Event Detection in Baseball Video Using Superimposed Caption Recognition

Dongqing Zhang
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA
212-854-7473
dqzhang@ee.columbia.edu

Shih-Fu Chang
Department of Electrical Engineering
Columbia University
New York, NY 10027, USA
212-854-6894
sfchang@ee.columbia.edu

ABSTRACT

We have developed a novel system for baseball video event detection and summarization using superimposed caption text detection and recognition. The system detects different types of semantic level events in baseball video including scoring and last pitch of each batter. The system has two components: event detection and event boundary detection. Event detection is realized by change detection and recognition of *game stat* texts (such as text information showing in score box). Event boundary detection is achieved using our previously developed algorithm, which detects the pitch view as the event beginning and non-active view as potential endings of the event. One unique contribution of the system is its capability to accurately detect the semantic level events by combining video text recognition with camera view recognition. Another unique feature is the real-time processing speed by taking advantage of compressed-domain approaches in part of the algorithms such as caption detection. To the best of our knowledge, this is the first system achieving accurate detection of multiple types of high-level semantic events in baseball videos.

Keywords

Videotext detection, recognition, sports video event detection, highlight extraction, summarization, retrieval.

1. INTRODUCTION

Sports video is a popular component in any broadcast television media. It has a large audience base and extensive production sources. As online video services and applications become gradually adopted, new tools and systems for adding innovative functionalities are needed. One specific need is to generate Table of Contents and summaries of long sports programs. Such information is useful for helping users navigate through large collections of sports video sources and access segments of video with specific interest to users.

Previous systems use audio-visual features to detect the events in sports videos. Gong et al [1] developed system for soccer video parsing, which is based on location recognition and ball presence detection. Sudhir et al [2] focused on tennis video analysis, They use court line detection and player tracking, which are combined using high level reasoning scheme to extract tennis play events. R.Yong et al [3] presented a system for baseball video highlight detection. Their system use audio features to detect the excited speech and pitch hit detection. A multi-channel fusion process is conducted to generate the highlights. D. Zhong et al [4] uses object segmentation and layout matching to achieve pitching view detection. Domain models are also learned and used for detecting canonical views in baseball and tennis videos. L. Xie [5] et al presents a system for structure analysis of soccer video by play and break classification. They use color, motion features and Hidden Markov Models.

These systems achieve encouraging results for video indexing. However the limitation of these systems is they are unable to detect the high-level semantic events, such as *score* and *strike out*. Furthermore, they are unable to provide the *game stat* information associated with the events, such as inning, score, strike-ball count. For video summarization services, it is important to provide such crucial information to end-users. In this paper, we present a system to extract the semantic events by superimposed caption recognition. Superimposed caption is a kind of captioned text that is overplayed on the video in the production process. The superimposed captions provide important information about the status of the game including score, ball count, out number, inning etc. Figure 1 shows the examples of superimposed caption text in baseball. It's typically called score box information in baseball also. Similar score boxes (with different types of information) can be found in other sports such as basketball, soccer, tennis etc.

Our system has two comportments: Event extraction and event boundary detection. Event extraction is to extract the events and their associated *game stat* information using videotext detection and recognition module. Event boundary detection is based on video view recognition. We expanded our previously proposed view recognition technique to detect the pitching view preceding the event, and the non-active view succeeding the event. After event detection, a visualization user interface is automatically

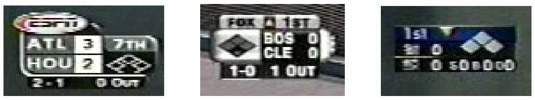


Figure 1. Score boxes in baseball videos.

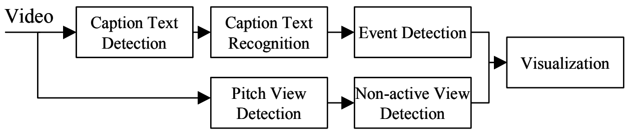


Figure 2. The system architecture.

generated to enable user random access to the video events. The architecture of the system is shown in Figure 2.

2. BASEBALL EVENTS AND THEIR SYNTACTIC STRUCTURE

Events in a baseball game fall into the following three categories: *score*, *last pitch (or batter change)*, and *base change*. Score is the event when a base runner successfully reaches the home plate and the score state is changed. Last pitch is defined as the event that is associated with the last pitching of a batter. After the last pitch of a batter, a new batter comes into play or the current inning is ended. Last pitch can be one of the following subcategories: *out*, *run*, *walk* and *score*. *Out* refers to an event that the batter is called out without going to the base (e.g., strike out or put out), *run* is the event the batter advances to the base, and *walk* is the event that the batter proceeds to the first base after the pitcher throws four balls. While *out*, *run*, *walk* occur exclusively, score may occur concurrently with each of these events. *Base change* event is the change of status of the three bases, it may or may not occur with *last pitch* or *score* event at the same time. The *base change* event excluding *last pitch* and *score* is most probably a *steal*, in which a base runner attempts to advance to the next base during pitching. In our system, we attempts to detect two types of events, namely *score*, and *last pitch,* because to the viewers, these two events are most important.

Comparing with other sports games, like soccer and basketball video, baseball video has its particular temporal syntactic structure. A baseball video segment containing above defined events typically can be decomposed into a sequence of semantic elements: pitching view, event, non-active view, replay and caption text change, as shown in figure 3. Pitching view is typically the beginning point of every event. Important events are usually accompanied with camera pan or audience cheering. At the end of an event, the camera will change focus on the players or the audience. This is the camera view we call non-active view. Typically a non-active view does not have intensive camera motion and does not cover a large portion of the field. At the end of every event, the caption box will be updated to show the latest *game stats*. The caption change includes the change of score, out number or ball count numbers. For different channels, the production rules may have slight difference. For example, in some channels event replay is shown before caption change, while in others, replay may be shown after caption change. Such variations have to be addressed during the determination of event boundaries.

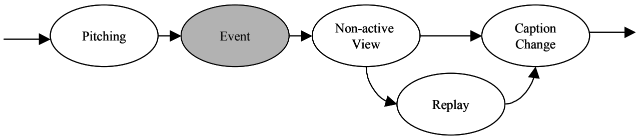


Figure 3. Syntactic structure of baseball event

3. SUPERIMPOSED CAPTION RECONGITION AND EVENT DETECTION

Based on the syntactic model described above, our system detect the occurrence of each event by detecting the incidents of caption change. Repeated captions without new information are not deemed caption change. Given a changed caption with new information, the caption text is recognized to extract the type of events and the specific state of the game. Furthermore, the pitching view and non-active views surrounding the event are detected to determine the beginning and ending points of the event.

3.1 Our Prior Work on Caption Recognition

Previously we have developed the algorithms for superimposed caption detection and recognition [6]. The system uses an initialization component to adapt to different video sources. Its output includes the bounding box, representative feature mask of the text bearing area etc. Such information is used subsequently in detecting candidate image frames that may contain text, and extracting caption keyframes in which change in superimposed text occurs. Candidate text bearing frames are further processed to extract and segment words, which are then processed for recognition. In [6], we also use a transition graph model to enhance the recognition performance of the *game stat* characters, including ball count, out, score and inning. The transition graph model models the temporal transition relationship of the *game stat* characters, and correct the falsely recognized characters by incorporating domain knowledge. For example, the baseball domain rules restrict the progressions of ball-strike counts and scores. If the current count is 1-1, then the following count is most likely 1-2 or 2-1. The model parameters are trained from training sequence. Our previous system achieved 92% accuracy for recognizing characters appearing in the caption box.

The minor modification here is we use temporal frame sampling instead of identifying caption keyframes by caption text change detection. We found frame sampling is more robust to noise for event detection by using redundant frames. The frame sampling interval is set to 4 to 8 I-frames.

3.2 Text region type classification

Slightly different from the previous system [6], here we need to recognize the types of each detected word region in the caption area. We need to distinguish different types of word regions in order to determine the type of information change in the caption area. Although for a given broadcast channel, the layout of information in the score box is fixed, there may be a great degree of variations among broadcast channels. In our system, we aim at automatic determination of word type information by using the change statistics and layout of the words.

Word regions in baseball may be of the following types: score regions, team name regions, ball count regions, out number regions, and inning regions. They cannot be identified by pure character recognition since most of them are just digits. To ensure that the system can be run on the fly, region type identification also is conducted using only the initial segment of the video. After the word region types are determined, they are assumed to be fixed for the remaining part of the video. The regions are identified as different types using text change frequency and production rule based classification. For ball count, out number, we use detection by change frequency. The incoming regions are aligned with the regions in previous sampled frames; overlapped regions are grouped into a cluster. The overlapping measurement of two region r_1, r_2 is defined as following:

$$O(r_1,r_2)=a(r_1\cap r_2)\times 2/\left[a(r_1)+a(r_2)\right] \qquad (1)$$

where $a(r)$ is the area of region *r*. If the overlap amount is significant (e.g., $O(r_1,r_2)>0.6$), then the two regions are regarded as the same type of region and put into the same cluster.

The change detection in a cluster is performed by first converting the image content within the region to the Zernike moment [6], then computing the distance between the Zernike moment features. If the distance is larger than a threshold, a region change is counted. After certain time, the change counts of clusters are sorted. The two most frequently changed regions (or one region, if there are more than one characters in one region) are identified as ball count region, the next frequently changed region is identified as out. For score, only the layout rule is used. Because the score change may not occur in the whole game. Furthermore, the layout of score region has much less variations than other types of regions in different games, thus the rule based method is viable. Two regions are identified as score regions, if they have the same heights, and exactly aligned in vertical direction. This condition is sufficient to identify score regions in different videos and robust to noise. Even if sometimes there are false identified score regions and the recognition algorithm output false results. They can be corrected by transition graph model. Inning region identification is realized by checking the region change when an out number reset occur.

After region identification, each cluster will be labeled by a region type. The region classification for incoming region then is achieved by matching the incoming region with the representative region in the saved clusters using the overlapping measurement and thresholding.

3.3 Event Detection

Two types of events are detected by the system, namely *score*, and *last pitch*. The *score* events are detected when either scores of the two teams changes. The *last pitch* event is detected when the strike ball count number is reset to 0-0 or 0-1, 1-0 (sometimes the first pitch of the next batter is missed due to replay of the highlight of the previous batter). The out number change is used as a complementary clue for *last pitch* event: if there is no ball count reset but the number of out changes, a *last pitch* event should have occurred. In this case the *last pitch* event must be missed by the system because of certain reason, for example word region extraction failure. The process can be illustrated as the decision tree shown in figure 4.

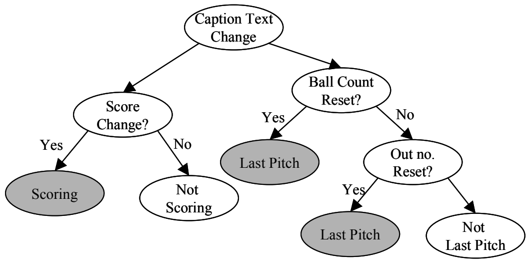


Figure 4. Decision tree for event detection.

Event detection by text change detection without domain model is sensitive to the noise. Our system achieves superior reliability of change detection by using transition graph model (TGM), which stabilize the recognition and reduce the false alarms. The following shows an example of strike ball count recognition. We can see that the false recognitions are corrected by TGM, and the false alarms caused by false recognition can also be eliminated.

Without TGM: 1-2 1-2 2-2 2-2 2-2 2-2 3-2 **1-0** 3-2 **1-0** 3-2 3-2 3-2 3-2 3-2 1-0 1-0 1-0 1-0

With TGM: 1-2 1-2 2-2 2-2 2-2 2-2 3-2 **3-2** 3-2 **3-2** 3-2 3-2 3-2 3-2 3-2 1-0 1-0 1-0 1-0

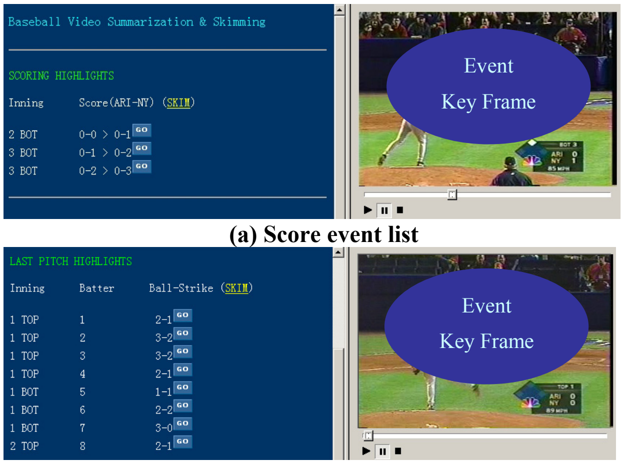
After text change detection, the time stamp of the text change point (TCP) and its preceding text (PT, which is defined as the latest text with the same semantic type appearing in the previous sampled frames) are recorded for event boundary detection, while the recognized text information is saved as *game stat* information associated with the text change event.

4. EVENT BOUNDARY DETECTION

Once we detect occurrences of events, we need to determine the exact beginning time and end time of the event in the video. Such boundary information is needed for event-based random access and editing. Caption detection does not provide such boundary information. Usually when an updated caption is shown, the corresponding event has passed. As we analyzed in section 2, a baseball video event begins with a pitching view and ends with a non-active view (we call the segment marked by such beginning and ending points as a pitch event segment). We adopt the algorithm that is developed by Zhong and Chang [4] for pitch view and non-active view detection. Pitch views are detected using color histogram matching, region segmentation, and layout analysis. Non-active view detection is realized using color and motion features. Basically, for a non-active view, the green color ratio and motion intensity is significantly lower than those in active views. To associate the score or ball count changes with its corresponding event segments, the pitch event segment nearest to PT before the TCP is taken to associate with the given text change. We use the nearest pitch event segment to PT instead of TCP because in some videos, the ball count "0-0" is not displayed, and the pitch event before TCP corresponds to the first pitch of the new batter instead of the last pitch of previous batter.

5. BROWSER FOR BASEBALL EVENTS

A baseball event browser has been developed to list all events, as shown in figure 5. The window shows the lists of score and last pitch events and also shows the video in play. User can randomly access the baseball events by clicking the "GO" button nest to each listed event. The browser also provide user with the functionality "SKIM", which if clicked will show a condensed version of video containing all extracted *score* or *last pitch* events. This is implemented by concatenating video segments corresponding to all of the events. It's interesting to compare the duration of the event skims (a few minutes) to the original length of the program (a few hours).



Baseball Video Summarization & Skimming

SCORING HIGHLIGHTS

Inning	Score(ARI-NY) (SKIM)
2 BOT	0-0 > 0-1 GO
3 BOT	0-1 > 0-2 GO
3 BOT	0-2 > 0-3 GO

(a) Score event list

LAST PITCH HIGHLIGHTS

Inning	Batter	Ball-Strike (SKIM)
1 TOP	1	2-1 GO
1 TOP	2	3-2 GO
1 TOP	3	3-2 GO
1 TOP	4	2-1 GO
1 BOT	5	1-1 GO
1 BOT	6	2-2 GO
1 BOT	7	3-0 GO
2 TOP	8	2-1 GO

(b) Last pitch event list

Figure 5. the video browsing user interface for baseball event detection.

6. PERFORMANCE OF THE SYSTEM

We evaluated the developed system on four baseball videos, three of them are US baseball video (multiple channels), and one of them is Taiwan baseball video. We deliberately include videos from different countries to test robustness of the system against variations in locations, layout styles, and even fonts of the captions. All of them are of CIF format with 352x240 resolution and 30 fps. The overall length of the video is about 6 hours.

The production rule of US video and Taiwan video has slight difference, for example "0-0" strike-ball counts do not appear in US video, thus we use two sets of parameters for the transition graph models used in the recognition error correction stage. However both parameters are trained from the same training sequence set except that the "0-0" counts are added manually into the training sequences for Taiwan video. The training sequence is extracted from another US video, which is about two hour long.

For the US baseball videos, the caption boxes appear in different positions, and their layouts have large variations, however our system handle the layout variation very well by region type identification. Table 1 lists the results for event detection of these four baseball videos. Table 2 lists the performance of *game stat* text recognition of the detected events.

Table 1. Event detection results

Evt	US 1		US 2		US 3		TW 1	
	M	F	M	F	M	F	M	F
Score	0/2	0/2	0/3	0/3	0/5	0/5	0/1	0/1
Last Pitch	1/26	0/25	1/25	3/27	1/14	2/15	2/28	1/27

Legends: in each cell, #1/#2 means #1 out of total of #2
M: Miss, FA: False alarm, TW: Taiwan, Evt: Event

Table 2. Performance of Game stat character recognition

Score	Ball Count	Out	Inning
100%	97%	100%	77%

The results show that the system achieved promising recall and precision rate. We found the accuracy is contributed largely by the use of transition graph model, which removes the noise in character recognition and reduces the false alarms of event detection. The results also show that inning number recognition has less ideal recognition performance. This is because in some videos, inning numbers are connected with other contents within the caption box, for example decorating lines. Consequently, they are missed by word region extraction due to region size filter. The false recognition is also caused by false character segmentation sometimes.

7. CONCLUSION

This paper describes a system for baseball video event detection and summarization by superimposed caption detection and recognition. The system currently extracts two important events in a baseball games, namely *score* and *last pitch*. We use video text detection and recognition to detect the gamestate text change in video caption box. The recognition is enhanced by word semantic type classification and temporal transition modeling based on domain knowledge. The overall results from testing 6 hours of videos of different channels and countries show the potential and usefulness of the system. The uniqueness of this system is its capability to detect multiple types of semantic level events, high caption recognition accuracy by incorporating domain knowledge models, real-time performance by using compressed-domain processing in the front end, and finally accurate event boundary extraction by using visual-based view recognition. The underlying methodologies can be extended to other sports domains such as basketball, tennis and soccer.

8. REFERENCES

[1] Y. Gong, T.S. Lim, and H.C. Chua, Automatic Parsing of TV Soccer Programs, IEEE International Conference on Multimedia Computing and Systems, May, 1995, pp. 167 - 174.

[2] G. Sudhir, J.C.M Lee, A.K. Jain, Automatic classification of tennis video for high-level content-based retrieval, Proceedings of IEEE International Workshop on Content-Based Access of Image and Video Database, 1998, pp. 81 – 90.

[3] Yong Rui; Anoop Gupta; Alex Acero; Automatically Extracting Highlights for TV Baseball Programs, Proc. ACM Multimedia, Oct. 2000, Los Angeles USA, pp. 105 –115.

[4] Di Zhong and Shih-Fu Chang, "Structure Analysis of Sports Video Using Domain Models", IEEE International Conference on Multimedia and Expo, August 22-25, 2001, Waseda University, Tokyo, Japan.

[5] L. Xie, S.F. Chang, et al. Algorithms and Systems for Segmentation and Structure Analysis in Soccer Video, IEEE International Conference on Multimedia and Expo, August 2002.

[6] D. Zhang, R.K. Rajendran, S.F. Chang, General and domain-specific techniques for detecting and recognizing superimposed text in video, IEEE 2002 International Conference on image processing, September 22-25, 2002.