AUTOMATIC LOCATION OF TEXT IN VIDEO FRAMES

Xian-Sheng Hua [1], *Xiang-Rong Chen* [2], *Liu Wenyin* [2], *Hong-Jiang Zhang* [2]

[1] National Laboratory on Machine Perception
Peking University, Beijing 100071, China
xshua@yahoo.com

[2] Microsoft Research China
No. 49 Zhichun Road, Beijing 100080, China
{wyliu, hjzhang}@microsoft.com

ABSTRACT

A new automatic text location approach for videos is proposed. First of all, the corner points of the selected video frames are detected. After deleting some isolate corners, we merge the remaining corners to form candidate text regions. The regions are then decomposed vertically and horizontally using edge maps of the video frames to get candidate text lines. Finally, a text box verification step based on the features derived from edge maps is taken to significantly reduce false alarms. Experimental results show that the new text location scheme proposed in this paper is accurate.

1. INTRODUCTION

The rapid growth of video data leads to an urgent demand for efficient and true content-based browsing and retrieving systems. In response to such needs, various video content analysis schemes using one or a combination of image, audio, and textual information in the videos have been proposed to parse, index, or abstract massive amount of data [1][2]. Among these information sources, text present in the video frames can provide important supplemental information for indexing and retrieval.

Many efforts have been done for text location in videos and images [2][3][4][5][6]. Current text detection approaches can be classified into two categories. The first category is connected component based methods, which can locate text quickly but have difficulties when text is embedded in complex background or touches other graphical objects [3]. The second category is texture classification based [4][5][6]. However, it is hard to find accurate boundaries of text areas and false alarms often exist in "text-like" texture areas. In addition, there are also several methods that try to locate text areas in the DCT compressed domain [2], but essentially they are texture-based schemes.

Our proposed text detection scheme is based on the observation that text regions typically are rich of corners and edges. We find candidate text regions by corner detection, refinement, and merging. The regions are then decomposed into single text lines using vertical and horizontal edge maps of the video frames. At last, we reduce some false alarms based on feature analysis of the detected text lines. **Figure 1** shows the flow chart of the proposed text location algorithm.

The rest of this paper is organized as follows. Section 2 introduces the corner detection algorithm, following by a detailed description on text bounding box extraction approach in Section 3, and on text box verification procedures in Section 4. Section 5 presents the experimental evaluation of the proposed approach and conclusion remarks are in Section 5.

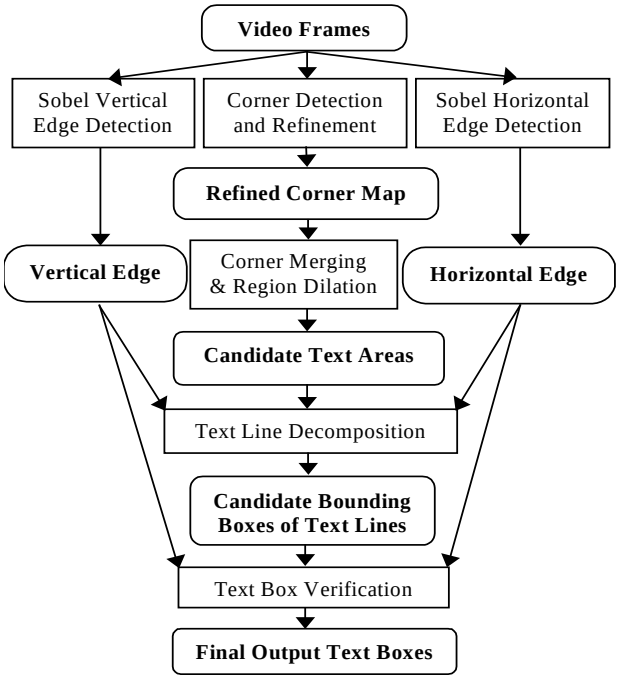


Figure 1. Flow chart of the proposed text location algorithm.

2. CORNER DETECTION

2.1 Corner Map of The Video Frame

It is observed that corners are rich in the text areas of video frames. "Corner" is a two-dimensional feature point in an image where the region boundary has high curvature [7]. **Figure 2** shows an example frame named "Coming Up". We apply the SUSAN corner detector [7] on this video frame to generate the corresponding corner map, shown in **Figure 3**. The black dots in **Figure 3** represent the corners, and the rectangle boxes illustrate the real text bounding boxes (ground truth). From **Figure 3**, we can also see that some isolate corners can be deleted because they are not "text corners", as to be shown in Section 2.2.

2.2 Corner Refinement

Since corners in the text area is nearly uniformly distributed, and generally text areas are rectangles, those corners that have few neighbor corners can be deleted.

Figure 2. Original video frame: "Coming Up".

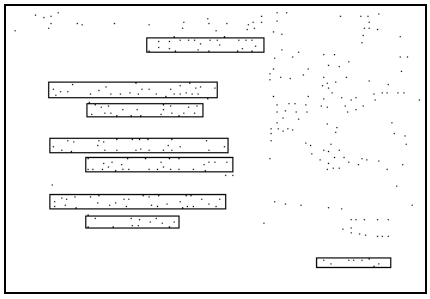


Figure 3. Corner maps of "Coming Up".

Denoting the number of corners in a $h \times w$ neighborhood area of corner P as $Corner(P, h, w)$, then the corners that totally satisfy the following constraints are considered as "text corners":

$$Corner(P, h_i, w_i) \geq n_i, \; i = 1, 2, 3. \quad (1)$$

In our experiments, we use the values in **Table 1** for these different parameters in the above constraints (These parameters are determined automatically, as to be explained in Section 5):

Table 1. Corner refinement parameters

i	h_i	w_i	n_i
1	5	10	2
2	5	15	3
3	5	50	6

The corners that do not satisfy one or more of the constraints will be deleted from the corner map. **Figure 4** shows the refined corner map. It is seen that there are still "non-text corners" in the complex background areas, which will be eliminated in later steps.

3. TEXT BOUNDING BOX EXTRACTION

3.1 Corner Merging and Region Dilation

In order to form candidate text regions, those corners that are close to each other horizontally or vertically are merged. The merging distance is determined by w_1, which is defined in Section 2.2. Suppose P and Q are two corners with coordinates in the corner map as (P_x, P_y) and (Q_x, Q_y) respectively. If

$$d(P, Q) = \min\left(|P_x - Q_x|, |P_y - Q_y|\right) \leq w_1, \quad (2)$$

all pixels in the rectangle determined by P and Q (**Figure 5**) are merged to generate the candidate text area.

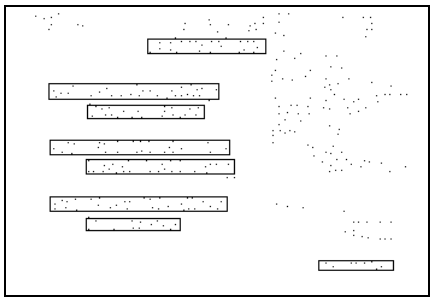


Figure 4. Refined corner map of "Coming Up".

Figure 5. If $d(P, Q) \leq w_1$, the entire shadowed area is part of a candidate text area.

For the merged area may be smaller than the actual one, it should be dilated several times. **Figure 6** shows the merged and dilated corner map, i.e., candidate text areas.

Figure 6. Candidate text areas: After merging and dilation.

3.2 Region Decomposition

To extract single text lines from candidate text areas, a vertical and a horizontal decomposition procedure using Sobel edge maps are preformed. Generally, regions are decomposed by analyzing the projection profiles of edge intensity maps [1][6]. In our scheme, we use an essentially similar method, but do not need to handle projection profiles. Furthermore, our decomposition method can obtain tighter and more accurate bounding boxes of the text areas (as to shown later in this section and Section 5), while texture-based (e.g., [2][5]) methods often get larger bounding boxes.

The Sobel vertical edge detector responses are scaled to [0, 255] and the edge threshold is 65 (i.e., those responses greater than 65 are edge points). We can get one or more line segments, which are denoted as l_i, i= 1, 2, …, when one horizontal scan line crosses a candidate text area, as shown in **Figure 7**(a).

Denote the number of vertical edge points in the top and bottom r lines of l_i (including l_i) as $ETop_v(l_i, r)$ and $EBtm_v(l_i, r)$, respectively, as shown in **Figure 7**(b). The length of l_i is denoted as $|l_i|$. We define the number ($EN_v(l_i, r)$) and the density ($ED_v(l_i, r)$) of the edge points in the $r \times l_i$ scan area as follows:

$$EN_v(l_i, r) = \max(ETop_v(l_i, r), EBtm_v(l_i, r)) \quad (3)$$

$$ED_v(l_i, r) = EN_v(l_i, r) / (|l_i| \times r) \quad (4)$$

If the line segment l_i does not satisfy one or more of the following constraints, it will be deleted from the candidate text area:

$$EN_v(l_i, r) \geq \min\_vedge\_number, \quad (5)$$

$$ED_v(l_i, r) \geq \min\_vedge\_density, \quad (6)$$

where r = 3, min_*vedge_number* = 15, min_*vedge_density* = 0.1 in our experiments.

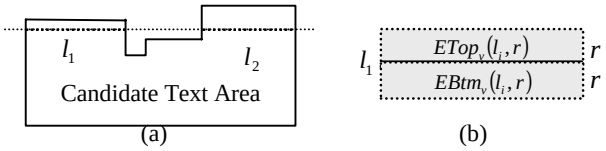


Figure 7. Candidate text area decomposition.

For horizontal decomposition, it is the same as vertical decomposition, except we use horizontal edge map and vertical scan lines. That is, the vertical line segment l_i will be deleted if it does not satisfy one or more of the following conditions:

$$EN_h(l_i, r) \geq \min\_hedge\_number, \quad (7)$$

$$ED_h(l_i, r) \geq \min\_hedge\_density, \quad (8)$$

where r = 8, min_*hedge_number* = 5, min_*hedge_density* = 0.05 in our experiments.

After several (4 in our experiments) iterations of decomposition vertically and horizontally, each area is then expanded to its bounding rectangle. **Figure 8** shows the text line decomposition result of **Figure 6**. It is seen that the candidate text lines are very close to the real text boxes. Though there are still false alarms, most of them can be eliminated in the next step.

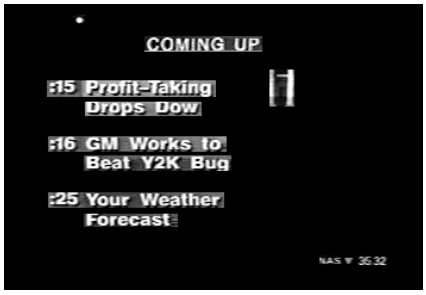


Figure 8. Candidate text lines of "Coming Up".

4. TEXT BOX VERIFICATION

A verification procedure is performed to remove false alarms from the text lines resulting from the procedure in Section 3. The method in [6] uses the following simple rules to remove some of the false alarms: (1) the height of text box should be larger than *min_text_height* and smaller than *max_text_height*; (2) the horizontal vertical aspect ratio must be larger than *min_hv_ratio*.

The above two rules can only filter out some non-text areas that is very small, or very large, or horizontal-vertical ratio is too small. Other false alarms still remain. We have derived a set of features from Sobel edge maps to further reduce false alarms, as described in this section.

4.1 Fill Factor Constraints

Suppose A is $d_x \times d_y$ candidate text box. The set of all edge points in A is denoted as $E(A,e)$, where $e \in \{h,v,a\}$, which represent horizontal edge, vertical edge and overall edge respectively. And $|E(A,e)|$ is the number of elements in $E(A,e)$. We check whether A satisfies the following constraints:

$$m \leq \frac{|E(A,h)|}{|E(A,v)|} \leq M \quad (9)$$

$$\frac{|E(A,e)|}{d_x \times d_y} \geq \min\_edge\_fill\_factor(e) \quad (10)$$

In our experiments, m = 0.5, M=2, min_*edge_fill_factor(e)* = 0.13, 0.2, 0.33 for $e = h$, v, and a, respectively.

It should be mentioned here that constraint (9) can filter out candidate text areas that contain a large number of horizontal edge points but few vertical edge points, or areas that contain lots of vertical edge points but few horizontal edge points.

4.2 Center Offset Ratio Constraints

Suppose edge point $P(P_x, P_y) \in E(A,e)$. The center offset ratio of the edge points in A is defined as follows:

$$COR(A,e,c) = \frac{\sum_{P \in E(A,e)} P_c}{|E(A,e)| \times d_c}, \quad (11)$$

where $e \in \{h,v,a\}$ denote the type of edge map, $c \in \{x,y\}$ denote the coordinate directions. The center offset ratio constraints is defined as follows:

$$COR(A,e,c) \leq \max\_center\_offset\_ratio(e,c), \quad (12)$$

where the values of max_*center_offset_ratio(e,c)* are assigned as in Table 2.

Table 2. Maximum value of center offset ratio.

c \ e	h	v	a
x	0.11	0.12	0.08
y	0.2	0.15	0.13

The candidate text lines that do not satisfy one or more of the constraints in (9), (10), and (12) will be regarded as false alarms. **Figure 9** shows the result after verification, which is the final output of the text detection approach proposed in this paper.

Figure 9. Final text detection result of "Coming Up".

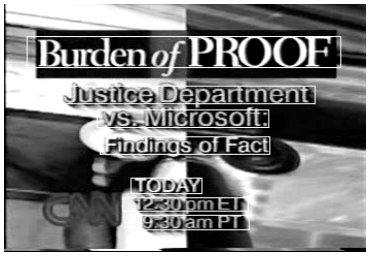
5. EXPERIMENTAL RESULTS

We evaluate our method using some CNN news videos. The total length of these videos is about 40 minutes. The testing data contain 90 video clips and each clip consists of 5 consecutive frames. The ground truths of these test data were manually labeled and recognized by us. The locations of the text lines and the corresponding real ASCII text strings in each frame are stored into a ground-truth file for automated evaluation. The evaluation result of the proposed text detection approach is listed in **Table 3**.

Table 3. Evaluation result of our text detection approach.

Frames	Text Lines	Detected	False Alarms
90	244	229 (93.85%)	18

Since usually the same text area will not change much over several consecutive frames, we can keep those detected text lines that have been detected in all the 5 frames as the final detection results, while others are regarded as non-text areas. The false alarms can be further reduced from 18 to 10 by this inter-frame verification procedure.



(a) Correct detection results in very complex background.

(b) Large and texture text was missed.

(c) False alarms in complex background.

Figure 10. More detection results.

Figure 10 shows some more detection results, including some examples that text areas are not correctly detected. We notice that most false alarms occur in the texture-rich video frames, while most missed text areas occur in the areas that have large or blurred text. This is because our method is based on corners and edges. There are lots of corners and edges in texture areas, while there are few corners and edges in the large or blurred text areas. If we adjust the parameters properly, false alarms can be remarkably reduced, while there is a slight increase in number of the missed text areas for some video frames.

Actually, the parameters determined in this paper are those that can yield the best results, determined by a comprehensive and objective performance evaluation using a protocol similar to [8]. However, this performance evaluation is not presented in this paper due to space constraint.

6. CONCLUSTION

In this paper, we have presented a new text location approach for news videos. This scheme is based on the observation that text regions typically are rich of corners and edges. Experimental results show that the new method is accurate. In addition, our method is expected to be more accurate for Chinese text detection, because there are more and denser corners and edges in Chinese text areas.

7. REFERENCES

[1] W. Qi, et. al. "Integrating Visual, Audio and Text Analysis for News Video". *7th IEEE International Conference on Image Processing (ICIP 2000)*, Vancouver, British Columbia, Canada, 10-13 September 2000.

[2] Y. Zhong, H. J. Zhang, A. K. Jain. "Automatic Caption Localization in Compressed video". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, pp. 385-392, April 2000.

[3] A. K. Jain, B. Yu, "Automatic Text Location In Images and Video Frames". *Pattern Recognition*. Vol. 31, No.12, pp. 2055-2076, 1998.

[4] V. Wu, R. Manmatha, E. M. Riseman. "Finding Text in Images" *20th Int'l ACM SIGIR Conf.Research and Development in Infromation Retrieval*, pp. 3-12.

[5] H.P. Li, D. Doermann. "Automatic Text Detection and Tracking in Digital Video". *IEEE Transactions on Image Processing*, Vol. 9, No. 1, January 2000.

[6] A. Wernicke, R. Lienhart. "On the Segmentation of Text in Videos". *IEEE Int. Conference on Multimedia and Expo (ICME2000)*, pp. 1511-1514, July 2000.

[7] S. M. Smith, J. M. Brady. "SUSAN - A New Approach to Low Level Image Processing". *International Journal of Computer Vision*. 23(1), pp. 45-78, May 1997.

[8] W.Y. Liu, D. Dori, "A Proposed Scheme for Performance Evaluation of Graphics/Text Separation Algorithms", *Graphics Recognition -- Algorithms and Systems*, eds. K. Tombre and A. Chhabra, *Lecture Notes in Computer Science*, Vol. 1389, pp. 359-371, Springer, April, 1998.