Baseball Event Detection Using Game-Specific Feature Sets and Rules

Chih-Hao Liang
Dept. of CSIE
National Taiwan Univ.
Taipei, Taiwan
jiho@cmlab.csie.ntu.edu.tw

Wei-Ta Chu
Dept. of CSIE
National Taiwan Univ.
Taipei, Taiwan
wtchu@cmlab.csie.ntu.edu.tw

Jin-Hau Kuo
Dept. of CSIE
National Taiwan Univ.
Taipei, Taiwan
david@cmlab.csie.ntu.edu.tw

Ja-Ling Wu
Graduate Inst. of NM
National Taiwan Univ.
Taipei, Taiwan
wjl@cmlab.csie.ntu.edu.tw

Wen-Huang Cheng
Graduate Inst. of NM
National Taiwan Univ.
Taipei, Taiwan
wisley@cmlab.csie.ntu.edu.tw

***Abstract*—A framework for scrutinizing baseball videos is proposed. By applying the well-defined baseball rules, this work exactly identifies what happens in a game rather than roughly finding some interesting parts. After extracting the information changes on the superimposed caption, a rule-based decision is applied to detect meaningful events. Only three types of information, including *number of outs*, *number of scores*, and *base-occupation situation*, have to be considered in the detection process. The experimental results show the effectiveness of this framework and demonstrate some research opportunities about generating semantic-level summary or indexing for sports videos.**

I. Introduction

Sports video analysis is emerging in recent years due to the rapid advance of digital home applications and large amount of commercial benefits. Good video analyses facilitate efficient access of sports games and provide users with excellent audiovisual entertainment. The urgent requirements of high-quality sports video applications therefore prompt multimedia researchers to study in various aspects.

Conventional baseball video analyses include shot classification and highlight detection. The former studies focus on classifying video shots into one of several classical views, such as pitch view, face close-up, in-field, out-field, and audience view. Because the positions of cameras are fixed in the game and the ways of showing game progress are similar in different TV channels, many shot classification methods are proposed based on field color, face detection, camera motion, or edge detection [1, 2]. Accurate shot classification eases users with efficient browsing and retrieval. On the other hand, the studies of highlight detection try to shrink a lengthy game into a compact summary. In general, the highlight events such as scoring, hits, or nice plays only occur in a few parts of a game. Automatic highlight detection facilitates viewing the whole game in an efficient way. Many techniques based on the results of shot classification [3] or audiovisual clues [4] have been proposed.

The approaches described above let us view baseball games in more efficient ways and at different granularities. However, from the viewpoint of baseball fans, these kinds of analytic results are still not good enough. For example, he/she may want to see the detailed performance of his/her favorite player. 'Hit' and 'home run' or 'sacrificed fly' and 'fly out', which are roughly classified as highlights in previous approaches, don't mean the same thing for the player or for the fans. Therefore, a good baseball video analysis system should indicate what events exactly happen and their corresponding timestamps. Some studies [5] start to pay attention on this issue, but the results they provided such as 'outfield hit' and 'outfield out' are still far from automatically generating a professional baseball scoreboard [6].

We propose a framework that particularly detects baseball events for efficient access. On the basis of the information changes in the superimposed caption, game specific features are extracted and analyzed. They are classified into different subsets corresponding to specific events through consulting official baseball rules [6]. This work provides a great help to scrutinize a game and matches the real needs of baseball fans.

The rest of this paper is organized as follows. Section II gives the overview of the proposed approach. Based on the results of game status extraction described in Section III, we address the main idea of game-specific sets and rules in Section IV. Section V shows the experimental results and Section VI concludes this work.

II. System Overview

A. Pitch Scene Block

To obtain the structure of a baseball video, we treat the video segment between two pitch shots as a basic unit and name it *Pitch Scene Block* (PSB), as illustrated in Figure 1.

This work was partially supported by the CIET-NTU(MOE) and National Science Council of R.O.C. under the contract No. NSC93-2622-E-002-033, NSC93-2752-E-002-006-PAE, NSC93-2213-E-002-006.

According to our observation, all possible events occur in PSBs. We take Figure 1(b) as an example to describe the concept of PSB. In this PSB, several pitches are invoked by the pitcher before the batter hits out the ball. After the last pitch, the ball is hit out to the outfield, and the view switches to the outfield. After that, the ball would be returned to infield and the view changes again to show the situation of base occupation. In this example, a PSB contains five shots and conveys a meaningful event, i.e., a 'hit'. For detecting meaningful events in a baseball game, we adopt the information in the superimposed caption, which is almost shown in each shot of PSBs and changes explicitly when some events occur. Therefore, the caption information is extracted to represent the game status of each PSB, and the status changes between two consecutive PSBs are exploited to model various events.

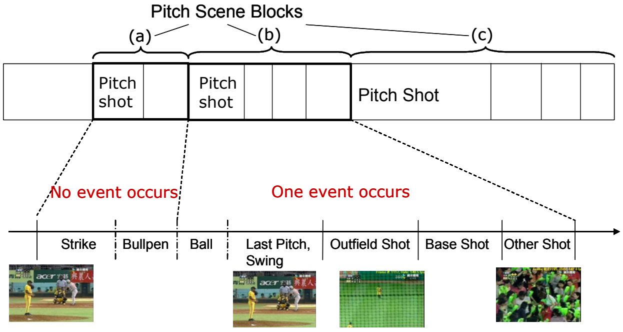

Figure 1. Examples of PSBs.

B. System Framework

Based on the visual analysis and game-specific features, the proposed framework is illustrated in Figure 2. The analysis starts with shot boundary detection and keyframe extraction. Because there is often large scale of motion in sports video, we apply a simple color-based shot boundary algorithm to segment videos. For each shot, we extract a keyframe to represent its status. Because the caption information we care are *number of outs*, *number of scores*, and *base-occupation situation* that remain unchanged in the same shot, we can just select any frame as the representative of a shot.

After keyframe extraction, the basic analysis unit (PSB) could be determined by detecting pitch shots. Two features, i.e., field-color percentage and field-color distribution, are extracted from each keyframe. A shot is declared as a pitch shot candidate if its keyframe's field-color percentage ranges from 20% to 45%, which cover most situations in different sports channels or stadiums. These candidates are further verified by checking the color profiles in horizontal and vertical axes. Because the pitcher is always in a pitch shot, we can find a valley in the vertical profile. Furthermore, the field pixels would distribute in the bottom part of the screen, so histogram values are larger in the bottom of the horizontal profiles. Details of shot boundary and pitch shot detection please refer to [7].

The game statuses displayed on the caption are extracted for each pitch shot. The information changes between two consecutive PSBs implicitly indicate the game progress and provide the clues for us to scrutinize this video. With the help of baseball rules, the event detection module discriminates a given feature into some subset, which represents one kind of events in baseball games. The results of event detection could be used in generating baseball digest and scoreboard, or further applied to event-based retrieval.

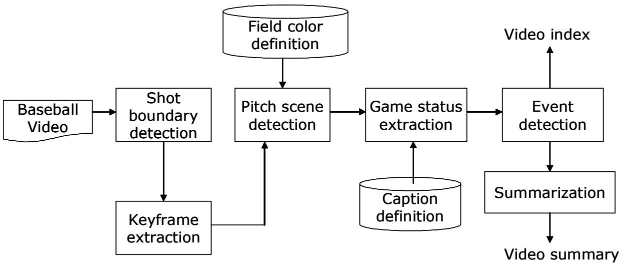

Figure 2. The proposed framework.

III. GAME STATUS EXTRACTION

According to the baseball rules, the occurrence of one meaningful event is accompanied with the changes of *base occupation*, *outs*, or *scores*. These changes can be detected by comparing the caption information between two consecutive PSBs. Two types of expression methods are usually used in caption: text and symbol. The text information often expresses the *number of scores* and *number of outs*, while the symbol information expresses the *base-occupation situation*. Of course, how to express them depends on the broadcast style of different TV channels, and how to automatically determine the displayed styles is beyond the scope of this paper.

For text information extraction, the text region is first indicated manually. To clearly show the instantaneous information, the pixels representing scores and outs always have higher intensity than others in the pre-indicated text region. We segment them out as character pixels and conduct character recognition. Zernike moments are then extracted to represent character pixels and facilitate character recognition [8]. They have been shown to give excellent performance in character recognition. After recognizing the displayed numbers, the information of scores and outs are determined.

For symbol information, we just employ the intensity-based approach similar to character pixels segmentation. In the pre-indicated region, the base-occupation situation is displayed according to whether the corresponding base is highlighted or not.

Detecting the information on superimposed caption is the shortcut to analyze baseball videos at high semantic level. Through checking the changes of game statuses, this framework effectively infers what happened in a game without performing high-complexity analysis, such as motion and edge estimation, in video frames.

IV. Game-Specific Feature Set

A. Game-Specific Features

For each PSB, the information extracted from caption includes *number of outs*, *number of scores*, and *base-occupation situation*. The changes of these statuses between two consecutive PSBs are jointly considered. They are:

(a) $o_{i,i+1}$, the difference of *number of outs* between the i^{th} and the $i+1^{th}$ PSBs, where $o_{i,i+1} \in \{0, 1, 2\}$. We don't deal with the situation of $o_{i,i+1} = 3$ because, in almost all TV channels, commercials are instantaneously inserted when three batters are out and the status resets to zero at the next inning.

(b) $s_{i,i+1}$, the difference of *number of scores* between the i^{th} and the $i+1^{th}$ PSBs, where $s_{i,i+1} \in \{0, 1, 2, 3, 4\}$. The case of $s_{i,i+1} = 4$ denotes the occurrence of a 'grand slam'.

(c) b_i and b_{i+1}, the *base-occupation situations* in the i^{th} and the $i+1^{th}$ PSBs, where b_i and $b_{i+1} \in \{0, 1, ..., 7\}$. The number of occupied bases at two consecutive PSBs (n_i and n_{i+1}) are calculated. And to catch the difference between two PSBs, the value of $b_{i,i+1}$ (= $b_{i+1} - b_i$) is also considered. The meanings of feature values are listed in TABLE I.

The features described above are concatenated as a vector $f_{i,i+1}$ to represent the game progress between two consecutive PSBs. However, many cases are illegal in baseball games. We propose a two-level decision approach that filters out the illegal features and identifies the events implied by legal features.

TABLE I. Physical Meanings of Base-Occupation Features.

b_i	n_i	Physical Meaning
0	0	No base is occupied.
1	1	Only the first base is occupied.
2	1	Only the second base is occupied.
3	2	Both the first and the second bases are occupied.
4	1	Only the third base is occupied.
5	2	Both the first and the third based are occupied.
6	2	Both the second and the third bases are occupied.
7	3	All bases are occupied.

B. Feature Filtering

When an event occurs, there may be one or no batter reaching a base, and the runners (the players who occupy the bases) would be still at bases or out or reach the home plate to get scores. Therefore, when a legal event is invoked by a batter, one of the three situations might take place:

(a) The batter is out, whether he suffers strikeout, fly out, or touch out. This case contributes one to $o_{i,i+1}$.

(b) The batter reaches a base, but no other runners are capable of reaching the home plate. This case contributes one to the number of occupied bases ($n_{i,i+1} = n_{i+1} - n_i = 1$).

(c) The most complex case is that the batter reaches a base, and some runners reach the home plate to get scores. No matter how many runners getting scores, $n_{i,i+1} + s_{i,i+1} = 1$. For example, assume that the second and the third bases are occupied in the i^{th} PSB. The batter gives a double and reaches to the second base, and both two previous-base-occupied runners reach the home plate to get two scores. The information change is $(n_{i+1} - n_i) + s_{i,i+1} = (1\text{-}2) + 2 = 1$.

According to these observations, a general decision rule for legal features can be mathematically expressed as:

$$f_{i,i+1} = \begin{cases} \textit{legal}, & \text{if } (n_{i,i+1} + s_{i,i+1} + o_{i,i+1}) = 0 \text{ or } 1, \\ \textit{illegal}, & \text{otherwise.} \end{cases}$$

The value of (n_{i+1} + $s_{i,i+1}$ + $o_{i,i+1}$), denoted as $\alpha_{i,i+1}$, indicates whether the batter changes or not. If the value is 0 (no batter changes), nothing happens in the i^{th} PSB. If the value is 1 (one batter changes), the batter is out or reaches some base, and a new batter comes at the $i+1^{th}$ PSB.

Furthermore, according to the baseball rule, no runner can go back to the previous base. We check the base-occupation situations in two consecutive PSBs and filter out this kind of illegal features. For example, it would not happen if $b_i = 2$ and $b_{i+1} = 1$ in the case of $s_{i,i+1} = 0$ and $o_{i,i+1} = 0$. (It's impossible that the occupied base is back in case of no score and number of outs changes.) This kind of error often appeared in the pure base-occupation detection approaches. Thus we can apply this rule to additionally compensate the errors occurred in feature extraction.

C. Event Identification

Twelve events are considered in this work: Hits (H), Doubles (2B), Home runs (HR), Stolen Bases (SB), Caught Stealing (CS), Fly Outs (AO), Strikeouts (SO), Base on Balls (Walk, BB), Sacrifice Bunts (SAC), Sacrifice Flies (SF), Double Plays (DP), and Triple Plays (TP). Although they still don't cover all events in baseball games, they explicitly state what happens in a game and greatly expand the visibility of baseball videos.

Given a legal feature vector, we can view the process of event identification as classifying it into a subset, which represents one baseball event. The given feature vector is first classified as one of the four types of events by checking whether the batter changes ($\alpha_{i,i+1} = 0$ or 1) and whether the number of outs ($o_{i,i+1}$) increases. Details of event categories are listed in TABLE II.

To further classify the feature vectors in each type, necessary condition rules derived from official baseball regulations are applied. The necessary condition rules indicate that a specific event is necessary for some information changes in caption. For example, a double in the i^{th} PSB is necessary for $b_{i+1} = 2$ or 6 in the $i+1^{th}$ PSB. If there is no occupied base in the i^{th} PSB, the batter hits out the ball and reaches the second base, and the caption in the $i+1^{th}$ PSB shows that the second base (and only this base) is occupied ($b_{i+1} = 2$). If there are some occupied bases in the i^{th} PSB, the batter hits out the ball and reaches the second base, one (the second) or two (the second and the third) bases would be occupied ($b_{i+1} = 2$ or 6) in the $i+1^{th}$ PSB. No other cases are legal for 2B in baseball games. By considering the necessary condition of various events, we define the rules for explicitly classifying feature vectors into a specific event. Details of rules are listed from TABLE III to TABLE VI.

Note that not all feature vectors can be mapped to one exact event. For example, in the case of no increasing in $o_{i,i+1}$ and $s_{i,i+1}$ and $b_{i+1} = 1$ in the next PSB, this approach cannot clearly determine whether a hit or a BB occurs. As shown in TABLE III, the feature set of BB is larger than that of H. Furthermore, we only tackle the case of one event in one PSB. If a hit (one event) followed by a defense error (another event) lets the batter reach the second base, the proposed approach would misdetect it as a double.

TABLE II. EVENT CATEGORIES.

	Batter change **($\alpha_{i,i+1} = 1$)**	**Batter not change** **($\alpha_{i,i+1} = 0$)**
No out **($o_{i,i+1} = 0$)**	H, 2B, 3B, HR, BB *Type 1*	SB, nothing *Type 2*
Out **($o_{i,i+1} \neq 0$)**	DP, AO, SO, SAC, SF *Type 3*	CS *Type 4*

TABLE III. NECESSARY CONDITION FOR TYPE1 EVENTS.

Event	**Necessary Condition**
H	$b_{i+1} = 1$ or 3 or 5 or 7
2B	$b_{i+1} = 2$ or 6
3B	$b_{i+1} = 4$
HR	$b_{i+1} = 0$
BB	(($b_{i+1} = 1$ or 3 or 5) and $s_{i,i+1} = 0$) or $b_{i+1} = 7$

TABLE IV. NECESSARY CONDITION FOR TYPE2 EVENTS.

Event	**Necessary Condition**
SB	$b_{i,i+1} \neq 0$
nothing	$b_{i,i+1} = 0$

TABLE V. NECESSARY CONDITION FOR TYPE3 EVENTS.

Event	**Necessary Condition**
DP	$o_{i,i+1} = 2$
AO	$b_{i,i+1} = 0$ and $o_{i,i+1} = 1$
SO	$b_{i,i+1} = 0$ and $o_{i,i+1} = 1$
SAC	($b_{i,i+1} > 0$ or $s_{i,i+1} > 0$) and $o_{i,i+1} = 1$
SF	$s_{i,i+1} > 0$ and $o_{i,i+1} = 1$

TABLE VI. NECESSARY CONDITION FOR TYPE4 EVENTS.

Event	**Necessary Condition**
CS	All other cases.

V. EXPERIMENTAL RESULTS

We use six videos with 120 minutes in total from one sports channel as the evaluation data. Note that we don't filter out commercials in advance. However, because the proposed framework only takes account of the information in pitch shots, the shots without superimposed caption are ignored and never degrade the detection performance.

TABLE VII shows the results of the event detection. Four metrics are listed: 'Correct' indicates the number of events that are correctly detected, while 'Error' indicates the number of falsely detected events. 'Contain' means that the PSB contains not only a meaningful event but also some 'nothing' events. 'Multiple' means that the PSB contains multiple events and the decided type is incorrect. The difference between 'Contain' and 'Multiple' is that 'nothing' events don't influence the result of event decision but other events will. Therefore, 'Contain' cases affect less in generating correct video summary or indexing. Overall, the performance is very promising and paves the way of constructing scoreboard automatically.

TABLE VII. PERFORMANCE OF EVENT DETECTION.

Video Clips	**Number of Events**	**Correct**	**Contain**	**Error**	**Multiple**
2003-1	15	12	1	2	0
2003-2	19	17	2	0	0
2003-3	34	28	3	3	0
2004-1	20	16	1	1	2
2004-2	21	18	3	0	0
2004-3	23	20	1	1	1

VI. CONCLUSION

We propose an approach to particularly detect baseball events. Instead of identifying what parts are highlighted, this work exactly identifies what happens in the game. On the basis of caption information and rule-based decision, the detection results greatly match fans' needs and enrich multimedia applications of baseball videos. Experiments show the convincing performance and demonstrate some clues for developing more advanced applications, such as video summarization and indexing. In the future, the combination of different audiovisual features will be conducted so as to settle the ambiguity problem in event detection.

REFERENCES

[1] W. Hua, M. Han, and Y. Gong, "Baseball scene classification using multimedia features," Proceedings of ICME, pp. 821–824, 2002.

[2] S.-C. Pei and F. Chen, "Semantics Scenes Detection and Classification in Sports Videos," Proceedings of IPPR Conference on Computer Vision, Graphics and Image Processing (CVGIP), pp. 210–217, 2003.

[3] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," Proceedings of ICIP, vol. 1, pp. 609–612, 2002.

[4] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for tv baseball programs," Proceedings of ACM Multimedia, pp. 105–115, 2000.

[5] M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method," Proceedings of ACM Multimedia, pp.347–350, 2002.

[6] Major League Baseball, http://www.mlb.com

[7] C.-H. Liang, J.-H. Kuo, W.-T. Chu, and J.-L. Wu, "Semantic units detection and summarization of baseball videos," Proceedings of IEEE International Midwest Symposium on Circuits and Systems, vol 1, pp. 297–300, 2004.

[8] A. Khotanzad and Y.-H. Hong, "Invariant Image Recognition by Zernike Moments," IEEE Trans. on PAMI, vol. 12, no. 5, pp. 489–497, 1990.