Detection of Score Changes in Sport Videos Using Textual Overlays

Reuven Berkun, Ezri Sonn and Dmitry Rudoy

Department of Electrical Engineering

Technion – Israel Institute of Technology

Haifa, Israel

Abstract - **Sport videos are very popular content among consumers. Many of such videos include overlaid textual data about the game, such as score or player names. But, when the video is viewed on a small screen textual information becomes illegible. In this work we propose a fully automatic system for detecting changes in overlaid textual information, such as the game score. Our system detects text in the video and then marks the spatial and temporal coordinates of where the changes that have occurred in the text. This enables us to make text legible on a small screen by displaying an enlarged score box at the time of score change events. The proposed method was tested on a dataset of different sport videos, collected from YouTube. It achieves good results in terms of accurate spatial score box localization and accurate temporal score-change detection.**

I. Introduction

In recent years, the world of media and digital content has revolutionized. An important example of this revolution is digital video content. Online services, such as YouTube, allow us to watch streaming video anywhere, anytime. Actually, a grand portion of internet traffic is made up of video content Another example is the growing field of Personalized TV, or Video On Demand, which enables the consumer to choose what to watch, not only on his TV set, but practically on any media device, such as laptops, tablets, or mobile phones.

One of the most popular video content genres is sport games, which often contain visual textual information overlaying the video frames acquired by the camera. Most of the textual information is concentrated in the game score box. When watching sports video on a mobile media device, which usually has a small screen and low display resolution, the information in the score box becomes illegible. In this work we deal with this problem by automatically detecting score changes events in order to enlarge the score box and display the interesting information to the viewer at the time of these events.

The system proposed in this work is composed of three main processing stages. In the first stage, the location of the score box is detected in the video frame. This stage is based on text detection using corner and edge density analysis and the novel Adaptive Threshold Factor method. In the second stage, text segmentation of the information inside the score box is performed, in order to separate the textual information into single characters, and achieve tight boundaries for each character. This is made by using binarization and connected component analysis. In the third stage, after finding the separate components, the video frames are scanned and each component inside the score box is examined for temporal changes. This stage uses temporal differentiating of the binary video frames, while removing irrelevant textual changes. The outputs of the system are the score box coordinates inside the video frame, and the time of the score changes.

The rest of the paper is organized as follows. Section II presents a review of related work in text and event detection. Section III gives a general description of the system, and describes the processing stages of the system –score box localization (Section A), segmentation of textual information in the score box (Section B), and detection of changes in the score box components (Section C). Experimental results are presented in Section IV, and Section V provides a conclusion and a discussion of future work.

II. Related Work

Much effort has been devoted to the analysis of sport video. In this section we review some of the work related to event detection in sport video, and to text detection and recognition in video.

Sport video broadcasting contains visual, audio and textual information, all of which can be used for content analysis to recognize interesting sporting events in the game, such as goal segments, changes in the game status, fouls and more. Various approaches have been proposed for event detection using visual data. One prominent feature of special events in sports videos is the appearance of slow-motion playback immediately following the event. Slow motion can be easily detected, and thus used to define interesting events [7]. Some approaches propose to detect close-up shots which usually follow interesting occurrences [4][8]. Another approach to video analysis uses the fact that most sports videos follow a certain structure of broadcasting. This approach is used in [2] to identify the different stages of baseball games by the

camera's point of view. Camera motion has also been used to identify goals by detecting changes in the direction of the camera motion [7]. In [8] it is proposed to use audio analysis to detect interesting events, which are accompanied by crowd applause or changes in the voice tone of the sportscaster. Others, such as [2][4][7], proposed to use superimposed captions to detect events and game statistics, using text recognition methods. Finally, a method of temporal models development, based on manual video observations, has been suggested in [10].

The various approaches to event detection mentioned above, suffer drawbacks in relation to the goal of our work. Methods based on camera motion or audio analysis, detect all kinds of events besides score changes, such as fouls, which we do not want to detect. Our approach to detect score changes is to find the score in the frame and scan it over time. On the other hand, the methods that do use overlaid text to detect events, use text recognition to translate the captions into text and then analyze the text, an operation of relatively high complexity. As explained in Section III, the method proposed here is simpler and more efficient since it only detects changes in the score, using very basic and low-cost functions such as edge and corner detection, thresholding, etc. It is not necessary to know what the score is but only that it has changed, and for this purpose we use text detection.

Text detection in video has many different uses and implementations, and is therefore a subject of extensive research. Some of the methods for text detection include texture analysis [6], neural networks [5], analysis of motion vectors and DCT coefficients in compressed video [6]. These methods usually demand prior processing of the video content and high computational complexity. In this work we chose to use edge and corner density analysis [1]. As we inferred, it is a proper approach for statistically characterizing text regions.

III. A System For Score Change Detection

In this section we describe the overall system for score change detection in sport videos, which is presented in Figure 1. Its input is a video sequences composed of grayscale frames. First, the system localizes the potential score boxes using text detection method, based on the approach by Hua et al. [1] (Section A). Then, the text inside all the candidate score boxes is segmented (Section B), and temporal changes of the text are detected (Section C). The output is the time of the detected change in the score together with the spatial coordinates of the altered text box.

A. Score box localization

The first step of detecting score changes is to define the spatial boundaries of the score box. Since the score box mainly contains textual information, which is what we are interested in, we will use text detection in order to locate the score box. Our approach is based on an algorithm by Hua et al. [1], which uses edge and corner density analysis for locating text in an image.

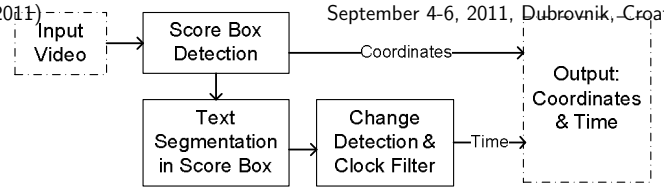


Figure 1. A scheme of the proposed system for score change detection.

1. Text detection

Text detection in [1] is performed on a still image, using corner and edge density analysis. This method is based upon the assumption that text areas in images are rich in corners and edges. First, a binary corner map is extracted, using the Harris corner detection method, due to its strong invariance to rotation, scale, illumination variance and image noise [11]. This map is refined by eliminating isolated corners which probably do not represent text areas, by minimum distance criteria. At this point candidate text regions are formed, by merging corners in close proximity to each other into rectangular areas. An example is shown on Figure 2, demonstrating the text detection flow on a frame of tennis match. The original frame and its resulting corner map are shown on Figure 2(a) and Figure 2(b).

Besides the corner density, text regions are rich with edges as well, as shown on Figure 2(c). Thus, the horizontal and vertical edge maps are produced using Sobel edge detector [12]. The candidate text regions are scanned vertically and horizontally, and each scanned line or column is checked for minimum number and density of edge points in its vicinity. Those lines or columns which do not satisfy the edge point constraints are deleted from the text region map. The purpose of this stage is to eliminate areas that do not satisfy both corner and edge criteria. This text line decomposition process is iterated several times for refinement, and the remaining text areas are then expanded to their bounding rectangles.

The final step in the algorithm is text box verification. Each text box is tested and verified using several criteria:

- Minimum and maximum height of text box.
- Minimal horizontal-vertical aspect ratio.
- Edge point fill factor constraints.
- Edge point center offset ratio.

In addition to these criteria, any text box located in the center of the frame is removed. The reason for this is that the score box almost always appears either at the top or the bottom parts of the frame. Figure 2(e) shows an example of a result of such verification.

2. Adaptive Threshold Factor

The text detection algorithm uses a large number of thresholds in its different stages. Naturally, the initial values for these thresholds specified in [1], cannot be suitable for all possible input images. This is the algorithm's main weakness, and requires special attention. By introducing an Adaptive Threshold Factor (ATF),

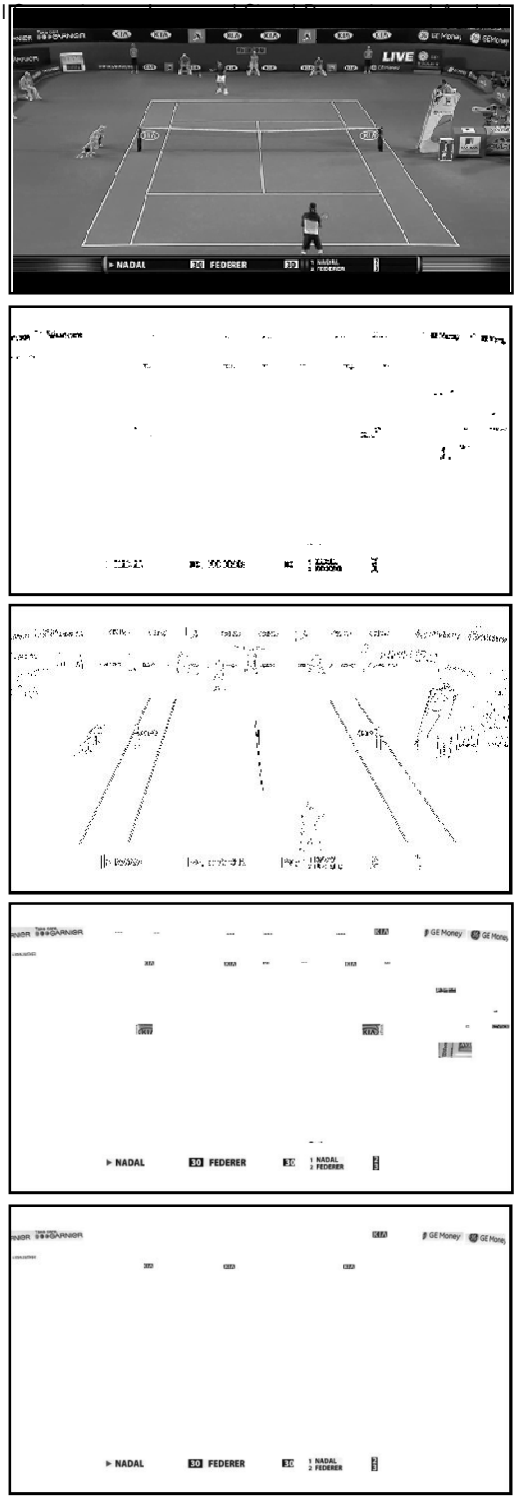


Figure 2. Text detection on a single video frame. (a) An input original video frame of a tennis match, and (b) its corresponding refined corner map (color-inverted). The regions with high density of corners are potential score boxes. (c) Vertical Sobel edge map..(d) Candidate text boxes after text line decomposition, and (e) final text boxes after verification. The final result contains most of the text in the frame, and particularly the text in the score box.

based on the image corner density map, we can adapt the basic thresholds to each image individually.

The ATF is calculated as follows. The binary corner map is low-pass filtered with a 15×15 uniform kernel. This action produces a corner density map, in which each pixel represents the number of corners in its 15×15 neighborhood. By building a histogram of the density map, we can get an estimate of the amount of corner-dense regions in the image. The ATF is inverse to the number of high-density pixels:

$$ATF = c_{ATF} \cdot \left(\frac{\sum_{n=10}^{225} hist[n]}{height \times width} \right)^{-1} \quad (1)$$

The value $c_{ATF} = 0.1$, was shown to produce the best results for most of the test videos.

The thresholds used in the text detection algorithm in [1] are then multiplied by the ATF, so that a high ATF will produce more flexible thresholds, and vice versa.

3. Temporal averaging of the video

From the example in Figure 2(e) we can see that besides detecting the score box text components, the algorithm also detects other textual information that appears in the video frame, such as commercials and banners in the tennis court. Since the score box is a superimposed object added to the video, it remains static over time, as opposed to the frame images which are dynamic and change rapidly, especially in sports videos. Based on the fact that the score box remains in a fixed location inside the video frame, temporal averaging is performed over a 1-second time period, before the text detection process. This action emphasizes any static objects in contrast to the rest of the frame, which becomes blurred and loses its edges and corners, thus making it easier for the text detection algorithm to identify the score box.

The results of applying the text detection to the averaged image can be seen in Figure 3. The textual data that is part of the scene, which had appeared in Figure 2(e), has now disappeared.

4. Final score box merging

In many cases, the score box is comprised of several text boxes, which are detected independently by the algorithm, as can be seen in Figure 3(b).

Therefore, we would like to merge these "sub-boxes" into one score box. However, in some sport videos more than one text box may appear (e.g., score box and channel logo), so a merging criteria must be defined.

Given two neighboring text boxes, as shown in Figure 4, the decision to merge them into one box is based on the distance between them and horizontal or vertical overlapping.

The horizontal merging conditions are:

$$\begin{cases} \Delta y > b_y \cdot \min(h_1, h_2) \\ [\Delta x < \max(w_1, w_2)] \; or \; [\Delta x < b_x \cdot w_{frame}] \end{cases} \quad (2)$$

Where we use $b_y = 0.5$, $b_x = 0.3$.

The vertical merging conditions are received by switching between Δx and Δy respectively.

At this point we have the final score box location, derived from a 1-second period of the video. In order to reduce false detections caused by short-lived texts such as commercials, the score box location process is repeated once more upon a different 1-second period, and the results are compared and combined.

It is possible for more than one "score box" to be detected. This can occur if beside the main score box, there are more text boxes such as channel logos, other game statistics, etc.

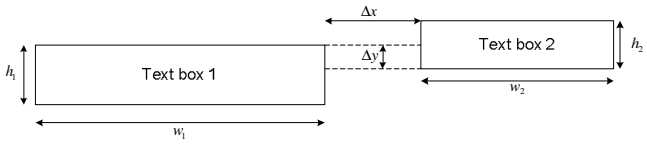
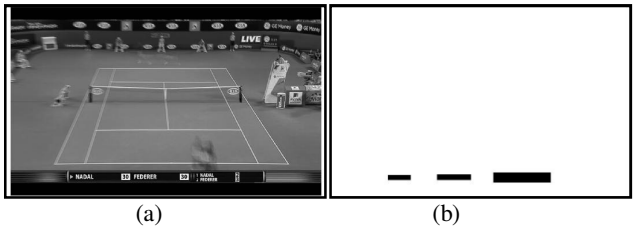


Figure 4. Example of neighboring text boxes

B. Segmentation of score box text

Once the score box is located and defined, the next step is to analyze the information inside the score box. This includes separating the textual data into its basic components and focusing on the information important to this system – the game score.



(a) (b)

Figure 3. (a) Averaged video frame. As one can see, the players and the dynamic objects in the frame are blurred. (b) Binary text box mask after text detection (color-inverted).

The textual data inside the score box is most commonly composed of: team names, game score, one or more clocks, symbols or logos.

The data can also be classified by the frequency in which it changes. For instance, the clock will change frequently and periodically, as opposed to the game score, which change infrequently (depending on the sport type), or the team names which do not change at all. If the score box contains semi-transparent background, then this part of the score box will change even more frequently than the clock, due to the dynamic character of the video.

1. Separating the score box into its components

As mentioned in section II, many event-detection systems use text recognition methods in order to interpret the data in the score box, and analyze the game status. Text recognition is a computationally expensive operation. In this work, however, we realize that it is not necessary to read and interpret the game score, but only to determine if the score has changed. Therefore, much simpler methods can be used in order to segment the score box.

First, we use an intensity based threshold operation to receive a binary image of the score box, as can be seen in Figure 5. This is based on the assumption that the text in the score box should be easily read, and therefore will either be bright text on dark background, or dark text on bright background. Next, the binary image of the score box is converted into a connected-component map.

Figure 5. A score box and its binary map.

2. Component inversion

In order to detect changes in the different components later on, it is important to achieve a high level of separation resolution, i.e., to segment each digit or letter as a separate component. The finer and more accurate the separation, the easier it will be to detect changes in each character of the text. An example of this is shown in Figure 6. The red rectangles represent different components of the score box. If a change occurs in the digit '2', it will be more pronounced when the digit fills the component (Figure 6(b)) than if the digit takes up a smaller part of the component (Figure 6(a)).

As a result, the case of dark text on a bright background must be treated differently than its opposite case. After applying the intensity threshold, bright text achieves a high level of separation, while dark text does not, as can be seen in figure 7 – the 3-digit score should be separated into three connected components, instead of one. In order to correct this, each component of the binary mask is inverted, and compared to its original version. The version with higher probability for finer separation is chosen to appear in the connected component map of the score box.

(a) (b)

Figure 6. (a) low and (b) high separation levels.Change detection will give better results for a higher level of separation.

C. Change detection

1. Detecting temporal changes

The change detection process is performed on each component of the score box separately. The basic operation used to detect changes is measuring the sum of absolute differences (SAD) between frames, using the binary version of the score box, so that every pixel has the same weight. The SAD result of each component is compared to a threshold.

$$\frac{SAD\left(n, n-\Delta n\right)}{\left(height \times width\right)_{component}} > threshold \tag{3}$$

Where $SAD\left(n, n-\Delta n\right)$ is the SAD of the nth video frame in comparison to the prior Δn frames.

This threshold is a function of the size of the component, even though the SAD is normalized by its size. This is done in order to deal with cases in which optimal character separation is not achieved, and a component may contain more than one character. Thus, a smaller component will receive a high threshold, while a larger component, in which every character constitutes a smaller fraction of its size, will receive a lower threshold.

Simple implementation of this method is susceptible to noise inherent in the video, which to a large part can be considered random and independent between frames. To improve noise robustness, the video first undergoes smoothing over time with a low pass filter:

$$\tilde{I}\left[n\right] = \alpha \cdot \tilde{I}\left[n-1\right] + \left(1-\alpha\right) \cdot I\left[n\right] \tag{4}$$

Where $I\left[n\right]$ is the original video, and $\tilde{I}\left[n\right]$ is the smoothed video. The smoothing factor is chosen to be $\alpha = 0.5$, in order to balance between noise reduction and maintaining image sharpness. In addition, changes in video usually occur over a number of consecutive frames, and therefore the SAD is performed between non-consecutive smoothed frames.

2. Classification of changes

The score box contains different elements, such as game score, clock, team names, etc. In order to identify the game score characters, we define an initialization time at the beginning of the game, in which we collect data about the number and frequency of changes. Assuming a maximum change frequency of 0.5Hz for the game score, elements with higher change frequency will be classified as uninteresting changes, or noise.

The first type of noise is clock changes – typically second, ten-second, and minute digits which appear in the score box. The second digit can be filtered by the maximum frequency constraint, but the rest must be filtered by checking for periodic changes. The second type of noise is usually due to transparent or semi-transparent background of either the score box or other logos which appear on screen. Unless the game score itself has a transparent background, the maximum frequency constraint can filter noise due to background transparency of other components.

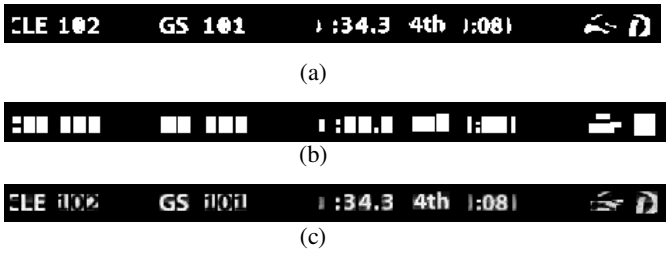


Figure 7. (a) Inversion of the dark text components after binarization. (b), (c) Final component mask of the score box.

The change detection function records the time of each change it detects, and continuously checks each component for clock or noise classification. Thus, the longer the algorithm runs, the more accurate and robust it becomes, reducing false alarms that may appear in the beginning.

IV. Results

The algorithm was tested on 24 video clips, including six different sport types (basketball, soccer, tennis, volleyball, hockey and football). The minimal resolution of the video clips was 1024×576 (Wide Screen TV).

The algorithm's test results are videos in which the detected score box is displayed in the center of the image at the time of detected score change, for a duration of 0.8 seconds. An example of the results can be seen in Figure 8.

The criteria by which we checked the successful operation of the system were:

- Detection of the score box (detection of more than one score box was not considered to be a false alarm, as the system allows this, as mentioned in section 3).
- Detection of score changes, if they occur.
- Identification and filtering of the game clock.
- Number of false alarms. A false alarm here is defined as a system "detection" of a score change when one did not actually occur (We defined the false alarm rate as the number of false score change declarations out of the number of the detected score boxes).

The results of the test are detailed in table 1.

As can be seen, the score box detection rate (correctly detected score boxes out of total number of score boxes) and score change detection rate give very good results. Some of the video clips, however, produced a number of false alarms. These false alarms were mainly due to either short clip length, or poor video quality. In the case of short video clips, the false alarms were a result of false detection of background transparency, or the minute digit

in the game clock, as a score change. One example of poor video quality was a jittering video clip due to primitive de-interlacing methods, which caused artificial changes in the video frames. For further examples see supplemental video.

Table 1. Experimental Results

Total video clips	24
Score boxes detected	23
Score box detection rate	**0.958**
Score changes occurred	23
Score changes detected	23
Score change detection rate	**1.00**
Number of clocks	27
Clocks detected	23
Clock detection rate	**0.852**
False score change declarations	4
False alarm rate	**0.174**

V. CONCLUSION

In this work we have presented a system for the detection of score changes in sport videos, implemented in three main processing stages: Score box localization based on edge and corner density analysis, score box segmentation using intensity based connected component analysis, and change detection by temporal analysis of size-dependant difference threshold. The results demonstrate the high performance and robustness of our system, while handling various types of sports. The new adaptive threshold factor presented gives greater flexibility and versatility for the text detection algorithm, enhancing the effectiveness of temporal averaging for score box detection Due to no prior assumptions about the input video, this system is not restricted to a single type of sport, nor a specific broadcasting method.

Figure 8. An example of the system's final output. The score change was detected and the score box transplanted in the center of the frame.

REFERENCES

[1] X.S. Hua, X.R. Chen, W.Y. Liu, H.J. Zhang, "Automatic location of text in video frames". ACM workshop on multimedia: multimedia information retrieval, pp 24–27, 2001.

[2] Q. Zhang, S.F. Chang, "Event Detection in Baseball Video Using Superimposed Caption Recognition," ACM Multimedia 2002, pp. 315-318, 2002.

[3] L. Agnihotri and N. Dimitrova, "Text Detection for Video Analysis," IEEE Workshop Content-Based Access of Image and Video Libraries, pp. 109-113, 1999.

[4] C. Xu, J. Wang, H. Lu, and Y. Zhang, "A novel framework for semantic annotation and personalized retrieval of sports video," IEEE Transactions on Multimedia, vol. 10, pp. 421–436, 2008.

[5] D. Chen, J.M. Odobez, H. Bourlard, "Text detection and recognition in images and video frames". Pattern Recognition 37, pp. 595–608, 2004.

[6] D. Zhang, R.K. Rajendran, and S.F. Chang, "General and domain specific techniques for detecting and recognizing superimposed text in video," IEEE 2002.

[7] V. Kobla, D. DeMenthon, and D. Doermann, "Identification of sports videos using replay, text, and camera motion features," SPIE Conference on Storage and Retrieval for Media Databases,Vol. 3972, pp. 332–343, 2000.

[8] D. Sadlier, N. O'Connor, "Event Detection in Field Sports Video Using Audio-Visual Features and a Support Vector Machine", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 15, No. 10, pp. 1225–1233, 2005.

[9] A. Padilla-Vivanco, A. Martinez-Ramirez and F. Granados-Agustin, "Digital Image Reconstruction by using Zernike Moments", Instituto Nacional de Astrofisica, Optica y Electronica, Tonantzintla, Puebla. Mexico.

[10] S. Nepal, U. Srinivasan, G. Reynolds, "Automatic Detection of `Goal' Segments in Basketball Videos," Proc. of ACM Multimedia, pp. 261-269, 2001.

[11] Derpanis, Konstantinos G. "The Harris Corner Detector", 2004.

[12] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis". New York: Academic Press, 1970.