Tagging Webcast Text in Baseball Videos by Video Segmentation and Text Alignment

Chih-Yi Chiu, Member, IEEE, Po-Chih Lin, Sheng-Yang Li, Tsung-Han Tsai, and Yu-Lung Tsai

Abstract—Sports video annotation, an active research area in the field of multimedia content understanding, is an essential process in applications like summarization, highlight extraction, event detection, and retrieval. This paper considers the issue in relation to the annotation of baseball videos. Conventional baseball video annotation frameworks are based primarily on video content analysis, such as scoreboard recognition and machine learning techniques, which require a substantial amount of human input to collect and organize training data. The performance of such frameworks might become unstable if they encounter audiovisual patterns not included in the training data. To address the issue, we propose a novel framework for baseball video annotation that aligns high-level webcast text with low-level video content. Several cues, which are derived from the video content and webcast text, are utilized for alignment by leveraging hierarchical agglomerative clustering and genetic algorithm optimization. In addition, we develop an unsupervised method to learn the pitch segment properties of baseball videos by Markov random walk, and thereby reduce the need for human intervention substantially. Our experiments demonstrate that the proposed framework yields a robust result against a variety of video content and enhances the automaticity in baseball video annotation.

Index Terms—Multimodal fusion, unsupervised learning, genetic algorithm, event detection, video annotation

I. INTRODUCTION

Many types of sport attract huge numbers of fans worldwide, and numerous sports games are broadcast and archived every day. Sometimes people cannot watch a live broadcast or rebroadcast of a game due to space and time limitations, but they would like to watch excerpts of the game, such as the summary and highlights. Fans might also want to search a video for the segments that show specific events like “the moment of advanced scoring” or “the performance of a particular player.” To increase accessibility, archived sports videos should be presented in a structured format with meaningful annotations. However, the huge amount of video content produced for sports games poses a great challenge. For example, Major League Baseball (MLB), a professional baseball organization in the United States and Canada, is comprised of 30 teams. Each team plays 162 games in a regular season, and each game usually lasts more than two hours. Since about 5,000 hours of video data is generated every season, manual annotation would obviously be too cumbersome and impractical. Content owners therefore require an automatic approach to annotate sports videos.

Thanks to the well-defined rules in sports games, the content of sports videos can be presented in a highly-formulaic audiovisual format. In addition, professional reports, logs, and commentaries, known as external source information, are often reported on games. Utilizing the knowledge and resources is an active research area in the field of multimedia content understanding. In general, video content analysis involves using low-level audiovisual cues to detect specific events in a sports game video, while source information is parsed for high-level sports keywords to provide comprehensive details of the game. The two-level features complement one another and therefore facilitate sports video analysis. Moreover, the correlation between the video content and the source information can be explored to associate video segments with textual annotations.

Multimodal fusion of video content and source information has become a major trend in annotating videos of soccer, American football, and basketball games; however, baseball video content is still analyzed in a conventional manner due to the nature of the game. Existing multimodal fusion frameworks have difficulty handling baseball videos. For example, the timestamp information of baseball games is not preserved in video content and source information. Thus, timestamp alignment methods [29][30][31] cannot be applied in baseball video analysis. Another example is that the attack-based alignment method [16] used for soccer and basketball videos cannot be applied for baseball videos. The method detects changes of the offensive team as a feature to align video content with source information. However, there is no such change in a baseball’s half-inning because the offensive team is always the same.

Existing baseball video annotation frameworks seldom exploit source information. In their frameworks, the annotation problem is mainly transformed to an event detection task, that is, to detect specific events in videos by analyzing low-level video content. There are generally two techniques for baseball event detection, including scoreboard recognition and machine learning. In baseball videos, a scoreboard is usually superimposed on video frames to present the audience with the current game status, including the inning number, score, base occupation, and counts of strikes/balls/outs. The scoreboard recognition technique first locates the scoreboard on video frames and
then extracts the above information to infer the happening events. Some empirical rules are applied to facilitate the process. For example, Hung and Hsieh [13] collected twelve scoreboard styles as the templates. If a testing video used the same scoreboard template, the game status can be easily obtained. Chu and Wu [6] and Huang and Chen [12] assumed the scoreboard information was usually displayed in high luminance to highlight the game status; thus, the game status was searched on the high luminance region. These methods can precisely determine most of baseball events if the scoreboard is successfully found in a video. However, there is a high risk to deal with a new scoreboard pattern that is different from the templates or violates the rules. Moreover, the scoreboard is not always superimposed on video frames; sometimes there is no scoreboard to be recognized.

The machine learning technique is also widely used in baseball event detection. Unlike the scoreboard recognition technique, the machine learning technique tends to characterize the context between shots by some probabilistic models. Chang et al. [4] conducted hidden Markov models (HMMs) to detect baseball events. They classified the video shots into predefined states, and HMMs were employed to learn the state transition probability. Four HMMs were designed for classifying four events: home run, catch, hit, and diamond play. The paradigm is generally followed by subsequent studies with some modifications [1][5][18]. The Bayesian belief network is also proposed to train event classifiers in some studies [13][23]. The states and their topology of an event model are usually defined heuristically. For example, Lien et al. [18] used an ergodic HMM to represent an event while Ando et al. [1] used a left-to-right HMM. However, the predefined event model limits its generality. Actually, the state transition and topology of an event are not unique because of various presentation styles. For example, a “homerun” event may be presented in either (1) pitch → outfield → crowd or (2) pitch → outfield → close-up. Besides, the predefined event models might cause some ambiguities. The two events “infield hit” and “infield out” both have the same state transitions: pitch → infield. Since the machine learning technique only use low-level audiovisual features, without the help of the high-level semantic information, it is difficult to distinguish some ambiguous cases. That is why the machine learning technique detects fewer events than the scoreboard recognition technique does. Overall speaking, the two event detection techniques require a great deal of human effort to collect, analyze, and label video content. They work well for the cases that obey the predefined rules and training models but have difficulty handling unknown cases.

In this study, we propose a novel annotation framework for baseball videos by fusing both video content and webcast text (i.e., a type of source information). Unlike the above event detection techniques, the annotation problem is transformed to find the alignment between video content and webcast text. Several cues are extracted from both video content and webcast text, including batter counts, batter image similarity, video length estimation, change of left-handed/right-handed batter, and event likelihood, and they are fused through the proposed hierarchical agglomerative clustering or genetic algorithm optimization. In addition, we develop an unsupervised method to learn the pitch segment properties by Markov random walk, and thereby enhance the adaptability in video content analysis.

The major merit of using webcast text is it provides the definite and comprehensive description of a game. We can obtain all occurred events from webcast text without applying the event detection techniques. While scoreboards and event models may vary from game to game, webcast text is organized in a standard structure can be parsed uniformly for every game. We consider that integrating webcast text in video content analysis can effectively improve the generality and robustness of the work. Experiment results on a variety of baseball video content validate our viewpoint.

The remainder of this paper is organized as follows. Section II contains a review of related works on sports video analysis and annotation. Section III provides some background information and an overview of the proposed framework. In Sections IV and V we discuss, respectively, the content analysis and multimodal alignment components of the framework. Section VI describes and discusses extensive evaluations conducted in experiments. In Section VII, we summarize our conclusion and future work.

II. RELATED WORK

Baseball video analysis and annotation has generated a great deal of interest among researchers in the field of multimedia content understanding. For example, to classify shot types, Ando et al. [1] integrated n-gram language models and scene length information in HMMs. Audio and motion features are widely used to extract highlights in baseball videos [2][5][11][21][28]. For event detection, Zhang and Chang [32] recognized scoreboard patterns and detected changes to infer baseball events (rule-based); Lien et al. [18] employed image, object, and motion features to classify scenes, which were then input to a HMM to detect events (model-based); Lie and Shia [17], Chu and Wu [6], and Hung and Hsieh [13] combined rule-based and model-based approaches to exploit their respective advantages. In tactic analysis, Takahashi et al. [25] located the ball trajectory, catcher’s position, and scoreboard information to identify the type of a pitch. Although the above studies investigate comprehensive audiovisual features and complex learning models, they seldom utilize the source information, e.g., closed captions and webcast text, to help analyze baseball video content. Tien et al. [26] and Huang and Chen [12] employed webcast text in their systems. However, the webcast text was only used to augment additional descriptions on detected events. Baseball events were detected through the scoreboard recognition technique. Then the webcast text was tagged to the corresponding event segments. Huang and Chen’s system also employed the webcast text to double check the correctness of the event detection result. Strictly speaking, the two systems do not well utilize webcast text to help analyze video content.

However, in recent years, integrating source information with video content has attracted increasing attention in the analysis of soccer, American football, and basketball videos. The closed caption and webcast text, two types of source in-
formation, are widely used to facilitate video content analysis. A closed caption is a transcript of the broadcasters’ and commentators’ remarks, while webcast text is the sentences organized in a well-defined syntactic structure. Webcast text has the following advantages over closed captions: wide availability, precise descriptions, and a parsing-friendly structure [29][30]. Babaguchi et al. [3] detected the game time shown in a textual overlay and matched it with the event time recorded in webcast text. Gupta and Mooney [10] exploited closed captions to automatically train an activity recognizer that can be used to re-rank the retrieval results. Xu and Chua [31] identified event boundaries in videos by aligning the timestamps extracted from the video content and webcast text. Xu et al. [29] developed methods to detect the game start and recognize digital clock in order to make the alignment more precise. Xu et al. [30] further constructed a conditional random field model for alignment. Liang et al. [16] proposed an attack-based sequence matching method that aligned the semantic sequences (encoded according to the attack direction and semantic events) of video content and webcast text. Although these studies demonstrate the benefits of fusing video content and source information in some sport categories, their frameworks cannot be applied to baseball games directly.

The alignment problem in baseball videos is similar to the topic of name-face matching in TV/movie videos. The face-name matching problem tries to find the relation between face images appeared in a video and character names recorded in a script. Zhang et al. [33] built the face and name affinity networks and applied a graph matching method to associate the two networks for alignment. Sang and Xu [22] and Liang et al. [15] used a bag-of-characters concept to represent the face and name histograms in each video shot and script scene. A HMM-based method was employed to learn the relation between face and name histograms and find the maximum a-posteriori probability as the optimal alignment. For comparison, we implement a related method and discuss the performance in the experimental section.

III. FRAMEWORK OVERVIEW

A. BACKGROUND ABOUT BASEBALL GAMES AND VIDEOS

A baseball game is played by two teams and comprises regular nine innings. Each inning is divided into a top half-inning and a bottom half-inning. In a half-inning, one team “bats” (the offensive side) and the other team “fields” (the defensive side). A half-inning ends when three offensive players are recorded out. The teams alternate between batting and fielding to start the next half-inning.

The video content of a half-inning can be dissected into two types of segments, called a pitch segment and an event segment. Figure 1 shows an example of the structure of the video content of a half-inning. The pitch segment shows a pitcher throwing the baseball toward the home plate to start play, and the subsequent event segment shows the batter’s action. Note that a batter usually occupies more than one pitch/event segment. The last event segment of the batter represents his at-bat event, which is the main body of webcast text.

Webcast text, a type of external source information based on a well-defined metadata structure, is used to annotate sports videos. It has been used to facilitate event detection in soccer, American football, and basketball videos [16][29][30][31]. According to Tjondronegoro et al.’s [27] definition, the above sports belong to the period-based category, where the play period has a time limit. Figure 2(a) shows soccer’s webcast text, where each event is associated with a timestamp. Baseball, on the other hand, belongs to the set-point-based category, i.e., the play period only ends when a certain condition is satisfied (three offensive players are out). Figure 2(b) and 2(c) shows two versions of baseball’s webcast text, which record each batter’s event without any time information. The basic version shown in Figure 2(b) is used in Chinese professional baseball league (CPBL); it contains the basic half-inning information including batters’ names, bats (left/right handed), and at-bat events. The detailed version shown in Figure 2(c) is used in MLB; it records each batter’s pitch count and each pitch’s speed, type, and event, in addition to the above-mentioned basic information.

B. THE PROPOSED FRAMEWORK

Let \( \Phi \) be a half-inning in a baseball game; \( \mathcal{V}_\phi = \{ (PS_i, ES_i), (PS_{i+1}, ES_{i+1}), \ldots, (PS_N, ES_N) \} \) be the video content of \( \Phi \) with \( N \) pairs of pitch segments (PS) and event segments (ES); and \( \mathcal{V}_\psi = \{ WT_1, WT_2, \ldots, WT_M \} \) be the webcast text of \( \Phi \) with \( M \) webcast text items (WT), each of which corresponds to a batter’s at bat event; \( N \geq M \). Our objective is to map each of \( N \) video segment pairs \( (PS_i, ES_i) \) to one of the text items \( \{ WT_m \} \), correctly denoted as \( PS_i \rightarrow WT_m \); it means the \( m \)th batter’s at-bat event \( WT_m \) can be tagged to the \( i \)th video sequence (PS, ES). We call this annotation task the alignment problem between video content and webcast text of baseball videos.

The alignment problem can be formulated as a bipartite graph that models the mapping relation between \( \mathcal{V}_\phi \) and \( \mathcal{V}_\psi \). The mapping is expressed by an onto function \( f : \mathcal{V}_\psi \rightarrow \mathcal{V}_\phi \), where each element in \( \mathcal{V}_\psi \) is mapped to at least one element in \( \mathcal{V}_\phi \). Note that the mapping cannot be a crossing, i.e.,

\[
\forall i, j \in [1, N], f_i(PS_i, ES_i) = WT_m \quad \text{and} \quad f_j(PS_j, ES_j) = WT_n, \quad m \leq n \quad \text{if} \quad i < j.
\]

Figure 3 shows an example of the mapping relation.
To deal with the alignment problem, we propose a novel baseball video annotation. Our framework consists of two main components, namely, content analysis and multimodal alignment, as shown in Figure 4. Given some video content \( V_\phi \) and webcast text \( W_\phi \), we analyze the content of \( V_\phi \) in two steps: pitch segment detection and batter extraction. Since pitching can be regarded as the start (initial state) of a baseball event, detecting pitch segments is an essential step in baseball video analysis. We develop a novel unsupervised method to detect such segments. By using hierarchical agglomerative clustering and Markov random walk, our method can capture the properties of pitch segments adaptively without applying any rules or training data beforehand. In the second step, the framework searches for clues to distinguish between the batters in the detected pitch segments. Face recognition techniques are not suitable in this case, because the images of batter faces are too small and it is difficult to compile a face corpus. Therefore, we try to extract the batter’s figure from an image. Each batter’s personal appearance and batting posture are unique characteristics that distinguish him from other batters. Figure 5 shows some examples of batter images. We use the histogram of oriented gradients (HoG) [8] to detect batter images. The HoG is a dense representation that samples the dense grids in an image to detect objects. The detected object is then characterized by the SIFT-Bag descriptor [34], which leverages the power of a sparse representation of scale invariant feature transform (SIFT) [19].

The second component, multimodal alignment, extracts several cues from the video content \( V_\phi \) and webcast text \( W_\phi \) to infer the best solution to the alignment problem. Liang et al. [16] proposed using a dynamic programming algorithm to align a video sequence with a text sequence. However, the method favors diagonal mapping between the two sequences. If one sequence is much longer than the other, the mapping cost might accumulate a great proportion of the penalty from non-diagonal traversing, which would bias the mapping result. Hence, the dynamic programming algorithm is not suitable for our case, where \( N \) pitch segments are aligned with \( M \) webcast text items, and \( N \gg M \) generally. We present two alternative methods to solve the alignment problem. The first method is batter clustering, which applies hierarchical agglomerative clustering to merge batter images based on their SIFT-Bag descriptors. The clustering stop criterion is set according to the number of batters recorded in \( V_\phi \). The second method utilizes a genetic algorithm. An objective function is formulated based on the similarity measures of video content and webcast text cues, including batter images, video length estimation, change of left-handed/right-handed handed, and event likelihood. The genetic algorithm searches for the best fit solution for the objective function through the evolutionary process.

IV. CONTENT ANALYSIS

In general, a broadcasting baseball video is composed of interlaced half-innings and commercials, and each half-inning comprises of pitch segments, event segments, and other content such as close-ups and crowd. Table I lists the statistics of the video content categories in a complete baseball game. In this paper, we first manually segment the baseball video into a set of half-innings and remove commercials. The next step is to locate the pitch and event segments in the half-inning. Some empirical rules are proposed in the literature [6][13][18]. For example, Lien et al.’s method [18] counted the soil and grass regions of a frame by the predefined HSV color ranges. If the total ratio of the soil and grass regions was greater than a threshold (25% in their method), the frame was classified to the pitch/event category. Their method yields 79% recall and 70% precision in our
video dataset. The performance is unstable because not all baseball videos conform to the predefined rules. To address the generality problem, we propose a novel method to locate pitch and event segments as follows.

### A. Pitch Segment Detection

Pitch segments is the most important content in baseball video analysis. While pitch segment is located correctly, its subsequent event segment can be obtained directly. However, detecting pitch segments is not a trivial task because the content of the segments might vary substantially between different innings and games. Figure 6 shows some variations including:

- left-handed/right-handed batters and pitchers;
- runners, basemen, and umpires;
- background ads and crowds;
- color layout of the infield grass and soil;
- the teams’ colors;
- camera views; and
- logos/scoreboards superimposed by TV channels.

![Fig. 6. Various pitch frames in baseball game videos.](image)

Existing pitch segment detection methods can be categorized as either rule-based or model-based approaches. The Rule-based approach defines rules through empirical observations [6][13][18]. As we have mentioned earlier, the major limitation of the rule-based approach is its generality. The predefined rules are only applicable in particular games but might fail in other games with different grass/soil color ranges. The Model-based approach employs a supervised learning technique that needs labeled training data to learn the pattern of pitch segments. For example, Lie and Shia [17] used a support vector machine (SVM); Shih and Huang [23] employed a Bayesian belief network; and Ando et al. [1] utilized HMMs. The performance of the model-based approaches is highly dependent on the coverage of training data. They might have difficulty handling different patterns.

We propose a novel unsupervised method for pitch segment detection. Given the video content \( V_\Phi \) of a half-inning \( \Phi \), we first divide \( V_\Phi \) into several sub-shots, and apply hierarchical agglomerative clustering to merge similar sub-shots into clusters. Then, by leveraging the pitch segment’s characteristics, we identify the pitch cluster through Markov random walk. Unlike the above-mentioned approaches, the pitch classifier is only built for \( \Phi \). As a result, it characterizes the pitch properties of \( \Phi \) effectively and excludes pitch variations from other innings. Besides, the classifier is trained via unsupervised learning; the need for human intervention is reduced substantially.

1) **Sub-Shot Segmentation and Clustering**

To obtain a good clustering result, we first divide a video into a set of sub-shots with homogeneous content. The self-similarity information [9] is employed for this task. The information is acquired by constructing a self-similarity matrix in which the \((i,j)\)th element indicates the similarity between the \(i\)th and \(j\)th frames. A checkerboard kernel is used to convolute with the self-similarity matrix along the main diagonal. Local maxima in the convoluted series that are greater than a predefined threshold are considered sub-shot boundaries.

We then apply hierarchical agglomerative clustering (HAC) to merge similar sub-shots. HAC is a bottom-up clustering algorithm that treats each object as a single cluster initially and merges pairs of clusters successively until a stop criterion is satisfied. We calculate the similarity of two clusters is calculated by the Bayesian information criterion (BIC) [20] and merge the two clusters with the smallest BIC to form a new cluster. The clustering process stops when no more clusters can be merged.

2) **Markov Random Walk**

Denote the sub-shots as \( SS_1, SS_2, \ldots, SS_P \) and the clusters as \( CL_1, CL_2, \ldots, CL_S \), where \( R \) and \( S \) are the numbers of sub-shots and clusters respectively. The clusters can be regarded as Markov states, and a state transition graph can be formed by traversing the sequence of sub-shots. Figure 7 shows an example of a state transition graph. We formulate the graph as a Markov random walk problem to obtain the steady state probabilities, each of which represents the visit frequency of a cluster (state). Since a pitch segment is the initial state of every baseball event, we assume that the cluster containing pitch segments should be visited the most often. The cluster with the highest visit frequency is identified as the pitch cluster.

Let \( P \) be a Markov chain characterized by an \( S \times S \) matrix. \( P \) is constructed from three transition probability matrices \( A, B, \) and \( C \). First, we define the matrix of inter-transition probabilities \( A = [A_{ij}] \) as follows:

\[
A_{ij} = \begin{cases} 
\frac{a_{CL_iCL_j}}{\sum_{k=1}^{S} a_{CL_iCL_k}} & \text{if } i \neq j; \\
0 & \text{otherwise,} 
\end{cases}
\]

where \( a_{CL_iCL_j} \) is the number of transitions between state \( CL_i \) and \( CL_j \). Next, we define the matrix of intra-transition probabilities \( B = [B_{ij}] \) as:

\[
B_{ij} = \text{the number of transitions from } CL_i \text{ to } CL_j 
\]

![Fig. 7. An example of a state transition graph with nine sub-shots (SS) in the video sequence and six clusters (CL) derived by HAC. The arrows indicate the transitions between clusters based on the video sequence order.](image)
\[ B_{ij} = \begin{cases} \frac{\text{size}(CL_i)}{\sum_{k=1}^{S} \text{size}(CL_k)} & \text{if } i = j; \\ \frac{(1 - B_{ij})}{(S - 1)} & \text{otherwise}, \end{cases} \]

where \( \text{size}(CL_i) \) denotes the number of video frames in \( CL_i \). In addition, a matrix of uniform transition probabilities \( C = [C_{ij}] \) is derived for every pair of \( CL_i \) and \( CL_j \):

\[ C_{ij} = 1/S. \]

\( C \) is used to ensure that the transition probability between any two states is greater than zero after certain transition steps. Let \( P \) be the linear combination of \( A, B, \) and \( C \):

\[ P = \alpha \cdot A + \beta \cdot B + \gamma \cdot C, \]

where \( 0 \leq \alpha, \beta, \gamma \leq 1 \) and \( \alpha + \beta + \gamma = 1 \). Thus, we can compute \( P \)'s principal left eigenvector, \( \pi \), with the eigenvalue 1:

\[ \pi \cdot P = 1 \cdot \pi \]

\( \pi \) can be interpreted as a \( 1 \times N \) vector of steady state probabilities, and the \( i \)th entry \( \pi_i \) is the visit frequency of state \( CL_i \).

Based on the assumption that the state containing pitch segments should be visited the most often, the state with the highest visit probability in \( \pi \) is regarded as the pitch cluster. The pitch cluster might not cover all pitch frames in the half-inning; some of them are distributed over other clusters. Thus, we train a classifier based of the pitch cluster by a uni-Gaussian function to compute each pitch frame’s similarity. Frames whose similarities are greater than a predefined threshold are considered pitch frames, and continuous pitch frames are grouped into a pitch segment. Figure 8 shows the classification result of a frame sequence. Note that the non-pitch frame sequence next to a detected pitch segment is considered the corresponding event segment of the pitch segment.

**Table II**

<table>
<thead>
<tr>
<th>Event</th>
<th>Strike out</th>
<th>Walk</th>
<th>Hit by pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword</td>
<td>Strike out</td>
<td>Walk</td>
<td>Hit by pitch</td>
</tr>
<tr>
<td></td>
<td>Called out on strikes</td>
<td>Walks</td>
<td>Hit by pitch</td>
</tr>
<tr>
<td></td>
<td>Infield ground out</td>
<td>Infield fly out</td>
<td>Infield hit</td>
</tr>
<tr>
<td>Keyword</td>
<td>Grounds out</td>
<td>Pops out</td>
<td>Singles to pitcher/baseman</td>
</tr>
<tr>
<td></td>
<td>Sacrifices bunt</td>
<td>Lines out to pitcher/baseman</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Grounds into double play</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event</td>
<td>Outfield fly out</td>
<td>Outfield hit</td>
<td>Homerun</td>
</tr>
<tr>
<td>Keyword</td>
<td>Sacrifices fly</td>
<td>Singles/triples to left/right center field</td>
<td>Homerun</td>
</tr>
<tr>
<td></td>
<td>Flies/lines out to left/right</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

V. MULTIMODAL ALIGNMENT

We propose two multimodal alignment methods: 1) a HAC-based batter clustering method (abbreviated as HAC), which applies HAC to merge batter images; and 2) a genetic algorithm optimization method (abbreviated as GAO), which seeks the best alignment solution through an evolution process.

To analyze webcast text, we write a simple parser with some prior knowledge to extract the information required for the alignment. We extract the number of batters, each batter’s pitch count, batting hand (left or right), and at-bat event. The former three properties can be obtained directly from specific tags. For the at-bat event, it is usually described in a natural language sentence. We define an event-keyword table, as summarized in Table II, which associates baseball events and keywords based on MLB’s webcast text. By detecting keyword phrases in the sentence, the corresponding event can be determined. The table is extendable to enhance its generality.
A. HAC-Based Batter Clustering

Batter clustering utilizes HAC to merge batter image sets based on their similarities in terms of visual features, i.e., SIFT-Bag descriptors. Suppose that the batter image sets \( \{ BI_i \} \) and the corresponding SIFT-Bag descriptors \( \{ SB_i \}, i = 1, 2, \ldots, N, \) of the half-inning \( \Phi \) are given. Initially, each batter image set is represented by a single cluster. Let \( \omega_0 \) and \( \omega_1 \) be the \( t \)th and \( r \)th clusters respectively; \( s \neq t \). Their combination similarity is defined as:

\[
\text{sim}_{\text{batter}}(\omega_s, \omega_t) = \begin{cases} 
\frac{1}{\| SB_{\omega_s} - SB_{\omega_t} \|_1} & \text{if } \omega_s \text{ and } \omega_t \text{ are temporally adjacent;} \\
0 & \text{otherwise,}
\end{cases}
\]

where \( \| \cdot \|_1 \) returns the 1-norm of the vector. Two clusters are said to be temporally adjacent if one is next to the other. Each time, the two clusters with the highest similarity are merged to form a new cluster. For a new cluster \( \omega_0 \) that is derived by merging \( \omega_s \) and \( \omega_t \), we update its SIFT-Bag descriptor \( SB_{\omega_0} = (SB_{\omega_s,1}, SB_{\omega_s,2}, \ldots, SB_{\omega_s,K}) \) as follows:

\[
SC_{\omega_0,k} = \text{cent}(\{SC_{\omega_s,k}, SC_{\omega_t,k}\}), k = 1, 2, \ldots, K.
\]

Suppose that, according to the webcast text, there were \( M \) batters in \( \Phi \). The clustering stop criterion is satisfied when \( M \) clusters are generated; and the \( M \) clusters correspond with the \( M \) batters. Denote the \( M \) clusters as \( \{ \omega_m^s \} \) where \( m = 1, 2, \ldots, M \). The alignment can be expressed as \( \{ PS_i | SB_i \in \omega_m^s \} \rightarrow WT_m \). That is, the \( m \)th batter’s at-bat event can be tagged to the pitch segments that associate with the \( m \)th cluster. Figure 9 shows an example of a dendrogram for three-batter clustering and the alignment with webcast text. The clustering is stopped when three clusters are obtained (indicated by the dashed line).

The time complexity of the HAC-based method is analyzed as follows. Suppose there are \( N \) batter image sets (clusters) to be merged. Since only the temporal adjacent clusters are computed at the first stage, the computation time is \( N-1 \). Next, when a new cluster is generated, we compute the new cluster’s combination similarity with its two adjacent clusters at the subsequent stages. The clustering procedure repeats until \( M \) clusters left. The time complexity of the HAC-based method is about \( O(\kappa N) \), where \( \kappa \) is a constant time spent to process a batter image set.

B. Genetic Algorithm Optimization

The second method uses a genetic algorithm (GA), which is a probabilistic global search technique. In GA, the solution domain is encoded as a set of chromosomes whose fitness scores are evaluated by an objective function. By mimicking the process of natural evolution, i.e., inheritance, mutation, selection, and crossover, GA assumes a population of chromosomes will evolve toward the fittest outcome, i.e., the best solution. One advantage of GA is its flexibility in modeling constraints of an objective function. We propose a novel objective function that assesses four similarity measures for multimodal alignment, including batter image similarity, video length estimation, changes of left-handed/right handed batters, and event likelihood.

The representation scheme for chromosomes is defined as follows. Let \( A \in \{ 1, \ldots, M \} \) be an alphabet set, where \( M \) is the number of batters in a half-inning \( \Phi \) (recorded in the webcast text). A chromosome \( x \in \phi^N \) is an alphabet string of length \( N \), where \( N \) is the number of pitch segments in \( \Phi \) (detected from the video content); \( x_i \), the \( i \)th alphabet of \( x \), represents the batter index of the \( i \)th pitch segment \( PS_i \) and \( x \) is a sorted string in ascending order, i.e., \( x_i \leq x_j \) for \( i < j \). We take an example for illustration. Given three batters and nine detected pitch segments in \( \Phi \), we have \( A = \{ 1, 2, 3 \} \) and \( N = 9 \). A chromosome "111223333" represents the candidate solution for the alignment: \( \{ PS_1, PS_2, PS_3 \} \rightarrow WT_1, \{ PS_4, PS_5 \} \rightarrow WT_2, \text{and } \{ PS_6, PS_7, PS_8, PS_9 \} \rightarrow WT_3 \).

In the initial stage, we randomly select a set of chromosomes as the first population. The evolutionary operations are applied to the population and suitable chromosomes are selected to produce the next population. The selection is based on an objective function \( g(x) \), which evaluates the fitness of each chromosome \( x \). The roulette-wheel scheme is employed to select chromosomes for a mating pool with probabilities proportional to their fitness, after which the one-point crossover operation is used to generate offspring chromosomes from the mating pool. The mutation operation randomly changes a chromosome’s alphabet in the mating pool with a predefined probability. A new chromosome that is generated through the crossover or mutation operation is sorted again to keep its alphabets in ascending order. We then apply the elitism mechanism to copy the best chromosomes generated so far into the new population.

The design of the objective function \( g(x) \) in GA is a critical issue. In this study, we define \( g(x) \) as a blending function:

\[
g(x) = a \cdot BS_x + b \cdot VL_x + c \cdot LR_x + d \cdot EL_x.
\]

\( BS_x, VL_x, LR_x, \) and \( EL_x \) are the measures of chromosome \( x \) with respect to batter image similarity, video length estimation, change of left/right handed batters, and event likelihood, respectively; \( a, b, c, \) and \( d \) are the corresponding coefficients in the interval \( [0, 1] \); and \( a + b + c + d = 1 \). These measures are detailed in the following.

1) Batter Image Similarity

This measure evaluates the intra-similarity among the batter image sets that belong to the same cluster label. Here, cluster
labels are assigned by a chromosome directly. Let \( \phi_{x,m} = \{ i \mid x_i = m \} \) be the index cluster of alphabet \( m \), and let \( \phi_x = \{ \phi_{x,m} \mid m = 1, 2, ..., M \} \) for the chromosome \( x \). We define the batter image similarity for the chromosome as

\[
BS_x = 1 - \frac{\sum_{m=1}^{M} |\phi_x - \phi_{x,m}|}{\sum_{m=1}^{M} |\phi_{x,m}|}
\]  

(10)

In the above equation, the numerator is the total intra-distance between each batter image set and its corresponding cluster, and the denominator is a normalized term that ensures the fraction is within the range \([0, 1]\). Note we subtract the fraction from one to recast the distance metric as a similarity metric.

\( BS \) is basically a modified version of batter clustering discussed in Section V.A. An example of chromosome \( x = "111223333" \) is shown in Figure 10. We have \( \phi_{x,1} = \{1, 2, 3\} \), \( \phi_{x,2} = \{4, 5\} \), and \( \phi_{x,3} = \{6, 7, 8, 9\} \). \( \phi_{x,1} \) means that the batter image sets \( B_1, B_2, \) and \( B_1 \) belong to the first batter according to \( x \). The sets’ intra-distance, i.e., \( \sum_{i \in \phi_{x,1}} \|SB_i - SB_{\phi_{x,1}}\| \), should be small if they represent the same batter. The batter image sets in \( \phi_{x,2} \) and \( \phi_{x,3} \) are calculated in the same manner.

\[
\begin{array}{c|cccc}
\text{Batter cluster} & 1 & 2 & 3 & 4 \\
\hline
\text{Chromosome} & \phi_{x,1} & \phi_{x,2} & \phi_{x,3} \\
SB_1 & SB_2 & SB_3 & SB_4 \\
SB_2 & SB_2 & SB_2 & SB_3 \\
SB_3 & SB_3 & SB_3 & SB_3 \\
\end{array}
\]

Fig. 10. An example of clusters of batter image sets and the chromosome \( x = "111223333." \)

2) Video Length Estimation

This measure is based on an intuitive assumption that a batter’s pitch count is proportional to the length of the video segment occupied by the batter. The pitch count recorded in webcast text is the number of pitches thrown by a pitcher. Let \( \tau \) be the total pitch count in \( \Phi \), and let \( r_m \) be the pitch count of the \( m \)th batter. For chromosome \( x \), the length of the pitch segment of the \( m \)th batter is estimated as follows:

\[
T_{\phi_{x,m}} = \sum_{i \in \phi_{x,m}} \text{duration}(PS_i).
\]  

(11)

where \( \text{duration}(PS) \) returns the duration (in seconds) of pitch segment \( PS \). We then define the video length similarity

\[
VL_x = 1 - \frac{\sum_{i=1}^{M} \|r_{\phi_{x,i}} - \frac{r_m \cdot \tau}{T}\|_2}{2T},
\]  

(12)

where \( T = \sum_{i=1}^{M} \text{duration}(PS_i) \); and \( \frac{r_m \cdot \tau}{T} \) is the pitch count ratio of the \( m \)th batter, which is used to estimate the length of the pitch segment of the \( m \)th batter. The numerator sums the differences between two estimated pitch segment lengths: one is from chromosome \( x \), and the other is from the pitch count ratio. The denominator is a normalized term.

\( VL \) measures the fitness of chromosome \( x \) by considering both video content and webcast text in terms of the length of the video segments. An example in which \( x = "111223333" \) is illustrated in Figure 11. \( r_1 \), \( r_2 \), and \( r_3 \) are the pitch counts for the first, second, and third batters respectively; and \( T = r_1 + r_2 + r_3 \).

For the first batter, the video segment length estimated by chromosome \( x \) is \( T_{PS_1} = \text{duration}(PS_1) + \text{duration}(PS_2) + \text{duration}(PS_3) \), and the estimated length using the pitch count ratio is \( \frac{r_1 \cdot r}{T} \cdot T \). A high \( VL \), indicates that the difference between the two estimates is small, and shows that chromosome \( x \) conforms to our pitch count assumption.

3) Change of Left/Right Handed Batters

A baseball player is either a left-handed or right-handed batter. The batting order of a team lineup is usually a combination results of the batter images expressed by

\[
U = \{ u \mid u = \min(\phi_{x,m}) \text{if } bat_m \neq bat_{m+1}, m \in [2, M]\}.
\]  

(13)

where \( bat_m = "L" \) or "R" represents left-handed or right-handed for the \( m \)th batter, respectively, recorded in webcast text. \( U \) contains the change points obtained according to the classification results of the batter images expressed by

\[
V = \{ v \mid v = i \text{ if classifyBat}(BI_i) \neq classifyBat(BI_{i+1}), i \in [1, N - 1]\}.
\]  

(14)

where function \( classifyBat(BI) = "L" \) or "R" returns the classification result for left-handed or right-handed, respectively, of the \( i \)th batter image set \( BI \). The classification function is implemented by the SVM-based HOG detectors presented in Section IV.B. The similarity between \( U \) and \( V \) is calculated by the Jaccard coefficient

\[
LR_x = \frac{|U \cap V|}{|U \cup V|},
\]  

(15)

where \( | \cdot | \) returns the cardinality of the set.

\( LR \) utilizes a useful property of batters derived from the video content and webcast text. The change of left-handed/right-handed batters sets a definite boundary between two adjacent batters in the lineup, as well as for the batter image sets. We use the example of chromosome \( x = "111223333." \) to illustrate the point, as shown in Figure 12. Suppose that, according to the webcast text, the three batters in
the half-inning are left-handed, right-handed, and left-handed respectively; \( U = \{3, 5\} \). Meanwhile, the classification result for the batter image sets is “LLRRRRLLL”; \( F = \{2, 5\} \). Thus we have \( L_R = 1/3 \).

4) Event Likelihood

The last measure considers baseball events. In the half-inning \( \Phi \), the action of each batter displayed in the video content and recorded in the webcast text can be classified into one of several predefined baseball events, such as a “homerun” or a “strike out.” The video content and webcast text are therefore related through their associated baseball events. Recall that in Section IV.A, we have located pitch segments \( \{PS_i\} \) and corresponding event segments \( \{ES_i\} \). The measure \( EL_x \) is defined as

\[
EL_x = \frac{\sum_{m=1}^{M} classifyEvent_{WT_m}(ES_{\text{max}}(\phi_{x,m}))}{M}
\]

(16)

where \( ES_{\text{max}}(\phi_{x,m}) \) is the event segment of the \( m \)th batter. Function \( classifyEvent_{WT_m}(ES) \) returns the likelihood that \( ES \) belongs to the at-bat event \( WT_m \). The classification function of each event category is trained by the SVM-based SIFT-Bag Kernel [34]. In this study, we adopt Lie and Shia’s basic categories to classify baseball events, namely, non-hitting, infield, and outfield [17]. In the testing phase, for the \( m \)th batter, we obtain his at-bat event \( WT_m \) from webcast text, and input his event segment \( ES_{\text{max}}(\phi_{x,m}) \) to the corresponding classifier.

The output likelihood is in the interval \([0, 1]\). The event segments might mix partial unrelated content such as close-ups and crowd. We empirically excerpt the first ten seconds of the event segment for classification since the unrelated content is usually located at the tail of the event segment.

Let us consider the example in Figure 13. Suppose the three batters’ events recorded in the webcast text are “out on strikes,” “flies out to center fielder,” and “grounds out,” which are categorized "non-hitting," "outfield," and "infield" categories respectively. For chromosome \( x = “111223333”\), the at-bat event segments are \( ES_3, ES_5, \) and \( ES_9 \). They are input to the corresponding classification functions and obtain the event likelihood of \( x: EL_{“111223333”} = \frac{(classifyEvent_{\text{non-hitting}}(ES_3) + classifyEvent_{\text{outfield}}(ES_5) + classifyEvent_{\text{infield}}(ES_9))}{3} \).

By combining the above-mentioned four measures, we obtain the objective function score of a chromosome \( x \). After several iterations of evolution, we select the best chromosome as the alignment result. Figure 14 shows an example of the GAO method and the alignment with webcast text.
VI. EXPERIMENT

To evaluate the proposed framework, we compiled a video dataset for use in several experiments. Dozens of baseball games played in the MLB 2008 regular season were recorded from TV broadcasts as our baseball video dataset. They vary in terms of the teams, stadiums, and broadcasting channels. The dataset’s format is 720×480 frame pixels and 29.97 frames per second. The corresponding webcast text was downloaded from the MLB official website (http://www.mlb.com).

A. Preliminary Study

First, we consider the performance of the main components of the proposed framework, namely, pitch segment detection, batter detection, left-handed/right-handed batter classification, and event classification.

1) Pitch Segment Detection

We selected fourteen half-innings from four games in the video dataset, as listed in Table III. The spatial color-based feature was extracted to represent video frames. We partitioned each video frame into 4×4 non-overlapping blocks, and computed each block’s average intensities of the Y, Cb, and Cr channels. Hence, a video frame was represented by a 48-dimensional feature vector. The parameters were configured by \( \eta = 2.5, \theta = 1, \rho = 0.01, \) and \((a, \beta, \gamma) = (0.4, 0.5, 0.1)\).

For comparison, we implemented two baseline methods: the rule-based method proposed by Chu and Wu [6] and a model-based method by using SVM. Four half-innings (marked * in Table III) were used to set the field ratio thresholds and train the SVM classifier. The SVM training accuracy can achieve 1.00 recall and 0.90 precision. Each method’s performance is represented by the recall (R) and precision (P) rates.

We observe that the SVM method outperformed Chu and Wu’s model. For Games 2008/06/10 and 2008/06/16, where partial video content of the games is used for training. For Games 2008/06/18 and 2008/06/24, however, Chu and Wu’s method outperforms the SVM method. Neither method yields a stable performance when they have to deal with various game styles. The proposed unsupervised method achieves a comparable result (and, in some cases, a better result) to the two baseline methods. In addition, our method’s performance is relatively stable on the above games, which shows our assumption for the pitch segment’s characteristic is very effective. Moreover, the method avoids the cumbersome task of observing and labeling video data manually.

2) Batter Detection and Left-Handed/Right-Handed Batter Classification

We trained two HOG detectors for left and right handed batters based on a SVM. 96 left and 96 right handed batter images were compiled as positive examples, each of which is 64×128 pixels. Negative examples were generated from the INRIA dataset, with total 117964×128 image blocks. 165 hard examples were added to negative examples in the second round training. 360 pitch frames were compiled for testing. The accuracy for the left handed batter detector is 0.91 recall and 1.00 precision, while for the right one is 0.97 recall and 0.98 precision. It shows that the detected batter images are near-perfect correct with a few misses by the proposed method.

3) Event Classification

Table IV lists the three event categories of baseball events and their numbers of training images. Three event classifiers were trained by the SVM-based SIFT-Bag kernel, where each event segment was represented as a SIFT-Bag descriptor with 256 global cluster centers (i.e., \( K = 256 \)). The testing data contained 290 frames of event segments for each category. The classification result is represented in a confusion matrix form in Table V. Some non-hitting events are misclassified into the infield category, and vice versa. We consider the reason is that both two categories are occurred at the infield area. Similarly, some non-hitting and infield events are misclassified by the outfield classifier because the video content of outfield events often comprise partial of the infield area. For example, a camera view of an outfield hit might pass through the infield area to the outfield area. Since we do not employ scoreboard recognition, we produce “rough” classifiers for three basic at-bat event categories trained by using their visual appearance only. Even so, our experiments show the classification power is sufficient for the alignment task.
B. Alignment Accuracy

1) HAC and GAO

Twenty half-innings of ten games and two full-length games were selected from our video dataset to evaluate the proposed HAC and GAO alignment methods. The configuration of GAO was set as follows: chromosome population size 32, crossover probability 0.75, mutation probability 0.0075, and maximal evolution iterations 20. Four GAO’s measures, including batter image similarity (BS), video length estimation (VL), change of left/right handed batters (LR), event likelihood (EL), were evaluated individually. Given the video content and webcast text of a half-inning as input, the proposed methods output a chromosome as the alignment solution, denoted as \( x^* = x_1^* x_2^* \ldots x_n^* \), where \( x_i^* \in \{1, 2, \ldots, M\} \) is the batter index of the \( i^{th} \) pitch segment, \( N \) is the number of pitch segments, and \( M \) is the number of batters in the half-inning. Let \( \hat{x} \) be the ground-truth of the half-inning. The alignment accuracy is defined as the inverse Hamming distance between \( \hat{x} \) and \( x^* \):

\[
HS(\hat{x}, x^*) = 1 - \frac{\sum_{i=1}^{N} I(x_i^* \neq \hat{x}_i)}{\text{max}(|\hat{x}|, |x^*|)}, \tag{17}
\]

where \( \oplus \) is the XOR logical operator. However, the above equation might have been slightly biased when there are offsets between the two strings. Hence, the Jaccard coefficient of \( \hat{x} \) and \( x^* \) is proposed to be an alternative evaluation metric:

\[
JC(\hat{x}, x^*) = \frac{\sum_{m=1}^{N} \varphi_{x,m} \cap \varphi_{\hat{x},m}}{\sum_{m=1}^{N} \varphi_{x,m} \cup \varphi_{\hat{x},m}}. \tag{18}
\]

Recall that \( \varphi \) is defined in Section V.B.

The experiment results are summarized in Table VI. The first column lists the dates of the ten test games, which are played by different teams, held in different stadiums, and broadcast by different channels. The second column gives the inning information, including the number of ground-truth pitch segments \( G \), the number of detected pitch segments \( N \), and the number of individual batters \( M \). The remaining columns show the accuracy rates of the HAC and GAO methods. Each field has two rates: the Hamming similarity and the Jaccard coefficient. The proposed pitch segment detection method achieves a very high accuracy rate. It identifies the pitch segments in the test data correctly, except for one missing in the half-inning 2008/07/11 – 3 bottom. The reason is that the broadcasting channel replayed the fourth batter segment when the fifth batter was batting. The pitch segment for the fifth batter is too short, so the proposed method fails to find the unusual case.

The accuracy rates of the HAC method are unstable. Since HAC is a greedy-based algorithm, it tends to get trapped in a local optimal solution. In particular, if two clusters that do not belong to the same batter are merged incorrectly, the new cluster might be very different from the original clusters and thus affect the subsequent clustering. Besides, HAC only employs SIFT-Bag descriptors of batter image sets to evaluate the similarity, so its performance depends to a large extent on the power of the SIFT-Bag descriptors. We observe that the descriptors might be seriously affected by abrupt changes, such as the crowd movement and batting posture switch, as shown in 2008/07/11 – 3 bottom. These cases might produce very different SIFT-Bag descriptors even for the same batter, and thereby increase the possibility of incorrect merging.

The GAO’s results comprise four measures’ individual performances. Overall, the VL measure outperforms the other three individual measures. The result supports our assumption that a batter’s pitch count is proportional to his batting time. In addition, the VL measure does not require any supervised learning process, so its performance is more consistent on various styles of video content. However, the measure is affected by frequent pickoffs – the action that holds base runners – so that the batter’s time at the home plate increases and thus biases the estimation of the length of video segments. Some examples of pickoff can be found in the following half-innings: 2008/06/27 – 4 bottom, 2008/07/05 – 5 top, 2008/07/11 – 3 bottom, and 2008/08/25 – 4 top.

Other measures are less effective than the VL measure. The performance of the LR is dependent on the number of changes of left-handed/right handed batters in a half-inning. Given \( n \) detected changes, we can separate the video content into \( n+1 \) different batters. If no more than one change point is detected, such as in 2008/06/27 – 2 top, 2008/06/30 – 4 bottom, and 2008/08/25 – 4 top, the effectiveness of the LR measure would degrade. A switch batter who bats both right-handed and left-handed also affects the LR measure; he might change his bat hand to face different pitchers. The BS measure is based on SIFT-Bag descriptors, so it has same problem as HAC, i.e., the robustness to abrupt changes. For the EL measure, since the accuracy of event classification shown in Table V is not very high, it somewhat degrades the alignment performance.

2) Integrated GAO

The next experiment investigates the integrating of these GAO measures. We evaluate five different combination methods of the GAO measures, including BS+EL, VL+LR, BS+LR+EL, BS+VL+EL, and ALL, where their configurations \((a, b, c, d)\) in (16) are set empirically by BS+EL: \((0.5, 0, 0, 0.5)\), VL+LR: \((0, 0.5, 0.5, 0)\), BS+LR+EL: \((0.3, 0, 0.4, 0.3)\), BS+VL+EL: \((0.3, 0.4, 0, 0.3)\), and ALL: \((0.05, 0.35, 0.3, 0.3)\). Table VII lists the average accuracy rates of the GAO combination methods. These combination methods simulate different multimodal fusion scenarios. For example, the information of the BS and EL measures can be obtained directly from webcast text, while that of VL and LR measures might be unavailable or incomplete in some cases. The BS+LR+EL method represents the situation of the basic version of webcast text shown in Figure 2(b), where the pitch count information is not available. The BS+VL+EL method does not use the LR measure for the case where the bat hand information is unavailable or some batter is a switch hitter. The ALL method putting all four measures together in optimization demonstrates the highest accuracy rate overall. The result manifests these measures can complement one another to yield a more robust performance when handling a variety of baseball video content.

3) Baseline Comparison

Three baseline methods were implemented for comparison. They are based on dynamic time warping (DTW), bipartite matching (BM), and hidden semi-Markov models (HSM). Recall in Section III.B, the alignment problem is to find the best
The mapping relation between two sequences $V_\phi$ and $W_\psi$, where $V_\phi = \{P_1, P_2, ..., P_N\}$ consists of $N$ pitch segments and $W_\psi = \{WT_1, WT_2, ..., WT_M\}$ consists of $M$ webcast text items. In these baseline methods, each of $P_i \in V_\phi$ and $WT_m \in W_\psi$ was represented by a 5-dimensional feature vector $(vl, lr, el_{non-hitting}, el_{field}, el_{outfield})$, where $vl$ was the pitch segment length ratio (for $P_i$) or pitch count ratio (for $WT_m$), $lr$ was an indicator of left/right handed batter determined by the batter classifier (for $P_i$) or recorded in webcast text (for $WT_m$), and $el_{non-hitting}$, $el_{field}$, and $el_{outfield}$ were the event likelihoods calculated by event classifiers (for $P_i$) or recorded in webcast text (for $WT_m$). The similarity between two feature vectors was calculated by the weighted cosine similarity; we set five weight scales 0.4, 0.3, 0.1, 0.1, and 0.1 to reflect the relative importance of the five feature elements, respectively.

The DTW method tries to find an optimal matching between two sequences of feature vectors allowing stretched and compressed alignment [7]. The mapping has to obey the following linking rules: 1) $P_i \rightarrow WT_1$; 2) $P_i \rightarrow WT_2$, 3) no crossing mapping; 4) each $PS_i$ maps to only one $WT_m$ i.e., each pitch segment corresponds to only one batter; and 5) each $WT_m$ associates with at least one $PS_i$.

The BM method is a greedy-based approach [7]. It process can be modeled by a finite state machine, as illustrated in Figure 16. A complete bipartite graph is first formed by connecting every $PS_i$ to every $WT_m$. We then select the edge with the highest similarity $PS_i \rightarrow WT_m$. If the edge obeys the same linking rules defined in the DTW method, it is moved to the answer set; otherwise it is directly removed from the bipartite graph. The process repeats until each $PS_i$ is connected to a $WT_m$ in the answer set.

The HSMM method is implemented based on Liang et al.’s method for TV video parsing [15]. They used HSMMD to align the video and script by learning the relation between video faces and script names. We adopted their method with some modifications to address the alignment problem between baseball video content and webcast text. Let $S = \{s_i \mid i = 1, 2, ..., N\}$ be the observed state sequence of pitch segments, $T = \{t_m \mid m = 1, 2, ..., M\}$ be the observed state sequence of webcast text, and $H = \{h_m \mid m = 1, 2, ..., M\}$ be the hidden state sequence, as shown in Figure 17. A simplified HSMM is represented as

$$P(S, T, H) = \prod_{i=1}^{N} P(h_m \mid t_m) \cdot P(s_i \mid h_m, t_m),$$

(19)

where $P(h_m \mid t_m)$ represents the duration probability of the nth batter, which is estimated by the pitch count ratio recorded in $WT_m$, $P(s_i \mid h_m, t_m)$ denotes the observation probability of the $i$th pitch segment given the $m$th batter, which is calculated by

$$P(s_i \mid h_m, t_m) = \exp\left(-\frac{\left(t_m-A_i\right)^2(t_m-A_i)}{2\sigma_m^2}\right) / \sqrt{2\pi\sigma_m^2}.$$

(20)

$A$ is an $M \times N$-dimensional relation matrix that associates $M$ batters with $N$ pitch segments, $\sigma_m$ is the covariance of the $m$th batter, $t_m$ is an $M \times 1$-dimensional pitch segment histogram at the $m$th state, and $s_i$ is a $N \times 1$-dimensional pitch segment histogram at the $i$th state. $A$ and $\sigma_m$ are iteratively estimated by an EM algorithm. To infer the most likely underlying state sequence, the Viterbi algorithm is employed to find the optimal answer.

### TABLE VI

<table>
<thead>
<tr>
<th>Game (Channel)</th>
<th>Inning (G/N/M)</th>
<th>HAC</th>
<th>GAO</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008/06/12</td>
<td>7 bot (9/3/1)</td>
<td>0.44/0.56</td>
<td>0.44/0.56</td>
</tr>
<tr>
<td>NYY vs. OAK (YES)</td>
<td>8 bot (14/4/3)</td>
<td>0.86/0.86</td>
<td>0.80/0.80</td>
</tr>
<tr>
<td>2008/06/24</td>
<td>2 top (13/13/4)</td>
<td>0.85/0.92</td>
<td>0.83/0.89</td>
</tr>
<tr>
<td>NYY vs. PIT (MY9)</td>
<td>5 top (10/10/3)</td>
<td>0.30/0.40</td>
<td>0.70/0.70</td>
</tr>
<tr>
<td>2008/06/27</td>
<td>2 top (14/14/3)</td>
<td>0.57/0.64</td>
<td>0.50/0.57</td>
</tr>
<tr>
<td>NYY vs. NYM (YES)</td>
<td>4 bot. (11/1/3)</td>
<td>0.64/0.64</td>
<td>0.64/0.64</td>
</tr>
<tr>
<td>2008/06/28</td>
<td>7 top (15/15/3)</td>
<td>1.00/1.00</td>
<td>0.90/0.90</td>
</tr>
<tr>
<td>NYY vs. NYM (FOX)</td>
<td>1 top (10/10/3)</td>
<td>0.63/0.69</td>
<td>0.70/0.70</td>
</tr>
<tr>
<td>2008/06/30</td>
<td>3 top (15/15/5)</td>
<td>0.53/0.67</td>
<td>0.53/0.67</td>
</tr>
<tr>
<td>TEX vs. NYY (YES)</td>
<td>4 bot. (17/17/5)</td>
<td>0.10/0.00</td>
<td>0.29/0.47</td>
</tr>
<tr>
<td>2008/07/05</td>
<td>4 bot. (16/16/4)</td>
<td>0.25/0.31</td>
<td>0.60/0.69</td>
</tr>
<tr>
<td>BOS vs. NYY (FOX)</td>
<td>5 top (12/12/4)</td>
<td>0.58/0.67</td>
<td>0.33/0.42</td>
</tr>
<tr>
<td>2008/07/06</td>
<td>1 bot. (12/12/3)</td>
<td>0.38/0.46</td>
<td>0.60/0.60</td>
</tr>
<tr>
<td>BOS vs. NYY (ESP)</td>
<td>3 bot. (20/20/3)</td>
<td>1.00/1.00</td>
<td>0.89/0.90</td>
</tr>
<tr>
<td>2008/07/11</td>
<td>2 bot. (17/17/4)</td>
<td>0.59/0.76</td>
<td>0.41/0.71</td>
</tr>
<tr>
<td>NYY vs. TOR (MY9)</td>
<td>3 bot. (19/18/6)</td>
<td>0.47/0.68</td>
<td>0.26/0.38</td>
</tr>
<tr>
<td>2008/08/25</td>
<td>4 top (16/16/5)</td>
<td>0.44/0.56</td>
<td>0.31/0.63</td>
</tr>
<tr>
<td>LAD vs. PHI (FSN)</td>
<td>4 bot. (8/8/3)</td>
<td>0.38/0.50</td>
<td>0.38/0.50</td>
</tr>
<tr>
<td>2008/09/13</td>
<td>1 top (11/11/4)</td>
<td>0.82/0.82</td>
<td>0.79/0.82</td>
</tr>
<tr>
<td>LAD vs. COL (FSN)</td>
<td>2 bot. (18/18/3)</td>
<td>0.44/0.50</td>
<td>0.83/0.83</td>
</tr>
<tr>
<td>2008/06/12</td>
<td>All 18 half-innings</td>
<td>0.58/0.66</td>
<td>0.70/0.80</td>
</tr>
<tr>
<td>NYY vs. OAK (YES)</td>
<td>All 17 half-innings</td>
<td>0.63/0.69</td>
<td>0.73/0.80</td>
</tr>
<tr>
<td>2008/06/24</td>
<td>All 17 half-innings</td>
<td>0.63/0.69</td>
<td>0.73/0.80</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.61/0.68</strong></td>
<td><strong>0.69/0.78</strong></td>
<td><strong>0.81/0.89</strong></td>
</tr>
</tbody>
</table>

### TABLE VII

The average accuracy rates of the GAO combination methods

<table>
<thead>
<tr>
<th>GAO</th>
<th>BS+EL</th>
<th>BS+LR+EL</th>
<th>BS+VL+EL</th>
<th>VL+LR</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS+EL</td>
<td>0.69/0.78</td>
<td>0.77/0.83</td>
<td>0.86/0.90</td>
<td>0.86/0.92</td>
<td>0.88/0.94</td>
</tr>
</tbody>
</table>
ever, our alignment problem is to match case where the two sequences lengths are approximate. However, the diagonal mapping between two sequences. It is suitable for the parameters. The DTW method favors the global optimal solution, the aggregation of all local optimal solutions does not guarantee to yield the global optimal solution. The HSMM method uses the EM algorithm to learn the hidden state parameters $A$ and $\sigma_m$. We observe that in some cases, these parameters are divergent during the learning process and the performance might be thus degraded.

4) Discussion

We investigate some worse cases in the GAO-ALL method. Table IX summarizes the average accuracy rates of different numbers of batters in a half-inning. A more number of batters usually increases the number of pitch segments, making the alignment problem more complicated. Thus, the accuracy generally degrades as the number of batters increases. The decline is slight except for a noticeable drop occurred when the number of batters is equal to or greater than six. A remedy is to carry out more evolution iterations in the GAO method to increase the probability of finding the best chromosome. For example, we adjust the maximum evolution iterations from 20 to 30. The accuracy rate of the half-inning 2008/07/11 – 3 bottom, which consists of six batters and eighteen pitch segments, improves from 0.79/0.84 to 0.89/0.89. Another half-inning 2008/06/24 – 6 bottom, which consists of six batters and thirteen pitch segments, also gains an improvement from 0.63/0.90 to 1.00/1.00.

We notice there is an error in the official webcast text for the half-inning 2008/06/27 – 4 bottom. In fact four batters bat at the half-inning, but the fourth batter’s at bat event is missed in the official recording. After we correct this error and ran our program again, the accuracy rate of the GAO-ALL method improves from 0.55/0.64 to 0.91/0.91.

VII. CONCLUSION AND FUTURE WORK

Conventional baseball annotation frameworks do not utilize webcast text effectively, and existing multimodal fusion frameworks do not fit baseball videos very well. To address the above issues, we present a novel framework for baseball video annotation based on aligning video content and webcast text. The alignment problem can be resolved by the proposed genetic algorithm optimization method, which integrates multimodal cues from both low-level video content and high-level webcast text. We also use Markov random walk for pitch segment detection to reduce the need for human interventions. Experiments show a robust result against a variety of video content and high automaticity in baseball annotation.

Webcast text preserves many detailed information occurred in a baseball game, for example, each batter’s pitch count and each pitch’s speed, type, and event. The future work can exploit each pitch’s information in webcast text to find its corresponding video segments, so that we can provide detailed event retrieval and browsing functions for baseball videos. The other direction is to extend the proposed alignment method to some potential applications, such as soccer video and text alignment [16] and face-name correspondence in TV videos [15].

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their thoughtful comments and suggestions that have advanced the quality of this article.

REFERENCES


