EXTRACTION OF TEXT FROM VIDEO CLIPS

1DEEPIKA SOOD, 2BALJIT SINGH

Dept. of CSE, B.B.S.B.E.C
Email: deepikasood07@yahoo.com, baljit.singh@bbsbec.ac.in

Abstract: Text in video is very compact and accurate clue for video indexing and summarization. In this paper, an algorithm is designed that the new Fourier-Statistical Features (FSF) in RGB space for detecting text in video frames of unconstrained background, different fonts, different scripts and different font sizes. This work consists of two parts namely automatic classification of text frames from a large database of text and non-text frames and FSF in RGB for text detection in the classified text frames. For text frame classification, presents novel features based on three visual cues; Max-Min method and sharpness in filter-edge maps to identify a true text frame. For text detection in video frames, presents the new Fourier transform based features in RGB space with statistical features and the computed FSF features from RGB bands are subject to Fuzzy C-means clustering to classify text pixels from the background of the frame. Text blocks of the classified text pixels are determined by analyzing the projection profiles and extract the text part from the video frame.

Keywords- FSF, Fuzzy C-means clustering, text detection, text frames classification.

I. INTRODUCTION

Nowadays, videos are playing a more important role in normal life. The rapid increase of digital video databases has led to the demand for user to query and index their interesting content efficiently and accurately. Manual annotation of video is greatly time consuming, expensive and unsuitable in the face of enormous video database. Most video text detection and extraction methods hold assumptions on text color, background contrast and font style. Moreover, few methods can handle multilingual text well since different languages may have quite different appearances. Text in video is a very compact and accurate clue for video indexing and summarization. The texts in video frames provide highly condensed information about the content of the video and it is helpful for video skimming, browsing and retrieval in large video databases. The increasing availability of online digital images and videos has rekindled interest in the problems of how to index multimedia information sources automatically, how to browse and how to manipulate them efficiently and accurately. Traditionally, images and video sequences have been manually annotated with a small number of keyword descriptors after visual inspection by a human reviewer. Unfortunately, manual annotation can be time-consuming, expensive and unsuitable in the face of enormous video database. At high level, text can be divided into two classes: Scene text and Graphic text. Scene text appears within the scene which is captured by the recording device. It is an integral part of the image and can be considered as a sample of the real world. Graphic text, on the other hand is a text that is mechanically added to video frames to supplement the visual and audio content and is often more structured and closely related to the subject than scene text. Studies on semantic image content in the form of text, face, vehicle, and human action have attracted some recent interest. Among them, text within a video frame is of particular interest as (i) it is very useful for describing the contents of a video; (ii) it can be easily extracted compared to other semantic contents, and (iii) it enables applications such as keyword-based frame search, automatic video logging, and text-based video indexing. This paper addresses this problem of detecting and extracting text from video. There are usually the following methods to extract the text in video: (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, it usually performs FFT, wavelet transform; (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. There are numerous applications of a text extraction system, including document analysis, vehicle license plate extraction, technical paper analysis and object-oriented data compression. To overcome the problem of complex background, texture based approaches are proposed: 1) To classify the text and non-text frames; 2) Apply Max-Min method to classify the text blocks; 3) Apply the novel features based on visual cue: Sharpness for classify the text blocks; 4) To detect and extract text only from classified text frames using the Fourier-Statistical Features (FSF) in RGB-space.
II. PROPOSED SYSTEM

Now will explain two stages in section II-A and II-B. Section II-A explain classification of text frames using 1) Max-Min method and 2) Visual Cue: sharpness of the edges. Section II-B then explain FSF in RGB to detect and extract text with the help of fuzzy C-means clustering. Figure 1 shows the overall block diagram of the proposed system. The video is a system input to the top layer which is converted into a video frames and resize each frames. For each frame it divides a given frame of 256×256 or 512×512 into equally sized non-overlapping blocks. The division of blocks aims to quickly find text features at the block level rather than the frame level, to further speed up, uses visual cue of straightness of the edges in the blocks to quickly classify the blocks into text blocks and non-text blocks before confirming the identity of a true text frame. If there is one or more blocks are classified as text blocks then that frame is considered as a text frame otherwise non-text frame shown in Fig. 2 (a) and (b). The classified text frames are given as input to the second stage. In the second stage it is accomplished by the use of Fourier Transform (FT) statistical features in the RGB space. 2D FFT is used to detect text present in the input text frame. I.e. in this, FFT is used on the three bands R, G and B for text detection purpose. Then apply Inverse Fourier Transforms (IFFT); by this it filter out the low frequency components from the reconstructed frame. Apply normalization for absolute values from IFT, and take average of RGB band. Automatically calculate threshold value by using co-efficient values of FT, and apply Maximum gradient method and finally extract the text part from the frame.

A. Max-Min Method to Classify the Text Frames

In Max-Min method max means 1 and min means 0. The straightness and cursiveness of the edges are computed as follows.

Let X = \{x_1, x_2, x_3, ..., x_n\} and Y = \{y_1, y_2, y_3, .... , y_n\} be the sets of x and y co-ordinates respectively, of the Sobel edge pixels. The straightness and cursiveness of an edge are defined as

\[ S_{-\text{Edge}} = \begin{cases} 1, & (C_x \in X) \cap (C_y \in Y) \\ 0, & \text{otherwise} \end{cases} \]  

Where \( C_x \) and \( C_y \) are the centroid of an edge, is defined as

\[ C_x = \frac{1}{n} \sum_{i=1}^{n} x_i \]  

And

\[ C_y = \frac{1}{n} \sum_{i=1}^{n} y_i \]  

Where n is the number of pixels in the edge. If \( S_{-\text{Edge}} = 1 \), the edge is considered a straight edge. otherwise, it is a cursive edge. Let \( N_{S-\text{Edge}} \) be the number of straight edges for each blocks. As Fig. 3 shows straightness for text-1 and non-text-7 block. We normalize this number between 0 and 1 for ease of comparison. Let \( NUM = \{n_1, n_2, n_3, ..., n_j\} \) be the set of normalized values, where j is the number of blocks.

The average of maximum and minimum value in normalized set is:

\[ \text{Average} = \frac{\max(\text{NUM}) + \min(\text{NUM})}{2} \]  

\[ \text{Text Block} = \begin{cases} 1, & \text{if}(n(j) \geq \text{Average}) \\ 0, & \text{otherwise} \end{cases} \]  

Where j is varies from 1 to number of blocks. Let \( C_1 \) be the Text block classified from Max-Min method.

![Fig 2. Text blocks and non-text blocks (a) Input (b) Blocks](Image)
2) Visual Cue: Sharpness of the Edges

Sharp edges are a distinctive feature of text versus non-text. The arithmetic mean filter is the simplest among different mean filters. Let \( W_x \) be the sub-image window size of \( m \times n \), and center point is \((x, y)\). The arithmetic mean filtering process computes the average value of the frame as follows:

\[
f_{\text{AMF}}(x,y) = \frac{1}{mn} \sum_{(s,t)\in W_{xy}} g(s,t)
\]

The best known order statistics filter is the median filter; the median filter replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel.

\[
f_{\text{MedF}}(x,y) = \text{median}_{(s,t)\in W_{xy}} \{g(s,t)\}
\]

The Arithmetic Filter (AMF) is used to blur the block and the Median Filter (MedF) to remove noise pixels. The Sobel and Canny operators are used to study the effect of filters on text and non-text blocks by counting the number of components in the filtered blocks. Fig. 4 shows (a) AMF (b) MF and (c) Diff of both of these on text block-1. Let \( \text{Diff}(x,y) \) be the difference between Median filter and Arithmetic filter. The Sobel edge operation is applied on \( \text{AMF}(x,y) \), i.e. \( \text{Sobel}_{\text{AMF}}(x,y) \) and the Canny edge operation is applied on \( \text{Diff}(x,y) \), i.e. \( \text{Canny}_{\text{Diff}}(x,y) \). The steps of filter operation for classifying text and non-text blocks are given below.

Let \( \text{Text Block} \) be the input block from previous method, then operations shown below.

\[
\begin{align*}
\text{AMF}_{\text{Text Block}}(x,y) &= f_{\text{AMF}}(\text{Text Block}(x,y)) \\
\text{Sobel}_{\text{AMF}}(x,y) &= \text{Sobel}(\text{AMF}_{\text{Text Block}}(x,y)) \\
\text{MedF}_{\text{Text Block}}(x,y) &= f_{\text{MedF}}(\text{Text Block}(x,y)) \\
\text{Diff}(x,y) &= \text{MedF}_{\text{Text Block}}(x,y)-\text{AMF}_{\text{Text Block}}(x,y) \\
\text{Canny}_{\text{Diff}}(x,y) &= \text{Canny}(\text{Diff}(x,y))
\end{align*}
\]

Let \( g(x,y) \) be the frame of \( B \)-band.

\[
\begin{align*}
\text{FFT}^B(x,y) &= \text{FFT}^B(g(x,y)) \\
\text{IFFT}^B(x,y) &= \text{IFFT}(\text{FFT}^B(x,y)) \\
\text{AIFFT}^B(x,y) &= |\text{IFFT}^B(x,y)|
\end{align*}
\]

Where, AIFT is the absolute values in IFFT.

\[
\text{NIFTT}^B(x,y) = \frac{\text{AIFFT}^B(x,y)}{\max(\text{AIFFT}^B(x,y))}
\]

The Sobel operator gives more edges in comparison with the Canny operator for the text block, for non-text block the Canny operator gives more edges in comparison with the Sobel operator.

Let \( \text{NUM}_{\text{Sobel}_{\text{AMF}}} \) and \( \text{NUM}_{\text{Canny}_{\text{Diff}}} \) be the number of edges in \( \text{Sobel}_{\text{AMF}}(x,y) \) and \( \text{Canny}_{\text{Diff}}(x,y) \), respectively.

\[
C_{\text{T}} = \begin{cases} 
\text{Text Block} & \text{if} \left( \text{NUM}_{\text{Sobel}_{\text{AMF}}} > \text{NUM}_{\text{Canny}_{\text{Diff}}} \right) \\
\text{Non Text Block} & \text{otherwise}
\end{cases}
\]

\( C_{\text{T}} \) be the Captured Text by the Sharpness of the Edges method. Fig. 5 shows classification based on cue on block-7. Let \( C_2 \) be the classified text blocks from Visual Cue: Sharpness of the Edges method.

B. Fourier Transform-Statistical Features in RGB Space for Text Extraction

This section describes the second stage of the proposed system. The input to the second stage is a text frame supplied by the first stage. In the second stage it is accomplished by the use of Fourier Transform (FT) statistical features in the RGB space. We use 2D FFT to detect text present in the input text frame shown in Fig. 6(a)-(f). I.e. in this, we use FFT on the three bands R, G and B for text detection purpose. Then we apply Inverse Fourier Transforms (IFFT); by this it filter out the low frequency components from the reconstructed frame.

Let \( g(x,y) \) be the frame of \( B \)-band.

\[
\begin{align*}
\text{FFT}^B(x,y) &= \text{FFT}^B(g(x,y)) \\
\text{IFFT}^B(x,y) &= \text{IFFT}(\text{FFT}^B(x,y)) \\
\text{AIFFT}^B(x,y) &= |\text{IFFT}^B(x,y)|
\end{align*}
\]

Where, AIFT is the absolute values in IFFT.

\[
\text{NIFTT}^B(x,y) = \frac{\text{AIFFT}^B(x,y)}{\max(\text{AIFFT}^B(x,y))}
\]
Where, NIFFFT is the Normalized values by dividing AIFFFT from maximum value of AIFFFT of the respected band.
The average band frame is calculated as
\[
\text{Avg}_{\text{RGB}}(x,y) = \frac{1}{2} \sum_{i=1}^{3} \text{NIFFT}_i(x,y)
\]
For RGB band i is from 1 to 3.
The Fuzzy C-Means algorithm is applied to classify the feature into two clusters: background and text candidates. Since Fuzzy C-Means is unsupervised, we propose to use the cluster’s mean as a basis of classification. The cluster that has the higher mean is classified as text. This is so because text pixels will have high contrast compared to the background pixels and contribute to the higher mean feature values. The result of Fuzzy C-Means algorithm; where the text cluster is shown in white against the background cluster shown in black. The text cluster is subjected to morphological operations, namely opening and dilation to get connected components and to reclassify too small objects into the background cluster.

III. EXPERIMENTAL RESULTS

The results are to illustrate that the algorithm can automatically classify text and non-text frames, detect and extract the text from classified text frames. For experimental purpose created our own dataset. In this dataset, there is variety of video frames, including frames taken from movies, news clips containing some scene texts.

A. Experimental Results on Classification of Text Frames

In the Table I No.F refers to the number of frames, TP true positive and FP false positive. We have selected 903 video frames which include 772 text frames and 131 non-text frames, which give 5462 text blocks of size 64x64 and 8980 non-text blocks. The approach implemented using MATLAB software is run on a PC with Pentium 1V 2.33 GHz processor. The approximate processing time for each video frame of size 256x256 is about 2.9 seconds for classifying a text frame. The actual processing time depends on the data structure and platform used by the approach. The proposed integrated approach is found indeed to be independent of font, contrast, font size, language, orientation and application. Its text frame identification capability can be deployed in event identification, exact event boundary identification and script identification. Performance: The performance of the system is evaluated at the frame level in term of Precision and recall rate. The precision and Recall Rate are given below.

The Recall Rate is defined as:
\[
\text{Recall Rate} = \frac{\text{True Positive}}{\text{Number of Frames}}
\]
The Precision Rate is defined as:
\[
\text{Precision Rate} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}
\]

<table>
<thead>
<tr>
<th>Data</th>
<th>No.F</th>
<th>TP</th>
<th>FP</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>772</td>
<td>732</td>
<td>41</td>
<td>94.82</td>
<td>94.69</td>
</tr>
<tr>
<td>Non-text</td>
<td>131</td>
<td>126</td>
<td>7</td>
<td>96.12</td>
<td>94.73</td>
</tr>
</tbody>
</table>

C. Experimental Results on Text Detection in Frame

In this section, Table II and III refers to results and performance for all text frames of text detection. In this experiment, selected 903 video frames from the above said sources which give 5462 actual number of text blocks and 772 actual number of text frames. Collected frames from different sources such as news containing graphics, scene text, low contrast text, complex background, different scripts, different fonts, colors, and font sizes. As Fig. 7 shows (a) Text image (b) result of intermediate image (c) detected the text region and (d) Extracted text region. 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.

1) 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. Also manually label each of the detected blocks as one of the following categories:
   1) Truly Detected text Blocks (TDB): a detected block that contains text fully or partially.
   2) Falsely Detected text Blocks (FDB): a detected block that does not contain text.
3) 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:

1) Detection rate (DR) = Number of TDB / Number of ATB.
2) False positive rate (FPR) = Number of FDB / Number of (TDB + FDB).
3) Misdetection rate (MDR) = Number of MDB/ Number of TDB.

CONCLUSION AND FUTURE SCOPE

In this paper we addressed two complex issues they are: Automatic classification of text and non-text frames without any constraints and Text detection, extraction in complex video frames. The text frame classification approach is based on the observation of sharp edges, straight appearances of edges and consistent proximity of edge distribution in the text blocks. The Fourier-Statistical Features (FSF) is proposed in the RGB space for accurate text detection in video frames. Experimental result shows that the text frame classification is needed for text detection approach; because non-text frames erroneously produce false positive detection in text detection approach. Thus a need to first select text frames for text detection to minimize false detection. The advantage of this approach is that it locates text even if the text is in different orientation. In addition, the text detection FSF approach increases the detection rate and decreases the false positive and misdetection rates, but in term of computational time this approach is slightly more expensive. The automatic classification of text and non-text frames scheme leads to a precision rate of 94.69% and 94.73% respectively, with the recall rate of 94.82% for classification of text frames, 96.12% for classification of non-text frames. The detection and extraction of text in video scheme leads to Detection Rate (DR) of 96.18%, False Positive Rate (FPR) of 41.17%, Miss Detection Rate (MDR) of 9.39% for English version and DR of 91.25%, FPR of 41.38%, MDR of 66.37% for Kannada version. Furthermore, a scheme has been developed to evaluate the process speed of the text detection, localization and extraction method and the average time is 0.29s. The performance of the approach demonstrated by extensive experimental results confirms the following: 1) It is capable of handling multilingual texts in the video. 2) It can extract both Scene text and Caption text. Further research work can be carried out for the following issues: 1) This algorithm resides on its assumption that all text is oriented in the same direction, which is by default horizontal. This makes algorithm not suitable to deal with frames with multiple styles, which means modification is still needed to cope with more sophisticated cases such as vertical and skew directions and multi-oriented text lines. 2) Optical Character Recognition (OCR) can be done for the obtained output to check the recognition performance.

Table II

<table>
<thead>
<tr>
<th>Parameters</th>
<th>English Quantity</th>
<th>Kannada Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTF</td>
<td>416</td>
<td>265</td>
</tr>
<tr>
<td>TDB</td>
<td>3913</td>
<td>1222</td>
</tr>
<tr>
<td>FDB</td>
<td>1611</td>
<td>870</td>
</tr>
<tr>
<td>MDB</td>
<td>382</td>
<td>896</td>
</tr>
<tr>
<td>ATB</td>
<td>4068</td>
<td>1350</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th>Languages</th>
<th>DR</th>
<th>FPR</th>
<th>MDR</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>96.18%</td>
<td>41.17%</td>
<td>9.39%</td>
</tr>
<tr>
<td>Kannada</td>
<td>91.25%</td>
<td>41.38%</td>
<td>66.37%</td>
</tr>
</tbody>
</table>

REFERENCES


★★★★