Robust News Video Text Detection Based on Edges and Line-deletion

SHWU-HUEY YEN[1,a], HSIAO-WEI CHANG[1,2,b], CHIA-JEN WANG[1,c], CHUN-WEI WANG[1,d]
Department of Computer Science and Information Engineering
Tamkang University[1]
151 Ying-Chuan Road, Tamsui, Taipei County 25137, Taiwan
REPUBLIC OF CHINA
Department of Computer Science and Information Engineering
China University of Science and Technology[2]
245, Sec. 3, Academia Road, Taipei City 11581, Taiwan
REPUBLIC OF CHINA
105390@mail.tku.edu.tw[a] changhw@cc.cust.edu.tw[b]
chiajen.wang@msa.hinet.net[c] 895410024@s95.tku.edu.tw[d]

Abstract: - This paper presents a robust and efficient text detection algorithm for news video. The proposed algorithm uses the temporal information of video and logical AND operation to remove most of irrelevant background. Then a window-based method by counting the black-and-white transitions is applied on the resulted edge map to obtain rough text blobs. Line deletion technique is used twice to refine the text blocks. The proposed algorithm is applicable to multiple languages (English, Japanese and Chinese), robust to text polarities (positive or negative), various character sizes (from 4×7 to 30×30), and text alignments (horizontal or vertical).Three metrics, recall (R), precision (P), and quality of bounding preciseness (Q), are adopted to measure the efficacy of text detection algorithms. According to the experimental results on various multilingual video sequences, the proposed algorithm has a 96% and above performance in all three metrics. Comparing to existing methods, our method has better performance especially in the quality of bounding preciseness that is crucial to later binarization process.

Key-Words: - Information retrieval, Multiple frames integration, Video text, Text detection, Canny edge map, Black-white transition, Line-deletion

1 Introduction

Video is a rich and convenient way to get information due to advanced and friendly multimedia techniques available. In order to help users locate the video content of interest to them, much research [1] has been dedicated to the subject of video indexing and information retrieval. Of the various video processing techniques, the use of text is the most habitually used in content understanding. Video text can be scene text or caption text [2]. Scene text appears unpredictably and disappears quite rapidly after being visually introduced. An example of scene text could be a store's name in a commercial; however, scene text is not that important of a factor when it comes to video content understanding. Alternately, caption text, static or scrolling, is superimposed in a later stage of videos producing. Most of static texts provide concise and direct description of the content presented in news video (e.g. the title of the current issue, names of anchorpersons), whereas scrolling texts are usually updated information (e.g. figures from stock markets, upcoming programs) which are not related to the current content. The procedure of video textual information extraction can be broadly divided into two categories; detection and recognition. Detection is used to label text regions and recognition is used to binarize the detected text regions and perform optical character recognition (OCR). Though text detection in a complex background is challenging, it is still critical and necessary since successful recognition is based on good detection. A good video text detection result must satisfy the following requirements: the text is masked or bounded by a box as tight/close as possible, the false detection is low, and the recall rate is high. Thus, three metrics, recall (*R*), precision (*P*), and quality of bounding preciseness (*Q*), can be used to measure the efficacy of text detection algorithms.

Many existing methods utilize a single frame to highlight video texts [3, 4, 5]. A disadvantage to this method is the difficulty to distinguish whether the detected edges are really from video texts. This problem is alleviated by the multiple frames integration method [6, 7, 8, 9]. Based on this, an

algorithm comprised of robust text features and a non-text line deletion technique to detect static caption texts on news video is proposed. The purpose is to design a text detection method that fulfills the metrics, R, P, and Q, without placing prior constraint on the videos.

The rest of this paper is organized as follows: Section 2 reviews the related work, Section 3 describes our text detection algorithm, Section 4 will highlight our findings, and the final section gives our conclusion.

2 Related Work

Video text detection methods can be classified into three classes [5]. The first class is texture-based [6, 9, 10]. It assumes that texts in images have distinct textural properties which can be used to distinguish them from non-text like the horizontal energy in wavelet transform. Generally speaking, the texture based method is more robust than the connected component based method in detecting texts in a complex background, but the high computational cost is a concern. The connected component based method is to segment an image into components by grouping edge pixels, then delete or merge components by geometrical characteristics of text blocks. This approach is intuitive but sensitive to complex backgrounds. The second class presumes that a text string contains a uniform color [11, 12]. Color-reduction is first applied followed by segmentation in a selected color channel or color space. Connected component analysis is then used to detect text regions. The third class is edge-based [5, 13, 14, 15]. This method utilizes stroke density and the contrast characteristics of the text. In general, edge detection [16] is first applied and then the horizontal profile projection histogram [17, 18] is constructed. A candidate text region is identified if the histogram bins are tall enough [5]. The main drawback of this method is the difficulty of finding a proper threshold. If a threshold is high, then it cannot detect short text strings; contrarily, it may detect errantly. Besides the text detection problem, how well the detected regions fit texts is also important. Although not discussed elaborately in the textural information extraction problem, if precise and tight text localizations are achieved, then the neighbor areas are also clearly defined and can be used to further improve the detection result. As in [19], edges from neighboring areas are used to verify whether a candidate block is a text block; in [9], a morphological reconstruction is applied on neighboring areas to remove irrelevant backgrounds. Precise text edges result in better binarization and recognition in video texts.

3 The Proposed Method

In general, a viewer needs 2 seconds or more to process a complex scene [20, 21]. Thus, if videos are played f frames per second, we are interested in detecting video texts staying on a fixed location for at least $2f$ consecutive frames. Let k be the nearest integer that is not less than f. We define every consecutive k frames to be one round, i.e., the first round is made of frames 1~k, and the second round is made of frames (k+1)~2k, etc. It can be shown that any 2k consecutive positive integers must have 2 integers congruent to r modulo k that are k apart for any r = 0, 1, …, k-1 (the proof is in the Appendix). Hence, any video text lasting for 2 seconds or more must appear on frames (m-1)k and mk for some positive integer m (by letting r=0 and two frames being k frames apart). Thus the same text appears on the fixed position for every frame on the m^{th} round which is made of frames (m-1)k, (m-1)k+1, …, mk. To simplify the calculation, for the m^{th} round, we define four reference frames on frames (m-1)·k+i, i= 1, $\lfloor k/3 \rfloor$, 2$\lfloor k/3 \rfloor$, 3$\lfloor k/3 \rfloor$, and the intersection of these reference frames can be used for checking whether there is any static text in that round. The flowchart of the proposed approach is given in Fig. 1 and the detailed implementation in one round is delineated below, in which one can assume h and w are the height and width of the character size that will be the focus for the remainder of the discussion.

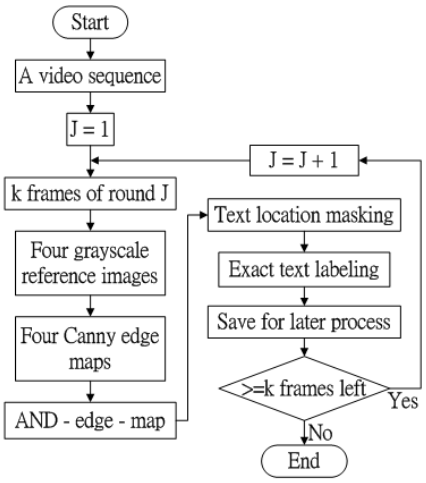


Fig. 1. The flowchart of the proposed approach.

Step 1: Get four reference frames from the given one round of video frames and transform them into grayscale images. There are several ways to convert

color images into grayscale images. We use Eq.(1) to accomplish this.

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \quad (1)$$

where Y is the intensity value and R, G, B are the values on red, green, blue channels of the pixel. Figures 2 and 3 show the reference frames before and after the conversion.

(a) (b)

(c) (d)

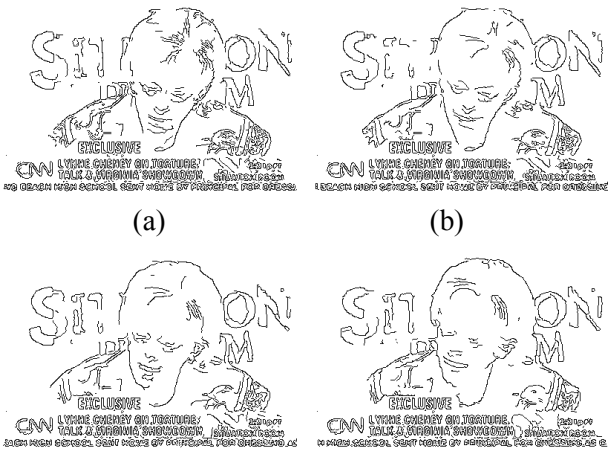
Fig. 2. Four color reference frames.

(a) (b)

(c) (d)

Fig. 3. Four grayscale reference images.

Step 2: Execute the edge detection on the grayscale reference images. The Canny edge detector is applied on each grayscale image yielding an edge map. A simple line (horizontal or vertical) deletion is implemented to remove long lines which are unlikely to be characters. From left to right and top to bottom of the edge map, a horizontal line (and/or a vertical line) is removed if its length exceeds the presumed width w (height h) of a character. The edge map after line deletion is called a Canny edge map. Figure 4 shows four Canny edge maps from Fig. 3.



(a) (b)

(c) (d)

Fig. 4. Four Canny edge maps after removing lines those are too long.

Step 3: Do logical AND on Canny edge maps. Note that after AND operation, a position (i, j) is true (an edge pixel) if all four Canny edge maps are true at (i, j). Thus, using AND, the video texts are kept if they are the same texts on the same location, whereas edge pixels not on the same location for all four Canny edge maps will be eliminated. We call the resulted image an AND-edge-map. Figure 5 shows the AND-edge-map to demonstrate the effect of taking logical AND operation. Most of the background edge pixels and the scrolling texts on the bottom of the Fig. 4 (a) ~ (d) are removed, but the static video texts are preserved.

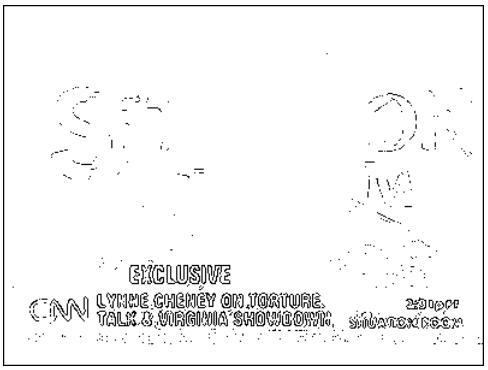


Fig. 5. The AND-edge-map: the result after taking AND operation on four Canny edge maps of Fig. 4.

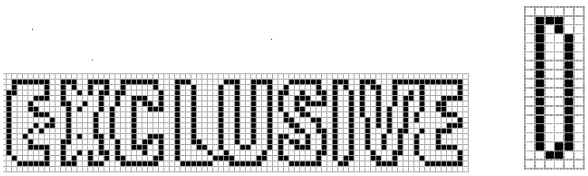
Step 4: Mask text location. A three-stage technique is designed to find the text mask. First, a rough text blob is obtained utilizing the number of black and white transitions column-wise and row-wise, then non-text noises and isolated noises are removed, finally the morphological operation is applied for

compensation to obtain the mask text region. Details are given below.

(a) A window of the size $w \times h$ (presumed character size) slides from left to right (per column) and top to bottom (per row) on the AND-edge-map. The value of *BWT* represents the transitions from black to white or from white to black for every row and every column inside the window as shown in Eq. (2).

$$BWT=\sum_{i=0}^{h-1}(\sum_{j=1}^{w-1}|b(i,j)-b(i,j-1)|)+\sum_{j=0}^{w-1}(\sum_{i=1}^{h-1}|b(i,j)-b(i-1,j)|), \quad (2)$$

where $w \times h$ is the window size, $b(\cdot)=1$ if it is black and 0 otherwise. If *BWT* is larger than the threshold T_{BWT}, this window is masked. The union of all masked windows is the rough text blob. The threshold T_{BWT} depends on the character size, i.e., $T_{BWT} = \beta \cdot (w \cdot h)$ with β a constant. Observe the "I" on "EXCLUSIVE" of the AND-edge-map on Fig. 5 as an example where "I" is the one with the least black and white transitions in 26 English letters. Figure 6(a) shows the enlarged version. In this example, each letter has dimension approximately 10×18 and the *BWT* in Fig. 6(b) is 72 which is 0.4 of 10×18. Besides, the edge pixels for characters may become less since the logical AND operation. The experimental results on many multilingual video texts are satisfactory when β is 0.35~0.37.

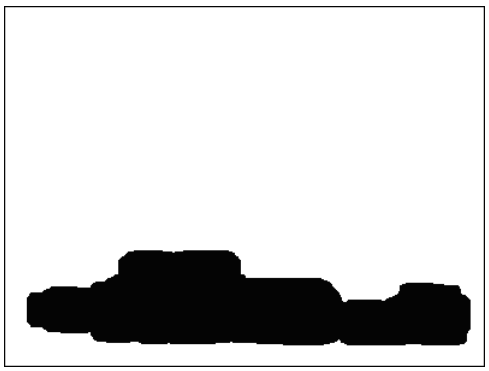


(a) (b)

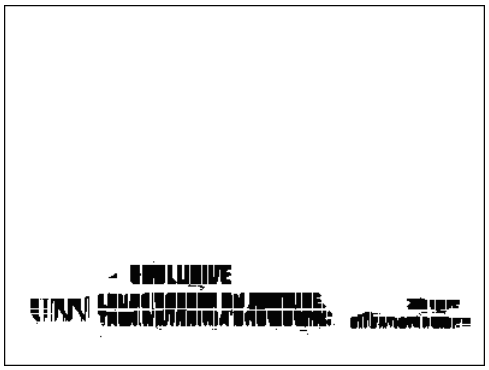
Fig. 6. (a) Edge of the string. (b) Edge of English letter "I".

(b) Every rough text blob is examined from left to right and top to bottom for every masked pixel to remove the non-text ones. For a masked point located on (i, j) position, a horizontal line segment of length w comprising points on (i, j), …, $(i, j+w-1)$ will be eliminated if neither of these points is an edge point on the AND-edge-map, and a vertical line segment of length h comprising points on (i, j), …, $(i+h-1, j)$ will be eliminated if neither of them is an edge point on the AND-edge-map. A simple connected component analysis is then followed to remove isolated pixels.

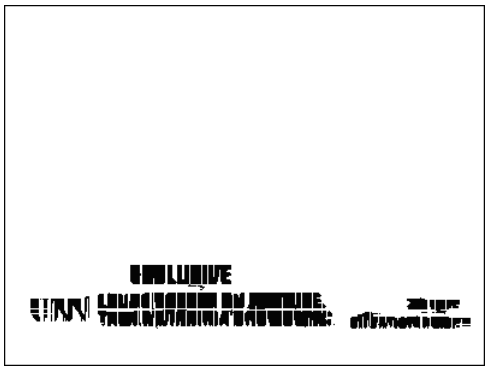
(c) Due to various contrasts caused by different backgrounds in reference frames, the results of the Canny edge detector in reference frames of the same text in the same location may differ in a few pixels. This causes characters to lose some pixels in the AND-operation. To remedy this problem, a morphological closing with a horizontal structuring element of size $\lceil w/3 \rceil$ is applied first to fill holes then followed by a dilation with SE size $\lceil w/4 \rceil \times \lceil h/4 \rceil$ to connect characters. Figure 7 shows the text location masking result: (a) is the obtained rough text blob; (b) is after non-text line deletion where the reference AND-edge-map is Fig. 5; (c) shows the result of isolated noise removed; (d) shows the compensated mask text region. We call this image the masked image M.



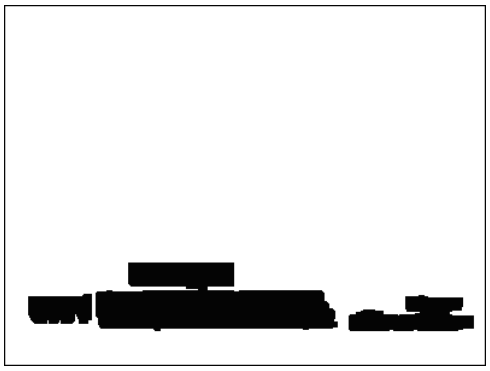
(a) The obtained rough text blob



(b) Non-text pixels removal



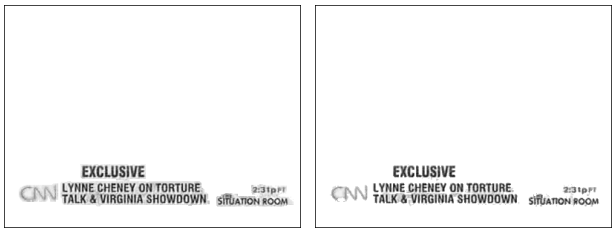
(c) Isolated noises removal



(d) Morphological compensation

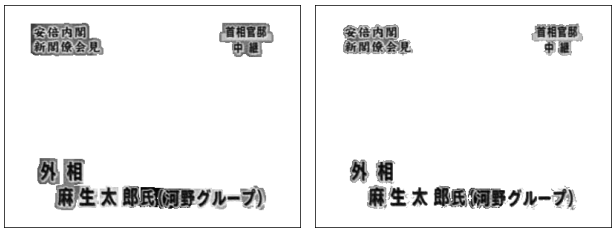
Fig. 7. The result of the masked region of Fig. 5.

Step 5: Exact text location. In this step, a rough text region and its edge map are first obtained by overlaying the masked region M with one of the four grayscale reference images and its corresponding Canny edge map, e.g. the first grayscale reference image and the first Canny edge map. It makes little difference which reference image is chosen. Note that in the rough text image and its edge map, most of the pixels are white because most of the pixels are not masked in M. To do the refinement, we examine every white pixel in the rough text image from left to right and top to bottom. Similarly, leftwards points on (i, j-1), (i, j-2), …, are examined until the first edge point is found and converted into white pixels on the rough text image when necessary; downwards and rightwards points are examined and converted (if necessary) likewise. Figure 8 shows the result: (a) is the rough text image using the grayscale reference image in Fig. 3(a), and (b) is after refinement where the corresponding edge map on Fig. 4(a) is used for refinement. Comparing images in Fig. 8 (a) and (b), we can see most of background pixels are cleared, e.g. "CNN" on the lower left corner and "THE SITUATION ROOM" on the lower right corner. In Fig. 9 we give another example to demonstrate the effectiveness of the refinement. Finally, to label the detected texts, we do a simple binarization on the refined text image followed by a morphological operation to connect texts that are close to each other, and use a rectangular box to inscribe the connected text blob. These rectangular boxes are called detected boxes. A detected box which has too few edge pixels (< 50) will be deleted to eliminate unlikely text boxes.



(a) The rough text image (b) after refinement

Fig. 8. The result of text extraction.



(a) (b)

Fig. 9. Effectiveness of text refinement (a) before and (b) after.

4 The Experimental Results

The proposed text detection algorithm was evaluated on multilingual videos clips from CNN, ESPN (USA), NHK (Japan) and ETTV, TVBS (Taiwan) for a total of 30 minutes 18 seconds. All of these videos have a resolution of 400×300 and a frame rate 29.97 per second (thus $k = \lceil 29.97 \rceil = 30$). The presumed largest character size $w \times h$ was set to be 20×20 and β in T_{BWT} (threshold for BWT in (2)) was set to be 0.35.

(a) (b) (c)

Fig. 10. Three bounding boxes for the same text from large to small.

4.1 Ground Truth Data and Evaluating Metrics

As the text is in gray scale, it is not trivial to define the ground truth bounding. Take "EXCLUSIVE" as an example in Fig. 10. The dimensions of bounding boxes are 88×20, 86×18, and 84×16 for (a), (b), (c) respectively. Any of them are perfect to serve as a ground truth, but the area in (c) is only 76.4% of that in (a). To accommodate these cases, we take the medium bounding box, the one in (b), as the ground truth and apply the same standard to all data in our experiments. A detected box truly detects the texts if the area ratio r, defined in Eq. (3), is at least 50%.

$$r = \frac{Area(D\_BOX \cap G\_BOX)}{Area(D\_BOX \cup G\_BOX)}, \quad (3)$$

where *D_BOX* is a detected box like yellow boxes in Fig. 11, *G_BOX* is the ground truth text box like blue boxes in Fig. 11. The area ratios *r* of Fig. 11 are 40.0%, 46.4%, 61.1%, 74.3%, and 82.3% for (a), (b), (c), (d), (e), respectively. Thus the text is not truly detected in (a) and (b), and is detected in (c), (d), (e). To focus the bounding preciseness, if the area ratio *r* in Eq. (3) is at least 80% we say the text is accurately detected. Accordingly, a detected box may detect texts (*D_BOX_T*) or may not detect (*D_BOX_F*). The former, *D_BOX_T*, contains detected boxes with $r \geq 50\%$; the latter, *D_BOX_F*, comprises those boxes with $r < 50\%$ or false detections (no text at all). Thus, in Fig. 11, (a) and (b) belong to the set of *D_BOX_F*, (c), (d), (e) belong to the set of *D_BOX_T*, and only (e) is accurately detected. We use recall (*R*), precision (*P*), and quality of bounding preciseness (*Q*) to measure the efficacy of algorithms as in Eq.s (4), (5), (6).

$$R = \frac{\#(D\_BOX\_T)}{\#(G\_BOX)}, \quad (4)$$

$$P = \frac{\#(D\_BOX\_T)}{\#(D\_BOX)} = \frac{\#(D\_BOX\_T)}{\#(D\_BOX\_T) + \#(D\_BOX\_F)}, \quad (5)$$

$$Q = \frac{\#(Acu\_D\_BOX\_T)}{\#(D\_BOX\_T)}, \quad (6)$$

where *Acu_D_BOX_T* indicates those boxes really detect texts and their area ratios $r \geq 80\%$.

To define a rigid ground truth data is challenging too. To our belief, the aim of video text analysis is to make the detected text correctly binarized and then recognized by an OCR. Take characters "TALK & VIRGINIA SHOWDOWN" in Fig. 12 as an example. We accept all (a)~(d) cases to be ground truths since text "TALK & VIRGINIA SHOWDOWN" is included. However, cases (a) and (b) are better because detected regions should be as tight as possible to exact edge pixels to reduce the influences of irrelevant background in binarization stage. Without rigid ground truth data may cause a problem in calculating recall *R* since #(*G_BOX*) is unclear. Yet, it has no impact on our algorithm because it never misses any true video texts in experiments done so far. To avoid confusion, we only measure the *R*, *P*, and *Q* for our method and discuss the effects on the existing methods.

(a) (b) (c) (d) (e)

Fig. 11. Detected boxes (in yellow) for a ground truth text (in blue) (a) and (b) fail to detect, (c)~(e) truly detect it but only (e) accurately detects the text.

(a) (b)

(c) (d)

Fig. 12. The ambiguities in defining a ground truth for "TALK & VIRGINIA SHOWDOWN".

4.2 Experimental Results and Discussions

Tables 1 and 2 depict the experimental results. Our proposed method reached 97.45%, 97.39%, and 95.97% on *R*, *P*, and *Q*. These figures are excellent compared to existing research. Some detection results are shown in Fig. 13. The bottom text lines in (a) and (f), the left text column and the bottom text line in (e) are scrolling texts. Since tested videos are comprised of English, Japanese, and Chinese, it is evident that our method is applicable to multi-languages. We summarize the contributions of the proposed algorithm in the following:

(1) Precise boxes: This will greatly benefit the later binarization process.

(2) Positive and negative text polarities: As observed in Fig. 13(e), there are dark texts in a high intensity background (positive polarity) and white texts in a dark background (lower center area) (negative polarity), and more examples can be found in Fig. 13(c).

(3) The alignment and the length of text strings: As mentioned, the horizontal profile projection is a common approach to determine text locations; however, it may fail when a text is vertically aligned or a text contains too few characters. In the proposed method, sliding windows and *BWT* values are used to determine possible text candidates and this successfully solve the problem. As illustrated in Fig. 13, there are correctly detected horizontal and vertical texts in one frame (Fig. 13(f)), and strings that have only a few characters (wuai xiang- the second text line from the bottom of Fig. 13(d) and "LIVE" in Fig. 13(e)). In addition, we do not make any assumptions on text locations. Thus, the proposed method can detect text in any positions on the frame.

(4) Sizes, fonts, and multi-color characters: In Fig. 13(e), two characters on the lower right (zui xin) have a size of 30×30 for each character, "09:06" (on the right of the bottom text line) has a size of approximately 10×10 for each digit, and in Fig. 13(b) "THE" in "THE SITUATION ROOM" has three letters with a box of 11×7 (appox. 4×7 for one letter). These results show that characters can be detected with sizes ranging from $(1.5)w \times (1.5)h$ to $(1/3)w \times (1/3)h$ where $w \times h$ is a presumed character size (20×20 in our case). In Fig. 13 (f), observe the five artistic styled characters on the upper left corner, the colors in the first two (bian an) are transferring from white to green, and the colors for the other three (da xuan pan) are changing between white and yellow.

Nevertheless, we still have some false detection and boxes failing to be detected. As in Fig. 13 (c), the "2" on the lower right is inscribed in two boxes, one of them coincides with the inserted yellow background and it is considered as a false detection since "2" is detected in the other box. The same situation happened in the next round on ESPN video clip, and this is how two false detections occur. Among our experiments, the worst experimental result is the ESPN video due to the small, isolated, and sparsely located characters. We repeated the experiment on the ESPN video with a presumed character size of $w \times h$ to be halved (10×10). As shown in Fig. 14, among detected boxes, only "IAAF" has the area ratio r less than 50%, and figures "1.98" and "2.03" show r to be 77% and 71%, respectively, the remaining boxes are all accurately detected ($r \geq 80\%$), and neither show a false detection nor appear missing. It has a much better result compared to Fig. 13(c). This indicates that the presumed character size may have to be adjusted for some special videos. The other problem in our method occurred in TVBS (Taiwan) news videos. Both in ETTV and TVBS, the area ratios of r are not as good as the others. This is due to an image, such as a logo, intentionally attached to the characters. As in the vertical text line in Fig. 13(f), the first character is connected to a diamond image causing the ratio to be below 50% (the diamond is related to the content of the news), and a similar reason is observed in "IAAF" in Fig. 14.

Some results from existing methods are shown in Fig. 15. Since the source videos are not available, we chose test videos carefully so that they are as comparable as possible. In all these existing methods, the bounding preciseness is not as good as ours. This confirms that the non-text line deletion technique in step (2) and the refinement in step (5) are really useful. By testing on comparable videos, our algorithm performed satisfactorily with impressive P, Q, R values for all videos. It is worth noticing that the experimental results in Fig. 15 are taken from [6, 8, 13, 15], which are examples depicting the accomplishments of their methods. These examples show that the proposed algorithm has advantages over these existing methods, especially in the quality of bounding preciseness.

Table 1. Results on different video sequences

Video Sources	Type/Language	Length min sec	# of *G_BOX*	*D_BOX_F*		*D_BOX_T*	
				# of False Alarms	# of boxes with $r < 50\%$	# of boxes with $50\% \leq r < 80\%$	# of boxes with $r \geq 80\%$
CNN	News/English	9'16"	404	0	4	9	391
ESPN	Sport/English	5'07"	111	2	11	29	71
NHK	News/Japanese	5'35"	310	0	0	0	310
ETTV	News/Chinese	5'29"	939	0	0	27	912
TVBS	News/Chinese	4'51"	1340	0	64	57	1219
Total	--/--	30'18"	3104	2	79	122	2903

Table 2. Results on R, P, and Q

Channel	Recall(R)	Precision(P)	Quality(Q)
CNN	400/404 = 99.01%	400/404 = 99.01%	391/400 = 97.75%
ESPN	100/111 = 90.09%	100/113 = 88.50%	71/100 = 71.00%
NHK	310/310 = 100.00%	310/310 = 100.00%	310/310 = 100.00%
ETTV	939/939 = 100.00%	939/939 = 100.00%	912/939 = 97.12%
TVBS	1276/1340 = 95.22%	1276/1340 = 95.22%	1219/1276 = 95.53%
Average	3025/3104 = 97.45%	3025/3106 = 97.39%	2903/3025 = 95.97%

(a) (b) (c) (d) (e) (f)

Fig. 13. Some results of our method: (a) and (b) are CNN videos, (c) ESPN video, (d) NHK video from Japan, (e) ETTV and (f) TVBS are two different news videos from Taiwan.

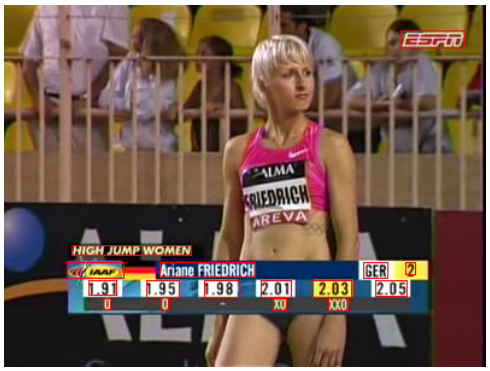


Fig. 14. The detection result with presumed character size to be 10×10.

(a) (b) (c) (d)

Fig. 15. Some results of other's method: (a) [6], (b) [8], (c) [13], (d) [15].

5 Conclusion

In this paper, we proposed a text detection algorithm that is applicable to multilingual news videos. Since we are interested in detecting static video text, logical AND operation on multiple Canny edge maps can remove most of irrelevant background. By this way, the noise-sensitivity problem of edge-based methods is alleviated, whereas the robustness on different text colors can still be retained. Different from the methods utilizing the horizontal profile projection histogram [17, 18], the proposed algorithm uses a window-based scanning method with the *BWT* to locate rough text blobs. Therefore, the method is robust to fonts or sizes, alignments, or length of text strings. Finally, a non-text line deletion technique combining edge maps is used twice to achieve very precise bounding boxes on detected texts. The algorithm has no complex calculation and is very efficient since it only processes 4 frames out of $\lceil f \rceil$ frames if the video is played on a frame rate of f frames per second. The proposed algorithm was tested on multilingual news videos (English, Japanese and Chinese) including different text polarities (positive

and negative), different character sizes (from 4x7 to 30x30), short text strings, and different text alignments (horizontal and vertical). This algorithm has excellent performances in recall (R), precision (P), and quality of bounding preciseness (Q) which are the best compared to existing experimental results that we have known so far.

Acknowledgment

This paper is an extension of the paper "Precise News Video Text Detection/Localization Based on Multiple Frames Integration" published in [22].

Appendix

Lemma. Any $2k$ consecutive positive integers must have 2 integers congruent to r modulo k that are k apart for every $r = 0, 1, \ldots, k-1$.

Proof: Assume $n, n+1, ..., n+k-1, n+k, \ldots, n+2k-1$ are $2k$ consecutive positive integers and $n \equiv r'$ (mod k) for some integer r' in $0, 1, \ldots, k-1$.

Then $(n+k) = (qk+r')+k = (q+1)k+r'$ for some integer q; thus,

$n \equiv (n+k) \equiv r'$ (mod k) (*)

Hence, there are two integers n and $n+k$ both congruent to r' modulo k. And, $(n+1) = (qk+r')+1 = qk+(r'+1)$ implies $(n+1) \equiv (r'+1)$ (mod k). As in (*), $(n+1) \equiv (n+k+1) \equiv (r'+1)$ (mod k).

There are two integers $(n+1)$ and $(n+k+1)$ both congruent to $(r'+1)$ modulo k.

Similarly, we can get

$(n+2) \equiv (n+k+2) \equiv (r'+2)$ (mod k),

:

:

$(n+k-1) \equiv (n+2k-1) \equiv (r'+k-1)$ (mod k).

And these k consecutive numbers r', $r'+1, \ldots, r'+k-1$ are exactly equivalent to $0, 1, \ldots, k-1$ in some order when considering them as remainders of k.

References:

[1] Chang, H., Automatic Web Image Annotation for Image Retrieval Systems, *12th WSEAS International Conference on SYSTEMS*, 2008, pp. 670-674.

[2] Jung, K., Kim, K. I., Jain, A. K., Text information extraction in images and video: A survey, *Pattern Recognition*, Vol. 37, No. 5, 2004, pp. 977-997.

[3] Liu, Y., Lu, H., Xue, X., Tan, Y., Effective video text detection using line feature, *8th International Conference on Control, Automation, Robotics and Vision* (ICARCV), Vol. 2, 2004, pp. 1528-1532.

[4] Wang, J., Zhou, Y., An unsupervised approach for video text localization, *IEICE TRANS. INF. & SYST.*, Vol. E89-D, Issue: 4, 2006, pp. 1582-1585.

[5] Tsai, T., Chen, Y., Fang, C., A two-directional videotext extractor for rapid and elaborate design, *Pattern Recognition*, Vol. 42, Issue 7, 2009, pp. 1496-1510.

[6] Wang, R., Jin, W., Wu, L., A novel video caption detection approach using multi-frame integration, *Pattern Recognition*, 1*7th International Conference on ICPR2004*, Vol. 1, 2004, pp. 449-452.

[7] Mi, C., Xu, Y., Lu, H., Xue, X., A novel video text extraction approach based on multiple frames, *IEEE International Conference on Information, Communications and signal Processing* (ICICS 2005), Vol. 2005, No. 1689133, 2005, pp. 678-682.

[8] Huang, X., Ma, H., Yuan, H., A novel video text detection and localization approach, *PCM 2008*. LNCS 5353, 2008, pp. 525-534.

[9] Yen, S., Wang, C., Yeh, J., Lin, M., Lin, H., Text extraction in video images, *The Second IEEE International Conference on Secure System Integration and Reliability Improvement* (SSIRI2008), 2008, pp. 189-190.

[10] Sun, H., Zhao, N., Xu, X., Extraction of text under complex background using wavelet transform and support vector machine, *IEEE International Conference on Mechatronics and Automation* (ICMNA 2006), Vol. 2006, No. 4026310, 2006, pp. 1493-1497.

[11] Jain, A. K., Yu, B., Automatic text location in images and video frames, *Pattern Recognition*, Vol. 31, No. 12, 1998, pp. 2055-2076.

[12] Mariano, V. Y., Kasturi, R., Locating uniform-colored text in video frames. *Proc. 15th Int. Conf. Pattern Recognition*, vol. 4, 2000, pp. 539-542.

[13] Hua, X., Chen, X., Liu, W., Zhang, H., Automatic location of text in video frames, *Proceeding of ACM Multimedia Workshops (MIR2001)*, 2001, pp. 24-27.

[14] Lyu, M. R., Song, J., Cai, M., A comprehensive method for multilingual video text detection, localization, and extraction,

IEEE Trans. on Circuits And Systems For Video Technology, Vol. 15, Issue: 2, 2005, pp. 243-255.

[15] Anthimopoulos, M., Gatos, B., Pratikakis, I., A hybrid system for text detection in video frames, *Document Analysis Systems, DAS '08. The Eighth IAPR International Workshop*, 2008, pp. 286-292.

[16] Safi, A., Azam, M., Kiani, S., Daudpota, N., Online Vehicles License Plate Detection and Recognition System using Image Processing Techniques, *Proceedings of the 5th WSEAS International Conference on Applied Computer Science*, 2006, pp. 793-800.

[17] Brook, S., Aghbari, Z., Holistic Approach for Classifying and Retrieving Personal Arabic Handwritten Documents, *7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES* (AIKED'08), 2008, pp. 565-570.

[18] Choi, S., Yun, J., Koo, K., Choi, J., Kim, S., Text Region Extraction Algorithm On Steel Making Process, *8th WSEAS Int. Conf. on ROBOTICS, CONTROL and MANUFACTURING TECHNOLOGY* (ROCOM '08), 2008, pp. 24-28.

[19] Zhang, J., Goldgof, D., Kasturi, R., A new edge-based text verification approach for video, *19th International Conference on Pattern Recognition*, ICPR2008, 2008, pp. 1-4.

[20] Lindsay, P. H., Norman, D.A., *Introduction Into Psychology-Human Information Reception and Processing (in German)*. Springer-Verlag, Berlin, Germany. 1991.

[21] Pfeiffer, S., Lienhart, R., Fischer, S., Effelsberg, W., Abstraction digital movies automatically, *J. Vis. Comm. Image Represent.*, Vol. 7, No. 4, 1996, pp. 345-353.

[22] Yen, S., Chang, H., Precise News Video Text Detection/Localization Based on Multiple Frames Integration, *10th WSEAS Int. Conf. on SIGNAL PROCESSING, COMPUTATIONAL GEOMETRY and ARTIFICIAL VISION* (ISCGAV '10), 2010.