Robust Extraction of Text in Video

S. Antani D. Crandall R. Kasturi
Department of Computer Science and Engineering
The Pennsylvania State University, University Park, PA 16802
{antani,crandall,kasturi}@cse.psu.edu

Abstract

Despite advances in the archiving of digital video, we are still unable to efficiently search and retrieve the portions that interest us. Video indexing by shot segmentation has been a proposed solution and several research efforts are seen in the literature. Shot segmentation alone cannot solve the problem of content based access to video. Recognition of text in video has been proposed as an additional feature. Several research efforts are found in the literature for text extraction from complex images and video with applications for video indexing. In this paper we present an update of our system for detection and extraction of unconstrained variety of text from general purpose video. The text detection results from a variety of methods are fused and each single text instance is now segmented to enable it for OCR. Problems in segmenting text from video are similar to those faced detection and localization phases. Video has low resolution and the text often has poor contrast with a changing background. The proposed system applies a variety of methods and takes advantage of the temporal redundancy in video resulting in good text segmentation.

1 Introduction

The use of digital video is becoming increasingly ubiquitous. Efficient access to such digital video involves indexing, retrieval, querying and browsing, which will require automated methods to understand its content. Content-based information retrieval from such digital video databases is a challenging problem and is thus rapidly gaining widespread research interest. For retrieval purposes, the video can either be annotated and indexed manually, which is a cumbersome task, or the users of this video data can rely on content-based methods for indexing (and retrieval from) the video database. Several automated methods have been developed which attempt to access image and video data by content from media databases [3]. Traditional approaches to address this problem have been to apply temporal shot segmentation methods to video. This approach has its limitations. There is a considerable amount of text occurring in video that is a useful source of information which can be used to improve the indexing of video. The presence of text in a scene, to some extent, naturally describes its content. If this text information can be harnessed, it can be used along with the temporal segmentation methods to provide a much truer form of content-based access to the video data.

This paper describes updates to the prototype system for detection, localization and extraction of text from video. This text recognition system [2] applies a battery of methods to video frames in both the compressed and uncompressed domains. The localized text needs to be segmented and then recognized. The text in video is found in various sizes, fonts, with different textures and color gradients on the character stroke. Typically, the text background is changing and often has poor contrast with the text. Segmentation methods developed for this purpose are discussed. Section 2 briefly describes the design of the system for detection of text from video. Section 3.1 describes earlier efforts and Section 3.2 details the segmentation module. Section 4 describes the results and details research in progress.

2. System Description

The video text extraction problem is divided into three main tasks—detection, localization, and segmentation. The recognition (OCR) stage is assumed to lie outside our system. The main components are implemented as POSIX threads. The detection/localization stage consists of a battery of methods for localizing text in the frame. Some methods use the MPEG DCT coefficients, while others use the uncompressed frame. Currently, we have included work from Gargi *et al* [2], Chaddha *et al* [4], LeBourgeois [7] and Mitrea and de With [11]. The spatio-temporal decision fusion module aggregates the decisions of the multiple localization algorithms over multiple frames, defining tight bounding regions around text instances. To improve results, the tracking stage can be used to provide additional input to the spatio-temporal decision-fusion stage. The segmen-

1051-4651/00 $10.00 @ 2000 IEEE

tation module contains the methods to binarize a localized text instance resulting from the fusion process, making it suitable for OCR. The system is designed to take advantage of the temporal nature of video and uses the fact that the text data lasts over several frames for providing robust text detection.

3. Text Segmentation

Most work in text segmentation and recognition has been with high-resolution document images. However video frames have a much lower resolution and suffer from blurring effects of lossy compression. The background of a video frame is often complex with many objects with text-like features. A single video frame can contain several text strings, each having a different color and orientation. This makes the segmentation of text in video a challenging problem.

The segmentation module operates on the bounding boxes determined by earlier stages in the system. The output of segmentation is a binary image of the text in each bounding box, with the text pixels in white and the background pixels in black. The development of this module has focused on making it as general as possible. It should be capable of binarizing both artificial caption text as well as scene text occurring naturally in a video frame. To accommodate scene text, the module should also be capable of segmenting low-contrast and unevenly illuminated text which is quite common in general purpose video.

3.1 Previous Work

This section briefly lists some earlier work found in the literature which deals with extracting text from complex images or video. In the method by Wu *et al* [15], the segmentation stage smoothes the gray scale image and thresholds it at values determined by the first valley on either end of the gray scale histogram. The algorithm does not determine whether the text is lighter or darker but instead generates two outputs, one for each case. Ohya *et al* [12] use a combined detection/segmentation stage to extract characters from scene images using a local thresholding scheme. Lee and Kankanhalli [8] also use a combined detection/segmentation stage. After quantizing the gray levels in the image, detection is performed by searching for strokes with the same gray level. Each potential character is thresholded using the gray level of its boundary. Post processing removes components with suspicious aspect ratios, low contrast, and fill ratios.

Messelodi and Modena [10] present a system for extracting text from book covers. They use a simple global thresholding scheme at the tails of each side of the histogram. Method by LeBourgeois [7] assumes that the dominant portion of the image histogram is the background. Binarization is performed using a maximum–entropy thresholding scheme with an additional stage to split characters inadvertently connected by the thresholding. Approximate character sizes are known a priori. Winger *et al* [14] use a modified form of Niblack's Multiple and Variable Thresholding scheme, which employs variable thresholds based mean local pixel intensity. After calculating the variance,the proposed modified scheme uses a different multiplier and exponent. The authors present the case that using smaller exponent than 0.5 in the Niblack's Point Operator,smaller variances are enhanced in relation to larger variances. Our implementation of the method did not result in significant enhancement of low contrast text.

Agnihotri and Dimitrova [1] use separate detection and segmentation steps to extract text from color video frames. After some preprocessing steps, thresholding is performed on the red plane at the average pixel value of the image. The average of the borders of the text region is also computed and assumed to be closer in value to the average background of the region than text.

Determining the polarity of text is a challenging problem. Several approaches have been taken to address it. Jang and Hong [5] make a priori assumptions about the polarity. Wu *et al* [15] do not make any decisions about the correct polarity, while in [13, 8, 10, 14] both light and dark components are processed and heuristics are used to select characters to form a final image. This results in a very noisy final image. In effect, the false alarms from the positive and negative images are propagated to the final output. However, unlike the other methods, the proposed segmentation module separates localization from segmentation giving better results.

3.2 Description of Segmentation Module

Each text instance is individually examined by the segmentation module. Separating segmentation from detection and localization allows the module to make assumptions about the homogeneity of text within each localized region, while allowing segmentation of text with different characteristics within a frame. Without much loss of generality, the segmentation module assumes that the text and background in a localized region has consistent enough gray levels that all characters are either lighter than or darker than the background.

The bounding box is pre-processed with a contrast-stretching step applied on the luma plane, thereby enhancing the low contrast text. After preprocessing the logical level binarization algorithm proposed by Kamel and Zhao [6] is applied. The algorithm takes two parameters, a maximum stroke width *W* and a minimum difference with the background *T*. We have empirically determined that *W=10* and *T=5* work well for any video frame. The *T* parameter allows characters with low contrast to the background to be extracted. The method assumes that characters are darker than the background. To process regions where the characters are light, the gray scale inverse can be taken, resulting in two segmented text regions. The decision of determining the correct polarity of text is delayed until later stages.

After applying the binarization algorithm, a connected components method is applied. Very small or large components and components that do not have aspect rations characteristic of text are filtered out. This step is applied to both the positive and negative images. A score is assigned to each polar image based on their text-like characteristics. The image with the higher score is selected and the other is discarded. To compute the score, similarity of component heights, widths, and aspect ratios are used. The correct polarity tends to have neatly separated components, whereas the components from the other image represent the text shadow and the holes in the text characters. This causes the bounding boxes of components to overlap. A final measure is determining collinearity of the components, which is typical of most text. The system applies a combination of these measures in a vote to determine the image with the correct polarity.

The logical level thresholding scheme works very well for most text. However, it fails to capture the detail of very small fonts. To overcome this problem, a simple linear interpolation step is applied before the segmentation. This could be performed only on localized text regions smaller than a given size. This step dramatically improves the quality segmentation results for small characters. In addition, the module also refines the results obtained from the logical level algorithm. The topographical analysis method by Lee and Kim [9] is applied to the original gray scale image region. Only pixels that were selected by the logical level method and also correspond to a peak or ridge in the topographical analysis are used as the final binarization. This thins out the characters making the segmented region more readable.

4. Results and Directions

We have also investigated using the results of the segmentation module as feedback to the detection/localization modules. Detection algorithms tend to give false positives for objects with text-like characteristics, such as uniformity of size, color, stroke width, spacing, etc. We have investigated using the shapes of connected components generated by the segmentation algorithm in a region to reduce these false positives. At present, the segmentation module examines the shapes of the connected components within each region's binarized output using a contour-following algorithm, and then parameterizes them into polar coordinates. Zhu and Chirlian's [16] algorithm is used to find the critical points. If a region entirely consists of simple shapes (few critical points, no holes), the system concludes that the region is probably noise and it is discarded. Video also contains useful temporal information. For example, we have found that noise and color bleeding due to lossy video compression can be partially alleviated by performing temporal averaging over a few frames before performing segmentation. Of course this assumes that the text remains stationary from frame to frame. The text tracking module [2] which we have previously developed can be used to determine whether temporal averaging is appropriate for a given subsequence of video. Some results from segmentation are presented in Figure 1 and 2.

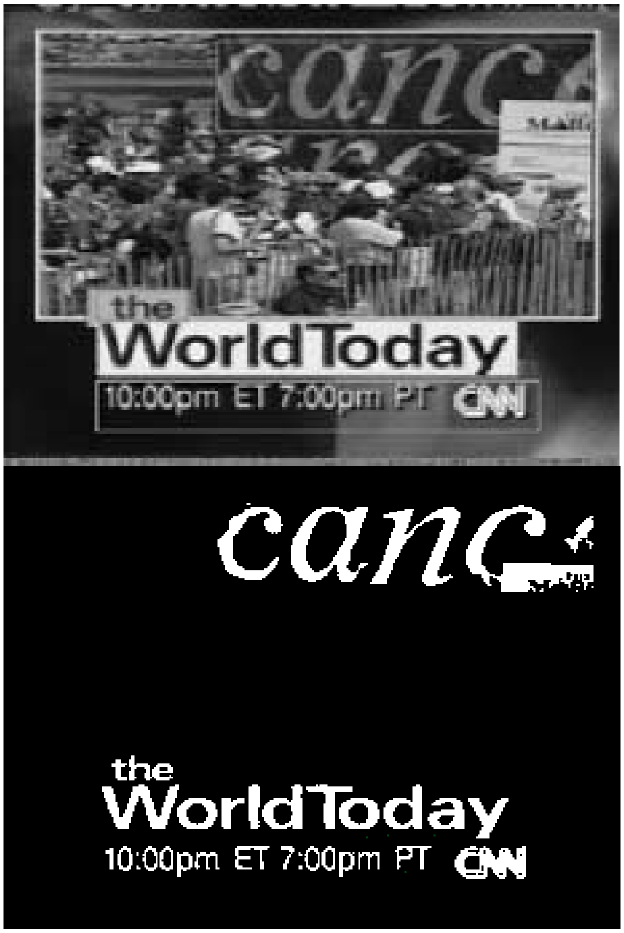

Figure 1. Segmentation Results

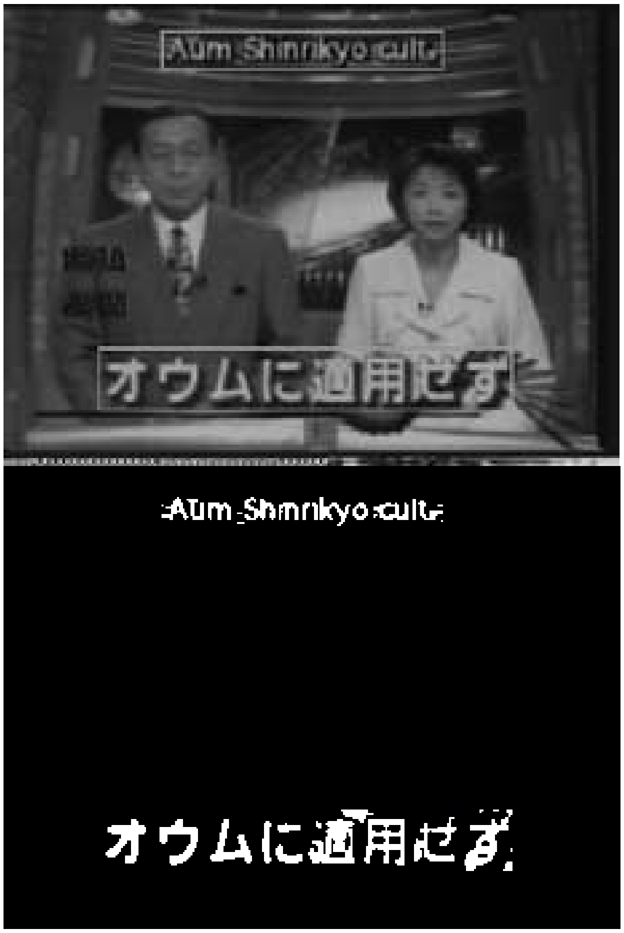

Figure 2. Segmentation Results

References

[1] L. Agnihotri and N. Dimitrova. Text Detection for Video Analysis. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1999.

[2] S. Antani, U. Gargi, D. Crandall, T. Gandhi, and R. Kasturi. Extraction of text in video. Technical Report CSE-99-016, Department of Computer Science and Engineering, Penn State University, 1999.

[3] S. Antani, R. Kasturi, and R. Jain. Pattern Recognition Methods in Image and Video Databases: Past, Present and Future. In *Joint IAPR International Workshops SSPR and SPR*, number 1451 in Lecture Notes in Computer Science, pages 31–58, 1998.

[4] N. Chaddha, R. Sharma, A. Agrawal, and A. Gupta. Text Segmentation in Mixed–Mode Images. In *28th Asilomar Conference on Signals, Systems and Computers*, pages 1356–1361, October 1994.

[5] J.-H. Jang and K.-S. Hong. Binarization of noisy gray-scale character images by thin line modeling. *Pattern Recognition*, 32(5):743–752, 1999.

[6] M. Kamel and A. Zhao. Extraction of Binary Character/Graphics Images from Grayscale Document Images. *Computer Vision, Graphics, and Image Processing*, 55(3):203–217, May 1993.

[7] F. LeBourgeois. Robust Multifont OCR System from Gray Level Images. In *International Conference on Document Analysis and Recognition*, volume 1, pages 1–5, 1997.

[8] C.-M. Lee and A. Kankanhalli. Automatic Extraction of Characters in Complex Scene Images. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(1):67–82, February 1995.

[9] S.-W. Lee and Y. Kim. Direct Extraction of Topographical Features for Gray Scale Character Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):724–728, July 1995.

[10] S. Messelodi and C. Modena. Automatic Identification and Skew Estimation of Text Lines in Real Scene Images. *Pattern Recognition*, 32(5):791–810, May 1999.

[11] M.v.d.Schaar-Mitrea and P. de With. Compression of Mixed Video and Graphics Images for TV Systems. In *SPIE Visual Communications and Image Processing*, pages 213–221, 1998.

[12] J. Ohya, A. Shio, and S. Akamatsu. Recognizing Characters in Scene Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:214–224, 1994.

[13] J.-C. Shim, C. Dorai, and R. Bolle. Automatic Text Extraction from Video for Content-Based Annotation and Retrieval. In *Proc. International Conference on Pattern Recognition*, pages 618–620, 1998.

[14] L. Winger, M. Jernigan, and J. Robinson. Character Segmentation and Thresholding in Low-Contrast Scene Images. In *Proceedings of SPIE*, volume 2660, pages 286–296, 1996.

[15] V. Wu, R. Manmatha, and E. Riseman. Finding Text in Images. In *2nd ACM Intl Conference on Digital LIbraries DL'97*, 1997.

[16] P. Zhu and P. Chirlian. On Critical-Point Detection of Digital Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):737–748, August 1995.