A System for Automatic Text Detection in Video

Ullas Gargi David Crandall Sameer Antani Tarak Gandhi Ryan Keener Rangachar Kasturi
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
{gargi, antani, kasturi}@cse.psu.edu

Abstract

Video indexing is an important problem that has occupied recent research efforts. The text appearing in video can provide semantic information about the scene content. Detecting and recognizing text events can provide indices into the video for content-based querying. We describe a system for detecting, tracking, and extracting artificial and scene text in MPEG–1 video. Preliminary results are presented.

1 Introduction

Content-based indexing of digital video has gained research importance in recent years. Detected text in video can provide a useful index with or with recognition [3]. The text appearing in a video can be either *scene text* which occurs naturally in the recorded 3-D scene and is distorted by perspective projection, or *artificial text*, the text overlaid on the video frame during editing.

Detection of text from color video images has some unique problems. Video frames are typically low resolution and compressed (which can cause color bleeding between text and background). The text in a video frame can be low-contrast, multi-colored (which means large-scale assumptions on the color of the text or background components cannot be made), and multi-font. It can be translucent and with a changing complex background. Its location can be changing from frame to frame. Scene text also undergoes the perspective deformation.

1.1 Previous Work

Shim et al [13] detect artificial text from MPEG video by identifying homogeneous regions in intensity images, perform segmentation by interactive threshold selection, and apply heuristics to eliminate non-text regions. Ohya et al [11] operate on gray scale images to perform character segmentation by local thresholding followed by pairing of nearby regions with similar gray levels. Jain and Yu [4] extract text from video frames and compressed images by background color separation and applying the heuristic that the text has high contrast with the background and is of the foreground color. Lienhart and Stuber [8] have developed a system for automatic text recognition. The system uses both spatial and temporal heuristics to segment and track (possibly scrolling) caption text in video.

1.2 Proposed Approach

From our study of the published algorithms, we find that no one algorithm is robust for detection of an unconstrained variety of text appearing in video. We present a system which uses a battery of different methods employing a variety of heuristics for detecting, localizing and segmenting both artificial and scene text and takes advantage of the temporal nature of video. Recognition of multi-font text is an entirely different research problem which is not addressed by our system.

The system performs the following main tasks—detection & localization, tracking, decision-fusion, and segmentation. There is also a specialized module to detect and segment scene text that satisfies certain constraints. The architecture of the system is shown in Figure 1. Modules for detection, decision-fusion, and segmentation as well as the novel algorithm for extracting scene text are described in Sections 2 through 5. Results are presented in Section 6.

2 Detection & Localization

The text 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. Localization improves performance of segmentation and provides a raw index if segmentation fails. Currently, we have included work from Gargi et al [3], Chaddha et al [1], Winger et al [14], LeBourgeois [7] and Mitrea and deWith [9]. Methods described in [1] and [9] have been modified as described below resulting in improved performance.

2.1 DCT energy localization

The original method described in [1] computes an energy measure of a set of DCT coefficients of intra-coded blocks as a texture measure. This method is initially applied using a high energy threshold. The threshold is successively lowered in subsequent iterations, resulting in region growing from initial high-probability seed blocks. Only those blocks which have an 8-neighbor in the result from the earlier iteration are retained. This improves detection without

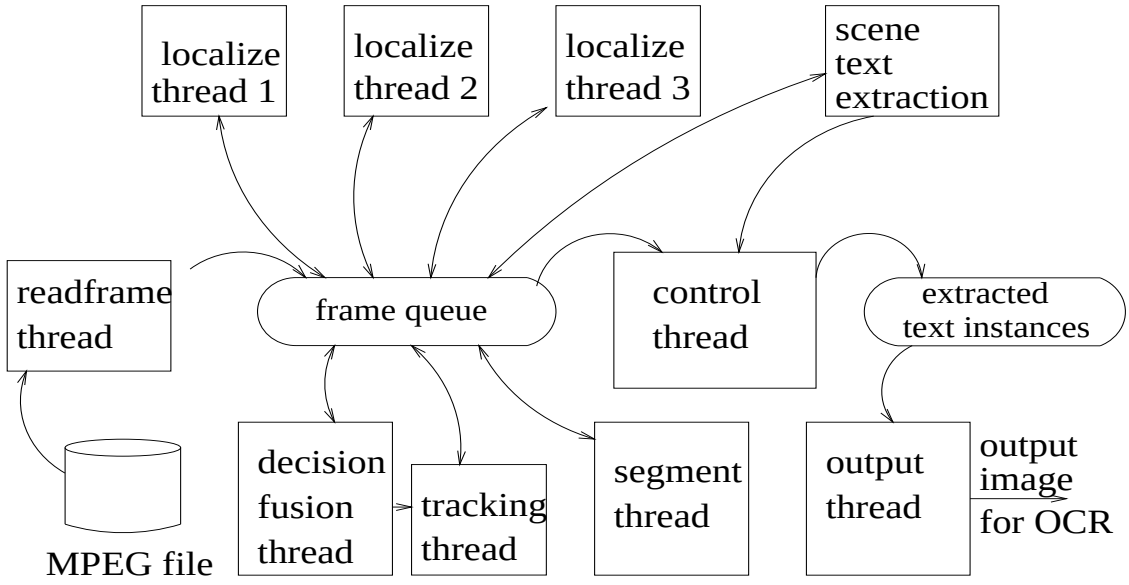

Figure 1: *Implementation architecture of the system*

excessive false positives. Regions with undesirable aspect ratios or due to large object boundaries are discarded. This identifies the text regions along MPEG block boundaries.

2.2 Graphics/Text block classification

Mitrea and de With [9] proposed a simple algorithm to classify video frame blocks into graphics or video based on the dynamic range and variation of gray levels within the block. Their application was improving image compression rates. We modified this method slightly and used it to classify blocks as text or non-text followed by morphological filtering of text regions.

3 Spatio-Temporal Decision-Fusion

The localization algorithms have different failure modes. The decision-fusion module when complete will use the text regions localized over multiple frames by multiple algorithms and fuse them into a final region based on agreement between algorithms, motion of text (if a localized text box moves, it may not be text unless the motion is not random and corresponds to tracking information), tracking (to test whether a moving text box is a false alarm or due to moving text, the fusion module can spawn a tracking thread and compare its output to the series of localized boxes) and static morphology (if the shape of a localized text region changes between frames and this change is not consistent with occlusion information output by the tracking module, the region may be discarded).

Both artificial text and scene text strings may be broken up, the former due to imperfect localization, the latter due also to occlusion. In a subsequent frame, the missing characters might be localized. The control module needs to look at all the localized text regions and stitch together those belonging to the same text string using information output by the tracker.

3.1 Tracking

The current method implemented in the tracking stage processes only P and I frames, ignoring B frames completely. The location of the text region in the subsequent frame is predicted based on its current velocity. If the next frame is a P frame, we look at the motion vectors of all of the macroblocks that correspond to the predicted bounding box region. All motion vectors that are labeled “flat” using an “edginess” criterion [12], are ignored. This is because the flat macroblocks tend to have noisy motion vectors. Adjacent motion vectors are compared to reject those which are spatially inconsistent (high angular difference [10]). A mode of the remaining vectors is taken to identify the vector which occurs in most macroblocks. The original bounding box is then moved by the amount indicated by the vector. Finally a least-square-error search of a small neighborhood is performed to locate the exact matching pixels. We found the gradient intensity instead of the actual values to be more robust for this step. This algorithm appears to be fast and effective and can be used to track an initial user-defined text box as well as for decision-fusion.

4 Segmentation

The segmentation stage is passed a bounding box around a localized text region possibly generated from multiple frames by the decision-fusion module. Because of the small extent, tight constraints can be applied. We have tried a few approaches at segmentation; LeBourgeois’ method [7] for inter-character segmentation fails for overlapping or italic characters. Various background/foreground detection meth-

ods (e.g. those proposed in [4] and [7]) assume that the background is the largest component of the histogram. With complex backgrounds this assumption is invalid. Using the logical level method of Kamel and Zhao [5] which was proposed for low-contrast check images, we obtained good results by using it on both original and inverted images and choosing one of the results based on the size and distribution of the connected components.

5 Scene Text Detection

Scene text typically exists on a planar surface in a 3-D scene with unconstrained orientation. As the camera or the object moves, motion is induced in the image plane. This motion of the text features should satisfy planar motion in 3-D. We propose to exploit this to separate text features from features due to other objects. These objects, which are likely to be at different random depths, do not satisfy the planar constraint. A sequence of images can be used to segment different planar surfaces in the image, and also to remove the outliers corresponding to clutter which does not fit any such surface, or is in motion with respect to such a surface. Devadiga [2] has developed a recursive motion-based segmentation algorithm to segment image into regions corresponding to planar and other objects. We propose to use a similar algorithm to identify text regions. Once the parameters of the plane are estimated, the perspective effect of the camera on the characters can be compensated.

5.1 Planar motion model

Suppose that the point given by (X, Y, Z) in the 3-D coordinate system in which the Z axis passes through the optical axis of the sensor lies on a planar surface. The image velocity for any such point on a planar surface can be written as:

$$\begin{aligned} \hat{u} &= a_1 + a_3 x + a_5 y + a_7 x^2 + a_8 xy \\ \hat{v} &= a_2 + a_4 x + a_6 y + a_7 xy + a_8 y^2 \end{aligned}$$

The eight coefficients $(a_1, \ldots, a_8)$, the planar motion model parameters are functions of the relative linear and angular velocities (V and W), as well as the parameter for the planar surface. Thus, the instantaneous motion of a planar surface undergoing rigid motion can be described as a second order function of image coordinates.

Hence, if the image velocity at a number of points on the planar surface is known, each point results in two equations containing eight unknown parameters. A least squares fit can be used to compute the planar model which best fits the points. Four or more points on the same plane are required for this purpose.

The relative motion parameters V and W between the camera and the scene need not be known in advance to solve these equations. The planar motion parameters can be used to compute the equation of the plane as well as the camera motion parameters, giving two solutions (except for a scale factor) in most cases [6]. The equation of the plane can then be used to determine how the text on the plane gets distorted by perspective effects, and to apply correction to compensate for this distortion.

6 Preliminary Results

6.1 Text Detection and Tracking

Figure 2 shows the results of localization by two algorithms ([3] and [9]), indicating the need for data fusion. Figure 3 shows localized and then segmented text; all three instances in the first frame and five instances in the second frame are detected, even the extremely low-contrast text, but the segmentation needs improvement. The localization is actually finer than the bounding boxes shown. Figure 4 shows two examples of scene text tracking: first, two initial user–delineated bounding boxes at frame 34, tracking across occlusion (frames 70 and 112), and text region modification as it leaves the frame (frames 139); second, multiple scrolling captions being tracked.

6.2 Scene text Parameter Estimation

The application of the planar fit has been tested on a number of simulated image sequences containing text or other patterns on planar surfaces. The camera translation and rotation, motion parameters of the camera, and the plane parameters were pre-specified. Using a sequence of images, the planar motion parameters were obtained. The error in the plane normal vector extracted from these parameters was calculated to be $3.56°$. Using the plane normal vector, the perspective deformation can be corrected by image warping methods.

7 Summary

We describe a system for detecting text in video. The system is to be able to detect unconstrained scene and artificial text in MPEG video. The text may be moving or have poor contrast in cluttered backgrounds. Results of localization, tracking and segmentation of artificial and some scene text are presented. Further work involves building the decision-fusion module, handling occlusion, improving the scene text extraction, and enhancing the text for OCR.

References

[1] 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.

[2] S. Devadiga. *Detection of Obstacles in Monocular Image Sequences.* PhD thesis, The Pennsylvania State University, Computer Science and Engineering Department, August 1997.

[3] U. Gargi, S. Antani, and R. Kasturi. Indexing Text Events in Digital Video. In *Proc. International Conference on Pattern Recognition*, volume 1, pages 916–918, August 1998.

[4] A. K. Jain and B. Yu. Automatic Text Location in Images and Video Frames. *Pattern Recognition*, 31(12):2055–.126, 1998.

[5] 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.

[6] K. Kanatani. *Geometric Computation for Machine Vision*. Oxford Science Publication, Walton Street, Oxford, UK, 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] R. Lienhart and F. Stuber. Automatic Text Recognition in Digital Videos. In *Proceedings of SPIE*, volume 2666, pages 180–188, 1996.

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

[10] Y. Nakajima, A. Yoneyama, H. Yanagihara, and M. Sugano. Moving Object Detection from MPEG Coded Data. In *Proceedings of SPIE*, volume 3.12, pages 988–996, 1998.

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

[12] M. Pilu. On Using Raw MPEG Motion Vectors to Determine Global Camera Motion. In *Proceedings of SPIE*, volume 3.12, pages 448–459, 1998.

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

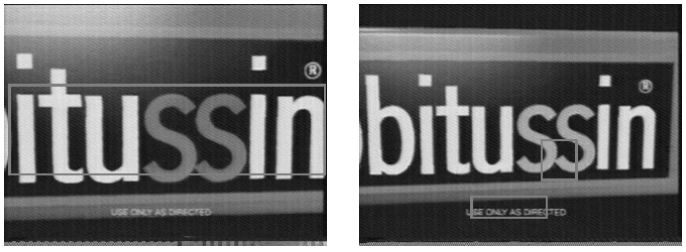

Figure 2: Results of text localization.

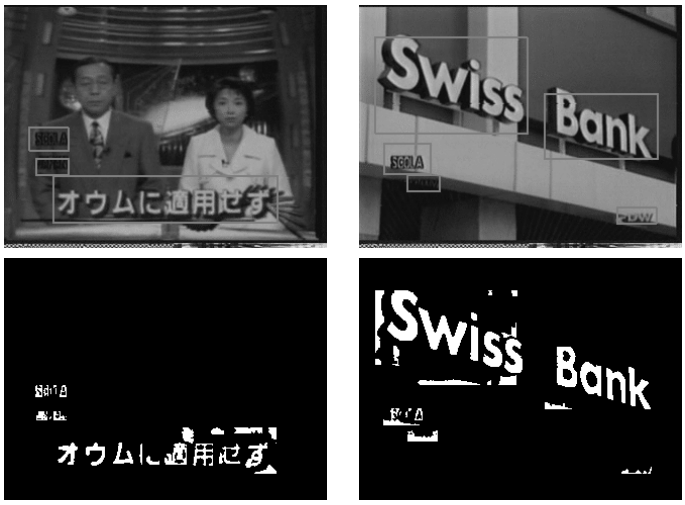

Figure 3: Segmentation of localized text.

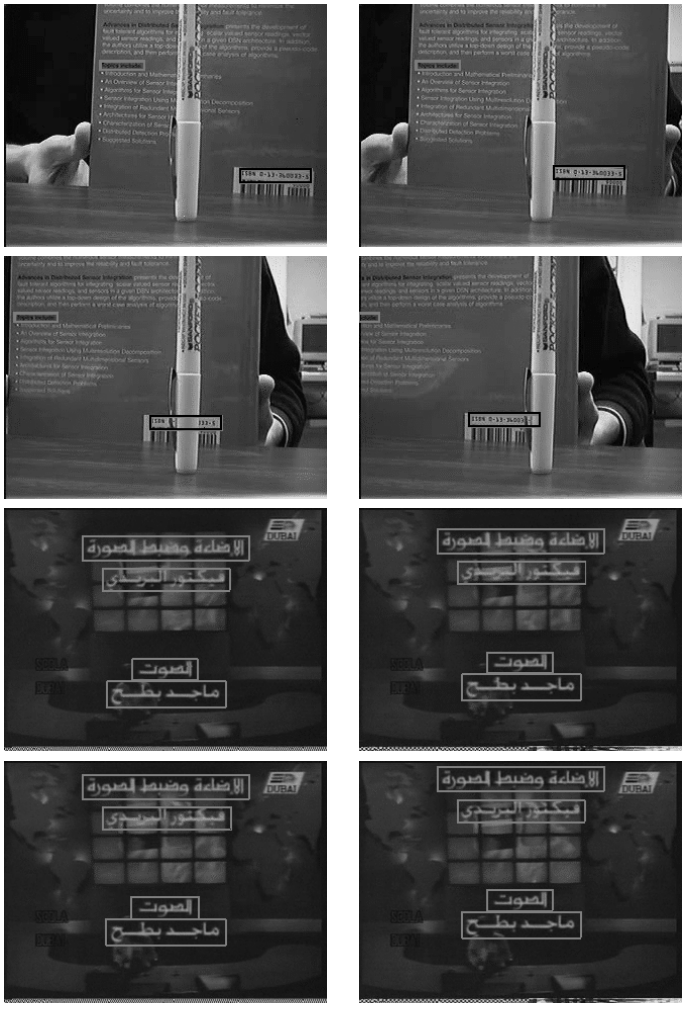

Figure 4: Tracking moving text.