Detection, Extraction and Segmentation of Video Text in Complex Background


Abstract – Automatic text detection in video is an important task for efficient and accurate indexing and retrieval of multimedia data such as events identification, events boundary identification etc. This paper uses method based on edge and connected component analysis, and performs quantification, analysis with the help of mathematical statistical method, then get the candidate text regions, combined with the feature of the text texture remove some non-text areas from candidate areas, and finally do binary processing and segment the text for better understanding in complex backgrounds and recognize the text with OCR software package.

Keywords – Connected Component Analysis (CCA), Detection, Extraction, Localization and Segmentation;

I. INTRODUCTION

With more and more digital devices, video has now become the most popular media type in our daily life. Since text embedded in video contains much semantic information related the video content, it plays an important role in content-based multimedia indexing and retrieval systems. Many text detection approaches have been proposed in the past several years; however, due to low resolution and complex backgrounds of videos and various sizes, colors, styles and alignments of text, text detection and extraction is still challenging.

Two types of text in video are: (1) caption/graphics/artificial text which is artificially superimposed on the video at the time of editing, and (2) scene text which naturally occurs in the field of view of the camera during video capture. Clearly, the detection of scene text is a challenging task due to varying lighting, complex movement and transformation.

Generally, text extraction in videos can be divided into the following stages: 1) Text detection, finding regions in a video frame that contain text; 2) Text localization, grouping text regions into text instances and generating a set of tight bounding boxes around all text instances; 3) Text tracking, following a text event as it moves or changes over time and determining the temporal and spatial locations and extents of text events; 4) Text binarization, binarizing the text bounded by text regions and marking text as one binary level and background as the other; 5) Text recognition, performing OCR on the binarized text image. Occasionally binarization step is eliminated in favor of applying OCR on color/gray level images. Video Text Retrieval System is shown in figure 1 below, which includes the above steps mentioned.

Text detection and extraction in video is usually addressed by the following methods: (1) method based on edge extraction; it can quickly locate the text area, there is relatively high accuracy if video frame contains strong edge information; (2) method based on texture, which is more versatile than the other methods, but it usually performs FFT, wavelet transform, so it must be time-consuming, poor in real-time performance; (3) method based on time-domain characteristics, it use the appearance and disappearance of video caption text to detect text area, because the appearance and disappearance of video caption text can cause the change of the gray value in text area, but no change in the non-text area. This method is limited in many occasions, e.g. the camera shake strongly addition text is moving regularly. But this paper uses method based on edge and connected component analysis. Next step is the segmentation process, the localized text is segmented

Fig 1: Video Text Retrieval System

ISSN (Print) : 2319 – 2526, Volume-2, Issue-5, 2013
into lines, where the multiple line lines are segmented into single line text, then the characters from the line are segmented using connected component analysis based method. If any touching characters occur in the detected text line, then that line is further processed for morphological operations.

II. RELATED WORK

Video text detection methods can be classified into two classes. The first class treats text as a type of texture. These methods usually divide a whole image into blocks. They first use various approaches, e.g., Gabor filter [1], spatial variance[2], or wavelet transform [3], to calculate the texture features of blocks. Then they employ proper classifiers, e.g., neural network [3] or a support vector machine [4], to classify textblocks and nontext blocks. Some methods [5], [6] assume that the text strokes have a certain contrast against the background. Therefore, those areas with dense edges are detected as text regions. Textures can also be detected in the MPEG-2-compressed domain. Gargi et al. [7] proposed to use the number of intracoded blocks in each P- and B- frame to detect the appearance of captions. However, the texts appearing in I-frames are not handled. Zhong et al.[8] and Lim et al[8] proposed to utilize DCT coefficients to detect text regions. The second class [9]-[11] assumes that a text string contains a uniform color. Thus, at first they perform color reduction and segmentation in some selected color channel ([10] only used the red channel) or color space (Lab space was chosen in [11]), and then they perform connected-component analyses to detect text regions.

Generally, there are two paradigms for the text localization: bottom-up paradigm and top-down paradigm. The bottom-up paradigm groups small text regions into entire text strings based on some heuristic rules and geometric constraints. Region growing is a conventional technique for this purpose. Wu et al. [2] proposed a non-region-based approach, which uses edges in the text regions to form strokes, and then strokes are aggregated to form chips (text strings). The top-down paradigm is based on splitting the image regions (the initial region is the whole image) alternately in horizontal and vertical directions based on the texture, color, or edge distribution. Gao and Tang [12] proposed to use horizontal and vertical projections of edges to localize text strings; however, it can only handle captions and cannot deal with complex text layouts; however, they obtained initial bounding boxes by region growing. Therefore, if the initial bounding boxes are over-segmented, which happens for Chinese texts, there will be no chance to recover the correct text regions.

The text extraction methods fall into two groups. One group includes color-based methods, and the other includes stroke-based methods. The former holds the assumption that the text pixels are of different color from the background pixels, so that they can be segmented by thresholding. Besides the validity of the assumption, another difficulty in this approach is the color polarity of text, i.e., light or dark, must be determined. Antani et al. [14] proposed a color-based method that first generates two segmented results in both polarities for each text string and then selects the one with a higher score on text-like characteristics as the final result. The stroke-based methods, on the other hand, employ some filters to output only those pixels likely on the strokes to the final results, such as the asymmetric filter, the four-direction character extraction filter [5], and the topographical feature mask. These filters are intended to enhance the stripe (i.e., stroke-like) shapes and to suppress others; however, the intersection of strokes may also be suppressed due to the lack of stripe shape.[17], [18] proposed the wavelet transform technique and gradient features for text extraction respectively.

III. PROPOSED METHOD

Figure 3.1 shows the architecture of the proposed system. The proposed architecture is divided into two modules namely Text Detection / Extraction and Segmentation.

![Fig. 3.1: Overall Architecture of the Proposed Model](image)

Video

Video Frames

Text Detection

Segmentation

Line Segmentation

Character Segmentation

Video is the system input to the top layer of the architecture, and then the video is converted into video...
frames. In the first module of architecture, text detection process will detect text present in video. In second module, segmentation of video text presents in the video frame obtained after text detection is done. In proposed architecture, segmentation is divided into line segmentation and character segmentation.

A. Text Detection

Text detection and extraction in video is usually addressed by three main approaches namely, connected component based, texture based, edge and gradient based. These methods solve the problem to some extent but still there is room for improvements especially for large datasets containing both graphics and scene text. Recently, integrating wavelet and color features is new way for text detection in video. Thus, we take advantage of color and wavelet decomposition as color of text component usually will have uniform color but not in contrast. Wavelet decomposition generally enhances the high contrast pixels by suppressing low contrast pixels.

Text detection in video frame is done by method comprising of wavelet decomposition and color features namely R, G and B. The wavelet decomposition is applied on three color bands separately to obtain three high frequency sub bands (LH, HL and HH) and then the average of the three sub bands for each color band is computed further to enhance the text pixels in video frame. To take advantage of wavelet and color information, we again take the average of the three average images (AoA) obtained by the former step to increase the gap between text and non text pixels. Proposed method assumes text lines in video are in horizontal direction. In Figure 3.4 shows the wavelet and color features images or results.

After obtaining wavelet and color features, we use Laplacian method for text detection. In first step, the input image is converted to grayscale and filtered by a 3 × 3 Laplacian mask to detect the discontinuities in four directions: horizontal, vertical, up-left and up-right. Because the mask produces two values for every edge, the Laplacian-filtered image contains both positive and negative values. The transitions between these values correspond to the transitions between text and background. In order to capture the relationship between positive and negative values, we use the maximum gradient difference (MGD), defined as the difference between the maximum and minimum values within a local 1 × N window.

The MGD map is obtained by moving the window over the image. Text regions typically have larger MGD values than non-text regions because they have many positive and negative peaks. Therefore, we normalize the MGD map to the range [0, 1] and convert the frame into binary frame, then Fuzzy C-means is applied to classify the feature into two clusters: background and text candidates, this is the final step for text detection. After text detection in the video frame, next step is to segment the text detected. Following figure 3.2 shows the text detection steps carried out in video text detection procedure.

B. Segmentation

Text segmentation and recognition in images and video frames, which aims at integrating advanced optical character recognition (OCR) and text-based searching technologies, is now recognized as a key component in the development of advanced image and video annotation and retrieval systems. Unfortunately, text characters contained in images and videos can be any gray-scale value (not always white), low-resolution, variable size and embedded in complex backgrounds. Experiments show that applying conventional OCR technology directly leads to poor recognition rates. Therefore, efficient segmentation of text characters from the background is necessary to fill the gap between image and video documents and the input of a standard OCR system. Segmentation of a video text into its basic entities namely text lines and characters, is a critical stage towards recognition. The difficulties that arise in video text make the segmentation procedure a challenging task. There are many problems encountered in the segmentation, these include the difference in the skew angle between lines, characters or even along the same text line, overlapping words and adjacent text lines and touching characters.
(i) Line Segmentation

In line segmentation, the text detected video frame obtained from text detection is taken as input, then make use of mathematical statistical method based on projection profiles to segment lines in video text. In horizontal and vertical projection profile, we scan each pixel value from top to bottom and then from left to right, i.e., row by row and column by column respectively, and count the number of white spots in a row and column, so we can get the horizontal and vertical projection of the whole image. Projection is a characteristic function, which changes two-dimensional distribution of pixels into two one-dimensional functions (x-axis and y-axis).

After obtaining horizontal and vertical projection of the image, compute top, bottom, left and right of the text line based on projection profile analysis so only the line of text is to be segmented, no other noises or non text regions are part of the segmentation. This process is repeated until all the lines of text in a frame or image are segmented. Line segmentation procedure is shown in figure 3.3.

(ii) Character Segmentation

Character segmentation is done by using Connected Component Analysis (CCA) and Vertical Projection Profile as shown in figure 3.4. Connected component analysis is done, based on the "Bounding Box" (BB), i.e., the location and dimensions of each connected component. Vertical projection profile is carried out when the connected component fails to segment a character.

In character segmentation first, we take a line of text obtained from line segmentation as input, then perform preprocessing operations like smoothing and then the line of text is processed for connected component analysis, where line of text is segmented into characters. Then we compute width of each bounding box and compute average of all the bounding boxes which contains characters. Then we compare each and every bounding box with the average value, if all the
bounding boxes are equal to the average width, then it is clear that all characters in the line are segmented correctly, else if the box violates the average width, that particular box is processed to next step called vertical projection profile analysis. This step is to verify that all bounding boxes contains single character only, no bounding box can contain more than one character. The vertical projection of a line consists of a simple running count of the black pixels in each column. Vertical projection profile analysis in our method works like computing each pixel value from top to bottom of the image vertically, when there exists pixel value less than the threshold value, perform segmentation at that particular location vertically.

Consider an image contains a word “TEXT”, first we compute the pixel values from top to bottom of the image vertically, then the vertical projection profile method first encounters the character “T”. When the scan is performed, pixels representing “T” has more value than other pixels, so when there exists a low pixel value compared to threshold continuously, we perform segmentation at that particular location like this same procedure is repeated until all the characters in the frame or in a bounding box are segmented. Hence the characters which failed in connected component analysis are segmented in vertical projection profile analysis.

IV. EXPERIMENTAL RESULTS

We create our own database as there is no benchmark data for text detection in video. The created database consists of 772 video frames, collected from different sources such as news containing graphics, scene text, low contrast text, complex background, different scripts, different fonts, colors, and font sizes. The dataset includes news of sports in which both scene text may appear with distortion, news of web source in which small font and low contrast text may appear with perspective distortion, news of business in which text may appear with different formats and layout and different colors, and finally news of scene text in which text appears with severe perspective distortion and complex background.

The following snapshots show the candidate text region generation. The red color box shows the generated text regions.

Fig. 4.1: Candidate Text Region Generation.

The following snapshot shows the line segmentation in which, two lines of text exist in a video frame are segmented separately.

Fig. 4.2: Detected image
The following snapshot shows the character segmentation in which, a line of text obtained from line segmentation is segmented into characters separately. As shown in Figure 4.4, the input line “HANDICRAFT” consists of ten characters, after applying character segmentation methods, all the characters exist in the line are segmented separately.

**Metrics for Evaluation:** The detection rate, false positive rate and misdetection rate as decision parameters and metrics in our experiments. The detected text blocks are represented by their bounding boxes. To judge the correctness of the text blocks detected, manually count Actual Text Blocks (ATB) in the frames in the dataset [17]. Also manually label each of the detected blocks as one of the following categories:

- **Truly Detected text Blocks (TDB):** a detected block that contains text fully or partially.
- **Falsely Detected text Blocks (FDB):** a detected block that does not contain text.
- **Text block with missing data (MDB):** a truly detected text block that misses some characters.

Based on the number of blocks in each of the categories mentioned above, the following metrics are calculated to evaluate the performance of the approaches:

- **Detection rate (DR)** = Number of TDB / Number of ATB.
- **False positive rate (FPR)** = Number of FDB / Number of (TDB + FDB).

Misdetection rate (MDR) = Number of MDB / Number of TDB.

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>QUANTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTF</td>
<td>416</td>
</tr>
<tr>
<td>TDB</td>
<td>3913</td>
</tr>
<tr>
<td>FDB</td>
<td>1611</td>
</tr>
<tr>
<td>MDB</td>
<td>382</td>
</tr>
<tr>
<td>ATB</td>
<td>4068</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Frames</td>
<td>100</td>
</tr>
<tr>
<td>Actual Text Lines</td>
<td>257</td>
</tr>
<tr>
<td>Correctly Segmented Text Lines</td>
<td>239</td>
</tr>
<tr>
<td>False Segmented Text Lines</td>
<td>98</td>
</tr>
<tr>
<td>Miss Segmented Text Lines</td>
<td>42</td>
</tr>
<tr>
<td>Efficiency</td>
<td>92.99%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Lines</td>
<td>100</td>
</tr>
<tr>
<td>No. of Characters</td>
<td>1040</td>
</tr>
<tr>
<td>Correctly Segmented Characters</td>
<td>920</td>
</tr>
<tr>
<td>False Segmented Character</td>
<td>70</td>
</tr>
<tr>
<td>Miss Segmented Characters</td>
<td>180</td>
</tr>
<tr>
<td>Efficiency</td>
<td>88.5%</td>
</tr>
</tbody>
</table>

**V. CONCLUSION**

The proposed text detection and segmentation methods have been tested on a number of real-life video clips. We use a set of quantitative evaluation measurements for the text detection, segmentation and recognition methods. The system is designed in such a
way that, the text in the video frame is detected and segmented automatically.

In proposed system, text detection makes use of Laplacian method based on wavelet and color features. The MGD map is obtained by moving the window over the image. Text regions typically have larger MGD values than non-text regions because they have many positive and negative peaks. Convert the input frame into binary frame and then Fuzzy C-means is applied to classify the feature into two clusters: background and text candidates.

Line segmentation is done by using horizontal projection profile and vertical projection profile analysis. Character segmentation is done by using Connected Component Analysis (CCA) and Vertical Projection Profile Analysis. Segmenting lines and characters correctly improves the accuracy of OCR.

Experiments and results show that, this application yield 92% of detection rate for text detection approach, 92.99% efficiency for line segmentation and 88.5% efficiency for character segmentation.

Further work can be carried out for the following issues: This application resides on its assumption that all text is oriented in the same direction, which is by default horizontal, modification is needed to cope with more sophisticated cases such as vertical and multi-oriented text lines. The small font size text presents in the video frame can be segmented. And the text present in video frame is sometimes skewed; in future, line and frame can be segmented. And the text present in video text lines. The small font size text presents in the video sophisticated cases such as vertical and multi-horizontal, modification is needed to cope with more sophisticated cases such as vertical and multi-oriented text lines. The small font size text presents in the video frame is detected and segmented automatically.

VI. REFERENCES


ISSN (Print) : 2319 – 2526, Volume-2, Issue-5, 2013

123