and the number of *VCR* respectively.

Figure 4 shows document vectors of different languages at different orientations. For each of two languages (Korean in Figure 4(a, b) and English in Figure 4(c, d)) currently under study, every 8 document vectors are given at each of two opposite orientations (correct orientation in Figure 4(a, c) and upside down orientation in Figure 4(b, d)), demonstrating the density and distribution of *VCR* of the underlying language and text orientation. As Figure 4 shows, The documents vectors of the same language at the same orientation are quite close to each other. On the other hand, the document vectors of different languages and orientations are totally different. In particular, the density of *VCR* (the first 8 elements of document vectors) is mainly affected by the underlying languages, but the distribution of *VCR* (the last 24 elements of document vectors) is affected by both languages and text orientation information.

2.3 Orientation Detection and Categorization

We accomplish document orientation detection and categorization based on the document vectorization technique described in the last subsection. For each of the four languages currently under study, a pair of vector templates is first constructed, each of which encodes the density and distribution of the *VCR* of text at the two opposite orientations. Orientation and langauge of the query image are then determined based on the distance between the query document vector and the constructed vector templates.

120 training documents are created for vector template construction where every 30 are printed in one of the four languages under study. The 120 training documents are collected from different sources and document text is printed in different styles and fonts with each document containing 30-40 text lines. Each testing document is scanned at the correct and upside down orientations and so produces two testing document images. The scanning resolutions are all set at 600 ppi with skew angles controlled close to zero. Lastly, a training set is constructed using the document vector of the 240 training images.

$$VRV_{training} = \{VRV_i, i = 1 \cdot \cdots \cdot N_c\} \tag{3}$$

where N_c is equal to 240 and VRV_i denotes the document vector of the i^{th} training image.

Based on the constructed training set, the vector templates can be simply estimated as the mean of the document vectors of each langauge at each of the two opposite orientations. For the i^{th} langauge under study, the vector templates at two opposite orientation can be estimated as:

$$VT_i^u = \sum_{j=1}^{N_u} VRV_j^u \quad and \quad VT_i^d = \sum_{j=1}^{N_d} VRV_j^d \tag{4}$$

where N_u and N_d give the number of training document vector of the i^{th} language at two opposite orientations. VRV_j^u and VRV_j^d refer to the j^{th} document vector of the i^{th} language at the two opposite orientations, respectively.

Based on the constructed document vector templates, orientation and language of the query image can be determined according to the distance between the query document vector and the 8 constructed vector templates (every two for each language at two opposite orientations). We evaluate the vector distances using Bray Curtis distance, which has a nice property that its value always lies between 0 and 1:

$$VD_i = \frac{\sum_{j=1}^{N_v}(|VRV^j - VT_i^j|)}{\sum_{j=1}^{N_v}(VRV^j) + \sum_{j=1}^{N_v}(VT_i^j)} \tag{5}$$

where N_v is equal to 32. VRV_j represents the j^{th} element of the query document vector. Parameter VT_j^i corresponds to the j^{th} element of the i^{th} learned document vector template. As a result, the orientation and language of the query image are determined to be the same as that of the vector template with the smallest Bray Curtis distance VD_i.

3. EXPERIMENTS AND DISCUSSIONS

492 testing documents are prepared to evaluate the performance of the proposed method and they are printed in the four different languages as listed in Section 1. Each testing document contains 15-40 text lines and document texts are printed in different fonts and styles. All testing documents are first scanned at 600 ppi with skew angle distributing arbitrarily within the range [0° 360°]. 492 document vector are then constructed. Lastly, document orientation and language are determined as described in the last section. Experimental results show that document orientation detection and categorization rates reach up to 97.56% and 99.59%, respectively.

The performance of the proposed method depends heavily on the number of characters within the query document image. As the number of characters becomes smaller, the query document vector may not reflect the real density and distribution of *VCR*, which introduces errors. To test the effect of character number, we create a set of testing images as given in Table 1, all of which are cropped from the 492 testing images described above. Table 1 gives experimental results where orientation detection is only counted when documents are correctly categorized. As Table 1 shows, the performance of the proposed method deteriorates quickly

Table 1: Accuracy of document orientation detection and categorization in relation to the number of text lines. (DODA: document orientation detection accuracy; DCA: document categorization accuracy)

No of text lines	No of test images	DCA	DODA
12	55	98.18%	98.18%
10	73	97.26%	95.89%
8	86	97.67%	96.51%
6	114	94.74%	92.11%
4	149	90.60%	88.59%
2	228	85.53%	83.77%
1	284	77.82%	73.94%

while the number of characters is becoming small. As query documents contain just one or two text lines, the orientation detection and categorization results become unacceptable.

On the other hand, though we focus on the study of documents of four languages, the proposed method can be easily extended to deal with documents of a new language. To add a new language to the system, the only thing required is to create a set of training images of that language and estimate a pair of corresponding vector templates. If the *VCR* density and distribution of the new language are far different from that of the existing ones, the orientation and category of the query document of the new langauge can normally be correctly determined. To test the extendability of the proposed method, we create 60 Bangla documents and use 20 for template construction and another 40 for testing. Experimental results show that orientation detection and categorization rates reach 95% and 97.5%, respectively.

4. CONCLUSION

This paper reports an orientation detection and categorization technique that is capable of detecting the orientation of document images and categorizing documents according to the underlying languages. In the proposed method, orientation detection and categorization are accomplished through document vectorization, which converts each document image into a document vector through the exploitation of the density and distribution of vertical component runs. Experimental results show the proposed method is accurate and easy for implementation.

As discussed in the last section, though the proposed method works fairly well when query documents contain a large number of characters, the performance may deteriorate as the number of characters becomes smaller. Character-level orientation detection and categorization techniques are required for documents that contain just one or a few text lines. Besides, for some languages with *VCR* density and distribution similar to that of the existing ones, the documents of that language cannot be processed by the proposed method either. Some other shape statistics are needed to to compensate the *VCR* inefficiency under such circumstance. We will study these two problems in future.

5. ACKNOWLEDGMENTS

This research is supported by Agency for Science, Technology and Research (A*STAR), Singapore, under grant no. 0421010085.

6. REFERENCES

[1] T. Akiyama and N. Hagita, Automated entry system for printed documents, Pattern Recognition, 23(11):1141–1154, 1990.

[2] D. S. Le and G. R. Thoma and H. Wechsler, Automated Page Orientation and Skew Angle Detection for Binary Document Images, Pattern Recognition, 27(10):1325–1344, 1994.

[3] , B. T. Ávila and R. D. Lins, A fast orientation and skew detection algorithm for monochromatic document images, ACM symposium on Document engineering, pages 118–126, 2005.

[4] D. Bloomberg and G. Kopec and L. Dasari, Measuring document image skew and orientation, SPIE 2422, pages 302–316, 1995.

[5] R. S. Caprari, Algorithm for text page up/down orientation determination, Pattern Recognition Letters, 21(4):311–317, 2000.

[6] A. Vailaya and H. Zhang and C. Yang and F. Liu and A. K. Jain, Automatic image orientation detection, IEEE Transactions on Image Processing, 11(7):746–755, 2002.

[7] S. Lyu, Automatic Image Orientation Determination with Natural Image Statistics, Proceedings of the 13th annual ACM international conference on Multimedia, pages 491–494, 2005.

[8] S. Lu and C. L. Tan, Script and language identification in degraded and distorted document images, Proceedings of the 21th National Conference on Artificial Intelligence (AAAI), 2006, Accepted.

[9] A. L. Spitz, Determination of Script and Language Content of Document Images, IEEE Transaction on Pattern Analysis and Machine Intelligence, 19(3):235–245, 1997.

[10] J. Hochberg and L. Kerns and P. Kelly and T. Thomas, Automatic Script Identification from Images Using Cluster-based Templates, IEEE Transaction on Pattern Analysis and Machine Intelligence, 19(2):176–181, 1997.

[11] T. N. Tan, Rotation Invariant Texture Features and Their Use in Automatic Script Identification, IEEE Transaction on Pattern Analysis and Machine Intelligence, 20(7):751–756, 1998.

[12] N. Otsu, A Threshold Selection Method from Graylevel Histogram, IEEE Transactions on System, Man, Cybernetics, 19(1):62–66, 1978.

[13] J. J. Hull and S. L. Taylor, Document image skew detection: Survey and annotated bibliography, Document Analysis Systems, pages 40–64, World Scientific, 1998.

[14] Y. Lu and C. L. Tan, A nearest-neighbor-chain based approach to skew estimation in document images, Pattern Recognition Letters, 24(14):2315–2323, 2003.