Precise News Video Text Detection/Localization Based on Multiple Frames Integration

SHWU-HUEY YEN¹, HSIAO-WEI CHANG¹,²
Department of Computer Science and Information Engineering
Tamkang University¹
151 Ying-Chuan Road, Tamsui, Taipei County 25137, Taiwan
REPUBLIC OF CHINA
Department of Computer Science and Information Engineering
China University of Science and Technology²
245, Sec. 3, Academia Road, Taipei City 11581, Taiwan
REPUBLIC OF CHINA
shyen@cs.tku.edu.tw  changhw@cc.cust.edu.tw

Abstract: - This paper presents a multiple frames integration based approach to detect and localize static caption texts on news videos. Utilizing the temporal information of videos, the algorithm includes robust text features and the non-text line deletion technique, and yields precise and tight localization for detected text regions. The Canny edge detector is first applied on reference frames and is followed by executing the logical AND to reduce the edges from the variation of the background including the scrolling texts. Next, rough text candidate regions are determined by calculating the number black-white transition (BWT). Finally, the text regions are refined by the non-text line deletion technique. The proposed algorithm is applicable to multiple languages and robust to text polarities, alignments, and character sizes (from 10×10 to 30×30). According to the experimental results on various multilingual video sequences, the proposed algorithm has a 96% and above performance in recall, precision, and quality of bounding preciseness.

Key-Words: - information retrieval, multiple frames integration, video text, text detection, Canny edge map, black-white transition

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 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. Extraction of text information from video includes detection, localization, extraction, and recognition. However, variations of text due to differences in size, style, and alignment, as well as low image contrast and complex background make the problem extremely challenging. Text in video can be scene text or caption text [1]. Scene text presents in the image as a part of it, e.g., a store’s name in a street scene. Caption text, static or scrolling, is superimposed in a later stage of videos producing. In news video, most static caption texts provide concise and direct description of the content presented, e.g., the title of the current issue, whereas, scrolling texts are usually updated information, e.g., figures from stock markets and upcoming programs, which are not related to the current content. In this paper, we focus on the detection and localization of static texts in news video which play a critical role to successful extraction and recognition for video text. In order to determine whether the text detection/localization that is good, the following requirements must be met; the false detection is low, the recall rate is high, and the text is bounded by a box as tight as possible. Thus, three metrics, precision ($P$), recall ($R$), 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 [2, 3, 4]. 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 [5, 6, 7, 8]. 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 of this new methodology is to design a general news video text detection method that fulfills the three metrics, $R$, $P$, and $Q$, without placing prior constraint on the videos.

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

2 Related Work

Connected component (CC) analysis combining with other methodologies is a common approach in solving the text detection problem. According to certain text features, possible text candidates are detected. Then, these text candidates are grouped as connected components and further processed by text properties. The method using CC analysis alone usually is sensitive to complex background. Video text detection methods can be classified into three classes [4]. The first class is texture-based [5, 8, 9]. It assumes that texts in images have distinct textural properties that can be used to distinguish them from non-text, such as the horizontal energy in wavelet transform. Generally speaking, the texture based method is more robust than the CC-based method in detecting texts in a complex background, but the high computational cost is a concern. The second class presumes that a text string contains a uniform color [10, 11]. 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 [12, 13, 14]. This method utilizes stroke density and the contrast characteristics of the text. In general, edge detection is first applied and then the horizontal profile projection histogram is constructed. A candidate text region is identified if the histogram bins are tall enough [4]. The main drawback of this method is the difficulty of finding a proper threshold. Besides the text detection problem, how well the localization of detected regions 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 recognition result. As in [15], edges from neighboring areas are used to verify whether a candidate block is a text block; in [8], a morphological reconstruction is applied on neighboring areas to remove irrelevant backgrounds.

3 The Proposed Method

Usually, people need 2 seconds or more to process a complex scene [16, 17]. Thus, if videos are played \( f \) frames per second, we are interested in video texts focusing 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 starting from frame one. On the \( s \)th round \((s \geq 1)\), four reference frames are defined to be frames \((s-1)k+i, i=1,\lfloor k/3\rfloor, 2\lfloor k/3\rfloor, 3\lfloor k/3\rfloor\). When divided by \( k \) for any \(2k\) consecutive positive integers there must have 2 integers, \( I \) and \( I+k\), that are of the same remainder \( r \) for any \( r = 0, 1, \ldots, k-1 \). Hence, there are two \( k \)-apart frames that are multiples of \( k \) (i.e., remainder to be zero) have the same video text if it lasts for 2 seconds or more. Let these two frames be frames \((m-1)k \) and \( mk \) for some positive integer \( m \), then the same texts appear on the fixed location for frames \((m-1)k, (m-1)k+1, \ldots, mk \). As defined, frames \((m-1)k+1 \) and \( mk \) constitute the \( m \)th round. We conclude that video texts lasting for 2 seconds or more must occur on the same location for at least one round and therefore on the four reference frames of that round. Thus the intersection of the reference frames can be used for checking whether there is any static text in one 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.](image)

**Step 1:** Get four reference frames from the given one round of video frames and transform them into grayscale images by Eq.(1).

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

where \( Y \) is the intensity value and \( R, G, B \) are the values on red, green, blue channels of the pixel. Figure 2 shows the reference frames after the conversion.

**Step 2:** Execute the edge detection on the grayscale images. The Canny edge detector is applied on each grayscale image yielding an edge map. A simple line (horizontal or vertical) deletion is followed if a line
is too long. 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 3 (a) shows one Canny edge map.

**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 only if they are the same texts on the same location. We call the resulted image an AND-edge-map. Figure 3(b) shows the AND-edge-map to demonstrate the effect of taking logical AND operation. The background edge pixels and the scrolling texts on the bottom of the Fig. 2 (a) – (d) are mostly removed, but the static video texts are preserved.

**Step 4:** Mask text location. A three-stage technique is designed to find the text mask. First, a rough mask is obtained, then non-text noises and isolated noises are removed, finally the morphological operation is applied for compensation. Details are giving below.

(a) A window the size of $w \times h$ (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).

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

where $h$ and $w$ are presumed height and width of the character 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 mask for candidate text blocks. The threshold $T_{BWT}$ depends on the character size, i.e., $T_{BWT} = \beta(w \cdot h)$ with $\beta$ a constant.

(b) The rough mask is examined from left to right and top to bottom for every masked pixel to remove the non-text ones. Consider a masked point located on $(i, j)$ position, a horizontal line segment of length $w$ comprising points on $(i, j), \ldots, (i, j+w-1)$ 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 $(i, j), \ldots, (i+h-1, j)$ will be eliminated if neither of them is an edge point on the AND-edge-map. Isolated masked points are further removed by a simple CC analysis.

(c) It is very possible that some text pixels are lost after AND-operation due to variations in background and contrast in reference frames. To alleviate this problem, a morphological compensated image $M$ is obtained by first applying a closing with a horizontal structuring element (SE) of size $\left\lceil \frac{w}{3} \right\rceil$ to fill holes among edge pixels. Isolated blobs of small size are removed followed by a dilation of SE size $\left\lceil \frac{w}{4} \right\rceil \times \left\lceil \frac{h}{4} \right\rceil$ to extend the text region. Figure 4 shows the result of masking text location for the AND-edge map on Fig. 3(b), (a) is the obtained rough mask image and (b) is the compensated mask text regions $M$.

**Step 5:** Exact text location. Overlay the $M$ on one of the reference images and its Canny edge map, then keep only points that are masked in $M$ from both images to obtain a rough text image and a corresponding text-edge-map. Figure 5(a) shows the
rough text image using the grayscale reference image in Fig. 2(a). Note that most of the pixels are white in the rough text image and the text-edge-map, 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. If a white pixel is located on (i, j) position of the rough text image, then points on the top, bottom, left, and right of (i, j) position of the (binary) text-edge-map will be examined. In checking the upwards direction, if (i-1, j), (i-2, j), …, (i-k+1, j) points are all white and (i-k, j) is the first edge pixel (black point), then every point on (i-1, j), (i-2, j), …, (i-k+1, j) in the rough text image must be non-text accordingly. Thus, any non-white pixels will be converted into white in the rough text image. 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 5(b) shows the refinement of 5(a). Comparing the results in Fig. 5 (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. 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) is unlikely to be text and will be deleted.

![Image 55x267 to 169x353](image)

(a) The rough text image (b) after refinement
Fig. 5. The result of text extraction.

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 approximately 30 minutes. 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 character size \(w\times h\) was set to be 20×20 which is comparable to common character sizes in news videos. To decide the parameter \(\beta\) in \(T_{BWT}\) (threshold for \(BWT\) in Eq.(2)), the letter “I”, the least black and white transitions in 26 English letters, is considered. The \(BWT\) of “I” is about 0.4 of the inscribed block area, and this number may become less after logical AND operation. Thus, \(\beta\) was set to be 0.35 in all of the experiments in this paper.

4.1 Evaluation Metrics
A detected box truly detects the texts if the area ratio \(r\), defined in Eq. (3), is at least 50%.

\[
r = \frac{\text{Area}(D\_BOX \cap G\_BOX)}{\text{Area}(D\_BOX \cup G\_BOX)},
\]

where \(D\_BOX\) is a detected box, like yellow boxes in Fig. 6, \(G\_BOX\) is the ground truth text box, like blue boxes in Fig. 6. The area ratios \(r\) of Fig. 6 are 40.0%, 46.4%, 61.1%, 74.3%, and 82.3% for (a), (b), (c), (d), (e), respectively. Thus the text is not detected in (a) and (b) (they are \(F\_BOX\)), and is detected in (c)–(e) (they are \(T\_BOX\)). To focus the bounding precision, if the area ratio \(r\) in Eq. (3) is at least 80% we say the text is accurately localized. Thus, in Fig. 6, only (e) is accurately localized (it is an \(A\_BOX\)). Accordingly, a \(D\_BOX\) may be a \(T\_BOX\) if \(r \geq 50\%\) (and \(A\_BOX\) if \(r \geq 80\%\)) or a \(F\_BOX\) if \(r < 50\%\) including false alarms, i.e., no text at all. We use recall \((R)\), precision \((P)\), and quality of bounding precision \((Q)\) to measure the efficacy of algorithms as in Eqs (4), (5), (6).

\[
R = \frac{\#(D\_BOX \cap T \_BOX)}{\#(G\_BOX)},
\]

\[
P = \frac{\#(D\_BOX \cap T \_BOX)}{\#(D\_BOX)} = \frac{\#(D\_BOX \cap T \_BOX)}{\#(D\_BOX \cap T \_BOX)+\#(D\_BOX \cap F \_BOX)},
\]

\[
Q = \frac{\#(Acu\_D\_BOX \cap T \_BOX)}{\#(D\_BOX \cap T \_BOX)},
\]

where \(D\_BOX, G\_BOX, T\_BOX, F\_BOX, A\_BOX\) are defined above.

![Image 311x257 to 553x277](image)

(a) (b) (c) (d) (e)
Fig. 6. Detected boxes (in yellow) for a ground truth text (in blue) that (a) and (b) fail to detect, (c)–(e) truly detect it but only (e) accurately matched.

4.2 Experimental Results and Discussions
Tables 1 and 2 depict the experimental results and some are shown in Fig. 7. The bottom text lines in (b), the left text column and the bottom text line in (c) are scrolling texts. In Fig. 7(b), the first character of the vertical text line on the right is intentionally connected to an image of diamond which is related to the content of the report. This causes the area ratio \(r\)
below 50% and thus it is considered as a $F$ _BOX_. In Fig. 7 (d), 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 alarm since “2” is detected in the other box. This happened in the duration of two rounds, and thus two false alarms are counted in Table 1 on ESPN video. Nevertheless, our proposed method reached 97.45%, 97.39%, and 95.97% on $R$, $P$, and $Q$. These figures are excellent compared to existing research. The results show that the detected boxes are all accurately localized. In Fig. 7(a) “THE” in “THE SITUATION ROOM” (on the lower right of the image) has a size of 11×7, in (c), two characters on the lower right (zui xin) have a size of 30×30 for each character, and “09:06” (below “zui xin”) has a size of approximately 10×10 for each digit. These cases demonstrate that the proposed algorithm can detect characters with sizes 0.5 ~ 1.5 times to the presumed character size $w \times h$ (20×20 in here). As the text polarity, in Fig. 7(c), there are dark texts in a high intensity background and white texts in a dark background (lower center area), and more examples can be found in (d). The proposed algorithm is also robust to length and alignment of text strings. There are correctly detected horizontal and vertical texts in one frame (Fig. 7(b)) and short text strings (“LIVE” on the upper left of Fig. 7(c)).

### 5 Conclusion

In this paper, we proposed a general text detection algorithm that is applicable to multilingual news videos without any constraints on text colors, fonts or sizes, alignments, or length of text strings. In our algorithm, logical AND operation on multiple Canny edge maps and a non-text line deletion technique together achieve very precise bounding boxes on detected texts. The algorithm has no complex calculation and is very efficient for it only processes 4 frames out of $f$ frames ($f$ is the frame rate per second). The proposed algorithm was tested on multilingual news videos (English, Japanese and Chinese) including different text polarities (positive

<table>
<thead>
<tr>
<th>Video Sources</th>
<th>Type/Language</th>
<th>Length min sec</th>
<th># of $G$ <em>BOX</em></th>
<th># of $F$ <em>BOX</em></th>
<th># of $T$ <em>BOX</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN News/English</td>
<td>9’16”</td>
<td>404</td>
<td>0</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>ESPN Sport/English</td>
<td>5’07”</td>
<td>111</td>
<td>2</td>
<td>11</td>
<td>29</td>
</tr>
<tr>
<td>NHK News/Japanese</td>
<td>5’35”</td>
<td>310</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ETTV News/Chinese</td>
<td>5’29”</td>
<td>939</td>
<td>0</td>
<td>0</td>
<td>27</td>
</tr>
<tr>
<td>TVBS News/Chinese</td>
<td>4’51”</td>
<td>1340</td>
<td>0</td>
<td>64</td>
<td>57</td>
</tr>
<tr>
<td>Total</td>
<td>--/--</td>
<td>3104</td>
<td>2</td>
<td>79</td>
<td>122</td>
</tr>
</tbody>
</table>

Table 1. Results on Different Video sources

<table>
<thead>
<tr>
<th>Channel</th>
<th>Recall</th>
<th>Precision</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>400/404 = 99.01%</td>
<td>400/404 = 99.01%</td>
<td>391/400 = 97.75%</td>
</tr>
<tr>
<td>ESPN</td>
<td>100/111 = 90.09%</td>
<td>100/113 = 88.50%</td>
<td>71/100 = 71.00%</td>
</tr>
<tr>
<td>NHK</td>
<td>310/310 = 100.00%</td>
<td>310/310 = 100.00%</td>
<td>310/310 = 100.00%</td>
</tr>
<tr>
<td>ETTV</td>
<td>939/939 = 100.00%</td>
<td>939/939 = 100.00%</td>
<td>912/939 = 97.12%</td>
</tr>
<tr>
<td>TVBS</td>
<td>1276/1340 = 95.22%</td>
<td>1276/1340 = 95.22%</td>
<td>1219/1276 = 95.53%</td>
</tr>
<tr>
<td>Average</td>
<td>3025/3104 = 97.45%</td>
<td>3025/3106 = 97.39%</td>
<td>2903/3025 = 95.97%</td>
</tr>
</tbody>
</table>

Table 2. Results on $R$, $P$, and $Q$
and negative), different character sizes (from 10x10 to 30x30), short text strings (only one word “LIVE” or logo “CNN”), 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. Among our experiments, the worst experimental result is the ESPN video due to small, isolated, and sparsely located figures. In fact, the localization for Fig.7(d) is much improved when the presumed character size $w \times h$ is set to be 10x10. Our future work will be working on an adaptation of the character sizes.

References: