

Script Identification of Camera-based Images

Linlin Li, Chew Lim Tan

School of Computing, National University of Singapore

{lilinlin,tancl}@comp.nus.edu.sg

Abstract

This paper reports a statistical script identification technique that determines the script of document images, especially camera-based images which suffer from perspective distortion. The identification technique represents a document image by a frequency vector of affine invariant signatures of characters, and identifies the script by comparing the vector with pre-prepared script templates. Experimental results show that our method is tolerant to moderate perspectives, document skew and various image noises.

1 Introduction

Script identification is to determine the script in which a document image is written. It plays an essential role in document image processing under a multilingual environment. Accurate and efficient script identification is crucial to downstream processing steps like document image recognition, indexing, and searching. Many methods have been reported for script identification. They can be classified into three categories. The first category is component based. Hochberg [4] proposed a method to identify the script by comparing characters of an unknown image to an exhausted list of characters of a certain script, which is gotten from training document images by clustering. Another category is based on the horizontal projection profile of certain features. In Spitz's method [6], upward concaves were used, while in Lu's work [5], vertical runs were employed. The third category is texture based. Busch [1] investigated the use of texture like gray-level co-occurrence, energy, and wavelets to differentiate scripts. However, all these approaches focused on images generated by scanners, where an image is the fronto-parallel projection of the document plane.

With the rapid development of camera technique, a camera becomes an important tool to take document image. It is well known that images taken by a camera suf-

fer from perspective distortion, which seldom appears in an image generated by a scanner. Hence, document image processing techniques developed specifically for scanned images no longer work on camera-based images. Thus script identification methods also face the same problem. The skew of a camera-based image is often more severe and unpredictable than that of a scanned image. Therefore, it is difficult for a component-based approach to train an appropriate representative character set from images of all possible skew angles. In addition, text-lines in an image are no longer parallel to each other, and thus the skew angle of each text-line is different from each other. Besides, characters within the same text-line no longer remain the same height, namely, characters near the camera lens are larger than those further away. These two attributes make the texture based approaches and the profile based approaches fail. Because the former are sensitive to skew. An accurate de-skewing processing is inevitable before extracting texture from images. The latter assumes that the Euclidian distance between each feature point and the base-line remains the same, but this assumption does not hold under perspective distortion.

In this paper, an effective method is presented in order to identify the script of camera-based images. In this method, a character is represented by an affine-invariant signature, and a document image is represented by a frequency vector in the signature frequency space.

2 Approach

The basic assumption of the identification method is that: for **each character** in the image, the perspective distortion can be approximated by an affine distortion. Theoretically, when the size of the perceived object depth is much smaller than the distance between the camera lens and the object, the perspective transformation can be approximated by an affine transformation. Practically, English characters printed on an A4 sheet is within a $2 \times 2 \text{ mm}^2$ bounding box. In order to take a photo of the whole sheet, the distance between

- (a) trapped underneath it, but the outlines of the lost
 (b) رئيس اتحاد الاذاعة والتلفزيون المصري الاسبق والاذاعي سعد
 (c) 装备。式地对舰导弹就是其中之一。开发期间在美国进

Figure 1. Three scripts: (a)English, (b)Arabic, (c)Chinese.

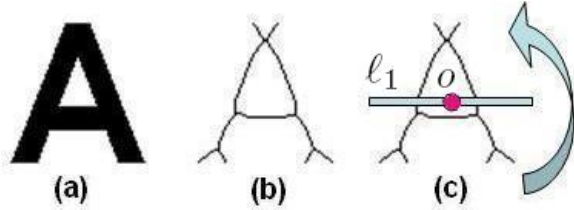


Figure 2. Signature generating process.

the camera and the projection center on the sheet is at least 30 mm. Because the image is taken for reading purpose, the camera projection angle is nearly perpendicular to the sheet plane, thus the object depth is very small. Hence, the affine assumptions holds in this case. In fact, a similar assumption has been widely employed in rectifying camera-based document images under in previous studies [2][7].

2.1 Signature Generating

The intuition of generating signatures is that: strokes of characters of different scripts have different complexity levels, which can be quantified by the number of intersections. Assume there is a vertical line passing through the centroid of a character, and the number of intersections between the line and strokes of the character is k . k of English characters, shown in figure 1(a), often ranges from 1 to 4; k of Arabic characters (figure 1(b)) is often equal to 1; Chinese characters (figure 1(c)) often have a larger k . Similar property has been employed in [5]. However, due to unpredictable skew, it is impossible to find the accurate direction perpendicular to the text-line. Hence, we propose a signature-generating method which is able to capture the intersections directly regardless of the text-line direction.

Assume there is a component C , such as character 'A' shown in figure 2(a), and the pixel sequence of the convex hull of C is $\{p_1, p_2, \dots, p_n\}$, where p_2 is the anti-clock-wise neighbor pixel of p_1 , and p_3 is the anti-clock-wise neighbor pixel of p_2 , and etc. The centroid of the convex-image of C is denoted by o . The signature of C is constructed as follows:

1. The centroid o of the convex image of C and the convex hull pixel sequence are first located. The skeleton of C , as shown in figure 2(b), is gotten by a thinning operation.
2. The line ℓ_1 defined by o and p_1 is found, shown as the bar in figure 2(c). There are several intersections between ℓ_1 and the skeleton of C . For each pair of these intersections, denoted by i_u and i_v , the length ratio λ_{uv} is calculated as:

$$\begin{cases} \lambda_{uv} = \frac{oi_u}{oi_v}, & i_u \text{ and } i_v \text{ are at different sides of } o \\ \lambda_{uv} = -\frac{oi_u}{oi_v}, & i_u \text{ and } i_v \text{ are at the same side of } o \end{cases} \quad (1)$$

where oi_u and oi_v are the Euclidean distances between o , i_u , and i_v respectively.

3. Repeat step 2 on $\{p_2, \dots, p_n\}$, namely, The bar is rotated 360 degrees around o . Length ratios are collected in the meantime.
4. A histogram is constructed for C to record the number of occurrences of length ratios λ_{uv} , if $|\lambda_{uv}| > 1$. In the experiment, we used a histogram starting with -5 and ending with 5, with n bins. In particular, bin i keeps a record of the number of length ratios within the range $(-5 + i \times \frac{10}{n}, -5 + (i + 1) \times \frac{10}{n}]$.
5. The signature of C is gotten by normalizing the histogram.

According to affine geometry, the number of intersections defined by the projection of ℓ_i and the projections of the skeleton keeps constant, and the length ratio of line segments on a given line remains constant, when C is under affine distortion. Also, It has been proved that the centroid of a convex polygon preserves under affine transformation [3], namely, the affine projection of o remains the centroid of the affine projection of the convex image. Therefore, the normalized ratio histogram (signature) remains constant when C is under affine transformation.

2.2 Script Template Generating

One template is generated for each candidate script. The script template is a frequency vector of signatures. For each script, a few training document images in fronto-parallel view are prepared. Signatures are extracted and classified into clusters by a hierarchy clustering algorithm, with cosine distance as the distance measure and 0.02 as the maximum radius of a cluster. The 30 biggest clusters are chosen, and a template comprises of the centroid and the size of the chosen clusters (the number of members in the cluster, denoted by $freq$). The format of a template is as: $\{(signature_1, freq), \dots, (signature_{30}, freq)\}$.

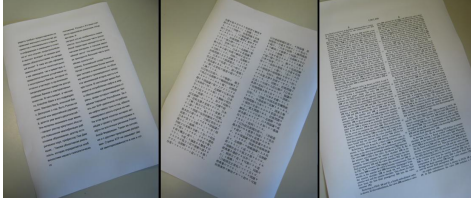


Figure 3. Testing images.

The similarity between an query document Q and a script template T is defined by:

$$sim(Q, T) = \frac{\sum freq(t_i, Q) \times t_i.freq}{\sum t_i.freq} \quad (2)$$

where t_i is a signature in the template. $freq(t_i, Q)$ is a function to find the frequency of t_i in Q . It works as follows: for each signature q_j of Q , if t_i is the nearest template signature of q_j , and the cosine distance between them is smaller than 0.02, the frequency of t_i in Q increases by 1. The script template which has the highest similarity with Q gives its script identity.

3 Experimental Results and Discussion

Ten scripts under study were 1:Arabic, 2:Chinese, 3:Cyrillic, 4:Greek, 5:Hebrew, 6:Japanese, 7:Korean, 8:Roman, 9:Thai, and 10:Bengali. These scripts were chosen because of their popularity. In the training data set, ten synthetic images in fronto-parallel view in each script were generated. In the testing data set, photos (3072×2304 pixels) of ten printed images in each script were taken by a camera. Since it is natural that a printed paper has some warping, this distortion was also kept in the photo. The criterion of taking the photos is that, all characters in the photo should be recognizable to people. Examples of the testing data is shown in figure 3.

3.1 Language Identification Results

In the experiment, Hochberg’s [4] was chosen as the baseline. In this method, connected components were first extracted, and were scaled to a 30×30 pixels size. A hierarchy clustering algorithm, with hamming distance as the distance measure, was employed to classify component images into clusters. The template comprised of the centroid of all clusters. The similarity between a template and a query was estimated by averaging the smallest hamming distance between each components in the query document and the template. We did not use the other methods mentioned in the introduction section because of their inability to deal with

Output	Ground Truth									
	1	2	3	4	5	6	7	8	9	10
1										
2						2				
3										
4		1			1					
5								2		
6		3								
7										
8										
9										
10										
Err.	0	4	0	0	1	2	0	2	0	0

Table 1. Confusion matrix of the proposed method.

Output	Ground Truth									
	1	2	3	4	5	6	7	8	9	10
1		1								
2	2		1			3			1	
3				2						
4		1	2		1					2
5		3		1		1		3		
6			1							
7										
8		1			3				2	2
9	2						1			
10										2
Err.	4	6	4	3	4	4	1	3	3	6

Table 2. Confusion matrix of Hochberg’s method.

skew as well as unparallel text lines caused by the perspective distortion.

Table 1 and 2 show the confusion matrices of language identification results of our and Hochberg’s method, respectively. An item in both tables is the number of documents in script i (ground truth) which were identified as script j (output). The proposed method was able to determine the scripts of testing images with 91% accuracy, while the baseline method was not able to deal with many of these images. We found that performance of Hochberg’s method highly depended on the skew: it worked good on those images with small skew (within $\pm 5^\circ$), but failed on those with severe skew. The performance of our method seemed to be affected by the type of script in which the testing image was,

	1	2	3	4	5	6	7	8	9	10
1	0.0000	0.6680	0.4661	0.8437	0.7364	0.8008	0.9381	0.9840	0.6318	0.9218
2	0.6773	0.0000	0.7051	0.5948	0.8042	0.3611	0.8678	0.9563	0.6330	0.9759
3	0.4661	0.7051	0.0000	0.8723	0.7524	0.9366	0.9787	0.5166	0.7224	0.9678
4	0.9102	0.5948	0.8723	0.0000	0.9297	0.9297	0.6533	0.9942	0.6953	0.9747
5	0.7364	0.8042	0.7524	0.9297	0.0000	0.9777	0.9903	0.2550	0.4355	0.8896
6	0.8008	0.3806	0.9366	0.6200	0.9777	0.0000	0.9841	0.8348	0.6838	0.9734
7	0.9381	0.8472	0.9787	0.5580	0.9773	0.9841	0.0000	0.9992	0.9830	0.9955
8	0.9816	0.9563	0.5412	0.9942	0.1473	0.8348	0.9992	0.0000	0.3385	0.9052
9	0.6258	0.7647	0.7224	0.6953	0.4658	0.7059	0.9830	0.3385	0.0000	0.9260
10	0.9218	0.9759	0.9678	0.9747	0.8896	0.9734	0.9955	0.9421	0.9260	0.0000

Table 3. Cosine distances between pairs of script templates.

which will be explained further in the next subsection.

It is worth noting that, the number of bins is an essential parameter for a better performance. The binarization process may suffer from pixel quantization; the centroid computation and skeletonizing steps may be tampered by noise. Hence length ratios may fluctuate accordingly. A signature with wide bins will be more tolerant to the fluctuation, but consequently it may have less discriminating power among characters. On the contrary, a signature with narrow bins is more discriminating, but it is more fragile to noise. $n = 20$ was set in the experiment.

3.2 Similarity Between Templates

The identification performance highly depends on the templates. If two templates are similar to each other, it is very likely that documents in one script are mistaken for another in the identification. Therefore, the similarity of templates are compared. Table 3 shows the cosine distance between each pair of template i and j . A cell on the diagonal is the distance between a template and itself, thus equal to 0.

Table 3 shows that two groups of templates are similar to each other: the first group is 2:Chinese and 6:Japanese; the second one is 5:Hebrew, 8:Roman, and 9:Thai. This explains why errors often occur within both groups in table 1. The table also indicates that the performance can be improved by preparing more discriminating templates of these scripts.

Although templates in the second group are closer to each other than those in the first group, errors occurred more frequently in the first group in the experiment. A possible reason is that, Chinese and Japanese both have thousands of frequently used characters, and a template with 30 signatures is not enough to incorporate them. Increasing the size of templates for Chinese and Japanese may help with this problem.

4 Conclusion

This paper proposes a script identification method for camera-based images based on statistics of an affine invariant signature. This method provides possibilities to train on frontal-parallel document images, but construct a classifier which is able to identify the script of images with perspective distortion. Further examination of the discriminating power of the method on more scripts will be done in future. In particular, the method showed some weakness in differentiating within a few script groups. The problem will be addressed by employing a template of more signatures or eliminating signatures shared by several templates in future.

Acknowledgment: This research is supported in part by IDM R&D grant R252-000-325-279.

References

- [1] A. Busch, W. Boles, and S. Sridharan. Texture for script identification. *PAMI*, 27(11):1720–1732, 2005.
- [2] P. Clark and M. Mirmehdi. Recognizing text in real scenes. *IJDAR*, 4(4):243–257, 2004.
- [3] C. Gope and N. Kehtarnavaz. Affine invariant comparison of point-sets using convex hulls and hausdorff distances. *Pattern Recognition*, 40(1):309–320, 2007.
- [4] J. Hochberg, L. Kerns, P. Kelly, and T. Thomas. Automatic script identification from images using cluster-based templates. *PAMI*, 19(2):176–181, 1997.
- [5] S. Lu and C. L. Tan. Script and language identification in degraded and distorted document images. *Proceedings of AAAI*, 2006.
- [6] A. Spitz. Determination of the script and language content of document images. *PAMI*, 19(3):235–245, 1997.
- [7] T. Yamaguchi, M. Maruyama, H. Miyao, and Y. Nakano. Digit recognition in a natural scene with skew and slant normalization. *IJDAR*, 7(2-3):168–177, 2005.