Video Text Extraction Using The Fusion of Color Gradient and Log-Gabor Filter

Zhike Zhang, Weiqiang Wang, Ke Lu
University of Chinese Academy of Sciences
Beijing, China
zhangzhike11@mails.ucas.ac.cn, wqwang@ucas.ac.cn, luk@ucas.ac.cn

***Abstract*—Video text which contains rich semantic information can be utilized for video indexing and summarization. However, compared with scanned documents, text recogniton for video text is still a challenging problem due to complex background. Segmenting text line into single characters before text extraction can achieve higher recognition accuracy, since background of single character is less complex compared with whole text line. Therefore, we first perform character segmentation, which can accurately locate the character gap in the text line. More specifically, we get a fusion map which fuses the results of color gradient and log-gabor filter. Then, candidate segmentation points are obtained by vertical projection analysis of the fusion map. We get segmentation points by finding minimum projection value of candidate points in a limited range. Finally, we get the binary image of the single character image by applying K-means clustering and combine their results to form binary image of the whole text line. The binary image is further refined by inward filling and the fusion map. The experimental results on a large amount of data show that the proposed method can contribute to better binarization result which leads to a higher character recognition rate of OCR engine.**

I. Introduction

With the rapid growth of videos on the Internet, there is an urgent demand for video indexing and summarization. In addition to image features such as colors, shapes, textures and objects embedded in images like people, vehicle, text information also plays an important role in video retrieval, since text may contain rich semantic information and text retrieval is relatively a mature technology. Video text is generally classified into two categories: the overlay text(added artificially during the editing process) and the scene text(existing in the real-world scenes). This paper focuses on the former type. With artificial edit, the overlay text is usually closely related to the video content, so it can be reliably used for video indexing and retrieval. However, due to the numerous difficulties, *e.g.* complex background, low resolution, unknown text color and so on, overlay text extraction for videos is not that easy as text extraction from scanned documents which has been well solved. The whole video text recognition should generally contain four steps: detection, localization, extraction, and recognition [1]. The detection step roughly identifies text regions and non-text regions. The localization step determines the accurate boundaries of text rows. The text extraction step removes background pixels in the text rows and the text pixels are left for recognition. The recognition step converts the binarized pixel text into the encoded text, which can be executed by OCR software and we use google's open source OCR software, Tesseract [15]. We focus on the text extraction, which can also be called binarazition here.

The text extraction methods can be classified into four classes: the first class is threshold-based methods which include global threshold [2] and local threshold [3]; the second uses cluster-based methods which cluster all pixels into several classes based on color similarity and select text classes from them, such as k-means [4], [5]; the third extracts text using some stroke-based filters [6], [7], [8] and the last one is based on a special MRF model [9], [10], [11] which can be solved by min-cut/max-flow algorithms. Like some methods [4], [13], [14], we perform character segmentation before text extraction. Huang *et al.* [4] used color gradient to get an edge map and then applied adaptive thresholding, inward filling, morphological close operations, holes filling to get a binary image. Segmentation points are obtained based on vertical projection of the binary image. Saidane *et al.* [13] used convolutional neural networks (CNN) to get segmentation results. Trung *et al.* [14] produced curved segmentation line by finding the minimum cost path based on gradient vector flow. Our approach is similar to Huang. However, our vertical projection analysis is based on the fusion map which fuses the results of color gradient and log-gabor filter and we get segmentation points by finding minimum projection value in a limited range instead of thresholding. Without a series of operations and thresholding, our approach is more simple and is able to avoid the selection of threshold. After character segmentation, we use K-means clustering to extract text pixels for single character, *i.e.* clustering pixels into several classes and selecting one as text class based on the fusion map. The binary image of the whole text line is combining the text extraction results of all characters and is refined by inward filling and the fusion map.

The rest of this paper is organized as follows. Section 2 presents the proposed method. The experimental results are reported in Section 3. Section 4 concludes the paper.

II. Proposed Method

The overview of the proposed approach is shown in Figure 1. For a given text line image, we compute color gradient and apply log-gabor filter to get a fusion map by fusing the results of color gradient map and log-gabor filter map. Then vertical projection analysis of the fusion map is performed. We treat local minimum points of vertical projection value as candidate segmentation points and get final segmentation points by finding minimum projection value points from candidates in a limited range. Subsequently we perform text extraction for single character using K-means clustering. The binary image of the whole text line is obtained by combining the text extraction

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.506

IEEE computer society

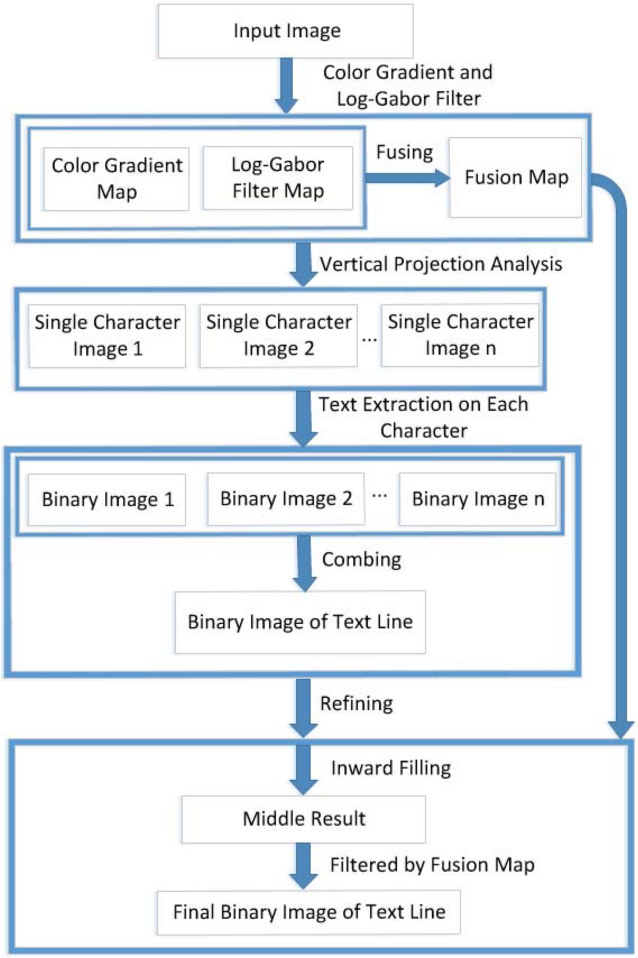


Fig. 1. The overview of the proposed approach.

of all characters. The result is refined by inward filling and fusion map.

A. The Fusion of Color Gradient and Log-Gabor Filter

1) Color Gradient Map: We use F to represnet a color image in the RGB space like:

$$F(x,y) = \left[\begin{array}{c} R(x,y) \\ G(x,y) \\ B(x,y) \end{array} \right], \tag{1}$$

and define g_{xx}, g_{yy}, g_{xy} as follows:

$$g_{xx} = (\frac{\partial R}{\partial x})^2 + (\frac{\partial G}{\partial x})^2 + (\frac{\partial B}{\partial x})^2, \tag{2}$$

$$g_{yy} = (\frac{\partial R}{\partial y})^2 + (\frac{\partial G}{\partial y})^2 + (\frac{\partial B}{\partial y})^2, \tag{3}$$

$$g_{xy} = \frac{\partial R}{\partial x}\frac{\partial R}{\partial y} + \frac{\partial G}{\partial x}\frac{\partial G}{\partial y} + \frac{\partial B}{\partial x}\frac{\partial B}{\partial y}. \tag{4}$$

With these symbols, the gradient orientation $\theta(x,y)$ and the gradient amplitude $f_\theta(x,y)$ can be calculated by the method of Di Zenzo [Di Zenzo, 1986]:

$$\theta(x,y) = \frac{1}{2}\arctan(\frac{2g_{xy}}{g_{xx} - g_{yy}}), \tag{5}$$

$$f_\theta(x,y) = \sqrt{\frac{1}{2}G}, \tag{6}$$

and G is given by:

$$G = (g_{xx} + g_{yy}) + (g_{xx} - g_{yy})\cos 2\theta + 2g_{xy}\sin 2\theta. \tag{7}$$

2) Log-Gabor Filter Map: Log-gabor filters proposed by Field [Field, 1987] can catch spatial information as well as frequency information in certain direction and circumvent some limitation that gabor filters suffer from. Log-gabor filters in frequency domain can be defined in polar coordinates by $H(f,\theta) = H_f \times H_\theta$ where H_f is the radial component and H_θ, the angular one:

$$H(f,\theta) = exp\{\frac{-[\ln(f/f_0)]^2}{2[\ln(\sigma_f)]^2}\} \times exp\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^2}\} \tag{8}$$

Here f_0, the central frequency, θ_0, the filter direction, σ_f which defines the radial bandwidth B in octaves with $B = 2\sqrt{2/\ln 2} * |ln(\sigma_f)|$ and σ_θ, which defines the angular bandwidth $\Delta\Omega = 2\sigma_\theta\sqrt{2\ln 2}$. We use two directions for the filter, the horizontal and the vertical one, *i.e.* $\theta_0 = \{0, \pi/2\}$ and get filter map M_g by fusing the results of both directions:

$$M_g(x,y) = \sqrt{M_g^h(x,y)^2 + M_g^v(x,y)^2}, \tag{9}$$

where $M_g(x,y)$ denotes log-gabor filter map value at pixel (x,y). For filter of each direction, we use a fixed angular bandwidth of $\Delta\Omega = \pi/2$. f_0 should be the reciprocal of the stroke width w_s, *i.e.* $f_0 = 1/w_s$, and σ_f is set to 0.65. The stroke width w_s can be estimated like [7]:

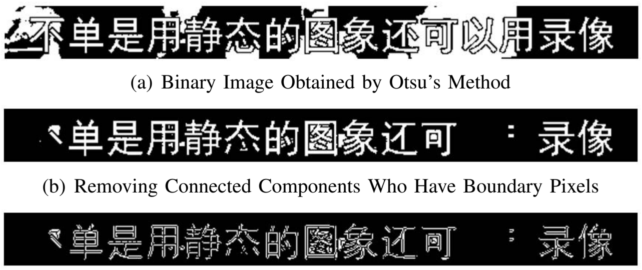
$$w_s = \frac{|B_f|}{|S|}. \tag{10}$$

Here B_f denotes foreground of binary image, S denotes skeleton of binary image and $|\cdot|$ is the size of the set. Here the binary image is obtained through method shown in Figure 2. We first get initial binary image using Otsu's method. To suppress the effect of noise, then we remove these connected components that have boundary pixels. In practice, we use the following Equation instead:

$$w_s = \alpha\frac{|B_f|}{|S|} + \beta, \tag{11}$$

since the binary image and its skeleton are approximatively calculated, the result obtained by Equation 10 may not be very accurate. Here α, β are set experimentally.

Some examples of Log-Gabor filter map can be seen in Figure 3. Here the examples include images with different stroke width. We can see that the filter map can capture the basic shape of the text and usually has large value at text pixels.

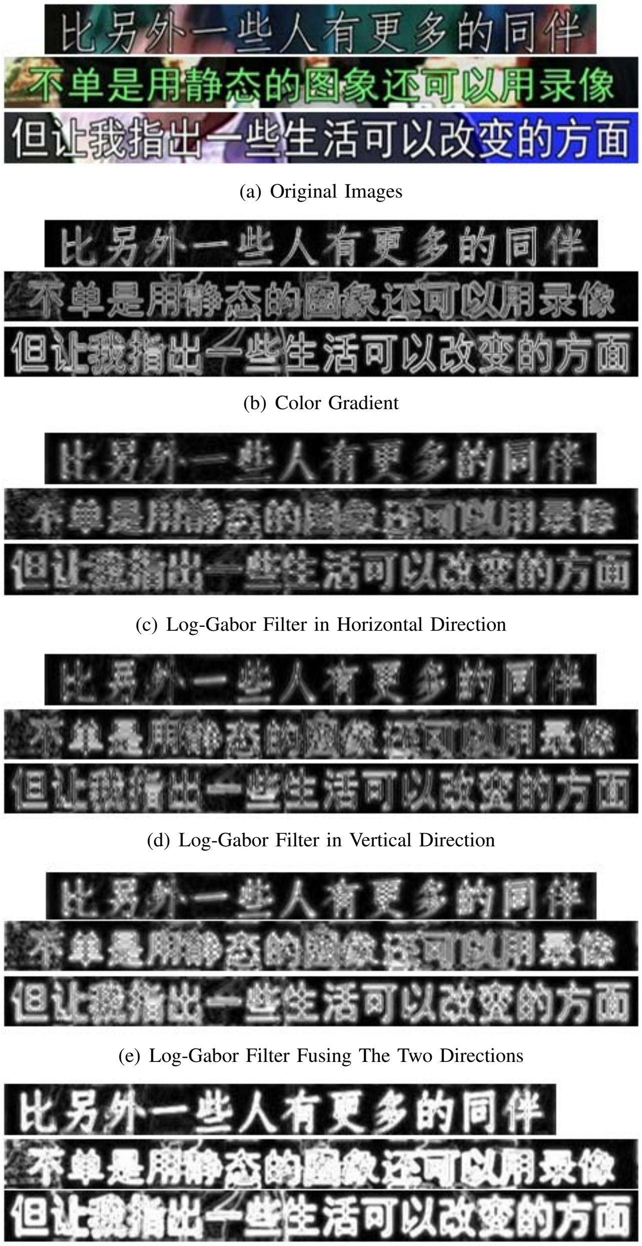


(a) Binary Image Obtained by Otsu's Method

(b) Removing Connected Components Who Have Boundary Pixels

(c) Binary Image and Its Skeleton Used for Estimating Stroke Width

Fig. 2. The method to get binary image and its skeleton used for estimating stroke width.


比另外一些人有更多的同伴
不单是用静态的图象还可以用录像
但让我指出一些生活可以改变的方面

(a) Original Images

(b) Color Gradient

(c) Log-Gabor Filter in Horizontal Direction

(d) Log-Gabor Filter in Vertical Direction

(e) Log-Gabor Filter Fusing The Two Directions

(f) Fusion Map Fusing Color Gradient and Log-Gabor Filter

Fig. 3. Some examples of color gradient and log-gabor filter.

3) Fusion Map: Intuitively, the color gradient captures edge information and the log-gabor filter map reflects stroke information. The fusion map $M(x, y)$ is simply got by:

$$M(x, y) = f_\theta(x, y) + M_g(x, y), \tag{12}$$

and the value of $M(x, y)$ is normalized to range [0,1] by setting the value greater than 1 to 1. Figure 3 also shows the results of the fusion map.

B. Character Segmentation

Assuming the size of the input image is $w \times h$ where w and h denote width and height respectively. We have got the fusion map M in previous phase, so the vertical projection of M can be obtained by:

$$P(i) = \sum_{j=1}^{h} M(i, j), \quad i \in \{1, ..., w\}. \tag{13}$$

We treat local minima of vertical projection as candidate segmentation point set S_c which can be defined as:

$$S_c = \{\ i \mid P(i) < P(i-1) \ and \ P(i) < P(i+1)\ \}. \tag{14}$$

The segmentation point set S_s includes these points whose projection values are minimum in a limited range:

$$S_s = \{\ i \in S_c \mid P(i) = \min_{k=l(i)}^{r(i)} P(k)\ \}, \tag{15}$$

where $l(i)$ and $r(i)$ are defined as:

$$l(i) = \max\{1, i - \frac{h}{2}\ \},\ r(i) = \min\{w, i + \frac{h}{2}\ \}. \tag{16}$$

After getting segmentation points, we estimate character width as the median value of adjacent segmentation point distances. We refine segmentation points by adding or removing segmentation point if the adjacent segmentation point distance is too big or too small comparing with the estimated character width. The examples are shown in Figure 4.

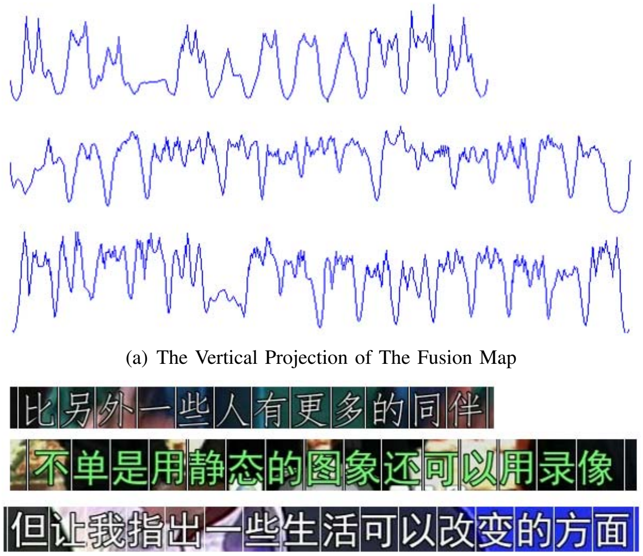
C. Text Extraction

In this phase, for each single character image segmented from text line image, we apply k-means clustering in the RGB space to cluster the pixels into k (k=4) classes and thus can get a binary image if we treat one class as foreground and the others as background. We choose text class based on following formula:

$$Score(i) = \frac{\sum_{(x,y) \in C_i} M(x, y)}{|C_i|}, \tag{17}$$

$$Score^{'}(i) = \alpha Score(i) + \beta(1 - \frac{|Skel(C_i)|}{|C_i|}), \tag{18}$$

Here i is the index of classes. C_i denotes the set of pixels in class i. $|\cdot|$ stands for the size of the set. α and β are parameters which are set to 0.7 and 0.3. $Skel(C_i)$ denotes the skeleton of the binary image corresponding to class i. We assume that the text class pixel has higher value of fusion map. However, the pixel at the text edge should also have higher value according to the definition of fusion map. Thus, we first select two classes with the two highest score values according to formula 17.



(a) The Vertical Projection of The Fusion Map

(b) The Segmentation Results

Fig. 4. The examples of character segmentation.

Then, we apply formula 18 to these two classes and select the one with higher value as text class since stroke width should be thicker than text edge. As shown in the figure 5, both of class c and d should have higher score values according to formula 17, but the score value of class c can be suppressed by formula 18. The binary image of whole text line is obtained by combining binary images of all characters. We use dam point labeling and inward filling [1] to remove boundary noise and apply fusion map to refine binary image by removing those connected components whose pixels have small average map value(Formula 19 and 20). Compared with imclearborder which removes all connected components that are connected to the image border, dam point labeling and inward filling can preserve text pixels when removing boundary noise. In Formula 19 and 20, L refers to connected component of B_f, B_f denotes foreground of binary image, $B_f^{'}$ is foreground of binary image after removing connected components with small response value, $M(x,y)$ is the fusion map value at pixel (x,y) like before. The final results of the extraction for examples are shown in figure 6.

$$B_f^{'} = \bigcup_{R(L)>T} L \tag{19}$$

$$R(L) = \frac{\sum_{(x,y)\in L} M(x,y)}{|L|} \tag{20}$$

III. Experiments

Since there is no standard dataset for overlay text in videos, we have used our tools to extract a large number of text lines from videos. Our tools can generate binary images of pixel level groundtruth simultaneously when extracting text lines. To be more persuasive, we collect data from a variety of sources including television series, movies, cartoon films and lectures

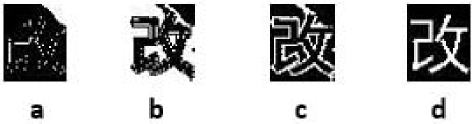


Fig. 5. From left to right: background class, background class, text edge class, text class.

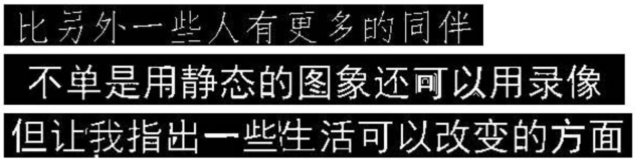

比另外一些人有更多的同伴
不单是用静态的图象还可以用录像
但让我指出一些生活可以改变的方面

Fig. 6. The results of text extraction for examples.

with different color, font and size, a total of 9549 images including 87282 characters. Some example images we collect for experiments and their groundtruth images generated by our tools are shown in Figure 7.

To evaluate the performance of the proposed algorithm, we show the results of character segmentation accuracy, recognition accuracy of OCR engine and pixel level accuracy of binary image.

A. Segmentation Accuracy

We randomly select 662 images from our data for evaluating segmentation accuracy. The total groundtruth gaps between characters is 5333. We use recall(R), Precision(P) and F-measure(F) as performance measures, and make following definitions:

Actual Gap(AG): groundtruth gaps

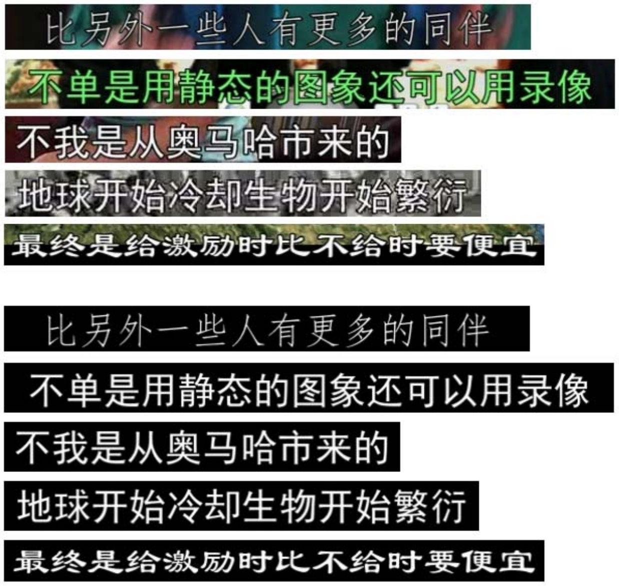


Fig. 7. Some example images and their groundtruth images.

TABLE I. SEGMENTATION ACCURACY

Method	Precision	Recall	F-score
Huang's method [4]	0.9045	0.9281	0.9161
Our method	0.9771	0.9632	0.9701

TABLE II. OCR ACCURACY WITH DIFFERENT TEXT EXTRACTION METHODS(IN %)

Method	CRR	IRR
Otsu [2]	75.90	29.85
Niblack [3]	74.90	28.10
Lyu's method [1]	80.56	31.24
proposed	90.30	47.80

True Gap (TG): segmentation line goes through the gap
False Gap (FG): segmentation line goes through the character all these values are counted manually and the performance measures are calculated as follows:

- $R = TG/AG$
- $P = TG/(TG + FG)$
- $F = 2 \times P \times R/(P + R)$

We compare our character segmentation method with Huang's method and the result is shown in table I. Obviously, our method outperforms Huang's method.

B. OCR Accuracy

OCR accuracy can reflect the performance of text extraction methods, since better text extraction methods contribute to higher recognition accuracy of an OCR software. So we test OCR accuracy to verify performance of our algorithm. For this we feed the binarization results of all methods to Tesseract [15], Google's OCR engine. The performance measure is the character recognition rate (CRR) that the proportion of correct recognition characters to total groundtruth characters and the image recognition rate (IRR) that the proportion of correct recognition text line images to total groundtruth images. Here correct recognition text line image means all characters in text line are correctly recognized. As shown in table II, compared to the threshold-based method, proposed approach has significant improvement in OCR accuracy. Our method also outperforms Lyu's method.

C. Pixel Level Accuracy

We also compare various algorithms based on pixel accuracy. The results of well-known measures like precision, recall, f-score are shown in table III. As we can see in the table, Otsu's method can extract the majority of text pixels, which leads to high recall, more than 0.90. However, The recall of Niblack's method is lower. It is reasonable because methods based on local threshold tend to improve precision but lose some recall at the same time. The proposed method has lower recall but precision of our method is far higher than that of threshold-based methods. The lower recall of our method compared with threshold-based methods is because that our method clusters the pixels into 4 class and selects one as text class while threshold-based methods just classify the pixels into two classes. Another reason resulting in lower recall of our method is that some text connected components may be filtered due to their low fusion map value. But the high OCR accuracy of proposed method demonstrates that despite missing some foreground pixels relative to groundtruth, the characters in binary images of our method are still recognizable and it more lies in that characters are thinner than those of groundtruth.

TABLE III. PIXEL LEVEL ACCURACY WITH DIFFERENT TEXT EXTRACTION METHODS

Method	Precision	Recall	F-score
Otsu [2]	0.73	0.91	0.81
Niblack [3]	0.82	0.83	0.83
Lyu's method [1]	0.85	0.89	0.87
proposed	0.96	0.75	0.84

IV. CONCLUSION

We propose a new method for video text extraction based on character segmentation using the fusion of color gradient and log-gabor filter. Given a text line image extracted from video, we first generate a map fusing the results of color gradient and log-gabor filter. Subsequently, we segment the text line into single characters based on vertical projection analysis of the fusion map. We treat local minimum points of vertical projection value as candidate segmentation points and get final segmentation points by finding minimum projection value points from candidates in a limited range. Finally, we perform text extraction using k-means clustering in the RGB space for single character and combine the results of all characters to form text extraction of whole text line. The binary image of text line is refined by inward filling and fusion map. Maybe we will extend our work to scene text in the future.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61271434, No. 61232013, No. 61175115.

REFERENCES

[1] Michael R. Lyu, Jiqiang Song, Min Cai, *"A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction,"* IEEE Trans. on Circuits and Systems for Video Technology. 15(2):243-255, 2005.

[2] N. Otsu, *"A threshold selection method from gray level histogram,"* IEEE Transactions on System, Man, Cybernetics, vol. 19, no. 1, pp. 62-66, 1978.

[3] W. Niblack, *An Introduction to Digital Image Processing,* Englewood Cliffs, New Jersey: Prentice-Hall, 1986.

[4] X. Huang, H. Ma, and H. Zhang, *"A New Video Text Extraction Approach,"* in Proc. ICME, 2009, pp. 650-653.

[5] K. Kita and T. Wakahara, *"Binarization of color characters in scene images using k-means clustering and support vector machines,"* in Proc. ICPR, 2010, pp. 3183-3186.

[6] D. Chen, K. Shearer and H. Bourlard,*"Text Enhancement with Asymmetric Filter for Video OCR,"* ICIAP, 2001, pp. 192-197.

[7] Céline Mancas-Thillou and Bernard Gosselin,*"Character Segmentation-by-Recognition Using Log-Gabor Filters,"* in Proc. ICPR, 2006, pp. 901-904.

[8] Xiaodong Huang,*"A novel video text extraction approach based on Log-Gabor filters,"* International Congress on Image and Signal Processing, 2011, pp. 474-478.

[9] C. Wolf and D. S. Doermann, *"Binarization of low quality text using a markov random field model ,"* In Proc. ICPR 2002, pp. 160-163.

[10] X. Peng, S. Setlur, V. Govindaraju and R. Sitaram, *"Markov random field based binarization for hand-held devices captured document images ,"* in ICVGIP, 2010.

[11] A. Mishra, K. Alahari and C. V. Jawahar, *"An MRF Model for Binarization of Natural Scene Text ,"* In Proc. ICDAR 2011, pp. 11-16.

[12] Z. Saidane and C. Garcia, *"Robust Binarization for Video Text Recognition,"* In Proc. ICDAR 2007, pp. 874-879.

[13] Z. Saidane and C. Garcia, *"An automatic method for video character segmentation,"* In Proc. International Conference on Image Analysis and Recognition, 2008, pp. 557-566.

[14] Trung Quy Phan, Shivakumara P., Bolan Su, Tan C.L., *"A Gradient Vector Flow-Based Method for Video Character Segmentation,"* In Proc. ICDAR 2011, pp. 1024-1028.

[15] OCR Engine used:http://code.google.com/p/tesseract-ocr/.