

# INTEGRATION OF RULE-BASED AND MODEL-BASED DECISION METHODS FOR BASEBALL EVENT DETECTION

*Wei-Ta Chu*

Dept. of Computer Sci. & Information Eng.  
National Taiwan University  
Taipei, Taiwan

*Ja-Ling Wu*

Grad. Inst. of Networking & Multimedia  
National Taiwan University  
Taipei, Taiwan

## ABSTRACT

To exactly detect what events occur in baseball games, a framework that integrates rule-based and model-based decision methods is proposed. The rule-based decision module infers what happened by checking the information changes in the caption. The model-based decision module further classifies the events that could not be explicitly determined by checking caption information only. Thirteen events, including hit, double, home run, and so on, are considered in this work. The promising experimental results show the effectiveness of the proposed framework and facilitate the development of advanced video applications.

## 1. INTRODUCTION

Sports video analysis has attracted much attention due to its potentially commercial values and entertainment functionalities. Many techniques have been proposed to analyze various sports videos, such as baseball, tennis, and soccer. Among them, baseball games possess strict and clear structures<sup>1</sup> and facilitate analyzing game contents automatically.

Recently, the techniques of baseball video analysis mainly focus on two phases: shot classification and game highlight detection. Because the locations of cameras in baseball games are fixed, the presentation of several classical views provides important clues to analyze baseball videos. Thus, studies of shot classification concentrate on classifying shots into pitch, infield, outfield, audience, and close-up views [1] [2]. On the other hand, based on specific visual/aural characteristics, game highlights are automatically detected [3] and facilitate rough game summarization.

Although the techniques described above certainly parse baseball videos systematically, the reported results are far from practical usage from the viewpoint of baseball fans. For example, a fan would like to see the records and their corresponding video clips of a specific player. As a baseball fan, it's more important to see "what really happened

in this game" rather than "which parts of this game may be attracted." '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 and for the fans. Therefore, techniques of explicit event detection should be devised to build the applications that can really meet users' needs.

Zhang and Chang [4] proposed an event detection method based on the caption information. However, they focused on detection of the last pitch and scoring and still didn't take care the detailed baseball events. Han et al. [5] developed a baseball digest system based on maximum entropy method and detected seven classical baseball events. However, in the reported detection results, the detection accuracy of some events is not very promising.

According the terms of Major League Baseball [6], more than ten common events lead the game progress. In our previous work [7], we elaborately exploited official baseball rules and game-specific features to detect baseball events. In this paper, events such as 'hit and walk' that cannot be determined explicitly by baseball rules are further discriminated by a newly proposed model-based method. The novelty of this work is to advance the granularity of game analyses so that it facilitates bridging semantic gaps and greater video adaptation [8].

The rest of this paper is organized as follows. Section 2 describes the overall system framework. The idea of using baseball rules to detect events is reviewed in Section 3. Section 4 describes the newly proposed model-based method, which further discriminates the events that are not tackled by the rule-based method. Section 5 shows the detection performance and conclusions are given in Section 6.

## 2. SYSTEM FRAMEWORK

We propose a framework that explicitly identifies what happened in a baseball game, as shown in Fig. 1. First, a shot classification module is developed to classify video shots into pitch, infield, outfield, close-up, and other views. For

<sup>1</sup>One baseball game usually has nine innings, each inning has top and bottom half innings, and each half inning has three outs.

each pitch shot, the system recognizes number of outs, number of scores, and base-occupation situation from superimposed caption. With the help of official baseball rules, the rule-based decision module detects events occurred between two consecutive pitch shots. Moreover, some event pairs such as ‘hit and walk’ could be further discriminated by the model-based decision module, which exploits shot transition information and specific features elaborately designed. Details of these modules are described in the following sections.

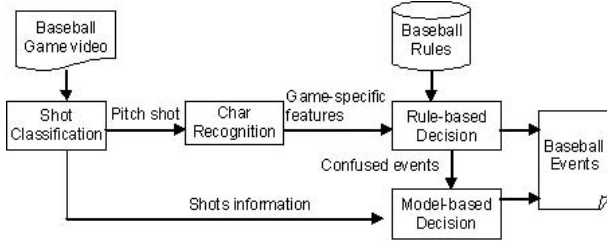


Fig. 1. The proposed system framework.

### 3. RULE-BASED DECISION

Conventional baseball video analyses focus on adopting visual or aural features to speculatively identify the highlighted parts [3]. However, we can exploit baseball rules to explicitly bridge the semantic gap between low-level features and high-level baseball events. For example, if no base is occupied in the  $i$ -th shot, and the score increases by one but still no base is occupied in the  $(i + 1)$ -th shot, we can infer that a home run (actually ‘solo home run’) occurs between these two shots. The caption information imposed and decided by human greatly assists us to detect what happened in a game. The informative caption data include ‘number of outs’, ‘number of score’, and ‘base-occupation situation.’ Each effective baseball event leads to a change of this information, such as ‘home run’ increases the score, ‘strikeout’ increases the outs, and ‘hit’ and ‘walk’ change the base-occupation situation.

#### 3.1. Caption Information Extraction

As shown in Fig. 1, we first perform shot change detection based on color histogram differences. On the basis of color distribution and edge information [7], the system further classifies each shot into five canonical views: pitch, infield, outfield, close-up, and other views. According to the common broadcasting style and baseball rules, the caption information of two consecutive pitch shots would be different if an event takes place within this duration. Therefore, we only extract caption information of pitch shots and

exploit information changes between two consecutive pitch shots to facilitate event detection.

Through the character recognition techniques, this system automatically recognizes ‘number of outs’ ( $o_i$ ) and ‘number of scores’ ( $s_i$ ) in the  $i$ -th pitch shot. Similarly, through detecting the region of high intensity, ‘base-occupation situation’ ( $b_i$ ) is also identified. The information differences ( $o_{i,i+1}$ ,  $s_{i,i+1}$ , and  $b_{i,i+1}$ ) are concatenated and denoted as  $f_{i,i+1}$  between the  $i$ -th and the  $(i + 1)$ -th pitch shots are then used to decide what happened within this duration.

#### 3.2. Rule-based Event Detection

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, we conclude that the changes of caption information are contributed by the status changes of runners and the batter. A general decision rule for legal situations can be mathematically expressed as:

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

where  $n_{i,i+1} = n_{i+1} - n_i$  is the change of number of occupied bases, and  $f_{i,i+1}$  denotes the game-specific features that clue the system in event detection. Furthermore, we abbreviate  $(n_{i,i+1} + s_{i,i+1} + o_{i,i+1})$  to  $\alpha_{i,i+1}$  to show whether the batter changes.

Given a legal feature vector, we can view the process of event identification as classifying it into a subset, which represents one baseball event, as illustrated in Fig. 2. The 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. Then, it is further classified by the necessary condition rules derived from official baseball rules. 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 pitch shot is necessary for that only the second base or both the second and third bases are occupied in the  $(i+1)$ -th pitch shot. If no base is occupied in the  $i$ -th pitch shot, the second base must be occupied in the  $(i + 1)$ -th pitch shot if a ‘double’ occurs (that’s the definition of a ‘double’). If there are more than one bases occupied in the  $i$ -th pitch shot, one (the second) or two (the second and the third) bases would be occupied in the  $(i + 1)$ -th pitch shot.

Thirteen events are considered in this work: Hit (H)<sup>2</sup>, Double (2B), Triple (3B), Home Run (HR), Stolen Base (SB), Caught Steal (CS), Fly Out (FO), Strikeout (SO), Base on Balls (BB, Walk), Sacrifice Bunt (SAC), Sacrifice Fly (SF), Double Play (DP), and Triple Play (TP). Although

<sup>2</sup>Generally speaking, ‘base hit’, ‘double’, ‘triple’, and ‘home run’ are all ‘hits’, but we only refer to ‘base hit’ as ‘hit’ in this paper.

they don't cover all possible events in baseball games, the detection results explicitly show what happened in games and greatly enhance the applications of broadcasting videos.

The rule-based decision method exactly detects most of the described events. However, some events cannot explicitly be discriminated by simply using baseball rules. For example, given that the first base is occupied and no score increases in the  $(i + 1)$ -th pitch shot, we could not precisely decide whether a 'hit' or a 'walk' occurs because they both incur the same information change. Similar cases occur in detecting 'fly out' and 'strikeout'. Therefore, we propose a model-based decision method to further detect confused events in finer granularity.

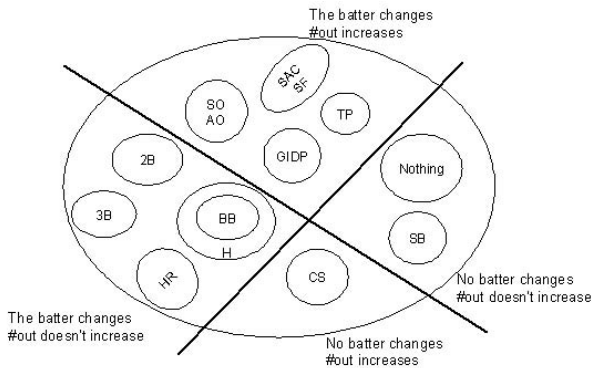


Fig. 2. Taxonomy of baseball events.

#### 4. MODEL-BASED DECISION

In this section, we describe how to utilize shot transition information as features to characterize baseball events. According to the conventional broadcasting style, shot transition information that includes the context of adjacent shots is elaborately extracted. The k-nearest models are then trained for event discrimination.

##### 4.1. Shot Context Features

According to the observation of broadcasting style and baseball rules, we elaborately propose the following features to characterize shot transition information. Note that these features are extracted within the duration from the previous effective event to current pitch shot, as shown in Fig. 3.

- *ConsecutivePF*: indicating whether a 'pitch-field' pair occurs within this duration. If the batter hits out the ball, this kind of shot pairs occur and indicate higher probability of the occurrence of 'hit' and 'fly out'. In Fig. 3, this shot pair occurs at the third-fourth shots, and *ConsecutivePF* = 1.

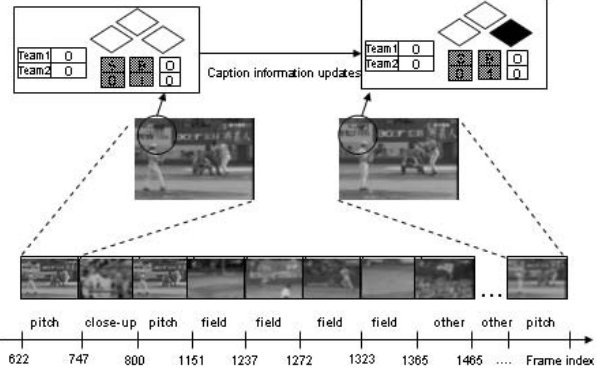


Fig. 3. An example of shot context feature extraction.

- *PitchBeforeFieldView*: indicating how many pitch shots before the first field shot. In general, more pitch shots occur before the first field shot in the events of 'walk' and 'strikeout'. In Fig. 3, *PitchBeforeFieldView* = 2.
- *DiffPitchField*: indicating the time difference between the last pitch shot and the first field shot. If the batter doesn't hit out the ball, i.e. *ConsecutivePF* = 0, *DiffPitchField* is often larger in 'walk' and 'strikeout' cases than that in 'hit' and 'fly out'.
- *FieldDuration*: indicating the time duration of the first field shot. When the ball is hit out, the duration of field shot is often short because the fielder should deal with the ball as soon as possible to prevent extra base hit. In contrast, the camera may switch to field view when no effective event occurs. But this kind of field shot often lasts long to show the overview of current situation. In Fig. 3, *FieldDuration* = 1237-1151 = 86 frames.

##### 4.2. K-nearest Neighbor Modeling

All the prescribed shot context features are normalized to the range [0,1] before training or testing. We manually selected twenty training sequences, ten of them are 'hits' and other ten sequences are 'walks', from the same TV channel to construct a 'hit-walk' classifier. K-nearest neighbor modeling is selected for each classifier due to its simplicity. Through the rule-based decision described in Section 3, the sequences decided as 'hit or walk' candidates are further discriminated by the classifier. The shot context features from the suspected sequence are then classified as a 'hit' or 'walk' event by the *k*-nearest neighbor algorithm. The same process is applied to detect fieldout or strikeout. In this work, *k* is empirically set as 8 for classification accuracy and efficiency.

## 5. EXPERIMENTAL RESULTS

We use three broadcasting baseball videos, with total length about nine hours, from two different TV channels as the evaluation data. They are recorded directly from TVs, and the commercials are not intentionally filtered out. Because the proposed framework only takes account of the caption information in pitch shots and shot transitions (pitch-field pairs), the commercials that are often classified as ‘other’ shots don’t significantly degrade the detection performance. This flexibility makes the proposed approach more practical in developing advanced applications.

Table 1. Detection results of six types of events.

Game	HB	2B	HR	O	SAC	DP
CPBL1-D	17	5	2	32	5	3
CPBL1-G	16	5	2	33	4	3
CPBL2-D	16	3	2	33	4	4
CPBL2-G	17	3	2	35	4	3
CPBL3-D	11	3	2	27	3	0
CPBL3-G	12	3	1	33	4	0

Table 2. Classification results of confused events.

Game	Hit	Walk	Strikeout	Field out
CPBL1-C	14	3	8	24
CPBL1-G	12	4	6	27
CPBL2-C	13	3	7	26
CPBL2-G	14	3	9	26
CPBL3-C	6	4	11	16
CPBL3-G	9	3	15	18

Table 1 shows the detection results of six types of events that frequently occurred in baseball games. The term ‘HB’ denotes the events of ‘hit’ or ‘walk’, and ‘O’ denotes ‘strikeout’ or ‘fly out’. The data in CPBLx-G rows indicate ground truth, and the data in CPBLx-D rows indicate the number of detected events. The detection results are very promising, while only one false alarm in ‘HB’ and ‘SAC’, and one miss in ‘O’ in the case of CPBL1. The detection performance of CPBL3 is slightly worse than other two cases because of worse shot classification or character recognition accuracy deriving from poorer video quality. Note that although only common events are shown in Table 1, other rare events could also be correctly detected by the proposed method. For example, the only ‘triple’ event in CPBL2 and the only ‘catch out’ event in CPBL3 are both correctly detected.

Table 2 shows the classification results for further discriminating confused events, i.e. ‘HB’ and ‘O’ events in Table 1. The data in CPBLx-C rows indicate the classified results, and the manually defined results are listed in CPBLx-G as the ground truth. The classification results are also very satisfactory in CPBL1 and CPBL2 games, while

present larger variations in the CPBL3 game. The shot transition patterns, which are adopted in constructing the event classifiers, often differ slightly in different TV channels and different situations. For example, different pitchers with different pitching strategies would differ the feature of *Pitch-BeforeFieldView*, and the accuracy of shot classification also affects the value of *ConsecutivePF*. However, through the simple modeling method, the proposed framework achieves satisfactory performance without being affected by game variations drastically.

## 6. CONCLUSIONS

We present a framework that integrates rule-based and model-based methods to detect what exactly happened in baseball games. On the basis of caption information changes between pitch shots, the rule-based decision module implements official baseball rules and infers which event occurs. Further, based on the canonical shot transition information in baseball games, the model-based decision module discriminates the events that could not be exactly determined by using baseball rules solely. The event detection results provide a significant foundation for developing more practical baseball video applications, such as automatic game digest generation and game summarization.

## 7. REFERENCES

- [1] M. Han W. Hua and Y. Gong, “Baseball scene classification using multimedia features,” Proc. ICME, 2002, pp. 821–824.
- [2] S.-C. Pei and F. Chen, “Semantic scenes detection and classification in sports videos,” Proc. of IPPR Conf. on Computer Vision, Graphics, and Image Processing, 2003, pp. 210–217.
- [3] A. Gupta Y. Rui and A. Acero, “Automatically extracting highlights for tv baseball programs,” Proc. of ACM Multimedia Conference, 2000, pp. 105–115.
- [4] D. Zhang and S.-F. Chang, “Event detection in baseball video using superimposed caption recognition,” Proc. of ACM Multimedia Conference, 2002, pp. 315–318.
- [5] W. Xu M. Han, W. Hua and Y. Gong, “An integrated baseball digest system using maximum entropy method,” Proc. of ACM Multimedia Conference, 2002, pp. 347–350.
- [6] Major League Baseball, <http://www.mlb.com>, 2005.
- [7] J.-H. Kuo C.-H. Liang, W.-T. Chu and J.-L. Wu, “Baseball event detection using game-specific feature sets and rules,” accepted by IEEE International Symposium on Circuits and Systems, 2005.
- [8] S.-F. Chang and A. Vetro, “Video adaptation: concepts, technologies, and open issues,” *Proceedings of the IEEE*, vol. 94, no. 1, pp. 148–158, 2005.