Automatic Text Extraction from Video for Content-Based Annotation and Retrieval

Jae-Chang Shim†, Chitra Dorai‡, Ruud Bolle‡
Andong National University†
388 Song-Chun Dong, Andong Kyungpook
760-749 South Korea
jcshim@anu.andong.ac.kr

IBM Thomas J. Watson Research Center‡
P.O. Box 704, Yorktown Heights
New York 10598, USA
{dorai,bolle}@watson.ibm.com

Abstract

Efficient content-based retrieval of image and video databases is an important application due to rapid proliferation of digital video data on the Internet and corporate intranets. Text either embedded or superimposed within video frames is very useful for describing the contents of the frames, as it enables both keyword and free-text based search, automatic video logging, and video cataloging. We have developed a scheme for automatically extracting text from digital images and videos for content annotation and retrieval. In this paper we present our approach to robust text extraction from video frames, which can handle complex image backgrounds, deal with different font sizes, font styles, and font appearances such as normal and inverse video. Our algorithm results in segmented characters that can be directly processed by an OCR system to produce ASCII text. Results from our experiments with over 5000 frames obtained from twelve MPEG video streams demonstrate the good performance of our system in terms of text identification accuracy and computational efficiency.

1. Introduction

The ongoing proliferation of digital image and video databases has led to an increasing demand for systems that can query and search large video databases efficiently and accurately for desired video clips. Manual annotation of video is extremely time consuming, expensive, and unscalable in the face of ever growing video databases. Therefore, automatic extraction of video descriptions is desirable in order to annotate and search large video databases. Text present in video frames is a valuable source of content information. Text is abundant in videos with program credits and title sequences. In news videos, text is often used as captions, and in sports videos game and player statistics are often superimposed on the frames in textual form. Video commercials ensure that the product and other shopping information is presented as readable text. When video text is automatically extracted, it not only provides keywords for annotation and search of image and video libraries but also aids in highlighting events which can then be used for summarizing a video. Text extracted can also be used in video categorization, cataloging of commercials, logging of key events, and efficient video digest construction.

Figure 1. Text in videos appears in different contexts, backgrounds, and font sizes.

Many of the existing approaches to extracting text from images and video [1, 5, 6, 7] suffer from one or more limitations such as locating only the bounding blocks of the text (therefore requiring human involvement to recognize the characters), sensitivity to font sizes and styles, restrictions on the appearance characteristics of text that can be handled, restrictions on the type of text that can be extracted (e.g. captions only), and inability to handle normal and inverse video modes of text.

In this paper, we present a novel computational scheme that not only locates the textual information in video, but also extracts it and generates images with segmented characters that can be directly supplied as input to any OCR system. Our algorithm employs a combination of region segmentation and feature-based refinement techniques to handle the variations in text font size, style, gray level contrast, and complex image backgrounds in which the text is embedded as shown in Figure 1. We also present a mechanism

to exploit the temporal persistence of the text over multiple consecutive frames which can enhance the performance of any video text extraction system.

2. Text Extraction from Video

Text in video appears as either *scene text* or as *superimposed text* [1]. Our system is designed to extract superimposed text and scene text that possesses typical text attributes. We do not assume any prior knowledge about frame resolution, text location, and font styles. Some common characteristics of text are exploited in our algorithm including monochromaticity of individual characters, size restrictions (characters cannot be too small to be read by humans or too big to occupy a large portion of the frame), and horizontal alignment of text.

2.1. Our Approach

The input to our system is a sequence of gray level images obtained by decompressing MPEG encoded video sequences. The terms, "image" and "frame" are used interchangeably in this paper. A basic process that is used repeatedly in our system is that of labeling the pixels in an image based on a given criterion (e.g. gray scale homogeneity) using contour traversal, thus partitioning the image into multiple regions; then grouping pixels belonging to a region by determining its interior and boundaries, and extracting region features such as its MBR (minimum bounding rectangle), area, mean gray level, etc. We have developed a fast and efficient algorithm that uses chain-code to perform these tasks collectively. We refer the reader to [3] for details on this new generalized region labeling (GRL) algorithm.

2.1.1. Extraction of Candidate Text Regions

The objective of this first step is to remove the background from an input gray scale image where the background is interpreted as containing non-text scene contents. The GRL algorithm is employed to extract homogenous regions from the input image. The criterion used to group pixels into a region is that the gray level difference between any pair of pixels within the region cannot exceed ± 10. Having obtained a number of homogenous regions in the labeled image, the non-text background regions are removed based on their spatial (width and height) proportions.

2.1.2. Refinement of Text Regions

Since OCR systems require text to be printed against a clean background for character recognition, a local thresholding operation based on an interactive selection method [2] is performed in each candidate region to separate the text from its surroundings and from other extraneous background pixels contained within its interior. Once thresholds are determined for all candidate regions, *positive* and *negative* images are computed, where the positive image contains region pixels whose gray levels are above their respective local thresholds and the negative image contains region pixels whose gray levels fall below their respective thresholds. A negative image will contain candidate text regions if text appears in inverse video mode in the input. All the remaining processing steps are performed on both positive and negative images and their results are combined at the end of the last stage. We further sharpen and separate the character region boundaries by performing a region boundary analysis based on the gray level contrast between the regions and their boundaries. This is necessary especially when characters within a text string appear connected with each other and need to be separated. For more details, we refer the reader to [4].

2.1.3. Verification of Text Characteristics

The candidate character regions that remain in the image are then subjected to a verification step where they are examined for typical text font characteristics. A candidate region is removed (i) if its area is less than 12 or its height less than 4 pixels, because OCR systems find it difficult to recognize small fonts; (ii) if the ratio of the area of its MBR to the region area (fill factor) is greater than 4; (iii) if the gray level contrast with the background is low [4].

2.1.4. Text Consistency Analysis

Neighboring text regions are examined for consistency to eliminate false positive regions. Unlike many other systems, ours attempts to ensure that regions adjacent in a line in the image exhibit the characteristics of a text string, thus verifying the global structure of a row of text in a local manner. This text consistency test includes (i) position analysis that checks intercharacter spacing, (ii) horizontal alignment verification of characters, and (iii) vertical proportion analysis of adjacent character regions. If all three conditions are satisfied, we retain the candidate word region as a text string. The final outputs are a binary image containing the text characters, which can be directly used as input to an OCR system to generate the text string in ASCII, and a text file containing the feature values of the character regions.

2.2. Interframe Analysis for Text Refinement

Since text in videos persists over multiple consecutive frames, intraframe processing is followed by interframe verification. The text regions in each set of five consecutive

Figure 2. Text extracted by our system.

frames are analyzed together to add missing characters and to delete regions incorrectly identified as text. This inter-frame analysis involves examination of the similarity of text regions in terms of their positions, intensities, and shape features and mitigates false alarm.

3. Experimental Results and Discussion

Our algorithm has been implemented in Visual C++ on a personal computer with a 133 MHz Pentium processor and 32 MB memory, running Windows 95. We tested our text extraction system on 12 video streams whose play-durations varied from 10 seconds to 1 minute. The total number of frames tested was over 5000. Figure 2 shows some of the results obtained from our system. Table 1 gives a quantitative account of the performance of our system on five sequences with ground truth. It can be observed from Table 1 that

Table 1. System performance results.

Video	Text/Non-text Frames	Total number of characters	% Miss rate
BK	467/901	11800	0.29
AB	441/1000	8118	0.85
N1	41/42	862	2.32
SM	159/395	9116	2.68
R1	55/259	1705	0

our system performs very well in terms of the miss rate, especially in video commercials where the characters appear much brighter than the background. With the SM sequence, we observed that our system missed 244 characters, mainly due to the dissolve nature of 12 frames containing text. This count reduces to 40 if the dissolve frames are excluded. Our system extracted text from each frame in about 1.7 seconds on the average; this can be improved with code optimization. These experiments were conducted without utilizing the interframe-based refinement (Section 2.2); adding the latter, we found that in a study with 100 frames containing a total of 140 character groups, the number of false positive text regions dropped from 21 to 2.

Each stage of our algorithm is designed to handle commonly encountered problems that arise in extracting text from video. Unlike many other systems, our system extracts text in reverse video easily by processing both the positive and negative images and combining the results. Our algorithm can be easily extended to handle vertical text by relaxing some of our requirements on horizontal alignment of text strings and also by processing a transformed version of the input image. Very large fonts can also be accommodated by modifying some of the size parameters. Our system handles scene text that is horizontally or diagonally oriented and whose font is within the allowed size. The system is stable as the parameters remain the same for all images.

4. Conclusions

We presented an effective and robust text extraction system for automatically obtaining text present in the videos. Text extracted from video is very useful as keywords for content annotation and searching. Our system outputs OCR-ready bitmaps and experimental results from 5000 video frames demonstrate its good performance. The algorithm is general enough to handle text that appears as part of a scene as well as superimposed text, and robust enough to handle the noise and artifacts introduced by block-based MPEG encoding of digital video. We also presented a novel mechanism that exploits the temporal extent of the text over multiple consecutive frames to improve the accuracy of text detection in video. Our future research is directed towards handling multi-colored text and developing OCR techniques specialized for video fonts.

References

[1] R. Lienhart. Automatic text recognition for video indexing. In *Proc. ACM Multimedia 96*, pages 11–20, Boston, MA, November 1996.

[2] T. Ridler and S. Calvard. Picture thresholding using an interactive selection method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8):630–632, 1978.

[3] J.-C. Shim and C. Dorai. A fast and generalized region labeling algorithm. To appear as an IBM Technical Report, 1998.

[4] J.-C. Shim, C. Dorai, and R. Bolle. Automatic text extraction from video for content-based annotation and retrieval. Technical report, RC 21087, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, January 1998.

[5] M. Smith and T. Kanade. Video skimming and characterization through the combination of image and language understanding techniques. In *Proc. CVPR*, pages 775–781, Puerto Rico, June 1997.

[6] V. Wu, R. Manmatha, and E. M. Riseman. Finding text in images. In *2nd ACM Intl. Conf. on Digital Libraries*, 1997.

[7] B.-L. Yeo and B. Liu. Visual content highlighting via automatic extraction of embedded captions on MPEG compressed video. In *Proc. SPIE Digital Video Compression: Algorithms and Technologies*, volume 2668, February 1996.