Text Area Detection from Video Frames

Xiangrong Chen, Hongjiang Zhang

Microsoft Research China
chxr@yahoo.com, hjzhang@microsoft.com

Abstract. Text area detection from video frame is an essential step for Video OCR. The key problem is the complex background of the video frames. This paper proposes a novel approach to this problem. First, we use the vertical edge information to detect candidate text areas. The horizontal edge information is then used to eliminate some of the false candidates. Finally, shape suppression technique is applied to further refine the results. Experimental results have shown the proposed approach is very effective in text area detection.

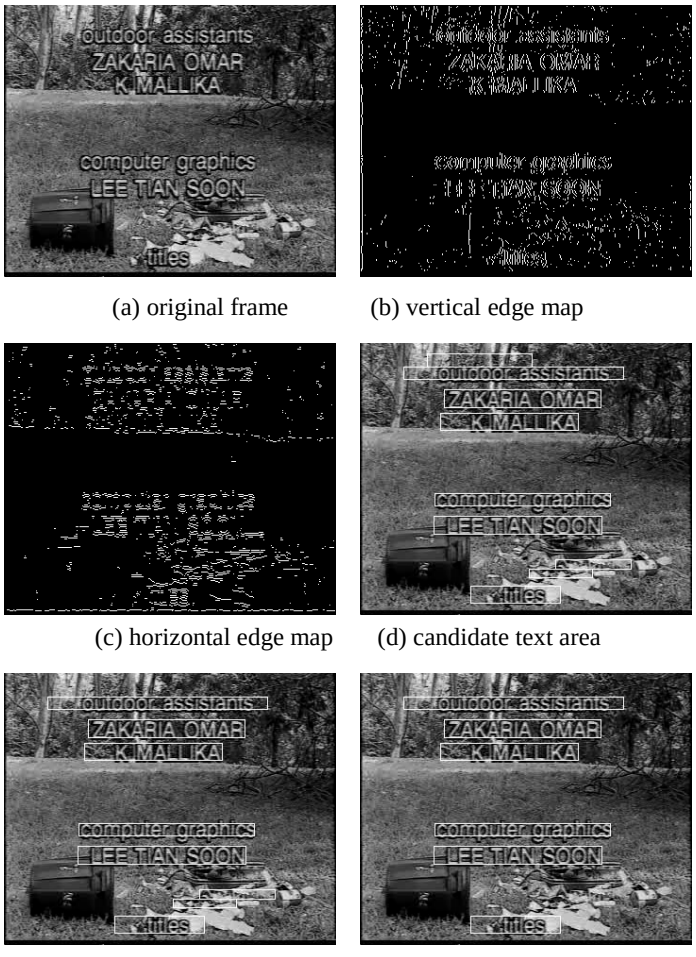
1 Introduction

Text embedded in video frames often carries the most important information, such as time, place, name or topics, etc. This information may do great help to video indexing and video content understanding. To extract text information from video sequence, which often referred as *video OCR*, the first essential step is to detect the text area in video frames. Comparing to traditional OCR application, the main difficulty to detect text area in video frames is complex backgrounds and low image resolution.

Fig. 1. Text regions detected by Li's method

There have been several published efforts in addressing the problem of text area detection in video. Li [1] used a hybrid wavelet/neural network segment approach based

on 16×16 pixels blocks. Zhong, et al, [2] located caption in compressed domain using 8×8 DCT coefficients blocks. The main flaw of these two block-based methods is their inaccurate area boundary, as shown in **Fig. 1**. Sato [3] applied a horizontal differential filter to the frame and detected text region that satisfies size, fill factor and horizontal-vertical aspect ratio constraints. However, this method also has no mechanism to constrain the boundary accurately. In addition, the above methods suffer when some non-text texture appears in the frame and thus increase the burden in the text recognition steps.



(a) original frame (b) vertical edge map

(c) horizontal edge map (d) candidate text area

(e) Horizontal edge alignment (f) shape suppression

Fig. 2. Examples of detection results by proposed method

In this paper, we propose a new approach to detecting text areas in video frames accurately and robustly in real time. As shown in **Fig. 2**, we first apply a horizontal and vertical Sobel differential calculator, followed by an edge thinning process on the original image, (a), to obtain a vertical edge map, (b), and a horizontal edge map, (c). From the vertical edge map (b), we obtain candidate text areas, shown as the white rectangles in (d). Then, by using horizontal edge alignment, false candidates can be eliminated, as shown in (e). Finally, we use a shape suppression technique based on Bayesian decision theory to avoid false candidates resulting from non-text texture areas, as shown in (f). Experimental results have shown the proposed approach is very efficient and accurate in text area detection.

The remaining of this paper is organized as following. In section 2, we describe the diagram and details of our proposed approach. Section 3 present the experimental evaluation of the approach and Section 4 conclude the paper.

2 Detect Text Area Using Edge Information

Typically, text regions in video frames are strongly textured. There are several methods to describe such textures introduced by text strings. We use edge information to characterize such textures in this paper. As most of the text strings in video align horizontally, we only address this case. **Fig. 3** shows the diagram of our proposed approach. However, the proposed approach only need to be modified slightly for detecting vertically aligned text string: to exchange the role of the horizontal edge map and the vertical edge map in Figure 3.

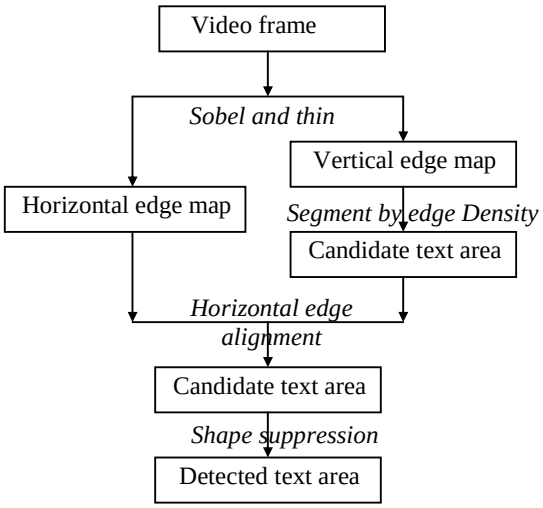


Fig. 3. Flow chart of the proposed method

2.1 Edge Map Generation

We apply a 3×3 horizontal and vertical Sobel filter on a video frame to obtain two edge maps of the frame: vertical and horizontal, as shown in **Fig. 3**. The Sobel operators are shown in **Fig. 4.** A non-maxima suppression is then used to thin the edges. Isolated edge points in the edge maps are then filtered out by a de-noising processing.

-1	0	1
-2	0	2
-1	0	1

(a) Horizontal

-1	-2	-1
0	0	0
1	2	1

(b) vertical

Fig. 4. Sobel operators

2.2 Finding Candidate Text Area

From the vertical edge map, we use edge density to find candidate text area. For a 352×288 PAL video frame, several heuristic rules are employed to embody the edge density: (1) Each scan line in the region should contain at least 6 edge points, (2) The edge density on each scan line in the region should be larger than 6 to 20, (3) The height of the region should be larger that 6 pixels. These heuristic rules can be adaptively modified for video frames of other systems and resolution.

2.3 Horizontal Edge Alignment Confirmation

For horizontal aligned text strings, there are many horizontal edges along the upper and bottom boundaries of the text area, as shown in **Fig. 2**(c). This observation motivates us to eliminate some false candidates by using of the horizontal edge alignment confirmation. In **Fig. 2** (e), the candidates resulted from tree trunk is cleared up by applying this method.

2.4 Shape Suppression

After the above process, there still remain two kinds of problems: false candidates and inaccurate left and right text area boundaries, both rising from non-text textures. These problems may increase the burden of the OCR procedures followed the text area detection. Therefore, it is desirable to do some preprocess to remove or at least reduce this problem. For this, we apply a shape suppression process based on Bayesian decision theory. The basic idea of shape suppression is to use shape information of character edge to reduce the effect of non-text texture. The probabilistic models required for the Bayesian approach are estimated with a Vector Quantization (VQ) framework [4].

VQ-based Bayesian Classifier

Considering n training samples from a class $\boldsymbol{c}$, a vector quantizer is used to extract m ($m<n$) codebook vectors, v_i ($1 \le i \le m$), from the n training samples. Given class $\boldsymbol{c}$, the class-conditional density of a feature vector $\boldsymbol{x}$, i.e., $\boldsymbol{f_X}(\boldsymbol{x}|\boldsymbol{c})$ can be approximated by a mixture of Gaussians with identity covariance matrices, each centered at a codebook vector, like below

$$\boldsymbol{f_X}(\boldsymbol{x}|\boldsymbol{c}) \propto \sum_{i=1}^{m} w_i * \exp(-\|x - v_i\|^2 / 2) \quad (1)$$

where w_i is the proportion of training samples assigned to v_i . The Bayesian classifier is then defined using the maximum a posteriori criterion as follows

$$\hat{c} = \arg\max_{c \in \Omega}\{p(c \mid x)\} = \arg\max_{c \in \Omega}\{p(x \mid c)\, p(c)\} \quad (2)$$

where $\Omega=\{c_1,c_2\}$ is the set of shape class and $p(c)$ represents the priori class probability.

Feature Selection

A sample in training the codebook can be each consecutive thin edge within the candidate region in vertical edge map. Each feature vector consists of four elements:

$$v = (N/H,\ D/H,\ V_N/H,\ H_N/H)^{\mathrm{T}} \quad (3)$$

where H is the pixel height of the candidate region; N the pixel number of the sample; D is the height of the gravity center of the edge according to the bottom of the candidate region; V_N and H_N the pixel number of the sample's vertical projection and horizontal projection, respectively.

Training Phase

In our method, c_1 represents text shape class and c_2 non-text shape class. Training samples of class c_1 are from the vertical edge map of all the characters, both upper and lower cases, and digitals. Training samples of class c_2 consists of two parts. One is from the real data selected manually. The other is from a "bootstrap" process like in method presented in [5]. That is, to add samples incorrectly classified as text shapes to the training sample set of non-text shape class. The codebook size for each training set is empirically set to 4.

3 Experimental Results

We have tested our method on 2 hours CNN TV programs and a 21 seconds segment of MPEG-7 test data. The data contains many different sources, including TV business

news, sport news, commercials, movies, weather reports, etc. With the proposed approach, over 95% of text regions have been detected correctly with a false detection rate lower than 5% in most cases except TV commercials. In TV commercials, characters in a same text region often vary to a large extent in both font and size, thus cannot pass the horizontal edge alignment confirmation. **Fig. 5** shows results of example video frames.

We implemented our algorithm using a Media SDK 6.0 on a DELL PIII-500 PC. The system can detect text region from MPEG1 file in real time, i.e., 25fps with 352×288 frame size.

(a) Business news (b) Sport news

(c) Weather report (d) Movie credit

Fig. 5. Experiment results

4 Conclusions

In this paper, we have presented an approach to text area detection from video frames. The approach consists of four main processes, including edge map generation, candidate extraction, horizontal edge alignment confirmation and shape suppression. Experimental have shown the proposed approach is robust and accurate. In addition, the computational complexity of the proposed approach is very low such that it can satisfy real-time applications.

At the present, the proposed approach operates on single video frames, thus it can also applied directly text area extraction of still images. To improve the performance of the proposed approach, we are actively investigating methods for integrating temporal information available from video sequences, such as tracking text areas across multiple frames.

References

1. Li, H.; Doermann, D.; Kia, O.: Automatic text detection and tracking in digital video, IEEE Trans. on Image Processing, 9(1) (2000) 147-156
2. Zhong, Y., Zhang, H., Jain, A.K.: Automatic Caption Extraction of Digital Videos, Proc. ICIP'99, Kobe, (1999) 24-27
3. Sato, T., Kanade, T.: Video OCR: Indexing digital news libraries by recognition of superimposed caption. ICCV Workshop on Image and Video retrieval (1998)
4. Gray R.M.: Vector Quantization, IEEE ASSP Magazine, 1(2) (1984) 4-29
5. Sung, K., Poggio, T.: Example-based learning for view-based human face detection. A.I.Memo 1521, CBCL Paper 112, MIT (1994)