Automatic Superimposed Text Localization from Video Using Temporal Information

Cheolkon Jung*, Joongkyu Kim* ***Regular Members***

ABSTRACT

The superimposed text in video brings important semantic clues into content analysis. In this paper, we present the new and fast superimposed text localization method in video segments. We detect the superimposed text by using temporal information contained in the video. To detect the superimposed text fast, we have minimized the candidate region of localizing superimposed texts by using the difference between consecutive frames. Experimental results are presented to demonstrate the good performance of the new superimposed text localization algorithm.

Key Words : Video Content Analysis, Text Localization, Superimposed texts

Ⅰ. Introduction

Much information is included in video as colour, motion, caption, audio, and so on. We call this information as video features. Using this information, we can understand the content without watching a video. So, we can save effort and understand the content of video quickly.

Current issues of video content analysis are video summary, video indexing, video search, video editing, video browsing, and video feature extraction. And video feature extraction is a basic research for video content analysis. With the rapid advances in digital technology, the quantity of video content is increased. Moreover computing speed is very fast, and so video feature extraction for video indexing is possible [1], [2].

In video features, the superimposed text has important information. The superimposed text is inserted by content provider intentionally to help the user to catch the video. We can use the superimposed text in video indexing. Also using the superimposed text, we can summarize news and search video contents that users find, easily (fig. 1).

Fig. 1. Superimposed texts

A number of algorithms to localize superimposed texts from video have been published. These algorithms can be classified into two groups. One group is to localize in the compressed domain [5], [9], and the other is in the decompressed domain [3], [4], [6]- [8], [10]-[15].

Zhong and et al [5] presented a method to localize video texts in JPEG compressed images and the I-frames of MPEG compressed videos. This method locates candidate video text regions directly in the DCT domain. So, this algorithm processes quickly. Tang and et al [3] presented a video text detection method based on a fuzzy-clustering neural network classifier. This algorithm uses both spatial and temporal information, and so has high precision. ienhart et

* 성균관대학교 정보통신공학부 (ckjung@ece.skku.ac.kr)
논문번호 : KICS2007-02-079, 접수일자 : 2007년 2월 22일, 최종논문접수일자 : 2007년 8월 1일

al [4] present a novel method for localizing text in complex images and videos. In this algorithm, input images and videos can be of any size due to a true multi-resolution approach. Wong et al [5] presented a robust method for localizing text in digitized color video. This method performs well on images corrupted with Gaussian noise, salt and pepper noise, and speckle noise. Dimitrova et al [6] presented a video text description scheme and automatic methods for detection of text in video segments. This algorithm is based on edge characterization and region analysis. Zhang et al [10] presented a video text detection and extraction algorithm for information retrieval in video databases. In this algorithm, a video text is detected by text (dis)appearance detection by taking full advantage of temporal information.

The methods to localize in decompressed domain have a superior performance, but take a long time.

We proposed the fast algorithm to localize the superimposed text in decompressed domain. Our algorithm has minimized the candidate region of localizing superimposed texts by using the difference between consecutive frames. And with a weighted edge filter, we can have increased a localizing performance.

Ⅱ. Presented Algorithm

2.1 Algorithm Overview

This paper provides an algorithm on localizing video text correctly and fast in a complicated background. First, we read *t*-th frame as the current frame in video, and then verify whether there is a text area detected in the current frame or not. If a text area does not exist in the same area as the (*t-1*)-th frame, we localize the text in the entire current frame. But if the text area is in the same area as the (*t-1*)-th frame, we localize the text from the remaining area excluding the text area which is detected already.

Finally, we check if this frame is the lst frame. If this frame is not the lst frame, the previous process is repeated. The following is the flowchart of the presented algorithm (fig. 2).

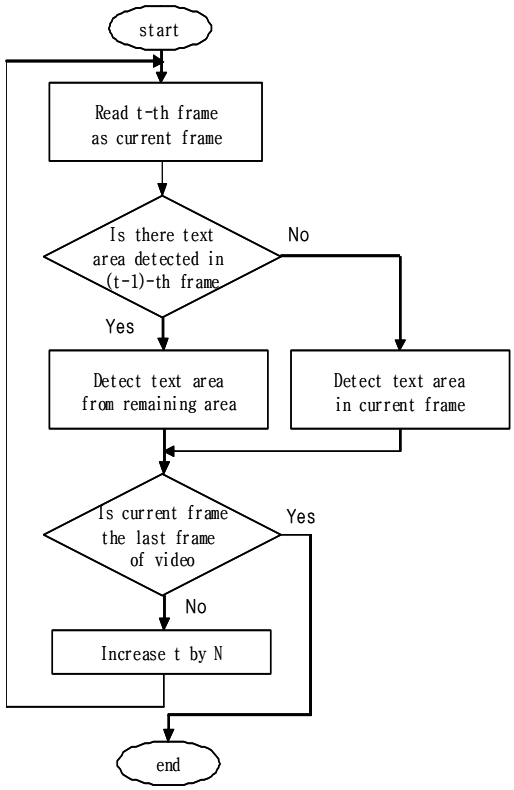

Fig. 2. Algorithm flowchart

Also we can explain how to detect text area in detail like this. As stated in section 1, we have minimized the candidate region of detecting text by using the difference between consecutive frames. And we used weighted edge filter to increase a detecting performance. The candidate regions of superimposed texts are selected by using inter-frame differences. Next, whether the candidate regions are superimposed texts or not is determined by the edge intensity per block. Then, the post-processing for localizing superimposed texts is done. Through this process, we can place the bounding box of superimposed texts in the frame. Finally, the frame numbers which the localized superimposed text start and finish at are determined. This procedure is explained in section 2.2-2.5.

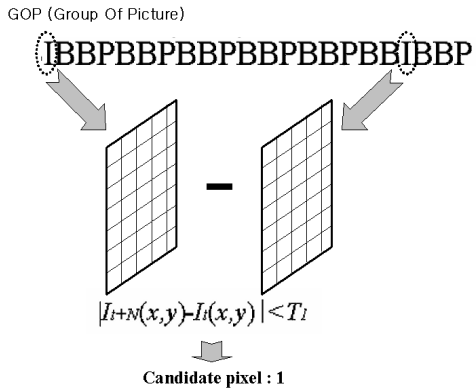

Fig. 3. Candidate pixel

2.2 Selecting the candidate regions of superimposed texts

We must spend a lot of time to localize superimposed texts in a decompressed domain. For this extra reason, time for decoding each frame is added in the computing time. So we have minimized the candidate regions for localizing superimposed texts by the differences of intensity between consecutive I-frames of the MPEG compressed domain.

The minimizing method of the candidate regions is as follows. First, the consecutive two I-frames are selected at *N*-time intervals, and then inter-frame difference in intensity is checked. Next, if the difference is higher than threshold T_I, we assign the pixel to 1, otherwise to 0 (fig. 3).

Then, we segment the frame per block and compute the number of 1's in each block. And if the number of 1's per block is higher than a prescribed threshold $T_{caption}$, this block is a candidate text block.

2.3 Determining whether the candidate regions are superimposed texts or not

In this section, we explain how to determine whether the candidate text block is a superimposed text block or not. By using a common edge filter, the non text block which comes from a smooth variation in intensity is determined as the superimposed text block. This is that the sum of gradient which comes from both smooth variation and abrupt variation is equal in block.

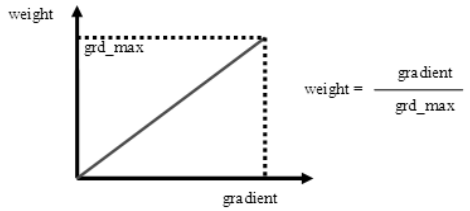

Fig. 4. Weight

So, we used the weighted gradient filter which multiplies the gradient by weight. Here, the weight is the value to divide the gradient of pixel by the maximum value of gradient (grd_max) (fig. 4).

Using the weighted edge filter, we can eliminate the non text block by a smooth variation in intensity.

2.4 Post-processing for localizing superimposed texts

Because the results of section 2.3 are unsatisfactory, we have to post-process to localize superimposed texts. Post-processing consists of three parts: bounding box generation, small region elimination by size filtering, and removing the box which has a small edge density. Bounding box is a rectangular box which is containing text, and generated by horizontal and vertical projection. A small region is a region where the number of pixels is smaller than the prescribed threshold. In general, a small region has nothing to do with text area, so we eliminate this region. Then, we have selected the box which the edge density is bigger than a threshold among the box to remain. So we have localized superimposed texts.

2.5 Determining starting and finishing frame numbers of the localized superimposed text

If the superimposed text is localized, we have to determine the starting frame number. To find the starting frame number, we compare the intensity of the localized region with the region

of a same position in the previous frame. If the difference between the two regions is higher than the threshold, we determine this frame as the starting frame of the localized superimposed text, where R_t is the text region of t-th I-frame.

And the process to find the finishing frame number of the localized superimposed text region is same as the starting frame number. To find the finishing frame number, we have to check whether the localized text exists in the next I-frame. If the localized text exists in the next I-frame, we do not have to find the finishing frame number, because the localized text is maintained within an interval between two I-frames. But if the localized text doesn't exist in the next I-frame, we have to find the finishing frame number. The process is same as for the starting frame number without comparing the region of next frame.

Ⅲ. Experimental Results

We used a total of 7 video sequences for testing. Each video sequence is about 7 min and has 12,600 frames which have easily recognizable superimposed texts. Text sequences consist of five newscasts, one movie, and one sportscast. File format of all sequences is MPEG-1 that frame size is 352x240. Measures for testing performance are recall, precision, and processing speed. Each measure is defined as follows:

- $\text{Recall}(\%) = \dfrac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{B}|}$
- $\text{Precision}(\%) = \dfrac{|\mathbf{A} \cap \mathbf{B}|}{|\mathbf{A}|}$
- $\text{Processing speed}(\text{fps}) = \dfrac{N}{\tau}$

where $\mathbf{A}$ and $\mathbf{B}$ are the sets representing the automatically created text boxes and the ground-truth text boxes, $|\mathbf{A}|$ and $|\mathbf{B}|$ are the number of text box in each set, $\mathbf{A} \cap \mathbf{B}$ is the set of joint boxes in $\mathbf{A}$ and $\mathbf{B}$, N is the number of total frames, τ is the processing time including decoding time, and fps is frame/sec.

(a) (a') (b) (b') (c) (c') (d) (d')

Fig. 5. Experimental results
(a)(b)(c) and (d) are the original images.
(a')(b')(c') and (d') the corresponding results

The results of our text localization algorithm are provided in Table 1. Column three of Table 1 shows the recall (Re), precision (Pr), and processing speed (V) for the 7 sequences. For test sequences, the average of recall rate was 98.08%, precision rate 93.17%, and processing speed 205.52 fps. The data from Table 1 show that the presented algorithm has superior performances about movies and sportscast as well as newscasts.

And the comparison with [10] by Zhang et al. of Microsoft China is provided in Table 2. The data from Table 2 show that the performance of the presented algorithm is similar in detection accuracy (recall and precision), but superior to [10] in processing speed. The reason why the detection accuracy is superior is because the presented algorithm used a weighted edge gradient. And the reason why the processing speed is superior is because the candidate region of localizing superimposed texts is minimized in this algorithm.

Table 1. Results of Text Localization

Test sequences	Re	Pr	V
KBS_030812.mpg	97.07	90.3	193.05
KBS_030815.mpg	98.62	87.85	225.94
MBC_030813.mpg	100	92.99	196.95
SBS_030814.mpg	98.36	89.71	235.79
NBC_030902.mpg	97.54	94.11	225.85
Water.mpg	97.09	96.98	171.02
Golf.mpg	97.55	99.03	190.05
Average	98.08	93.17	205.52

Table 2. Comparison with [10]

	Re	Pr	V
MS China [10]	94%	98%	100 fps
Presented Algorithm	98%	93%	205 fps

Above all, using this algorithm which has so fast processing speed, we can localize video text faster than real time processing (30 fps).

This shows that the presented algorithm has good performance in the localization of superimposed texts.

Ⅳ. Conclusion

In this paper, we present a new and fast superimposed text localization method in video sequences. We have used the temporal consistency of texts and minimized the candidate region of localizing superimposed texts by using the difference between consecutive frames. In result, recall rate, precision rate, and processing speed were 98%, 93%, and 205.52 fps about test sequences which have easily recognizable superimposed texts, respectively. These results show that our algorithm has a superior performance in the superimposed text localization.

Then the presented algorithm can be used for content based video indexing, summary, and search. For examples, it can be applied to news summary, news search, the extraction of important information in sports, sports summary, and so on.

References

〔1〕 Nevenka Dimitrova, Hong-Jiang Zhang, Behzard Shahraray, Ibrahim Sezan, Thomas Huang and Avideh Zakhor, "Application of video-content analysis and retrieval," *IEEE Trans. on multimedia*, Vol. 9, No. 3, pp. 42-55, July-Sept. 2002.

〔2〕 Yao Wang, Zhu Liu and Jin Cheng Huang, "Multimedia content analysis," *IEEE Signal Processing Magazine*, Vol. 17, No. 6, pp. 12 36, Nov. 2000,

〔3〕 Xiaoou Tang, Xinbo Gao, Jianzhuang Liu and Hongjiang Zhang, "A spatial-temporal approach for video caption detection and recognition," *IEEE Trans. on Neural Network*, Vol. 13, No. 4, July 2002.

〔4〕 Rainer Lienhart and Axel Wernicke, "Localizing and segmenting text in images and videos," *IEEE Trans. on CSVT*, Vol. 12, No. 4, April 2002.

〔5〕 Yu Zhong, Hongjiang Zhang and Anil K. Jain, "Automatic caption localization in compressed video," *IEEE Trans. on PAMI*, Vol. 22, No.4, April 2000.

〔6〕 Edward K. Wong and Minya Chen, "A new robust algorithm for video text extraction," *Pattern Recognition*, vol. 36, pp. 1397-1406, 2003.

〔7〕 Nevenka Dimitrova, Lalitha Agnihotri, Chitra Dorai and Ruud Bolle, "MPEG-7 Videotext description scheme for superimposed text in images and video," *Signal Processing: Image Communication*, Vol. 16, No. 1-2, pp. 137-155, Sept. 2000.

〔8〕 Min Cai, Jiqiang Song and Michael R. Lyu, "A new approach for video text detection," *IEEE conf. on image processing*, Vol. 1, pp. 117-120, 2002.

〔9〕 Seong Soo Chun, Hyeokman Kim, Jung-Rim Kim, Sangwook Oh and Sanghoon Sull, "Fast text caption localization on video using visual rhythm," *Lecture Notes in Computer Science*, VISUAL 2002, pp.259-268, March 2002.

〔10〕 Bo Luo, Xiaoou Tang, Jianzhuang Liu and Hongjiang Zhang, "Video caption detection and extraction using temporal information," *IEEE conf. on image processing*, Vol. 1, pp. 117-120, 2003.

〔11〕 C. Garcia and X. Apostolidis, "Text detection

and segmentation in complex color images," *IEEE conf. on Acoustics, Speech, and Signal Processing*, Vol. 4, pp. 2326 2329, 2000.

〔12〕 Jean-Marc Odobez and Datong Chen, "Robust video text segmentation and recognition with multiple hypotheses," *IEEE Conf. on Image Processing*, vol. 2, pp. 433-436, 2002.

〔13〕 Kongqiao Wang, Jari A. Kangas and Wenwen Li, "Character segmentation of color images from digital camera," *IEEE Conf. on Document Analysis and Recognition*, pp. 210 214, 2001.

〔14〕 Dongqing Zhang, Raj Kumar Rajendranand Shi-Fu Chang, "General and domain-specific techniques for detecting and recognizing superimposed text in video," *IEEE conf. on Image Processing*, Vol. 1, pp. 593- 596, 2002.

〔15〕 K. I. Kim, K. Jung, S. H. Park and H. J. Kim, "Supervised texture segmentation using support vector machines," *Electronics Letters*, Vol. 35, No. 22, Oct. 1999.

정 철 곤 (Cheolkon Jung) 정회원

1995년 2월 성균관대학교 전자공학과 학사

1997년 2월 성균관대학교 전자공학과 석사

2002년 8월 성균관대학교 전기전자컴퓨터공학부 박사

2002년 10월~2007년 2월 삼성종합기술원 전문연구원

2007년 9월~현재 성균관대학교 정보통신공학부 박사후 연구원

<관심분야> 멀티미디어 컨텐츠 분석, 멀티미디어 요약 및 검색, 영상처리, 컴퓨터비젼, 컴퓨터그래픽스, 디지털비디오처리

김 중 규 (Joongkyu Kim) 정회원

1980년 서울대학교 전자공학과 학사

1982년 서울대학교 전자공학과 석사

1989년 The University of Michigan, Ann Arbor, Department of Electrical and Computer Engineering Ph.D.

1980년~1981년 한국전자통신연구소 위촉연구원

1989년~1990년 University of Michigan, Post Doctoral Fellow

1990년~1991년 삼성전자 선임연구원

1992년~현재 성균관대학교 정보통신공학부 정교수

<관심분야> 적응신호처리, 레이더신호처리, 의학영상신호처리, 음향신호처리, 디지털비디오처리