Baseball Game Highlight and Event Detection

(R98942073) Wei-lun Chao

Abstract

In this final project report, I generate a video processing framework for baseball game highlight and event detection based on Major League Baseball (MLB) videos. Baseball game videos are first segmented into scene shots, and I treat the video segment between two pitch shots as the basic unit and name it Pitch Scene block (PSB) [9]. Audio analysis and shot changing rate in each PSB are used for highlight decision purpose, and I further classify these highlight segments into three general highlight events: home run, run scoring hit, and nice play. A number of key frames are extracted from each scene shots and classified into nine pre-defined frame types with the use of support vector machine (SVM), and I use this classified frame sequence of each highlight segments as the temporal feature for event classification based on hidden Markov models (HMM). The experimental results show that the framework achieves good segment extraction and event classification performance.

1. Introduction

With the ever-increasing entertaining video data, such as TV broadcasting, news, movies, and documentaries, entirely manual annotation for video database is neither feasible nor appropriate nowadays. Video archiving technology has been developed to cope with the rapidly growing database and to provide more efficient service and convenience for end users [4]. Automatic highlight extraction and event detection attract a great amount of attention recently not only because people desire to access video data more efficiently and actively, but also they perform as the medium for semantic-level video analysis. Among kinds of video, sport video are thought of containing only a part of exciting intervals while a large amount of domain-specific semantics such as sport rules, especially in baseball, basketball, and soccer games. From the “multimedia analysis and indexing” (MMAI) [14] course this semester, we have learned how to extract features and some kinds of machine learning techniques, and because the great interest in baseball, I decide to implement the baseball game highlight and event detection for my final project.

There have been many researches about sport and baseball analysis in recent years. In the scope of highlight extraction, Xie et al. [11] used HMMs for play / break segmentation in soccer games without definitely defining the meaning of states in models. Dominant color and motion information are the only two features used in

their work, and it showed the power of HMMs to bridge the semantic gap from low-level features. Rui et al. [3] exploited the announcers' excited speech and the ball-bat impact sound for baseball highlight extraction. They claimed that audio track features alone without relying on expensive-to-compute video track features is more suitable for highlight detection executed on the local set-top box using the limited compute power available. Xiong et al. [2] also used audio features while they first classified each 0.5 second window into one of the five types and then two of them are used for highlight extraction.

Besides using audio track for baseball highlight extraction, more multimodalities information have been exploited and extent the scope to event detection. Chang et al. [1] proposed a method based on HMMs and shot type classification for baseball event extraction detection including home run, nice catch, nice hit, and within diamond play. In their study, video data is first segmented into scene shots, and then the recognition is applied to the shot sequence based on HMMs in which each state represents a shot type. Zhang et al. [10] proposed an interesting idea to use the superimposed caption, especially the score boxes, in baseball videos for baseball event detection. Although this idea could directly exploit the semantic level events by the transition of numbers and states in the action area, it seems a little bit tricky and the caption area may not show all the time in the baseball video streams, so I didn't adopt this method in my project. H. S Chen [8] extracted the complete pitching action and could effectively reduce a baseball video into a denser pitch-by-pitch video summary. Readers could find more researches in [4, 5, 6, 7, 9].

Among all these techniques, there are primary three sources of information used as the features for highlight and event detection: analysis of video track, analysis of audio track, and the use of close caption information. I briefly introduce each of them and the corresponding references as follows.

- *Video track:* This class extracts features from video frames and video context for further recognition purpose. Widely-used features include dominant color, field color, motion information, field shape, and player and skin detection, etc. Related works are [1, 6, 7, 8, 11].
- *Audio track:* Audio track analyzes the audio signal in videos and classified each temporal segment (from 0.5 to a few seconds) into several pre-defined types. Based on the type of each segment or the corresponding probability of classification into each type, audio track can be used for low-complexity highlight extraction. Mel-frequency cepstral coefficients (MFCC), pitch tracking, zero-crossing rate, and energy are frequently-used low-level features, and related

works are [2, 3].

- *Close caption:* Close caption text information, speech recognition technique, and the accompanying score boxes of baseball videos are another class of information sources. It have been found that the changing of number of outs, number of scores, and base-occupation situations directly relate to near all baseball events [9], and there are some highly correlated spoken words to each kind of events. Symbolic works are [9, 10].

Besides independently using one of these three sources above, recent works tend to combine them with multimodal fusion or feature reduction / selection techniques for better performance and more functions [4, 5].

In my project implementation, I adopt a three-step framework which first segments videos into Pitch Scene block (PSB) mentioned in [9], recognizes each PSB into highlight or not, and finally classifies each highlight segment (PSB) into three highlight types: home run, run scoring hit, and nice play. I use both information of audio and video track and adopt heuristic defined threshold, SVM, and HMM for recognition purpose. Scene shot segmentation, feature extraction, shot boundary frame classification, key frame classification, and highlight and event decision are the essential steps in my work. Using this framework, we could not only separate baseball videos into pitch-by-pitch segment sequences, but also detect the event of each segment. More details about my project will be presented in following sections.

The project report is organized as follows: In section 2, I will introduce the procedure of my framework in detail, and section 3 talks about the video source I use, the video-to-frame transformation, and the shot detection technique. Adopted features from both audio and video track are described in section 4, and the semantic-level event classification is presented in section 5. Section 6 shows the experimental result and section 7 concludes this work and proposes some future works.

2. Overall approaches

The purpose of my project is to find a better segmentation of the baseball videos and detect highlights and three pre-defined events from these segments. The framework I use in this project is composed of three steps: Pitch Scene block (PSB) extraction, highlight decision, and baseball event classification, figure 2.1 shows the complete algorithm flowchart. Given a baseball video, I first implement the shot boundary extraction algorithm to separate this video into consecutive scene shot sequence. In order to fast detect the pitch shot (an example is shown in figure 2.2) from the shot sequence, I extract only one specific frame from each shot (the second

frame of each shot is used) for frame type classification: pitch shot or not. After this step, we could segment the video into a longer interval sequence, the PSB sequence, shown in figure 2.3. I use both the audio track information and shot changing rate for highlight decision, and later use the video track information to classify each highlight segment into one of the three pre-defined event.

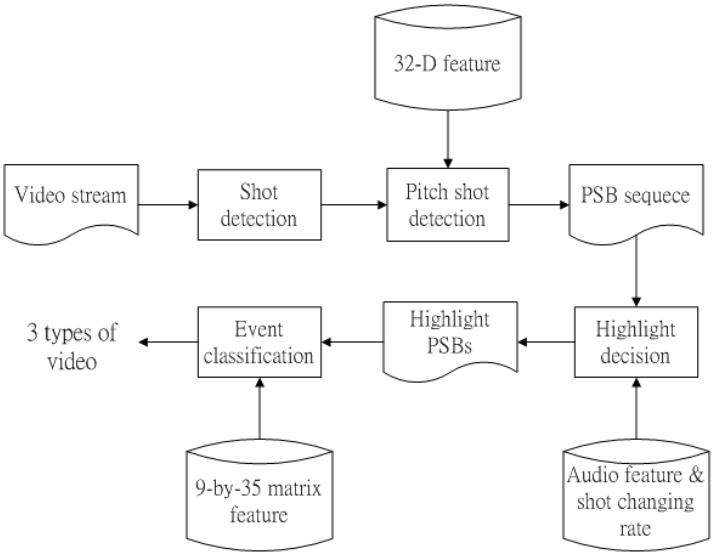


Figure 2.1: The complete flowchart of the algorithm I adopt.

Figure 2.2: An example of the pitch frame types in general MLB game videos.

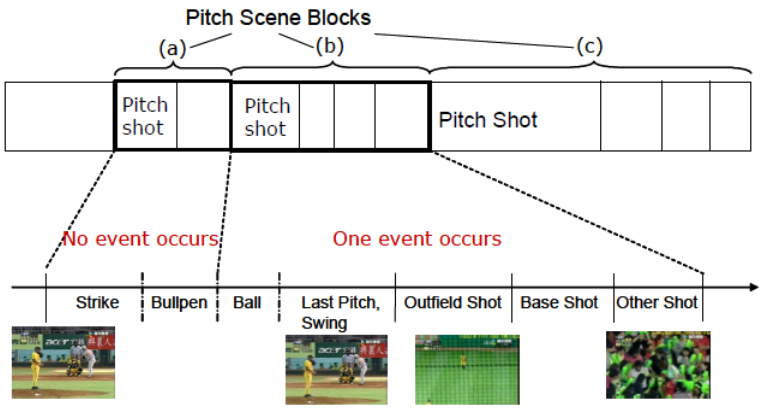


Figure 2.3: Example of PSBs [9].

2.1 Highlight decision

The audio signal of a PSB segment is first segmented into consecutive non-overlapped window with 0.5 second long, and each window is classified into one of the six pre-defined sound types with SVM algorithm: excited speech, cheering and applause, ball-bat impact sound, speech, and noise, and the former three are defined as highlight sound types. The shot changing rate is determined as follows:

$$R_{fc} = N_{shot} / T_{PSB} \quad (2.1)$$

, where N_{shot} and N_{frame} means the number of shots and the length (the unit is second) of each PSB. The ratio of highlight sound segments (RoH) in the 10%-50% (counting in the temporal order) interval of each PSB and the R_{fc} are used for highlight decision with a heuristic threshold, where a PSB is recognized as a highlight segment only when its RoH is larger than 75% and R_{fc} is larger than 0.6.

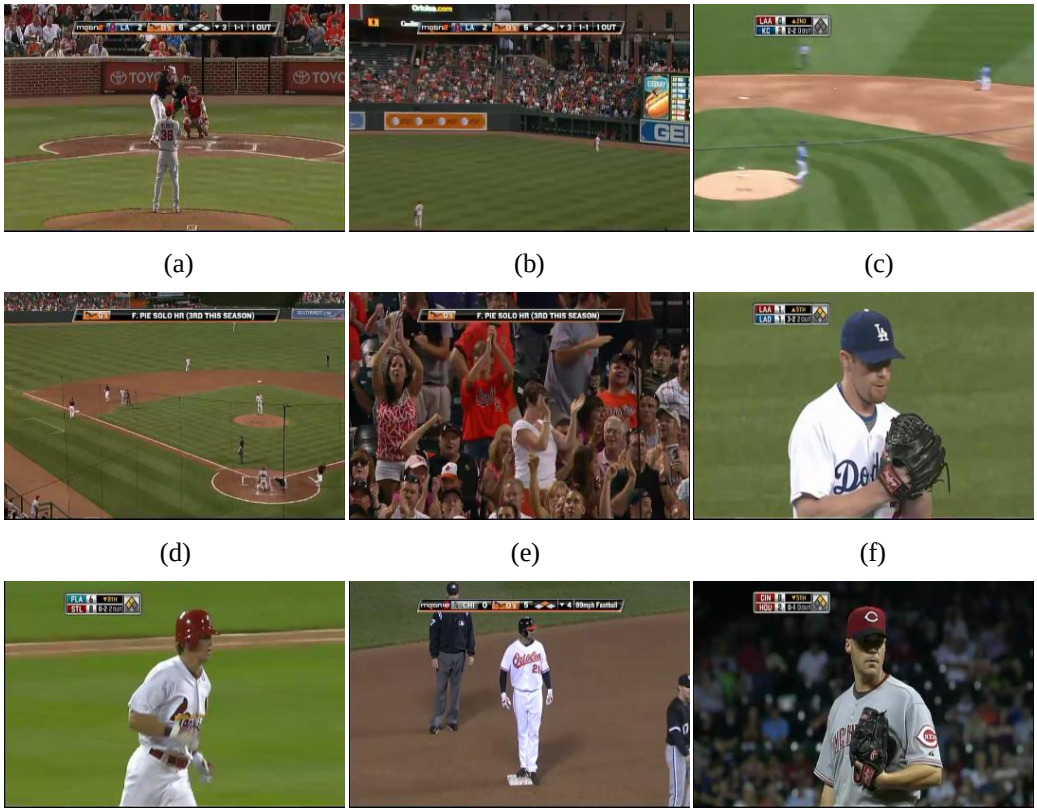
This is my original algorithm for highlight decision, but due to the difficulty to get full game videos and the bad classification results of sound type which will be mentioned in section 3 and section 5, I change the rule for highlight decision. With the observation of over 100 highlight videos of MLB games, I set a new threshold that a highlight segment is detected only when the number of shots in the corresponding PSB is more than 5.

2.2 Event classification

From the observation of highlight videos of MLB games extracted and annotated manually, I found these videos could be simply classified into three types: home run, run scoring hit, and nice play, where nice play contains a large variety of plays such as sliding catch, and double play, etc. Each shot is described with some pre-defined frame types and the HMMs are generated and learned for each event type for event classification purpose. The main reason I use HMMs combined with shot pre-defined frame types is based on the observations that (1) most highlights in baseball games are composed of certain types of scene shots and (2) those scene shots exhibit special transition context in time [1]. But instead of feeding HMMs with shot type sequence of each highlight PSB mentioned in [1, 5], I choose to use key frames sequence based on the observations that several shots are composed of kinds of views and it's hard to classify them into one certain type of shots. For example, the fielding shot with high camera motion usually contains audience view, filed view, and the player view, so it's better to define some frame types rather than shot types.

To build the input sequence for training and recognition of HMM, seven key frames

of each of the front five shots are extracted from each highlight PSB as a 35-frame sequence. Nine frame types are defined and shown in figure 2.4, and rather than hard classifying each key frame into one type, we exploit the recognition probability of a key frame into each of the nine types as the feature for each frame. Then the exact feature sequence of a highlight PSB fed into HMMs is a 9-by-35 matrix.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

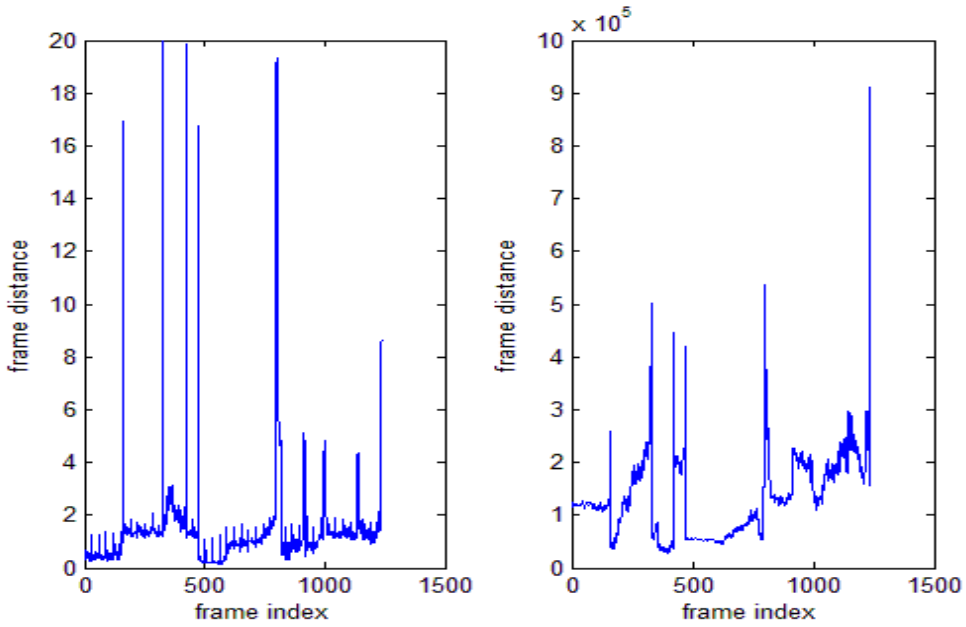
Figure 2.4: Examples of the nine pre-defined frame types: (a) pitching (b) outfield (c) fielding (d) filed (e) audience (f) close-up with filed-color background (g) running (h) base (i) closed-up with audience background.

3. Pre-processing steps

In this section, we mention the video source I use, the video-to-frame transformation, and the shot detection technique. Among several kinds of baseball games, I chose to use MLB game videos because MLB is the most popular baseball league in the world, and shot type transition seems more stable for each event among MLB game videos. It's really a difficult work to get full game videos, especially during the off-season time now. I did record four full game videos on TV but the format cannot be transformed into frames for further processing, so I decided to get videos from an on-line source [12], which offers highlight plays, for training and testing purpose. In order to facilitate the shot boundary detection algorithm, the frame

extraction function of KMPlayer [13] is used to transform a video into a frame sequence with 30 frames / s.

We have learned several shot boundary detection method from the MMAI course [14], and in this project I adopt a block-based histogram matching technique. Each frame in a frame sequence is first separated into non-overlapped blocks with equal size, and then histograms are extracted from each block and the L1 distance is computed between two consecutive frames for blocks at the same position. In my implementation, the frame size used in the project is 360-by-240, and then each frame is separated into 24 60-by-60 blocks. Totally 24 L1 distances are computed between two consecutive frames, and the 7^{th}-to-18^{th} largest distances are summed to represent the distance between two consecutive frames. This method is shown to be better than the motion-compensated block matching technique for baseball game videos in figure 3.1.



(a) (b)

Figure 3.1: The comparison of shot boundary detection performance between (a) the block-based histogram matching technique and (b) the motion-compensated block matching technique. It shows that the peaks are more obvious and easier to detect in (a) than in (b).

4. Feature extraction

I exploit features from both video track and audio track information in this project. Audio track features are used for highlight decision and video track features are used for both pitch frame recognition and the nine frame-type classification.

4.1 Audio track features

Four kinds of low-level audio features are used for audio type classification purpose, and the implementation is based on the well-known MIRtoolbox [15]. Features are extracted from each 0.03 second analysis window with 0.01 second overlap, while the texture window for audio type recognition is 0.5 second long without overlapping and takes the statistical moments of analysis windows as its feature.

- *Mel-frequency cepstral coefficients (MFCC):* MFCCs are perceptually motivated features based on STFT, and it has been mentioned that MFCC is useful for speech and music recognition. Here the first 13 delta-MFCCs are taken from each analysis window, and the mean and variance are computed for each texture window with totally 26 values.
- *Pitch:* Pitch is a popular audio feature, which measures the fundamental frequency of an audio signal. Here we expect that pitch could be used to distinguish excited speech from speech. The autocorrelation method is used for pitch tracking using MIRtoolbox built-in functions. I take only the main pitch with the highest peak in each analysis window, and compute the mean, variance, and the highest pitch as the texture-window feature.
- *E23:* This feature is used in [3] for speech and excited speech detection. The audio energy in the 630-4400 Hz sub-band is taken from each analysis window, and the mean and variance are computed for texture-window feature.
- *Zero-crossing rate*:

$$Z_t = \frac{1}{2}\sum_{n=1}^{N} |sign(x[n]) - sign(x[n-1])| \quad (4.1)$$

where the *sign* function is 1 for positive augments and 0 for negative augments and x[n] is the time domain signal at the n^{th} sample. Time domain zeros-crossing provides a measure of the noisiness of the signal. This feature is directly extracted from each texture window.

From the above four kinds of features and their statistical properties, a 32-dimensional feature vector is defined for each texture window.

4.2 Video track features

The video track features I currently used are field color, edge density, camera motion, player height, and face color. These features are extracted for each frame, and we explain them in more details.

- *Field color:* There are two main field colors in the baseball field, grass and soil. These two colors are detected using the color distribution on the HSV color

coordinate, and described as:

$$\text{Soil color: } 0.01 < H < 0.15, 0.25 < S < 0.8, V > 0.31 \quad (4.2)$$

$$\text{Grass color: } 0.15 < H < 0.46, 0.2 < S < 0.8, V > 0.31 \quad (4.3)$$

The distribution of these two colors contains a great amount of information for frame type classification as shown in figure 2.4. Here a video frame is separated into 9 non-overlapped blocks with equal size (120-by-80), and the ratio of pixels with soil or grass color is calculated for each block, besides, the ratio of soil, grass, and their sum are computed for the whole frame. Totally 21 feature values are extracted, and figure 4.1 shows an example of soil and grass color detection.

- *Edge density:* A video frame is also separated into 9 non-overlapped blocks with equal size, and the edge detection algorithm is executed in each block. The ratio of pixels recognized as edges are computed for each block. Figure 4.1 also shows an example of edge detection. This Edge density describes the pattern of highly textured region, for example, the audience or players.
- *Camera motion:* Camera motion contains important information for frame type classification, for example, running and fielding usually contain high motion, while the base and filed view contain slow motion. With the motion estimation algorithm, *N* motion vectors are extracted from two consecutive frames. I take the averaged motion along x-axis, averaged motion along y-axis, the mean squared of these two values, and the mean squared value of motion variance along x- and y-axis as the motion feature (totally four values). These four values are expected to recognize zoom-in, zoom-out, horizontal motion, and vertical motion of the camera.
- *Player height:* Player height is used for detecting camera focusing on players. Here I use a tricky method to get this quantity. Each pixel in a frame is first classified into field color and non-field color. Then this classified frame is projected onto the horizontal axis and each horizontal grid contains the ratio of field color of pixels with the same horizontal coordinate. The difference between the highest ratio and the 10^{th} least ratio is defined as the player height.
- *Face color:* Face color is used for closed-up recognition. I detect the face color density only on the central block of each frame because when the camera is focusing on a player, the player's face usually shows at the central location. The HSV color distribution is also used for face color detection defined as follows:

$$\text{Face color: } 0 < H < 0.14, 0.23 < S < 0.68 \quad (4.4)$$

There are totally 36 augments as the feature for each frame, where 21 from field color, 9 from edge density, 4 from camera motion, and 2 from player height and face color.

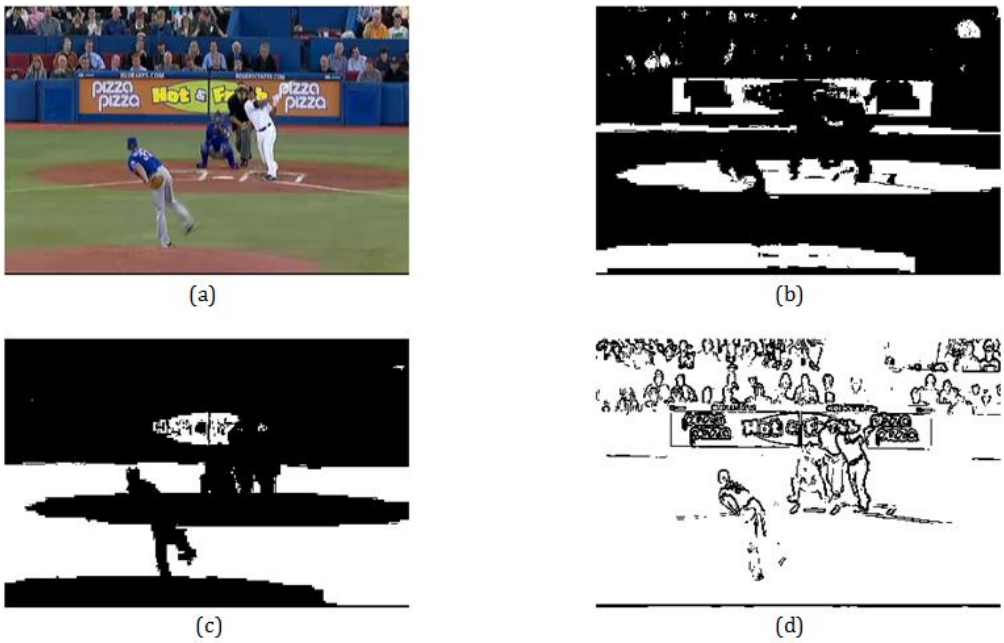


Figure 4.1: Examples for field color and edge detection. The image (a) is the original pitch frame, and (b) is the result of soil color detection. Grass color detection is shown in (3), and the edge detection result is shown in (4) with black point as the detected edges.

5. Semantic-level event classification

There are four recognition tasks: pitch shot recognition, highlight decision, nine frame-type classification, and baseball event classification, and each of them will be explained in the rest of this section.

5.1 Pitch shot recognition

From the observations of baseball videos, the pitch shot is the best starting point to extract a highlight segment. Rather than expensively computing the statistical property of the entire shot, I just select the second frame of each shot for pitch shot recognition. This is because a highlight segment always contains a short replay segment which usually starts from the pitch shot, while I don't want to cut this highlight segment into two segments. I found that this replayed pitch shots usually start with a high motion but short period team mark, which is shown in figure 5.1, and if only the second frame is used for recognition, these replayed pitch shot won't be detected. The LIBSVM [16] is adopted for recognition and only 32 augments of the video track features are used to represent each frame despite the camera motion feature.

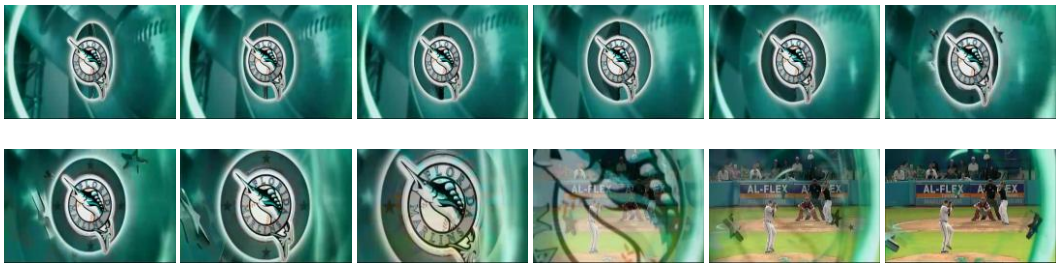


Figure 5.1: An example shows the first 12 frames (the order is from left to right and from top down) of the replayed

5.2 Highlight decision

The step for highlight decision is described in section 2.1, where the 32-dimensional feature is extracted from each 0.5 second window and is expected to be classified into the five pre-defined types: excited speech, cheering and applause, ball-bat impact sound, speech, and noise. While the classification result is not the case I expected, only the excited speech could be detected and makes this step out of function.

5.3 Nine frame-type classification

Instead of classifying shot types, in my implementation, I chose to select seven frames from each shot and classify them into nine pre-defined types as the shot representation. Here the total 36-dimensional video track feature is extracted for each frame, and the LIBSVM is used for classification purpose. While rather than directly giving the type index to each frame, I exploit the probability mode in the LIBSVM tool and then each frame is represented by 9 probabilities.

5.4 Baseball event classification

Only the first five shots are used for event classification since I found the context of the first five shots has the highest similarity among highlight segments of the same event. The 9-by-35 features are used to represent each highlight PSB and the Bayes Net Toolbox for Matlab [17] is used for building HMM models. Three HMMs (with 3, 4, and 5 states) are built and learned for each event, and during the testing process, the event corresponding to the highest likelihood model is selected as the recognized event.

6. Experimental result

Thirty highlight videos of each event type are downloaded and used for classifier training purpose. Each one of them is of 20 to 50 seconds long, and has been transformed into frame sequences with 30 frames per second. These totally ninety

videos involve different teams, stadiums, cable companies, and the day and night games. They may not start directly from the pitch shot while we can use the pitch shot detection algorithm to find the starting point of the PSB. Besides the training data, fifteen highlight videos of each event are used for testing purpose, and I'll show the result as follows.

6.1 Pitch shot detection

The 45 testing videos contain nearly 300 shots, so we test all the 300 second frames of each shot, and the result is shown in table 6.1.

Table 6.1: the result of pitch shot detection

Frame type annotated by human	Detected accuracy
Pitch frame	89%
Non-pitch frame	92%

The result shows that the pitch shot detection algorithm could find the pitch shot with a high accuracy, but I expect the result to be better if using more training samples.

6.2 Highlight decision

Due to the failure to use audio track for highlight detection, only the number of shots are used for highlight decision. The threshold is set as 5 and the result is shown in table 6.2.

Table 6.2: the result of highlight decision

Highlight types	Number of videos	Corrected decision	Wrong decision
Home run	13	13	0
Run scoring hit	13	9	4
Nice play	14	6	8

Here we discard the PSBs with wrong starting point due to the misrecognition of pitch shot, so the number of videos of each highlight type is no longer equal to 15. The result shows that using the hard defined threshold of number of shots in each PSB is not a good idea towards highlight decision, and the experiment is not complete because I only test on the annotated highlight videos but not the non-highlight videos. The worst case occurs at the nice play highlights, and it because that the nice plays sometimes don't have more than 5 shots. The failure of using audio track may come from several reasons. I didn't normalize the augments in the feature vector, and the

defined types may be too difficult to be classified. The manual annotation process is another hard task because I have to listen to each 0.5 second window for more than 20 minutes audio signal, and some of them are hard to be decided to one of the types.

6.3 Event classification

This is the most important test in my project: Event classification of the highlight PSBs, and the result is shown in table 6.3.

Table 6.3: the result of event classification

Highlight types	Number of highlights	Corrected decision	Wrong decision
Home run	13	12	1
Run scoring hit	9	7	2
Nice play	6	3	3

Here we also discard the PSBs that haven't been decided as highlight segments. The result shows that the classification for home runs and run scoring hit are really good, while the performance is still bad at the nice play highlights. And the main reason is probably that the variety of nice play segments is too large, which contains several kinds of plays such as sliding catch, nice running catch, double play and some funny plays.

7. Conclusion and future works

In this project, a three step framework for Pitch Scene block (PSB) extraction, highlight decision, and event classification for baseball game videos is proposed. Both the audio track and video track features are exploited and the SVM and HMM are used for frame and event classification. It is shown that the combination of low-level features with the machine learning algorithms could bridge the semantic gap and the high-level classifiers could be generated. There still exists a large space to be improved for future works. Only the highlight segments are tested in this project, and I expect that this algorithm could also be executed on full game videos not only filtering out the non-highlight segments but also filtering out the commercial programs between innings. Also I'd like to figure out the problem with audio track failures and enhance the event classification performance, and more accurate audio and highlight types are need to be defined.

I'd like to thank to Winston Hue and the TA for the awesome course this semester and the great project practice, and I hope to learn more in the advanced MMAI next semester.

Reference

[1] P. Chang, M. Han, and Y. Gong, "Extract highlights from baseball game video with hidden Markov models," in *Proc. IEEE Int. Conf. Image Processing*, Sept. 2002, pp. 609–612.

[2] Z. Xiong, R. Radhakrishnan, A. Divakaran, and T. S. Huang, "Audio events detection based highlights extraction from baseball, golf and soccer games in a unified framework," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, 2003, pp. 632–635.

[3] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs," in *Eighth ACM International Conference on Multimedia*, pp. 105–115, 2000.

[4] C. C. Cheng and C. T. Hsu, "Fusion of audio and motion information on HMM-based highlight extraction," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 585–599, 2006.

[5] W. Hua, M. Han, and Y. Gong, "Baseball scene classification using multimedia features," in *Proc. IEEE Int. Conf. Multimedia and Expo 2002 (ICME '02)*, vol. 1, Aug. 26–29, 2002, pp. 821–824.

[6] Lien, C. C., Chiang, C. L., Lee, C. H, "Scene-based event detection for baseball videos," Journal of *Visual Communication and Image Representation*, 1–14 (2007).

[7] R. Ando, K. Shinoda, S. Furui, and T. Mochizuki, "A robust scene recognition system for baseball broadcast using data-driven approach," in 6th *ACM international conference on Image and video retrieval (CIVR '07)*, Amsterdam, Netherlands, 2007.

[8] H. S Chen, H. T Chen, W. J Tsai, S. Y Lee, J. Y Yu (2007), "Pitch-by-pitch extraction from single view baseball video sequences," in: *Proc. ICME* 2007:1423–1426

[9] C. H. Liang, W. T. Chu, J. H. Kuo, J. L. Wu, and W. H Cheng, "Baseball event detection using game-specific feature sets and rules," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 3829–3832, 2005.

[10] Dongqing Zhang , Shih-Fu Chang, Event detection in baseball video using superimposed caption recognition, Proceedings of the 10th *ACM international conference on Multimedia*, December 01-06, 2002, Juan-les-Pins, France.

[11] L. Xie, S-F. Chang, A. Divakaran, and H. Sun, "Structure analysis of soccer video with hidden markov models," in *Proc. of ICASSP*, 2002.

[12] Mimi baseball website, http://mimi.twgg.org/

[13] KMPlayer, http://kmplayer.kde.org/

[14] W. Hsu, "Multimedia Analysis and Indexing – Course Website," 2009. [online] Available: http://www.csie.ntu.edu.tw/~winston/courses/mm.ana.idx/index.html [Accessed Jan. 17, 2009].

[15] MIRtoolbox, https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

[16] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

[17] Bayes Net Toolbox for Matlab, http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html