EFFICIENT VIDEO TEXT RECOGNITION USING MULTIPLE FRAME INTEGRATION

Xian-Sheng Hua [1], *Pei Yin** [2], *Hong-Jiang Zhang* [1]

[1]Microsoft Research Asia,
{i-xshua, hjzhang}@microsoft.com

[2] Dept. of Computer Science and Technology, Tsinghua Univ.
pascalyin@yahoo.com

ABSTRACT

Text superimposed on the video frames provides supplemental but important information for video indexing and retrieval. Many efforts have been made for videotext detection and recognition (Video OCR). The main difficulties of video OCR are the low resolution and the background complexity. In this paper, we present efficient schemes to deal with the second difficulty by sufficiently utilizing multiple frames that contain the same text to get every clear word from these frames. Firstly, we use multiple frame verification to reduce text detection false alarms. And then choose those frames where the text is most likely clear, thus it is more possible to be correctly recognized. We then detect and joint every clear text block from those frames to form a clearer "man-made" frame. Later we apply a block-based adaptive thresholding procedure on these "man-made" frames. Finally, the binarized frames are sent to OCR engine for recognition. Experiments show that the word recognition rate has been increased over 28% by these methods.

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 text information in 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 supplemental but important information for indexing and retrieval.

Many efforts have been made for text detection and recognition in videos and images [2-10]. The main difficulties lie in two aspects. One is the low resolution of the text, and the other is the complexity of the background. Some researchers have proposed methods to enhance the resolution by Shannon up-sampling， and to separate text from complex background just by adaptive thresholding [7]. For videos, the same text often exists in several consecutive frames. H. Li has used this information to clear the background of the videotext[8]. Firstly Li use text block registration to identify all the text blocks of the same text string in the consecutive frames. These text blocks are then averaged to get clearer text. Some other papers, such as[9], also average multiple frames or get the minimal brightness pixels of multiple frames to get clearer text (for white text).

However, frequently only parts of these frames have clear text. If we average all the frames that contain the same text, the results may become even worse. Furthermore, sometimes in one frame only part of the text is readable or clear in very few frames. In these cases, averaging or getting minimal brightness pixels over all these frames can not work well.

To deal with the above issue, in this paper, we propose four simple but efficient methods, named Multiple Frame Verification (MFV), High Contrast Frame (HCF) averaging, High Contrast Block (HCB) averaging and Block Adaptive Thresholding (BAT), to utilize multiple frame information to the utmost. By using these methods we can produce clearer text from very complex background, and the recognition rate has increased remarkably.

2. PROPOSED SCHEMES

As just mentioned, sometimes only parts of the frames have clear text and sometimes only part of the text is readable or clear in these frames. However, human beings can recognize all the text because we can integrate all parts into a whole text string even we do not see them at the same time. This phenomenon enlightens us that we can use similar methods to get clearer text from multiple frames.

Figure 1 shows the flow chart of the proposed schemes. In this part, firstly we briefly introduce the text detection method we used, and describe the MFV method, which is used to reduce false alarms. The method for identifying the frame set that contain the same text string is then presented. Later, we introduce HCF averaging and HCB averaging in detail, which are used to enhance the text quality using multiple frame integration. And last, the Block Adaptive Thresholding method is presented.

* The work reported in this paper was performed when this author was working at Microsoft Research Asia as a research intern.

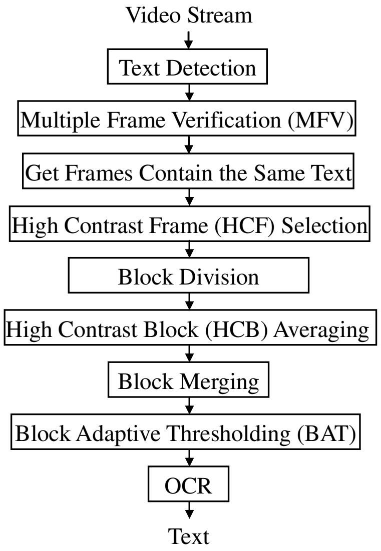

Video Stream
Text Detection
Multiple Frame Verification (MFV)
Get Frames Contain the Same Text
High Contrast Frame (HCF) Selection
Block Division
High Contrast Block (HCB) Averaging
Block Merging
Block Adaptive Thresholding (BAT)
OCR
Text

Figure 1. System flow chart.

2.1. Text detection

2.1.1. Text detection

We detect text in each frame using an improved version of the method proposed in [1] to detect text area in each frame. The differences between our approach and the one in [1] lie in two aspects. Firstly, we adopt the region decomposition method proposed in [10] to get more accurate text bounding boxes. Secondly, we utilize multiple frames to reduce false alarm, as to be explained in detail below.

2.1.2. Multiple Frame Verification (MFV)

We only regard the textboxes that exist in several consecutive frames as true textboxes, and others as false alarms. This verification procedure can reduce those false alarms that appear only in a few consecutive frames. Let

$$Clip = x_1\, x_2\, x_3 \ldots x_k \tag{1}$$

denotes the frame sequence of a video clip, and y_i denote the sub-sequence $x_i\, x_{i+1} \ldots x_{i+m-1}$, i.e.,

$$y_i = x_i\, x_{i+1} \ldots x_{i+m-1} \tag{2}$$

Among the m consecutive video frames, if at least one textbox can be detected in at least n frames, we consider the corresponding frames as the frames that actually contain text.

Suppose there are at least n frames that contain text in y_s, $y_{s+1} \ldots y_{s+p-1}$ but there are at most n-1 frames that contain text in y_{s-1} and y_{s+p}. Then

$$x_s x_{s+1} \ldots x_{(s+p-1)+(m-1)} \tag{3}$$

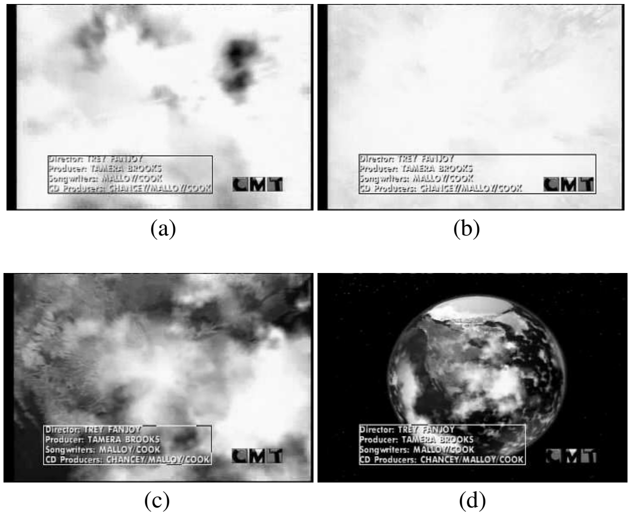
is regarded as a sub-sequence with text, i.e., text appears at frame s and disappears at frame (s+p+m-1). This information will be delivered to the next step to identify the frame set that contains the same text. In the above expressions, m and n are thresholds determined by experience.

2.1.3. Identify the frame set that contains the same text

In the previous step, we get frames that contain text. Now we need to identify the frame set containing the same text string. In [8], the authors proposed the idea of registering all textboxes of the same text string. We may use the same idea here. However, for still superimposed text, we can accomplish this goal approximately in a simpler way. We regard those textboxes whose locations are almost the same in consecutive frames as the same text string. Our experimental results have showed that in most cases, it produces correct results.

2.2. High Contrast Frame (HCF) Averaging

Some text-frames, i.e., frames that contain text, are not suitable for recognition because the background is too complex or the contrast is too low. Figure 2(a)(b) show two examples of this kind of frames, while Figure 2(c)(d) are more suitable for recognition. In this paper, all the video frames are extracted from CMT MTV or MTV2.

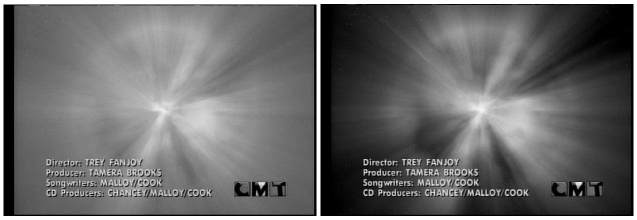


(a) (b)

(c) (d)

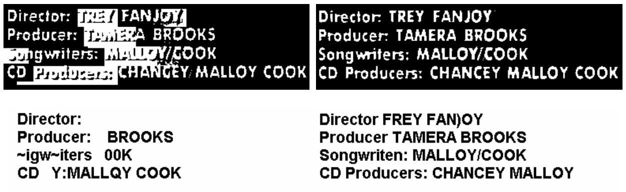
Figure 2. Not all the text-frames are suitable for OCR.

So we only select some "good" frames from the frame set to apply averaging on them. We only average those text-frames that the contrast of the neighbourhood of the textboxes is not very low. In Figure 2, the text bounding boxes shows the merged boxes we used to determine whether they are high contrast frames or not. In our experiments, we judge whether they are HCFs just by counting how many percent of "dark pixels" around these boxes since in most cases the text is white (if the text is not in white or pure color, we will need more complex method to estimate the contrast of the textboxes). An example is showed in Figure 3 and 4. This frame set contains 165 video frames.



(a) (b)

Figure 3. Result of HCF averaging. (a) Average on all text-frames. (b) Average on HCFs.

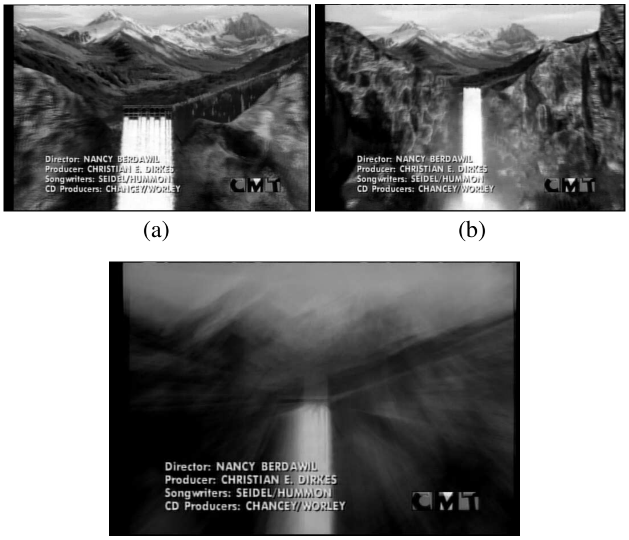


```
Director:                          Director FREY FAN)OY
Producer:   BROOKS                 Producer TAMERA BROOKS
~igw~iters  00K                    Songwriten: MALLOY/COOK
CD  Y:MALLQY COOK                  CD Producers: CHANCEY MALLOY
```

Figure 4. Thresholding (using BAT method, which is to be presented later) and recognition results after averaging on all text-frames (left column) and HCFs (right column) (Character/word recognition rate: left – 0.483/0.357, right – 0.966/0.857).

2.3. High Contrast Block (HCB) Averaging

Sometimes only part of the text is readable or clear in HCFs. In this case, HCF averaging method can not manage very well. Figure 5 shows an example.



(a) (b)

(c)

Figure 5. Sometimes HCF averaging does not work well. (a,b) Example text-frames. (c) Result of HCF averaging.

To solve this issue, we can segment the textbox into smaller blocks and select the corresponding "clearer" blocks in the frame set. Then we average the selected clearer blocks and merge the averaged results to form a whole textbox. By this method, we can recognize every word if it is ever clear in the frame set.

Firstly the textbox is divided into words by the text area decomposition methods in [10] on the HCF averaging frame. Figure 6 shows an example of words segmentation.

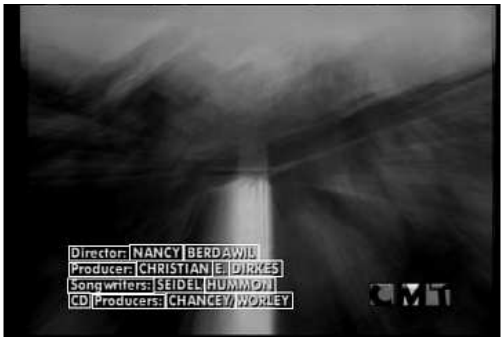
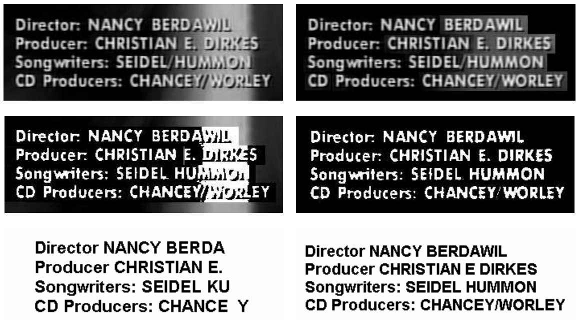


Figure 6. Words segmentation results.

Then, we find the "clear" blocks, i.e., blocks whose neighbourhood has high contrast with the blocks (HCB), in the text-frame set. All the corresponding HCBs are then averaged to get clearer blocks. And last, these clearer blocks are merged to form a whole textbox. Figure 7 show the HCF and HCB averaging results of the example mentioned above. It shows HCB averaging get quite clearer results compared with HCF averaging.



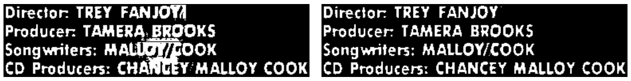
```
Director NANCY BERDA               Director NANCY BERDAWIL
Producer CHRISTIAN E.              Producer CHRISTIAN E DIRKES
Songwriters: SEIDEL KU             Songwriters: SEIDEL HUMMON
CD Producers: CHANCE  Y            CD Producers: CHANCEY/WORLEY
```

Figure 7. Comparison of HCF averaging and HCB averaging. The left column are averaging, thresholding (using BAT method, which is to be presented later) and recognition results using HCF averaging, while the right column is the corresponding results using HCB averaging (Character/word recognition rate: left – 0.783/0.643, right – 1.0/1.0) .

2.4. Block Adaptive Thresholding (BAT)

Since we have got word-box for each word, we can apply an adaptive thresholding (AT) method by threshold each word-box separately, instead of the whole textbox, to get better results while still keeping low computing complexity. For each word-box, we can use any automatic thresholding method here. In our experiments, we apply a K-Mean Clustering method to divide the pixels into two subsets. One sub-set is foreground, and the other is background. Figure 8 shows the comparison of EAT (AT performed in entire text area) vs. BAT (AT performed in each word box) method for the example mentioned in Section 2.2.

The binarized images are then sent to OCR engine for recognition. The final outputs are ASCII text strings.



(a) EAT (b) BAT

Figure 8. Comparison of thresholding methods.

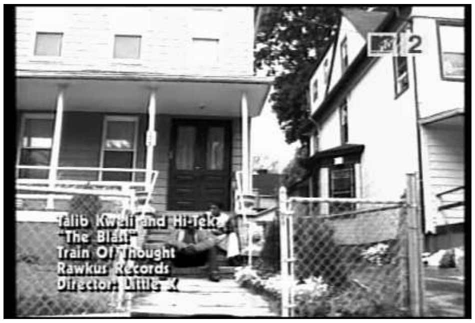
3. EXPERIMENTAL RESULTS

In our experiments, we select 6 MTV which can not get clear text by averaging all text-frames. There are 462 characters, 77 words in total. By using HCF and HCB averaging, the recognition rate of character increases 8.0% and 26.4%, recognition rate of words increases 1.3% and 28.5%, respectively. The detailed evaluation results are listed in Table 1, while Figure 9 shows a powerful example to illustrate the effectiveness of the proposed algorithm, in which we have recognized most of the characters and words in video frames with very complex background.

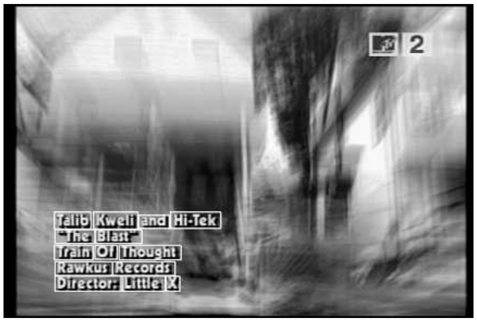
The Video OCR system based on the above schemes is fast enough. All tasks can be accomplished in real time with a few seconds delay on a PC (Dell-PIII866).

Table 1. Evaluation results

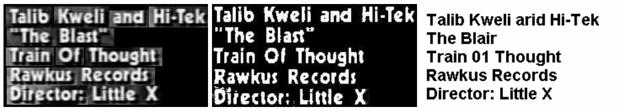
Averaging Method	Thresholding Method	Characters (Total: 462)		Words (Total: 77)	
		Correct	Rate	Correct	Rate
All	**BAT**	238	51.5%	33	42.9%
HCF	**BAT**	275	59.5%	34	44.2%
HCB	**EAT**	344	74.5%	52	67.5%
HCB	**BAT**	**360**	**77.9%**	**55**	**71.4%**



(a) One of original frames.



(b) Results of HCF averaging. The white bounding boxes illustrate the results of word segmentation.



```
Talib Kweli arid Hi-Tek
The Blair
Train 01 Thought
Rawkus Records
Director: Little X
```

(c) Results of HCB averaging, BAT thresholding and recognition. Totally there are 68 characters/14 words. Our algorithm can correctly recognize 65 characters/12 words, while without HCF/HCB averaging and BAT we can only correctly recognize 50 characters/7 words.

Figure 9. More example.

4. CONCLUSION

In this paper, we propose four simple but efficient methods to deal with background complexity in video OCR by efficiently integrating multiple frame information. By using these methods we can produce quite clear text for OCR, and the recognition rate has been increased about 26% for characters and 28% for words, respectively. These methods can also be adopted by any other Video OCR systems to increase recognition rate.

5. ACKNOWLEDGEMENT

Section 2.4 is inspired by the helpful discussion with Prof. Xiao-Ou Tang, IE Dept, CUHK, when he was at MSRA as a visiting researcher.

6. REFERENCES

[1] W. Qi, et al, “Integrating Visual, Audio and Text Analysis for News Video,” *Proc. of ICIP 2000*, Vancouver, Canada, 10-13 September 2000.

[2] Y. Zhong, H. J. Zhang, A. K. Jain, “Automatic Caption Localization in Compressed video,” *IEEE T-PAMI*, 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,” *Proc. 20th Int. ACM SIGIR Conf. on Research and Development in IR*, pp. 3-12, Philadelphia, 1997.

[5] H. Li, D. Doermann, “Automatic Text Detection and Tracking in Digital Video,” *IEEE Trans on Image Processing*, Vol. 9, No. 1, pp. 147-156, January 2000.

[6] A. Wernicke, R. Lienhart, “On the Segmentation of Text in Videos,” *Proc. IEEE 2000*, pp. 1511-1514, New York, U.S.A, July 2000.

[7] H. Li, O. Kia, D. Doerman, “Text Enhancement in Digital Video,” *Proceeding of SPIE, Document Recognition IV*, pp. 1-9, 1999.

[8] H. Li, D. Doermann, “Text Enhancement in Digital Video Using Multiple Frame Integration,” *Proceedings of ACM Multimedia 99*, pp. 19-22, 1999.

[9] J. Xi, et al., “A Video Text Detection and Recognition System,” *Proc.of ICME 2001*, pp 1080-1083, Waseda University, Tokyo, Japan, August, 2001.

[10] X.S. Hua, X.R. Chen, Liu Wenyin, H.J. Zhang, “Automatic Location of Text in Video Frames,” *Proceeding of ACM Multimedia 2001 Workshops: MIR2001*, pp. 24-27, Ottawa, Canada, October 5, 2001.