# Video Script Identification based on Text Lines

[a]Trung Quy Phan, [a]Palaiahnakote Shivakumara, [a]Zhang Ding [b]Shijian Lu and [a]Chew Lim Tan
[a]School of Computing, National University of Singapore, Singapore
[a]{phanquyt, shiva, zding and tancl}@comp.nus.edu.sg
[b]Institute for Infocomm Research, Singapore, [b]slu@i2r.a-star.edu.sg

*Abstract*—**In this paper, we present a new method for video script identification which is essential before choosing an appropriate OCR engine for identifying text lines when a video frame contains more than one language. The input for script identification is the text lines obtained by our text detection method. We extract upper and lower extreme points for each connected component of Canny edges of text lines. The extracted points are connected to study the behavior of upper and lower lines. The direction of each 10-pixel segment of the lines is determined using PCA. The average angle of the segments of the upper and lower lines is computed to study the smoothness and cursiveness of the lines. In addition, to discriminate the scripts accurately, the method divides a text line into five equal zones horizontally to study the smoothness and cursiveness of the upper and lower lines of each zone. We evaluate the method by conducting experiments on different combinations of languages such as English and Chinese, English and Tamil, Chinese and Tamil, and English, Chinese and Tamil.**

***Keywords-*** ***Video text line, Upper and lower points, Smoothness, Cursiveness, Video scrpt line identification***

## I. INTRODUCTION

Text information extraction from images and videos is an important task for automatic content-based indexing and retrieval. The process of text extraction consists of several steps: (1) Text detection aims to locate text regions in images/videos, (2) Text localization determines the exact boundary of text lines, (3) Text segmentation and binarization separate text pixels from the surrounding background pixels, which results in binary images where text appears as black pixels and background appears as white pixels or vice versa, (4) Text recognition performs OCR on binarized images and converts binarized images into ASCII text [1].

Several methods have been proposed for text detection and text localization in video [2-5]. These methods can be classified as connected component-based, texture-based, and gradient and edge-based methods. Although these methods achieve good accuracy for text detection and localization irrespective of fonts, font sizes, types of text and orientation, they do not have the ability to differentiate different languages in the same video frame because the main intention was to identify text boundary [2-5] and not to identify the script. There are text recognition methods which take care of segmentation and binarization of the text areas by proposing enhancement criteria to increase the contrast of text lines before feeding them to OCR [6-9]. However, when the frame contains more than two scripts or the dataset includes different languages then the performance degrades because the extracted features and OCR are usually designed for specific languages. Furthermore, in multi-lingual countries like Singapore and India, there is more than one official language and thus it is common to have multiple languages in the same video frame. In this work, we consider three scripts, namely English, Chinese and Tamil as they are the official scripts used in Singapore. Therefore, there is immense scope for identifying the scripts before selecting the appropriate OCR to improve the performance as it is hard to develop a universal OCR. There is a paper on video script recognition, which uses statistical and texture features with k-nearest neighbor classifier to identify Latin and Ideographic text in images and videos [1]. This method works well for English and Chinese but not for other scripts like Tamil. In addition, its performance depends on the classifier.

The problem of recognizing the scripts in printed materials is a familiar topic in document analysis. An overview of script identification methodologies based on structure and visual appearance is presented in [10]. It is noted from this review that the proposed methods work well for camera-based images but not for video frames since the latter has low contrast and complex background. The rotation invariant features for automatic script identification are proposed by Tan [11] based on Gabor filter. This work considers 6 scripts for identification. Busch et al. [12] have explored the combination of wavelet and Gabor features to identify the scripts. However, these approaches expect a large number of training samples to achieve good classification rate. Lu and Tan [13] have proposed a method for script identification in noisy and degraded document images based on document vectorization. Although the method is tolerant to various types of document degradations, it does not perform well for Tamil because of the complexity of the script. Texture features based on Gabor filter and Discrete Cosine Transform are used for script identification at the word level in many papers [14-16] where the methods expect high contrast document for segmentation of words. Similarly, a study of character shape for identifying scripts is proposed in [17-18]. These methods perform well as long as segmentation works well and the character shape is preserved. Online script identification is addressed in [19] where the spatial and temporal information is used to recognize the words and text lines. From our literature review, we observe that most of

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.250

the papers used Roman and Devanagiri as the common scripts and a few papers consider English, Chinese and Tamil for identification purpose [15]. In addition, the main focus of these methods is the identification of scripts in documents with plain background and high contrast but not the scripts in video. To the best of our knowledge, none of the papers addressed the problem of script identification in video, in particular the identification of English, Chinese and Tamil.

Therefore, this paper presents a new method for video script identification based on the smoothness and cursiveness of upper and lower lines in each zone of text lines in videos.

## II. PROPOSED METHOD

Since our objective is to identify the scripts, we use our developed text detection method [5] for text line detection from the video frames. Therefore, for this work, Canny edge map of text lines is inputted. The reason for choosing Canny is that Canny gives better edges compared to other edge detectors such as Sobel for low contrast image when the image background is relatively low compared to the whole image background. The proposed method is divided into two subsections – distinct feature extraction based on upper and lower extreme points of text lines is presented in Section A. Representatives for identification of scripts are computed in Section B.

### A. New Features for Script Identificaiton

It is observed from Edge map of English, Chinese and Tamil images that edge components in Chinese and Tamil are more complex than edge components in English because generally edge components of Chinese and Tamil are more cursive than edge components of English. Besides, we also observe that there are more sub components in each component of Chinese and more modifiers in Tamil text line image compared to edge components in English. To extract these observations and to study the local information, we divide the whole edge map of text line into five equal zones in horizontal direction. For each edge component in each zone, the method identifies upper and lower extreme points and connects them separately into upper and lower boundary lines for each zone (Fig. 1f and 1h).

To measure the smoothness and cursiveness of the upper and lower lines, we divide each line into a set of 10-pixel segments $\{s_i\}$. The reason for choosing 10 pixel segments here is that PCA gives correct angles for 10 pixel segments. The average angle of the whole line is computed as follows:

$$\vec{v_i} = PCA(s_i) \qquad (1)$$

$$\theta_i = arctan\left(\frac{|\vec{v_i}_y|}{|\vec{v_i}_x|}\right) \qquad (2)$$

$$\theta_{avg} = \frac{1}{N}\sum_{i=1}^{N}\theta_i \qquad (3)$$

$PCA(.)$ returns the first principal component of segment $s_i$. $\theta_i \in [0,\ 90)$ since we are interested in how much a segment deviate from the x-axis, i.e. 0 degrees. Therefore, we have extracted two values from each zone: the average upper line angle and the average lower line angle (Figs. 1, 2 and 3).

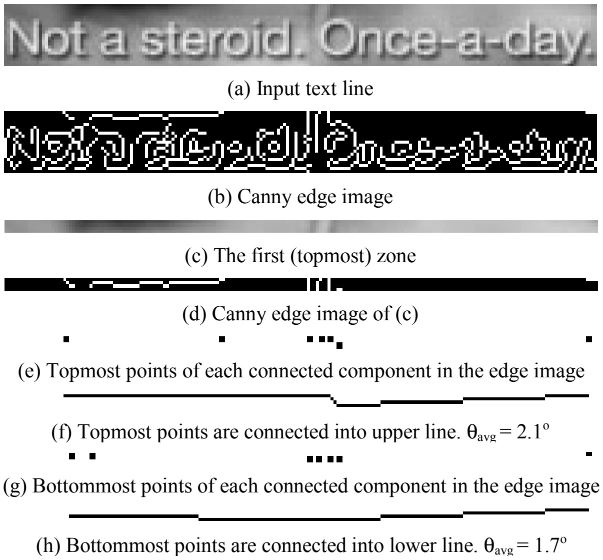
We then propose a simple measure that is the sum of angles of segments of upper and lower lines of the five zones to study. The sum is expected to give high value for Chinese and Tamil scripts compared to English script because of the less cursive nature in English upper and lower lines and the greater number of sub components in Chinese and modifiers in Tamil.

Figure 1(a) shows an input English text line with its Canny edge map in Figure 1(b). The first top most zone out of the five zones is shown in Figure 1(c) with its corresponding canny to image shown in Figure 1(d). The upper points for the line are shown in Figure 1(e) with their connecting line shown in Figure 1(f) with an average angle of 2.1 degree. Similarly, the bottommost points in this zone are shown in Figure 1(g) with their connecting line in Figure 1(h) with a computed angle of 1.7 degree. Figures 2 and 3 show similar sets of lines where we only show the top and bottom connecting lines for the first zone. One can notice from Figures 1-3 that the average angle of the upper and lower lines for English is lower than the average angle of the upper and lower lines for Chinese and Tamil text lines. Therefore, we use the average angle as a feature for the purpose of classification in this work.

### B. Representative for the Classes

We propose K-NN with K = 1 classifier to classify the scripts in this work. Since the problem is a two class problem, K=1 is enough. Before using K-NN classifier, we compute a representative for a class by averaging the features (sum of angles of segments of upper and lower lines of text line images) of all images in the class. In other words, given a training set for a particular class, the average number of features over all images in that class is the representative number of the class. Thus, a particular class is represented by a single number. Classifier takes absolute difference between feature and representatives to find a nearest neighbor. This is the advantage of our work compared to other script identification work where they use different classifier with large dimensional vector to classify the scripts. Given an unknown image to be recognized, it is given the label of the class whose representative number is closest to the sum of features of the unknown image.

Representing a class of images by a single value has been studied earlier in scene category classification [20]. We compute representatives for 10% and 50% training samples per class chosen randomly for English, Chinese and Tamil scripts as shown in Table 1 where one can notice that the difference between representative of English, Chinese and Tamil is huge compared to the difference between representative of Chinese and Tamil. This shows that this work classifies English and Chinese, and English and Tamil well but it fails for classifying Chinese and Tamil, and English, Chinese and Tamil (3 classes). Generally, most of the time the video comes with English and Chinese or Tamil but rarely, we can see Chinese and Tamil in one video and all three in one video.



(a) Input text line

(b) Canny edge image

(c) The first (topmost) zone

(d) Canny edge image of (c)

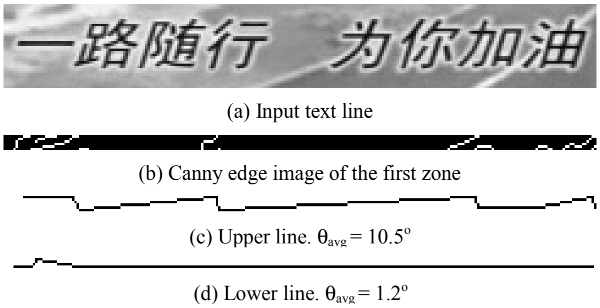
(e) Topmost points of each connected component in the edge image

(f) Topmost points are connected into upper line. $\theta_{avg} = 2.1^o$

(g) Bottommost points of each connected component in the edge image

(h) Bottommost points are connected into lower line. $\theta_{avg} = 1.7^o$

Figure 1. Sample upper and lower lines of an English text line. Note that in (e) and (g), if a column contains more than one extreme points, only the topmost one or bottommost one is selected, depending on whether the current line is upper line or lower line.



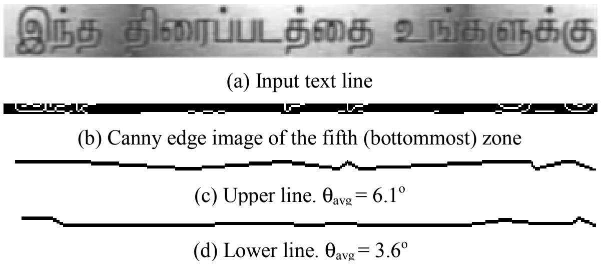
(a) Input text line

(b) Canny edge image of the first zone

(c) Upper line. $\theta_{avg} = 10.5^o$

(d) Lower line. $\theta_{avg} = 1.2^o$

Figure 2. Sample upper and lower lines of a Chinese text line. Chinese text lines have larger angles, i.e. more cursive, than English text lines.



(a) Input text line

(b) Canny edge image of the fifth (bottommost) zone

(c) Upper line. $\theta_{avg} = 6.1^o$

(d) Lower line. $\theta_{avg} = 3.6^o$

Figure 3. Sample upper and lower lines of a Tamil text line. The bottommost zone is shown here because Tamil text lines have large angles in this zone due to bottom modifiers.

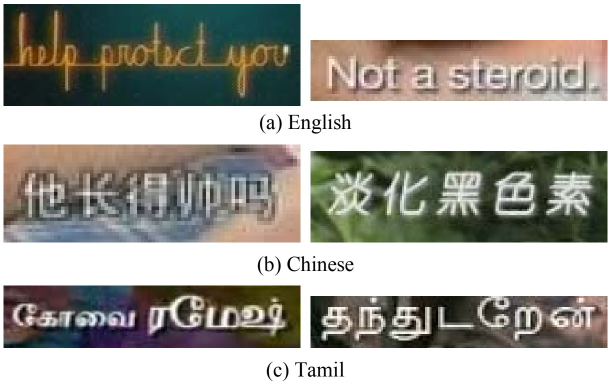
Hence, this work is useful in context to Singapore. Further, it is also observed from Table 1 that there is no much difference in representatives 50% samples and 10% samples per class. Therefore, we can conclude that the proposed method requires minimum supervision and number of training samples are not important for achieving accuracy. In Table 1, No. Images denote the number of images considered for computing representative values and Rep denotes representative. These representatives are used for experimentation and evaluation of the method in the next section.

Table 1. Representatives for classification

| Class | 50% samples | | 10% samples | |
|---|---|---|---|---|
| | No. Images | Rep | No. Images | Rep |
| English | 100 | 45.1 | 20 | 46.4 |
| Chinese | 75 | 85.0 | 15 | 83.0 |
| Tamil | 75 | 81.8 | 15 | 77.2 |

## III. EXPERIMENTAL RESULTS

As it is reported in Table 1, the database includes 200 English text lines, 150 Chinese text lines and 150 Tamil text lines. Since no benchmark database is available, we have created in total 500 images with three classes in this work. The sample images from our database for different scripts are shown in Figure 4. In order to show that our proposed method is effective and robust in terms of classification rate, we conduct experiments on different groups such as experiments on English and Chinese, English and Tamil, Chinese and Tamil and English, Chinese and Tamil and respective confusion matrix and classification rate are shown Table 2-5. We show that our method works well for identification of scripts by testing on 50% and 90% samples per class.



(a) English

(b) Chinese

(c) Tamil

Figure 4. Sample images in the dataset.

### A. Experiments on English and Chinese

The confusion matrix and classification rate for English and Chinese script classification are reported in Table 2.

From Table 2, it is observed that the proposed method gives good classification rate for two types of experimentation.

Table 2. Performance on English and Chinese scripts

| Matrix | 50% images per class for testing | | 90% images per class for testing | |
|---|---|---|---|---|
| | English | Chinese | English | Chinese |
| English | **69.0** | 31.0 | **77.2** | 22.7 |
| Chinese | 12.0 | **88.0** | 9.6 | **90.3** |
| Average | **78.5** | | **83.7** | |

### B. *Experiments on English and Tamil*

The confusion matrix and classification rate for English and Tamil scripts classification are reported in Table 3. Table 3 shows that the proposed method gives good classification rate for two types of experimentation to classify English and Tamil scripts.

Table 3. Performance on English and Tamil scripts

| Matrix | 50% images per class for testing | | 90% images per class for testing | |
|---|---|---|---|---|
| | English | Tamil | English | Tamil |
| English | **67.0** | 33.0 | **74.4** | 25.5 |
| Tamil | 4.0 | **96.0** | 15.5 | **84.4** |
| Average | **81.5** | | **79.4** | |

### C. *Experiments on Chinese and Tamil*

We also conduct experiments on Chinese and Tamil script classification to test our proposed method's performance. Table 4 shows that the proposed method is not good for classification of Chinese and Tamil scripts as it gives poor classification rate for 50% and 90% images per class for testing. Hence, the proposed method fails for identification of Chinese and Tamil scripts.

Table 4. Performance on Chinese and Tamil scripts

| Matrix | 50% images per class for testing | | 90% images per class for testing | |
|---|---|---|---|---|
| | Chinese | Tamil | Chinese | Tamil |
| Chinese | **60.0** | 40.0 | **60.7** | 39.2 |
| Tamil | 88.0 | **12.0** | 72.5 | **27.4** |
| Average | **36.0** | | **44.0** | |

### D. *Experiments on English, Chinese and Tamil*

In this experiment, we show that the proposed method is capable of identifying scripts when video contains three scripts in one frame. The confusion matrix and classification rate for the two three are given in Table 5. Table 5 shows that the proposed method gives good results for English (E) to Chinese (Ch) and English to Tamil (Ta) but not Chinese to Tamil and English, Chinese and Tamil. The reason for poor classification rate is that the extracted features are not good enough to classify Chinese and Tamil due to common cursiveness property. However, from the experimental results, we can conclude that the proposed method is good for English and a cursive script, e.g. Chinese and other Indian scripts. Hence the proposed method can be used for classifying English from other scripts as a two class problem.

Table 5. Performance of English, Chinese and Tamil scripts

| Matrix | 50% images per class for testing | | | 90% images per class for testing | | |
|---|---|---|---|---|---|---|
| | E | Ch | Ta | E | Ch | Ta |
| English | **67.0** | 13.0 | 20.0 | **74.4** | 12.2 | 13.3 |
| Chinese | 10.6 | **60.0** | 29.3 | 8.1 | **60.7** | 31.1 |
| Tamil | 4.0 | 88.0 | **8.0** | 15.5 | 72.5 | **11.8** |
| Average | **45.0** | | | **48.9** | | |

## IV. Conclusion and Future Work

We have presented new features based on visual clues over an edge map of English, Chinese and Tamil video text line images for identification of those scripts in this work. The extracted features reflect the smoothness and cursiveness properties of the scripts. It is expected that Chinese and Tamil scripts are more cursive than English script. The smoothness and cursiveness are studied with the help of upper and lower extreme points of each edge component of each zone of the text line image. Experimental results show that the proposed method is good for classifying two languages but not three languages in the same video frame. We plan to extend the method to classify more scripts with better accuracy.

## Acknowledgment

This research is supported in part by A*STAR grant R252-000-402-305.

## References

[1] J. Gllavata and B. Freisleben, "Script Recognition in Images with Complex Backgrounds", In Proc. IEEE International Symposium on Signal Processing and Information Technology, 2005, pp 589-594.

[2] J. Zang and R. Kasturi, "Extraction of Text Objects in Video Documents: Recent Progress", In Proc. DAS 2008, pp 5-17

[3] K. Jung, K.I. Kim and A.K. Jain, "Text information extraction in images and video: a survey", Pattern Recognition 37, 2004, pp. 977-997.

[4] D. Doermann, J. Liang and H. Li, "Progress in Camera-Based Document Image Analysis", In Proc. ICDAR 2003, pp 606-616.

[5] P. Shivakumara, T, Q. Phan and C. L. Tan, "A Laplacian Approach to Multi-Oriented Text Detection in Video", IEEE Transactions on PAMI 2011, pp 412-419.

[6] C. Wolf and J. M Jolion, "Extraction and Recognition of artificial text in multimedia documents", Pattern Analysis and Applications 2003, pp 309-326.

[7] D. Chen and J. M. Odobez, “Video text recognition using sequential Monte Carlo and error voting methods”, Pattern Recognition Letters 26, 2005, pp 1386-1403.

[8] X. Tang, X. Gao, J. Liu and H. Zhang, “A Spatial-Temporal Approach for Video Caption Detection and Recognition”, IEEE Transactions on Neural Networks 2002, pp 961-971.

[9] S. H. Lee and J. H. Kim, “Complementary combination of holistic and component analysis for recognition of low resolution video character images”, Patten Recognition Letters 29, 2008, pp 383-391.

[10] D. Ghosh, T. Dube and A. P. Shivaprasad, “Script Recognition-Rview”, IEEE Ttansactios on PAMI 2010, pp 2142-2161.

[11] T .N. Tan, “Rotation Invariant Texture Features and Their Use in Automatic Script Identification”, IEEE Transactions on PAMI 1998, pp 751-756.

[12] A. Busch, W. W. Boles and S. Sridharan, “Texture for Script Identification”, IEEE Transactions on PAMI 2005, pp 1720-1732.

[13] L. Shijian and C. L. Tan, “Script and Language Identification in Noisy and Degraded Document Images”, IEEE Transaction on PAMI 2008, pp 14-24.

[14] S. Jaeger, H. Ma and D. Doermann, “Identifying Script on Word-Level with Informational Confedence”, In Proc. ICDAR 2005, pp 416-420.

[15] P. B. Pati and A. G. Ramakrishnan, “Word level multi-script identification”, Pattern Recognition Letters 2008, pp 1218-1229.

[16] S. Chanda, S. Pal, K. Franke and U. Pal, “Two-stage Apporach for Word-wise Script Identification”, In Proc. ICDAR 2009, pp 926-930.

[17] S. Chanda, O. R. Terrades and U. Pal, “SVM Based Scheme for Thai and English Script Identification”, In Proc. ICDAR 2007, pp 551-555

[18] L. Li and C. L. Tan, “Script Identification of Camera-based Images”, In Proc. ICPR 2008.

[19] A. M. Namboodiri and A. K. Jain, “On-line Script Recognition”, In Proc. ICPR 2002, pp 736-739.

[20] P.Shivakumara, Deepu Rajan and Suresh A. Sadanathan, “Classification of Images: Are Rule based Systems effective when Classes are fixed and known?”, In Proc. ICPR 2008.