A NEW APPROACH FOR VIDEO TEXT DETECTION

Min Cai, Jiqiang Song, and Michael R. Lyu

Department of Computer Science & Engineering
The Chinese University of Hong Kong
Hong Kong SAR, China

ABSTRACT

Text detection is fundamental to video information retrieval and indexing. Existing methods cannot handle well those texts with different contrast or embedded in a complex background. To handle these difficulties, this paper proposes an efficient text detection approach, which is based on invariant features, such as edge strength, edge density, and horizontal distribution. First, it applies edge detection and uses a low threshold to filter out definitely non-text edges. Then, a local threshold is selected to both keep low-contrast text and simplify complex background of high-contrast text. Next, two text-area enhancement operators are proposed to highlight those areas with either high edge strength or high edge density. Finally, coarse-to-fine detection locates text regions efficiently. Experimental results show that this approach is robust for contrast, font-size, font-color, language, and background complexity.

1. INTRODUCTION

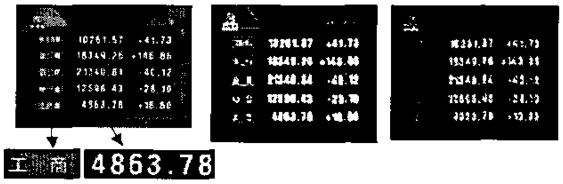
Efficient indexing and retrieval of digital video is an important function of video database. One powerful index for retrieval is the text appearing in videos. There are two kinds of text in video: scene text and artificial text. Artificial text is usually a carrier of important information and its appearance is carefully designed, whereas scene text is not significant to video indexing since it appears accidentally and is seldom intended. Therefore, we want to locate and extract the artificial text in video, and feed them to OCR engine. This procedure is generally called Video OCR. Nowadays commercial OCR engines cannot yet detect and recognize text embedded in complex background directly, e.g. a video image. Consequently, preprocessing, i.e., detecting accurate bounding box of text and simplifying the background, is very important.

Text detection in real-life videos and images is still an open problem. Existing text detection methods can be classified into two main categories. One category is connected-component-based method [1,2], and another is texture-analysis-based method [3-7]. Methods in the former category detect text regions by analyzing the spatial distribution of edges or homogeneous color/grayscale components that are segmented as text. For example, Zhong *et al* [1] extracted text as those connected components of monotonous color that follow certain size constraints and horizontal alignment constraints. Texture-analysis-based methods could be further divided into top-down approaches and bottom-up approaches. Classic top-down approach is based on splitting image regions alternately in horizontal and vertical direction based on texture, color or edge [3-5]. On the contrary, the bottom-up approach intends to find homogeneous regions from some seed regions. The region growing technique is applied to merge pixels belonging to the same cluster [6,7]. Existing methods do solve the problem to a certain extent, however, not perfectly. The difficulty comes from the variation of font-size, font-color, spacing, contrast and language of text, and mostly, the background complexity.

In this paper, we propose a new efficient video-text-detection approach that is capable of detecting text in a complex background, and is robust for font-size, font-color, and language. The rest of this paper is organized as follows. Section 2 analyzes the difficulties of existing methods and summarizes the features of video text that we utilize in detection. Section 3 describes the new approach. The experiment results are shown in Section 4. Section 5 draws our conclusions.

2. ANALYSIS

Although text detection has been studied for a long time, existing methods still have many difficulties when they are applied to real-life videos. First, when text is embedded in a complex background, the contrast of text, i.e., the difference between the color (or luminance) of text and its local background, varies in different areas of the image. Therefore, those methods that use a global threshold to separate text and background will miss low-contrast texts. For example, Figure (1-a) shows an image with different contrast texts. After applying Sobel operator, if threshold is 40 (Fig. 1-b), all texts are kept. However, if threshold is 65 as suggested in [8], the low-contrast texts disappear (Fig. 1-c). Second, the color of text is not uniform due to color bleeding and noise introduced by the lossy video compression (Fig. 2), which causes the failure of those methods that assume font color of a text string is uniform. Third,



a. Original frame b. Filtered by 40 c. Filtered by 65

Figure 1. Different contrast of texts in one frame

0-7803-7622-6/02/$17.00 ©2002 IEEE

Figure 2. Color difference within a character

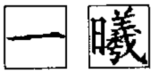


Figure 3. Various stroke density of Chinese characters

characters of different language have various stroke structures. The methods using stroke-density constraints may detect western language text, say English, successfully. However, they will fail to detect Asian language text correctly, say Chinese. For example, region growing methods use a small-size window (16×16 in [4], 3×3 in [6]) to scan the image and classify each window as text or non-text, and then merge adjacent text blocks to form text regions. Since every Chinese character occupies the same space, whereas the stroke number varies from 1 to more than 20 (Fig. 3), characters with few strokes will not be classified as text by stroke-density constraints. Fourth, methods utilizing domain-specific layout constraints, e.g. [4] assuming that text only appears in the center of the screen, limit their applications to different types of images. Finally, multi-resolution methods [6] are popular in handling the large range of font-size; however, they are too time-consuming to process a large number of video images.

To overcome above difficulties, we seek for the features of video text that are independent of contrast, font-color, font-size, and language.

1) Although the pixel color of a text string is not uniform, a recognizable text string in video does have dense sharp transitions of color or luminance, i.e. edges, against its local background. Therefore, edge is a more reliable feature than font-color. Edge has two properties: edge strength and edge density. When embedded in a simple and high-contrast background, a text string is noticeable by both edge strength and edge density. When embedded in a simple and low-contrast background, it is noticeable by mainly edge density. When embedded in a complex background full of edges of non-text objects, it is noticeable by mainly edge strength. Since a text string is improbable made of all characters with few strokes, its average edge density is significantly higher than that of background.

2) Characters are mostly upright and appear in clusters at a limited distance aligned to a horizontal line, and they show spatial cohesion -- characters of the same text string are of similar heights, orientation and spacing.

Based on edge strength, edge density and horizontal distribution, we design an efficient approach to detect multilingual text in complicated background.

3. ALGORITHMS

The goal of our approach is to detect both low-contrast texts and high-contrast texts without being affected by language and font-size. First, it converts the video image into edge map using a color edge detector [9] and uses a low global threshold to filter out definitely non-edge points. Then, a selective local thresholding is performed to simplify the complex background. Next, We design an edge-strength-smoothing operator and an edge-clustering-power operator to highlight those areas with high edge strength or edge density, i.e. text candidates. Finally, considering feature (2) is invariant with language and font-size, we employ a string-oriented coarse-to-fine detection method to locate text strings quickly.

3.1. Edge detection

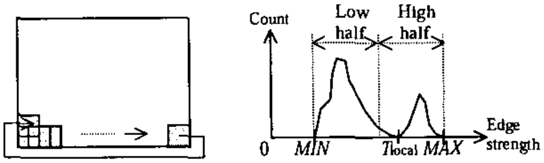
The color edge detector uses Sobel operator to detect edges in YUV color space. The final color edge map is the union of three edge maps of Y, U, and V channels. However, we do not use the fast entropic thresholding in [9] since the threshold is too high to keep low-contrast texts. Instead, we apply a low threshold determined by the histogram of edge strength to eliminate only definitely non-text points. First, locate the peak of ranks 0-20 in the histogram and get the average height around it; then, the low threshold is the first position after the peak whose height falls below 10% of the average height. After the global thresholding, the value of non-edge points is zero, while that of edge points is their individual edge strength.

3.2. Selective local thresholding

If the background is simple, a text string, even of low contrast, can be easily detected by a low threshold, whereas a text string embedded in a complex background needs a higher threshold to further simplify the background. Therefore, it is necessary to determine a proper threshold for each local area according to its background complexity.

We define a window of size *H*×*W* (*H* and *W* are proportional to image height and width respectively). The window scans the edge map step by step first in horizontal direction then in vertical direction (Fig. 4). In each step, the origin of window only moves *W*/2 in horizontal direction (or *H*/2 in vertical direction) so that the inaccuracy caused by splitting a character with window border can be compensated. The part of edge map covered by the window is the local area to be analyzed.

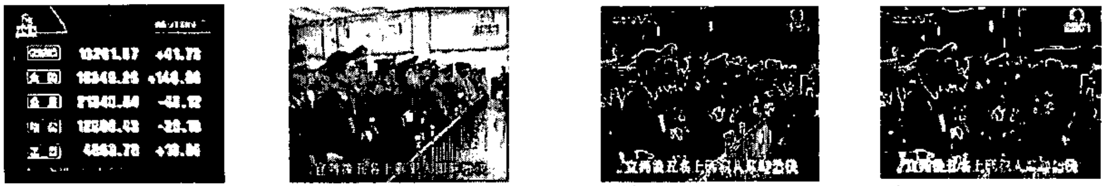
The background complexity is defined as follows. If a pixel is non-edge point, we call it *blank* point. The background complexity is *simple* on condition that the number of totally blank rows in this local area is not less than 10%×*H*. Otherwise, it is *complex*. A simple area does not need thresholding any more, while a complex area needs a much higher threshold. The new threshold is found from the local histogram of this area. Let *MAX* and *MIN* be the highest edge strength and the lowest edge strength respectively. We find the low peak at the low half of [*MIN*, *MAX*] and the high peak at the high half of that, and then determine the new threshold (T_{local}) as the lowest position between the low peak and the high peak. The edge points in this area whose strengths are lower than T_{local} are marked with a flag. After the entire edge map has been scanned, all edge points with the flag are removed. Applying selective local threshold, the low-contrast texts in simple background is kept, and meanwhile, the background of high-contrast texts is simplified. Figure (5-a) shows the thresholding result of Fig. (1-a). Both low-contrast texts and high-contrast texts are kept, since they are in a simple background. Figure (5-b) is an image with complex background. After global thresholding (Fig 5-c), the barrier around the high-contrast text remains, but the low-contrast logo of TV station



a. Window move

b. Local threshold selection

Figure 4. Local thresholding



a. Result of Fig. 1-a image b. A complex image c. Global thresholding d. Selective local thresholding

Figure 5. Effect of selective local thresholding

(top-right corner) is corrupted. However, using selective local threshold (Fig 5-d), the barrier is eliminated and the logo is kept.

3.3. Text area enhancement

Selective local thresholding picks out edge points that are noticeable in local background. However, only the edge strength feature is utilized. To further highlight the text area by its edge density feature, we design two operators, i.e. edge-strength-smoothing (ESS) operator and edge-clustering-power (ECP) operator, whose convolution kernels are shown in Figure (6). The weights in both kernels are derived from Euclidean distance. Since we use integer to speed up convolution, the convolution result is divided by the sum of weights (220 for ESS, and 100 for ECP). ESS weight is reversely proportional to the square of distance outwards from center. It reflects the average edge strength around the center edge point. Since local thresholding may decrease the edge density in text area, we first perform ESS operator on every point in the edge map, denoted *EM(x,y)*, using Equation (1) to increase the edge density.

$$\frac{1}{220}\times\begin{array}{|c|c|c|c|c|}\hline 2&4&5&4&2\\\hline 4&10&20&10&4\\\hline 5&20&40&20&5\\\hline 4&10&20&10&4\\\hline 2&4&5&4&2\\\hline\end{array}\qquad\qquad\frac{1}{100}\times\begin{array}{|c|c|c|c|c|}\hline 8&5&4&5&8\\\hline 5&2&1&2&5\\\hline 4&1&0&1&4\\\hline 5&2&1&2&5\\\hline 8&5&4&5&8\\\hline\end{array}$$

a. ESS kernel b. ECP kernel

Figure 6. Text area enhancement operators

$$ESS(x,y)=\frac{1}{220}\times\sum_{-2\le i\le 2;-2\le j\le 2}EM(x+i,y+j)\times ESS\_weight(i,j) \quad (1)$$

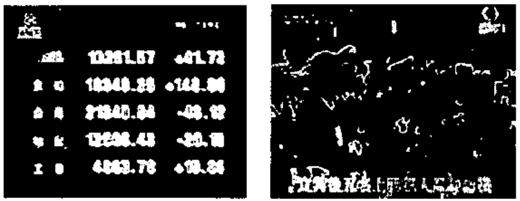
ESS map is a smoothed edge strength map. Then, we enhance the high edge-density areas by performing ECP operator on only non-zero points in ESS map, as shown in Equation (2).

$$ECP(x,y)=\frac{\frac{1}{100}\times\sum_{-2\le i\le 2;-2\le j\le 2}ESS(x+i,y+j)\times ECP\_weight(i,j)}{ESS(x,y)} \quad (2)$$

ECP weight is proportional to the square of distance outwards from center and the convolution result is divided by its own ESS value. ECP only reflects the edge density around the center point since the edge point is highlighted, whatever its ESS value is, if it has many neighboring edge points with higher or similar edge strengths. Finally, we integrate ESS value and ECP value by Equation (3) and update the edge map.

$$EM(x,y)=\alpha\times ESS(x,y)+(1-\alpha)\times ECP(x,y) \quad (3)$$

Usually, α is 0.5 to treat ESS and ECP equally. Now, the edge map highlights those areas with either high edge strength or high edge density. After binarized with a moderate threshold, e.g. 0.6, the edge map is ready for text detection. Figure (7) shows the results of text area enhancement of Figure (1-a) image and Figure (5-b) image respectively. In Figure (7-a), both low-contrast texts and high-contrast texts are highlighted due to their high edge density. In Figure (7-b), those isolated and low-contrast edge points are suppressed.



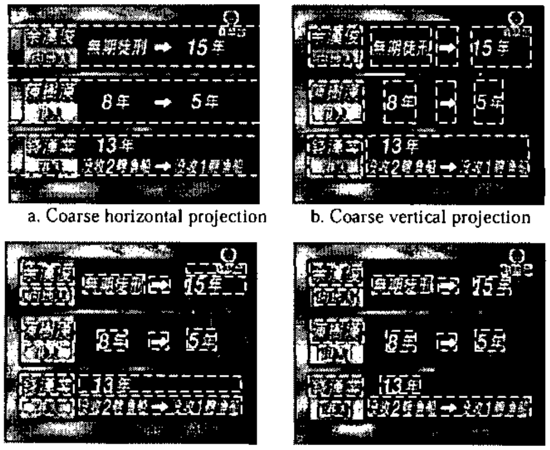
a. Result of Fig. 1-a image b. Result of Fig. 5-b image

Figure 7. Effect of text area enhancement

3.4. Coarse-to-fine detection

It is clear that the horizontal rectangular areas with high density indicate text strings. Projection is a more efficient way to find such high-density areas than other methods, say region growing methods, since text strings always produce sharp jumps in horizontal projection. However, in video images, text strings will not appear line by line. They are often overlapping in horizontal or vertical dimension, especially in financial reports. Since simple projection only reflects the distribution at one dimension, it cannot handle well such cases. Thus, we design a coarse-to-fine detection scheme to solve this problem.

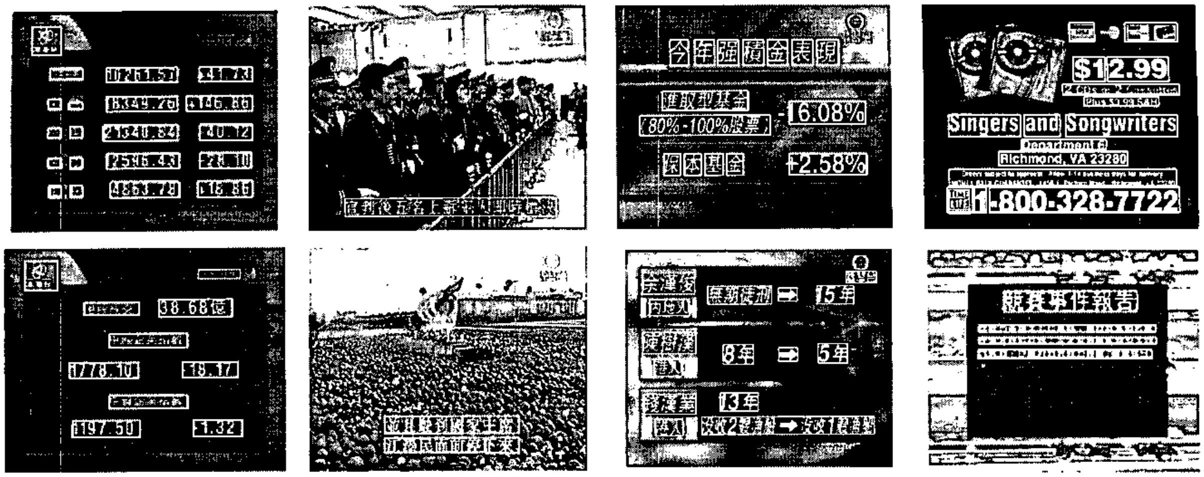
The idea of coarse-to-fine detection is to locate the text region progressively by two-phase projection (refer to Fig. 8). In the first phase, it segments the edge map roughly into text blocks using *coarse horizontal projection* and *coarse vertical projection*. Then, in the second phase, it locates the text-like region accurately by *fine horizontal projection* and *fine vertical*



a. Coarse horizontal projection b. Coarse vertical projection

c. Fine horizontal projection d. Fine vertical projection

Figure 8. Coarse-to-fine projection



a. Different contrast b. Complex background c. Overlapping horizontally/vertically d. Different font-size

Figure 9. Experiment results on real life video images

projection. Note that only the coarse horizontal projection is a global projection, others are all local projection in a rectangular area Finally, it checks the text-like regions by the rules of average density, peak distribution in the fine vertical projection, and density distribution to eliminate non-text regions. Since the preprocessing has highlighted the text-like areas significantly, text strings are located easily and quickly.

4. EXPERIMENTAL RESULTS

The proposed approach has been tested on real-life videos. Chinese videos are captured from the TVB news programs aired by the Hong Kong Jade station. Video resolution is 288×352 (PAL). English videos are captured from CNN news. The resolution is 240×352 (NTSC). Totally 150 minutes programs are used in the experiments. There are total 14,685 text strings in sampled frames. The overall detection rate is 93.6%. The false-alarm rate is 6.5%, and the missing rate is 1.8%.

Figure (9) shows the experimental results of four kinds of images selected from news, financial reports and advertisements. In Figure (9-a), Both low-contrast texts and high-contrast texts are successfully detected. The texts embedded in the complicated background (Fig. 9-b) are correctly located. Figure (9-c) shows that coarse-to-fine detection handles well the complex text-string distribution. Figure (9-d) demonstrates that this approach is robust to font-size and language.

5. CONCLUSIONS

This paper proposes an efficient approach to handle existing difficulties in video text detection, e.g., different contrast, font-size or complex background. It depends on invariant features, i.e., edge strength, edge density and horizontal distribution. To avoid the weakness of global thresholding, a selective local-thresholding technique is proposed to both keep low-contrast text and simplify complex background of high-contrast text. ESS operator and ECP operator are introduced to highlight those areas with either high edge strength or high edge density. Finally, the efficient coarse-to-fine detection locates text regions correctly and quickly. Experimental results that this approach is robust for contrast, font-size, font-color, language, and background complexity.

5. ACKNOWLEDGEMENT

The work described in this paper was fully supported by two grants from the Hong Kong Special Administrative Region: the Hong Kong Research Grants Council under Project No. CUHK4222/01E, and Innovation and Technology Fund, under Project No. ITS/29/00.

6. REFERENCES

[1] Y. Zhong, K. Karu, and A.K. Jain. "Locating text in complex color images", *Pattern Recognition*, 28(10): 1523-1535, 1995.

[2] A.K. Jain, and B. Yu, "Automatic text location in images and video frames", *Pattern Recognition*, 31(12): 2055-2076, 1998.

[3] H. Li, D. Doermann, and O. Kia, "Automatic Text Detection and Tracking in digital Video", *IEEE Trans. Image Processing*, 9(1): 147-156, 2000.

[4] X. Gao, and X. Tang, "Automatic News Video Caption Extraction and Recognition", In *Proc. of the 2nd Intl. Conf. IDEAL*, pp.425-430, 2000.

[5] T. Sato, T. Kanade, E. Hughes and M. Smith, "Video OCR for Digital News Archives", In *Proc. Of IEEE Intl. Workshop on Content-based Access of Image and Video Databases*, pp.52-60, Jan, 1998.

[6] V. Wu, R. Manmatha, and E.M. Riseman, "Finding Text In Images", in *Proc. Of the 2nd Intl. Conf. on Digital Libraries*. Philadaphia. PA. pp.1-10, July 1997.

[7] K. Sobottka, H. Bunke, and H. Kronenberg, "Identification of Text on Colored Book and Journal Covers", In *Proc.of the 5th Intl. Conf. Document Analysis and Recognition*, pp.57-62, 1999.

[8] X.S. Hua, X.R. Chen, W.Y. Liu and H.J. Zhong, "Automatic location of text in video frames", In *Proc. of the 3rd Intl Workshop on Multimedia Information Retrieval (MIR'01)*, Ottawa, Canada, October, 2001.

[9] J. Fan, D.K.Y. Yau, A.K. Elmagarmid, and W.G. Aref, "Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing", *IEEE Trans. on Image Processing*, 10(10): 1454-1466, 2001.