SEMANTIC UNITS DETECTION AND SUMMARIZATION OF BASEBALL VIDEOS

Chih-Hao Liang[1], Jin-Hau Kuo[1], Wei-Ta Chu[1], and Ja-Ling Wu[1,2], *Senior member, IEEE*
[1]Communication and Multimedia Laboratory, Department of Computer Science and Information Engineering
[2]Graduate Institute of Networking and Multimedia
National Taiwan University, Taipei, Taiwan, R.O.C.

***Abstract*-A framework for analyzing baseball videos and generation of game summary is proposed. Because of the well-defined rules of baseball games, the system efficiently detects semantic units by the domain-related knowledge, and therefore, automatically discovers the structure of a baseball game. After extracting the information changes that are caused by some semantic events on the superimposed caption, a rule-based decision tree is applied to detect meaningful events. Only three types of information, including number-of-outs, score, and base occupation status, are taken in the detection process, and thus the framework detects events and produces summarization in an efficient and effective manner. The experimental results show the effectiveness of this framework and some research opportunities about generating semantic-level summary for sports videos.**

I. INTRODUCTION

To freely access and efficiently use multimedia documents, a browsing tool that presents well-organized structure or provides effective summary is necessary. Recently, many studies on content-based video analysis and summarization have been conducted. They are mostly designed according to the physical features of the audio-video materials, and are applied to various types of videos, such as news, movie, and sports. Among these videos, sports games inherently possess well-organized structure and are widely investigated in event detection, summarization, and annotation [6][7].

By observing sports videos, important events, such as home-run hits and goals, generally occupy only some portions of the whole video. Therefore, detection of important events and production of compact summaries is an interesting and attractive research topic. For the applications of sports videos, the summary facilitates quick browsing and provides users a rough description of the whole story of the game. In this paper, we propose a framework to analyze baseball videos for creating effective video indices and summary. Baseball games are very popular and possess a well-defined structure according to game's rules. The structural components of a baseball game can be divided into *inning*, *half inning*, *batter*, and *pitch*. Since the camera locations in baseball videos are fixed, there are several typical scenes in a baseball game, such as pitch shot and the shot of left/right batter. Among them, the most important one is "pitch shot", as shown in Fig. 1. In order to clearly represent the behavior of the pitcher, catcher, and batter, there is a fixed camera behind the pitcher. A "pitch shot" may include multiple pitches until the batter hits out the ball and the screen switches to other shots, e.g., an outfielder catches the flying ball.

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 Fig. 2. By the observation of baseball videos, all possible events occur in PSBs. We take a 'single' hit as the example to describe the concept of PSB. In a PSB, several pitches may be invoked by the pitcher before the batter hits out the ball. At the last pitch to a batter, the ball is hit out to the outfield, and the displayed content switches to the outfield shot. After that, the ball would be returned to infield and the screen changes again to show the situation of infield. In the example of Fig. 2, the PSB contains four shots and conveys a meaningful event, i.e., a 'single'. For detecting the meaningful events in a baseball game, we adopt the information in the superimposed caption, which is almost shown in each shot of a PSB. The information change in the caption is exploited to model various events.

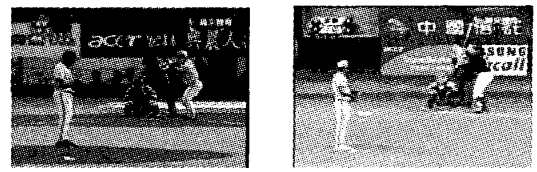


Fig. 1. Examples of *Pitch Shot.*

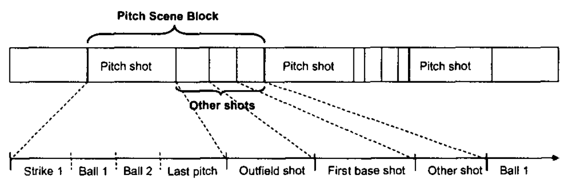


Fig. 2. The definition of *Pitch Scene Block* and the corresponding pitch shot.

Many researches in sport videos analysis are based on audio-visual features and syntactic events such as color, motion, specified audio event, field line detection, object detection, and slow motion detection [5][7]. These approaches exploit low-level features and provide some clues to semantic indexing or summarization that meets user's needs. Only a few high level semantic events, such as "play-stop", "goal", and "trace of specified objects" are defined and applied in [1]-[4]. However, in a well-structured baseball game, many other high level events occur (e.g., home-run hit, one-base hit, steal, and touch out) and drop many hints for us to analyze baseball videos automatically. Elaborate employment of these events facilitates us to analyze baseball videos in a semantic manner so that better summary could be produced. By recognizing the superimposed caption that describes the information of the game's progress, [6] presented an effective algorithm for detecting events in baseball videos. In [6], the last pitch of a pitch shot and the score status are detected first and then the event boundaries are found. We follow a similar approach to detect more semantic events of baseball videos and further identify event boundaries based on the definition of PSB. The difference between the proposed approach and [6] is that we acquire the game's progress only by *score*, *number-of-outs*, and *base occupation status* which remain the same in one single pitch shot but change if some meaningful events occur. Through detecting the change patterns between two PSBs, we develop a rule-based decision tree, which is motivated by the formal baseball rules, to detect meaningful events.

Two important issues are addressed in this paper[1]: event modeling and event boundary decision. As described above, we model the baseball events by a rule-based decision tree. The event boundaries are decided by the structure of *PSBs*. The result of this work can be applied to video summarization, indexing, and automatic game progress generation.

In Section II, we introduce baseball game's structure and present the system framework. Section III describes the algorithm for shot boundary detection and keyframe extraction. Then the methods for field color detection and pitch shot detection are addressed. The procedure to get game's progress is presented in Section IV. Section

[1] This work was partially supported by the National Science Council and the Ministry of Education of ROC under the contract No. NSC92-2622-E-002-002, NSC92-2213-E-002-023 and 89E-FA06-2-4-8.

0-7803-8346-X/04/$20.00 ©2004 IEEE

V describes the rule-based decision tree for detecting high level semantic events. Section VI shows our experimental results. Finally, Section VII concludes this paper.

II. THE DOMAIN-RELATED STRUCTURES IN BASEBALL VIDEOS

As prescribed, the structural components of one baseball game can be divided into *inning*, *half inning*, *batter*, and *pitch*. One batter usually faces several pitches. The last pitch may induce one of the following events: *out*, *run*, *walk*, and *score* which cause *batter change*. In addition, steal and caught steal events do not cause batter change.

It is important to show the information of the game's progress for audiences in time. Therefore, there are always superimposed captions overlaid on the videos. As shown in Fig. 3, the superimposed captions provide information about score, ball counts, number-of-outs, base occupation status, etc. The information difference between the superimposed captions in two PSBs, such as changes in score and base status, offers clues to detect semantic events. For example, if the difference of number-of-outs is two, a 'double play' event may occur.

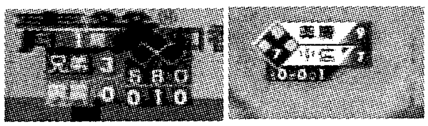


Fig. 3. The superimposed captions in baseball videos. They provide instantaneous information of *score*, *number-of-outs*, and *base occupation status*.

The block diagram of the proposed framework is shown in Fig. 4. The framework starts with shot boundary detection and keyframe extraction to offer basic content unit for analyzing. The keyframe of each shot is then analyzed so that the information of caption is extracted. Moreover, the shots other than game process, such as shots of commercials, are discarded. After computing the difference between the keyframes of two pitch shots, a rule-based decision tree is used to detect what happens within the PSB. Based on the event information, we can efficiently show the structure of baseball videos, and condense the original content to produce summary full of semantic meanings. Details of each module of the framework will be addressed in the following sections.

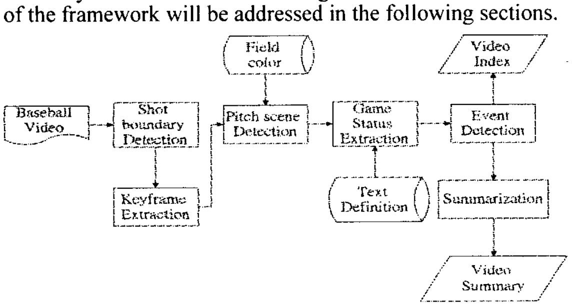


Fig. 4. Block diagram of the baseball video analyzing framework.

III. PITCH SHOT DETECTION

Shot boundary detection is the first step for analyzing video structures. Reliable shot boundary detection aids to detect PSBs accurately. Since the cameras in a baseball game are usually fixed, there are few dissolve and wipe effects in the game process. Therefore, we use an algorithm simplified from [7] to detect shot boundaries. Because sports videos often contain large motion, the shot detection algorithms based on motion analysis are not suitable. In this work, a color-based algorithm which is based on the color histogram and insensitive to motion is used to detect shot boundaries. After separating different shots, keyframes are extracted for each shot to be further analyzed. In our work, we can simply select any frame (e.g., the first frame) of a shot to be its keyframe, even in the pitch shots. Although the ball counts may change in one pitch shot, we only consider the information from 'outs' and 'base status' in this work and thus ignore the unimportant changes in a pitch shot.

The most important characteristic is the large range of field colors, including the colors of grass and soil. Therefore, the ratios of field colors are usually used to detect pitch shots [8]. In this work, we detect the field color in baseball games by a color matching method which has been successfully applied to detect field colors in soccer videos [7].

A. Color Matching

The field in a baseball video usually has two colors, one is grass and the other is soil. The field color may vary from stadium to stadium. Even in the same stadium, the field color may also vary due to weather and lighting. Therefore, we do not assume any specific color as the field color. In this work, the color matching process is taken in HSI color space, and the two field colors are indicated manually in advance. The pixels of field colors are detected according to the distance measures of the robust color cylindrical [7]. The color distance between a pixel j and the pre-defined field is calculated in HSI color space. That is,

$$d_{intensity}(j) = \left| I_j - \bar{I} \right|, \tag{1}$$

$$d_{chroma}(j) = \sqrt{(S_j)^2 + (\bar{S})^2 - 2S_j\bar{S}\cos(\theta(j))}, \tag{2}$$

$$d_{cylindrical}(j) = \sqrt{(d_{\text{intensity}}(j))^2 + (d_{\text{chroma}}(j))^2}, \tag{3}$$

$$\theta(j) = \begin{cases} \Delta(j) & if\,\Delta(j) <= 180^\circ \\ 360^\circ & otherwise \end{cases}, \tag{4}$$

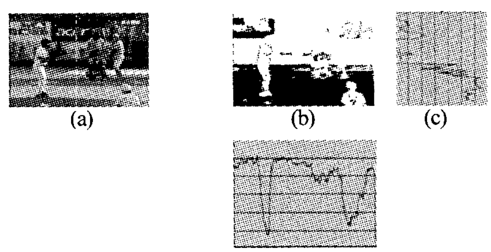
$$\Delta(j) = \left| \overline{Hue} - Hue_j \right|, \tag{5}$$

where Hue, S, I refer to hue, saturation, and intensity values, respectively. $\overline{Hue}$, $\bar{S}$, and $\bar{I}$ denote the three components of the pre-defined field color, and T is a threshold for color distance. A pixel is declared as field if

$$d_{cylindrical} < T. \tag{6}$$

B. Pitch Scene Detection

PSB is the basic analysis unit in our framework, and it can be found precisely by using the pitch scene detection algorithm given in [8]. Two features, i.e., field-color percentage and field-color distribution, are extracted to identify pitch shot. For each keyframe, the field-color percentage is calculated, and 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 distributions profiles in horizontal and vertical axes.



(a) (b) (c) (d)

Fig. 5. The vertical and horizontal projections of field colors. (a) The keyframe of a pitch shot. (b) The field color distribution after binarization. The black pixels denote the pixels of field color.(c) The horizontal histogram profiles. (d) The vertical histogram profiles.

As shown in Fig. 5, we project the field color pixels onto the horizontal and the vertical axes and construct two histograms profiles to describe the spatial layout. Because the pitcher is always in a pitch shot, we can find a valley in the vertical histogram profiles. Furthermore, the pixels with field color should distribute in the lower half part of the screen so that larger values would appear in the lower part of the horizontal histogram profiles.

IV. GAME STATUS EXTRACTION

To extract the baseball game's progress, the overlay superimposed caption provides a hint. The useful information for us is *score*, *number-of-outs*, and *base occupation status*.

A. Character Segmentation and Recognition

Firstly, we manually locate the rectangles where the useful information described above. It is time-invariant within one baseball game, especially in one specific sports channel. In order to show the instantaneous information, the *score*, and the *number-of-outs* always have higher intensity than others in the pre-indicated rectangle. We segment them out as character pixels and conduct character recognition.

Zernike moments are extracted to represent character pixels and facilitate character recognition [9]. The performance comparison in [10] shows that Zernike moments possessed the best performance in character segmentation and recognition. Zernike moments projects the character image to complex bases, which are complete orthogonal sets. A cosine distance metric is used to match the feature vector of input character to those of all characters in the database. The highest match is taken as the result of recognition. Zernike moment owns uniform scaling of image size by normalizing the character image into the interior of the unit circle. It also owns the translation uniformity with the aid of regular moments. Translation invariance is achieved by transforming the image into a new one whose horizontal and vertical first order regular moments are both equal to zero. These characteristics facilitate the improvement of the accuracy of character recognition, which is very important in detecting game status.

B. Game Status

After character recognition, the information about score and number-of-outs is available. In addition, the color of the empty base and location of three bases are pre-determined, and the *base occupation status* is detected using the same algorithm for detecting field color. Note that not every keyframe contains this information. The detection process is applied to every keyframe, including the keyframes in commercials or replay. Of course, the keyframes without this information are neglected in detecting meaningful events and would not affect the results of event detection.

Moreover, the prescribed information would not change in a PSB. According to our observation, only new PSB comes out once a meaningful event occurs. Therefore, we can just pick the information from any shot in the same PSB to represent the PSB's status. This observation leads to the fact that a meaningful event always causes some changes in status, thus we compute the status differences between two PSBs and apply them to detect meaningful events.

V. THE RULE-BASED DECISION TREE

Based on the information changes of *score*, *number-of-outs*, and *base occupation status* between two PSBs, we propose a rule-based decision tree to detect semantic events in baseball games. As shown in Fig. 6, the events detected by this process include hits (e.g. single, double, home-run, etc.), outs (e.g. double play, catch steal), and steal base. The definitions of the terms in Fig. 6 are given in Table I and Table II.

The rule-based decision tree plays the role of connecting the changes of the game's progress, and therefore, detects the events. The idea is that the events are classified first by how many 'outs' occurs. If the out number doesn't change, it means a 'hit' (including single, double, triple, and home-run), a 'walk' (walk or hit by pitch), or a 'stolen base' occurs. If the out number changes, different situations (including fly-out, catch-out, strike-out, or sacrifice out, etc.) are checked according to the number changes.

In the second step, numbers of changes in base status and score are taken into account. According to different numbers of change and base occupied situations, we finally detect the possible event occurs in one specific PSB.

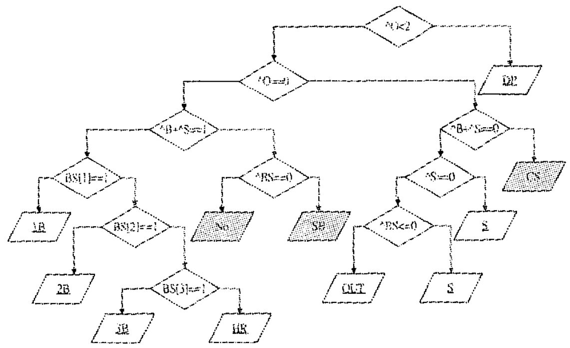


Fig. 6. The decision tree for event detection. The left arrow means that the condition is satisfied and the right means the condition is not satisfied. The events with underline mean that they will cause *batter change*.

For example, when a batter is credited with a double, he will safely reach the second base. That means the first base is empty, the second base is occupied, and there is no *number-of-outs* increasing. Since there is no out, the sums of base occupation or score will increase.

TABLE I
THE DEFINITIONS OF TERMS ABOUT A BASEBALL GAME'S PROGRESS

Term	Definition
^O	The difference of the number-of-outs between current and previous pitch scenes.
^B	The difference of the number of runners between current and previous pitch scenes.
^S	The difference of scores between current and previous pitch scenes.
BS	The base occupation status (BS). Three bits are used to indicate BS. BS[x]==1 means that the xth base is occupied, e.g. BS[3] stands for the significant bit.
^BS	The difference of base occupation status between current and previous pitch scenes.

TABLE II
THE DEFINITIONS OF TERMS ABOUT A BASEBALL GAME'S SEMANTIC EVENTS

Term	Definition	Batter Change
1B	Single, Walk	Yes
2B	Double	Yes
3B	Triple	Yes
HR	Home runs	Yes
SB	Stolen base	No
OUT	Fly out, Strike out	Yes
S	Sacrifice bunt, Sacrifice fly	Yes
DP	Ground into double play	Yes
No	Nothing happened	No
CS	Caught stealing	No

Some ambiguity problems may occur by just using the decision tree. More than one event would be considered the same because they satisfy the same conditions. For example, as shown in Table II, the events of 'single' and 'walk' could not be discriminated explicitly because they both have no change in 'out', change one in ^B+^S, and have one occupation in the first base. To solve the ambiguity, we have to take more features such as audio, speech, and motion into consideration. This, of course, is the most important topic of our future study.

After detecting those events causing *batter change*, the boundary between two successive batters is therefore detected. In the beginning of a half inning, the *number-of-outs* will be set to zero. Thus, one half inning is found if the difference of *number-of-outs* is negative. By detecting the boundaries of each batter and the half inning, the structures of a baseball video can be constructed as shown in Fig. 7.

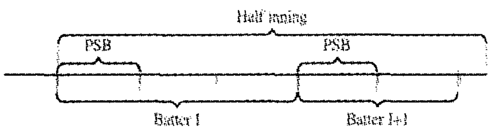


Fig. 7. The structures of a baseball video based on *PSBs*.

Each batter faces several pitches, but most of them are not interesting to audiences. To obtain a more compact summarization, we choose those meaningful *PSB*s (not only the last pitch of batters, but also the steal or caught steal of runners) to conduct the preliminary summarization. After these processes, an event-based summary is therefore produced by condensing the original video to a collection of meaningful events with PSBs.

VI. EXPERIMENTAL RESULTS

We use six baseball videos (352x240 pixels, 30fps) from one specified sports channel as the test data. The average length of a PSB is about 1.5 minutes. 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 in the process.

Table III shows the results of the semantic events detection and shows the acceptable performance. Four metrics are listed: "Hit" 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 will not influence the result of event decision but other events will. Therefore, "Contain" cases affect less in generating correct video summary.

The robustness of this event detection framework depends on the correctness of character recognition and the result of pitch shot detection. In TABLE III, some misses and false alarms come from the false detections of pitch shots. The information we take into consideration is extracted only from pitch shots. Therefore, the false or miss detection of pitch shots ("Multiple" cases) leads to information loss.

TABLE III
THE TOTAL EVENTS OF TESTED VIDEOS AND THEIR RECALLS AND PRECISIONS

	Number of PSBs	Hit	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

We produce the summary of baseball videos based on the events within *PSBs*. In baseball videos, there are many shots without any event, such as the shots of audiences or coaches. Therefore, we can condense a baseball game by just collecting the segments with meaningful events, including hits, sacrifices, stealing, and etc. to produce summary. This simple procedure can achieve good distillation ratio, especially in a game that has few meaningful events. Table IV reveals the resultant high distillation ratios of frames. Notice that the ground truths of all tests are manually defined through an experienced user. The summary also can be classified as offensive and defensive parts according to different genres of events. There are some features useful for performing summarization such as *scores* and highlights of baseball games. They would be the possible directions for improving this work.

VII. CONCLUSION AND FUTURE WORK

We use the concept of *PSB*s to detect semantic events and produce a compact summarization based on these meaningful units. This comes from the fact that there are some semantic events in PSBs, which can be applied to improve the condensing performance in producing a baseball video summary. The main contribution of this work lies in the proposal of a game's progress based rule decision tree, which is very useful for detecting semantic events within a *PSB*. Experiments show that a convincing result can be achieved. In the future, the combination of different features (such as audio and video features) will be conducted so as to settle the ambiguity problem in event detection based only on rule-based decision tree.

TABLE IV
THE EFFECTIVENESS OF THE EVENT BASED BASEBALL VIDEO SUMMARIZATION

	Total Frame	Number of Event	Event Frames	Compression Ratio
2003-1	26946	5	17793	0.66
2003-2	31351	6	15461	0.49
2003-3	48522	4	4696	0.1
2004-1	29144	4	8923	0.31
2004-2	27137	4	6918	0.25
2004-3	28000	5	9711	0.35

REFERENCES

[1] B. Li and M. Ibrahim Sezan, "Event detection and summarization in sports video," Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries, 2001, pp. 132-138.

[2] R. Leonardi and P. Migliorati, "Semantic indexing of multimedia documents," IEEE Multimedia, vol. 9, no. 2, 2002, pp. 44-51.

[3] S. Nepal, U. Srinivasan, and G. Reynolds, "Automatic detection of 'goal' segments in basketball videos," Proceedings of ACM Multimedia Conference, 2001, pp. 261-269.

[4] M. Xu, L.-Y. Duan, C.-S. Xu, and Q. Tian, "A fusion scheme of visual and auditory modalities for event detection in sports video," Proceedings of ICASSP, 2003, pp. 189-192.

[5] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," Proceedings of ACM Multimedia Conference, 2000, pp. 105 -115.

[6] D. Zhang and S.-F. Chang, "Event detection in baseball video using superimposed caption recognition," Proceedings of the tenth ACM Multimedia Conference, 2002, pp. 315-318.

[7] A. Ekin, M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," IEEE Trans. on Image Processing, vol. 12, no. 8, 2003, pp. 796-807.

[8] S.-C. Pei and F. Chen, "Semantic scenes detection and classification in sports videos," Proceedings of IPPR Conference on Computer Vision, Graphics and Image Processing, 2003, pp. 210-217.

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

[10] D. Zhang, R.K. Rajendran, and S.-F. Chang, "General and domain-specific techniques for detecting and recognizing superimposed text in video," Proceeding of International Conference on Image Processing, 2002, pp. 22-25.