J. Vis. Commun. Image R. 25 (2014) 285–295

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

A framework for video event classification by modeling temporal context of multimodal features using HMM

Hsuan-Sheng Chen *, Wen-Jiin Tsai

Department of Computer Science, National Chiao-Tung University, Taiwan 1001 University Road, Hsinchu 300, Taiwan, ROC

ARTICLE INFO

Article history:
Received 23 April 2013
Available online 7 December 2013

Keywords:
Multimedia system
Video semantic analysis
Baseball event classification
Co-occurrence symbol
Interval-based multimodal feature
HMM
Probabilistic temporal modeling
Multivariate temporal data classification

ABSTRACT

Semantic high-level event recognition of videos is one of most interesting issues for multimedia searching and indexing. Since low-level features are semantically distinct from high-level events, a hierarchical video analysis framework is needed, i.e., using mid-level features to provide clear linkages between low-level audio-visual features and high-level semantics. Therefore, this paper presents a framework for video event classification using temporal context of mid-level interval-based multimodal features. In the framework, a co-occurrence symbol transformation method is proposed to explore full temporal relations among multiple modalities in probabilistic HMM event classification. The results of our experiments on baseball video event classification demonstrate the superiority of the proposed approach.

1. Introduction

The amount of multimedia information has grown explosively over the past decade. To search and index through these data is a difficult task and dependent upon techniques that can recognize events in videos. Event mining has been an active research area with notable recent progress and a detail review can be found in [1]. The event application systems from these works make the delivery and searching of video contents more effective and efficient. However, most of current systems employed domain specific heuristics to model audiovisual feature patterns in event classification. [2–4] pointed out that there is a lack of a framework for event classification. In this paper, a framework of modeling interval based multimodal features is proposed for video event classification. In this framework, videos from different domains are represented by interval-based structures and reasoned by their temporal context.

Temporal context of interval-based features can be utilized as important cues for video classification. Take a fly ball event in baseball video for example as shown in Fig. 1, where the video starts with a pitching scene. After the hitter hits the ball, the video is changed into an outfield scene and the cameraman tracks the ball location, producing pan up camera motions. At the same time, the audience cheers loudly as the ball fly. After the ball was caught by the outfielder, the video is switched to a close up scene of the pitcher to show his reaction. This example indicates that the high-level event, fly ball, can be recognized by detecting interval-based features: pitching scene, outfield scene, close-up scene, camera pan-up as well as audience cheering in the video and verifying whether all these features meet the temporal context described above.

Researchers have made some progresses in modeling the temporal context structure inherent to high-level events. Zhu et al. [6] took into account the temporal continuity besides association rules. Therefore, they introduced multilevel sequential association mining to explore associations among the audio and visual cues. However, the temporal knowledge used in their work is rather simplified. Chen et al. [7,8] proposed a hierarchical temporal association analysis framework using multimodal data mining method. In their method, temporal analysis is used to identify significant temporal patterns. Unfortunately, since the temporal pattern adopted is sequential, this method cannot detect events with complex temporal relationships. Allen [9] defined all possible temporal relations between time intervals. According to Allen [9], the temporal relation between any two features can be described by one of 7 temporal logics: *before*, *meet*, *overlap*, *equal*, *during*, *start*, and *finish*. Snoek et al. [10] used a relaxed version of Allen's relations, called TIME, to detect events in two different domains: soccer and news. But the relaxed Allen's relations cannot convey temporal interval information precisely. Fleischmann et al. [11,12] proposed an unsupervised method based on Allen temporal algebra [9] by searching for commonly occurring event temporal relationships. The main contribution of their work is the temporal structures of complex events and the integration of external text-modal information. However, as pointed out by Wu et al. [13], there are two ambiguous problems in Allen's temporal algebra when dealing

* Corresponding author.
E-mail addresses: xschen@cs.nctu.edu.tw (H.-S. Chen), wjtsai@cs.nctu.edu.tw (W.-J. Tsai).

1047-3203/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.12.001

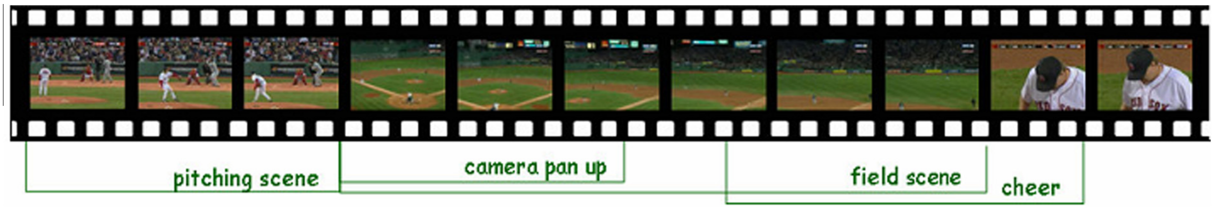
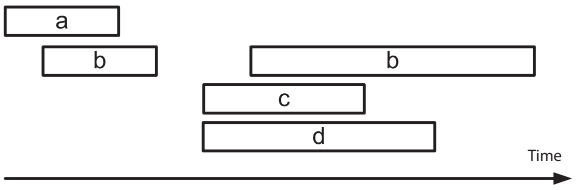


Fig. 1. Temporal compositions of interval-based features, including pitching scene, camera pan up, field scene, and cheer for a fly ball event in baseball video [5].

with more than two features. They are (1) different representations may exist for the same temporal relation among features, and (2) the same representation can stand for different temporal relations among features. To overcome these problems, Wu et al. [13] proposed a new non-ambiguous representation (called *temporal sequence*, or TS) by using feature type, start/end symbols, occurrence number, and temporal order. For example, the temporal relation "*x* meets *y*" in Allen's method is represented by $x^{+1} < x^{-1} = y^{+1} < y^{-1}$ in Wu et al.'s method, where x and y are feature types, < and = are temporal order relation, + and − mean feature start/end, and the number following the +/− is the occurrence times of this feature type. For m features, we can construct $2^m - 1$ TSs to describe all of their temporal relations: one TS for one relation. Given five features as shown in Fig. 2, there would be $2^5 - 1 = 31$ TSs among them. Due to space limit, we only show three TS examples in Fig. 2. Moreover, for two temporal sequences, if one is a subset of the other one, we say there is a *containment* between them. As shown in Fig. 2, ts_3 is contained in ts_1, but not contained in ts_2. The formal definition of temporal sequence can be found in [13] and thus, it is not described in details here.

By using Wu et al.'s representation, Dao et al. [14] proposed a temporal frequent pattern-based event classification (TFPEC) method for event detection of sports video. Their method has two stages: training and testing. In training stage, frequent pattern set (FPS) for each event are constructed using its training TSs, where the FPS is a set of frequently occurring feature patterns among the training sequences of an event class. Since any subset of a TS could be a frequent pattern candidate, all the subsets for each training TS are enumerated and verified by counting their occurrence number. If any subset occurs more than N times for an event class, it is called a frequent pattern, where the N is specified by users [13]. In TFPEC method, an event class is distinguished by its FPS. Therefore, for N event classes, there will be N frequent pattern sets ($FPS_1, \ldots, FPS_N$) after the training stage. In the testing stage, for a test sequence, *containment* check is performed between all its TSs and each FPS. This test sequence is classified to an event class (say i) if one of its TSs are contained by one of the patterns in FPS_i. If some of its TSs are contained by multiple FPSs, the event class corresponding to the one with the longest containment is selected. Although this method is able to represent events with complex temporal relations, their classification method of using containment match makes it suffer from multiple-recognition problem (i.e., one test data is recognized as multiple exclusive event classes), resulting in a poor accuracy rate. Therefore, probabilistic classification based on HMM would be adopted to solve this problem.



$$ts_1 : a^{+1} < b^{+1} < a^{-1} < b^{-1} < c^{+1} = d^{+1} < b^{+2} < c^{-1} < d^{-1} < b^{-2}$$
$$ts_2 : a^{+1} < b^{+1} < a^{-1} < b^{-1} < c^{+1} < b^{+2} < c^{-1} < b^{-2}$$
$$ts_3 : c^{+1} = d^{+1} < c^{-1} < d^{-1}$$

Fig. 2. TS examples denoting some temporal relations among five features.

HMM has been regarded as an effective tool to recognize continuous-time signals since it has been successfully used in many applications including speech recognition [15], gesture/action recognition [16,17] and biological sequence modeling [18]. A discrete HMM is characterized by a set of parameters $\lambda = (A, B, \pi)$, where *A* is the state transition probability matrix, B is the observation symbol probability matrix, and π is the initial state distribution. Using HMM for events classification involves training and testing stages. At training stage, each event class will have a HMM trained with corresponding sequences for it. At testing stage, maximum likelihood method is used to classify events. The details of HMM can be found in [15].

HMM was also introduced to video event analysis [19] such as soccer [20], baseball [21], and basketball [22]. Even though these works of using HMM can classify events in video content efficiently, most of them use information from one single modality such as scene transitions, camera motions, or object motions, etc. This kind of approaches is called the *single-model approach*. Effective event analysis, however, requires a *multimodal approach* in which information from different modalities are exploited [23–28]. However, multiple modalities produces more temporal relationships which make it hard to apply HMM which can only model temporally sequential data. To apply HMM, each video (no matter training video or testing video) must be transformed into a symbol sequence, with different symbols representing different features and the order of symbols in the sequence standing for the temporal order of the features in that video. Note that if all the features used for classification belong to one feature model, then the resulting symbols for this video sequence will be sequential, which can be fed into HMM directly. However, if the features used for classification are from more than one feature modal, these features may have overlap in time within the video and thus, cannot be represented as one sequential symbol sequence.

Huang et al. [29] proposed a method called *product-HMM* which applies multimodal information on HMM-based classification. In product-HMM, modalities are assumed to be mutually independent. Thus, for each video, different symbol sequences are generated independently for different feature modalities; and for each event class, multiple HMMs are trained, one for each feature modal. Taking three feature modalities: visual, motion and audio, as an example, instead of using one HMM λ_i for each event class, three HMMs $\lambda_i^V, \lambda_i^M, \lambda_i^A$ are trained for each event class, where λ_i^V is for visual model, λ_i^M for motion and λ_i^A for audio. At testing stage, symbol sequences of different feature models generated from test video are fed into corresponding HMMs of each event class. Since product-HMM assumes that feature models are mutually independent, the observation probability of event class i is computed as:

$$P(O|\lambda_i) = P(O^V|\lambda_i^V) \bullet P(O^M|\lambda_i^M) \bullet P(O^A|\lambda_i^A),$$

where $\lambda^V = (A^V, B^V, \pi^V)$, $\lambda^M = (A^M, B^M, \pi^M)$ and $\lambda^A = (A^A, B^A, \pi^A)$. The test video is classified to the class with maximum likelihood according to this probability. Note that although product-HMM provides a way of utilizing multimodal features in HMM-based event classification, it did not utilize full temporal relations among different modalities because modalities are assumed to be mutually independent.

In light of these discussions, this paper proposes a new method of applying multimodal features on HMM for video event classification. There are two main contributions in our approach. (1) A HMM-based event classification framework is proposed using multimodal interval features. (2) A co-occurrence symbol transformation method is proposed with the capability of exploiting full temporal relations among multiple modalities in HMM-based classification without suffering the multiple-recognition problem. In the remainder of this paper, detailed procedures of the proposed framework are introduced in Section 2. Section 3 gives performance evaluations from baseball event recognition and the results indicate that the performance and effectiveness of the proposed framework are satisfactory. Conclusions are drawn in Section 4.

2. Proposed method

As described, TFPEC-based methods exploit multimodal features in event classification by using temporal sequence representation. But they suffer from a multiple-recognition problem (i.e., one test data is recognized as multiple exclusive event classes since one frequent pattern could occur in multiple event classes) by using frequent pattern match, resulting in poor accuracy rate of recognition. This could happen easily especially when some event classes have parts of frequent patterns in common. With probabilistic model and maximum likelihood decision making, although HMM-based methods will not encounter the multiple-recognition problem, it is hard to exploit temporal relationship among multimodal features. The assumption that feature models are independent of each other may result in poor recognition performance because temporal relationship is not fully utilized.

The proposed event classification method aims at taking advantages of both TFPEC- and HMM-based methods. In order to achieve a better recognition performance, multimodal features are exploited in our approach; and to overcome the multiple-recognition problem, probability based HMM rather than pattern-match based TFPEC is adopted. To utilize full temporal relationship of multimodal features in HMM, a *co-occurrence symbol transformation method* is proposed, which encodes multimodal features as a sequence of symbols. The sequence not only keeps full temporal relationship among features, but also can be fed to HMM directly. The conceptual diagram of the proposed system is shown in Fig. 3. It consists of three functional blocks: *interval-based multimodal feature representations*, *co-occurrence symbol coding transformation*, and *event classification using HMM*. First, input videos are processed to find selected multimodal features and these features are represented in temporal structure called *temporal database*. Then, they are transformed to the proposed *co-occurrence symbol sequences* which exploit temporal context of multimodal features in a manner that HMM can use. These sequences are then used as the inputs of HMM for video event recognition based on probabilistic classification. The details of each functional block are described in the following subsections.

2.1. Interval-based multimodal feature representation

Our approach adopts time interval-based temporal representation to denote video data using selected features and the resulting data is called a *temporal database*. To generate a temporal database, the event classes for recognition and the corresponding *interval-based features* should be determined first. Both of them are decided based on domain specific knowledge. Here, we use baseball application as an example to explain this module as shown in Fig. 4.

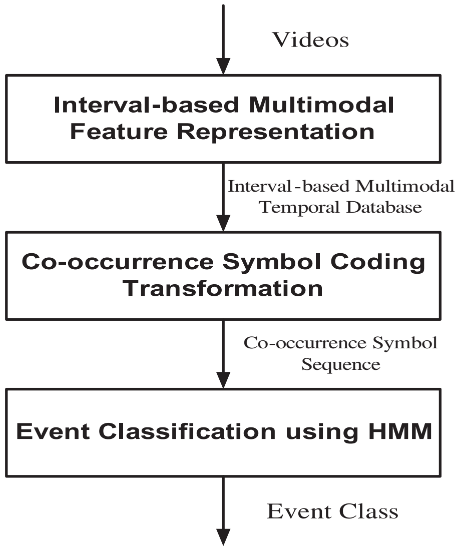

Videos
Interval-based Multimodal Feature Representation
Interval-based Multimodal Temporal Database
Co-occurrence Symbol Coding Transformation
Co-occurrence Symbol Sequence
Event Classification using HMM
Event Class

Fig. 3. Conceptual diagram of the proposed event detection system.

High-level events such as homerun, outfield hit, outfield out, infield out, strike out, walk, etc. are the typical baseball event classes for recognition. The features used to recognize these high-level events can be low-level features such as color and texture or mid-level features such as video shot, scene, camera motion, etc. Compared with low-level features, since mid-level features are more semantically related to high-level events, they are widely employed in the recent researches and the methods for mid-level feature detection have been developed well and proven efficient in sports domain [30–32]. Fig. 4 shows the interval-based mid-level feature detection in baseball from different sources: visual shot, camera motion, audio, and misc. Eight baseball shots including Pitching, Infield, Outfield, Audience, Base, Close-up, Running, and Misc can be detected and classified using a decision tree classifier based on feature vectors composed of different low level visual features [12]. In addition to video shots, camera motions such as Pan Right, Pan Left, Zoom In, Zoom Out, Tilt Up and Tile Down can be detected by the methods in [33] using an affine model. Audio features such as exciting commentator speech, audience cheering can be detected by segmenting, classifying and clustering audio frames represented by feature vectors composed of different low-level audio features [12]. The interval-based features utilized in the proposed framework can be low-level or mid-level. Since mid-level feature detection is somewhat mature now, we adopt mid-level features in the proposed framework to illustrate interval-based representation and high-level event recognition. When the mid-level features are selected from different source models such as visual scene shot, camera motion, audio, object, replay, and misc, they are called *multimodal mid-level features*.

Since mid-level features are strongly application-related, how to select them properly will not be discussed in details in this paper. Here we only focus on proposing a framework based on selected mid-level features. Assume seven baseball mid-level feature types (Close-up, Outfield, Audience, Zoom In, Pan Right, Excited Speech, Cheering) from three modalities (Baseball Shot, Camera Motion, Audio) are selected. Fig. 5(b) shows an example of temporal database for the video in Fig. 5(a). A mid-level feature selection for baseball event detection in Section 3.1 can be referenced for better comprehension.

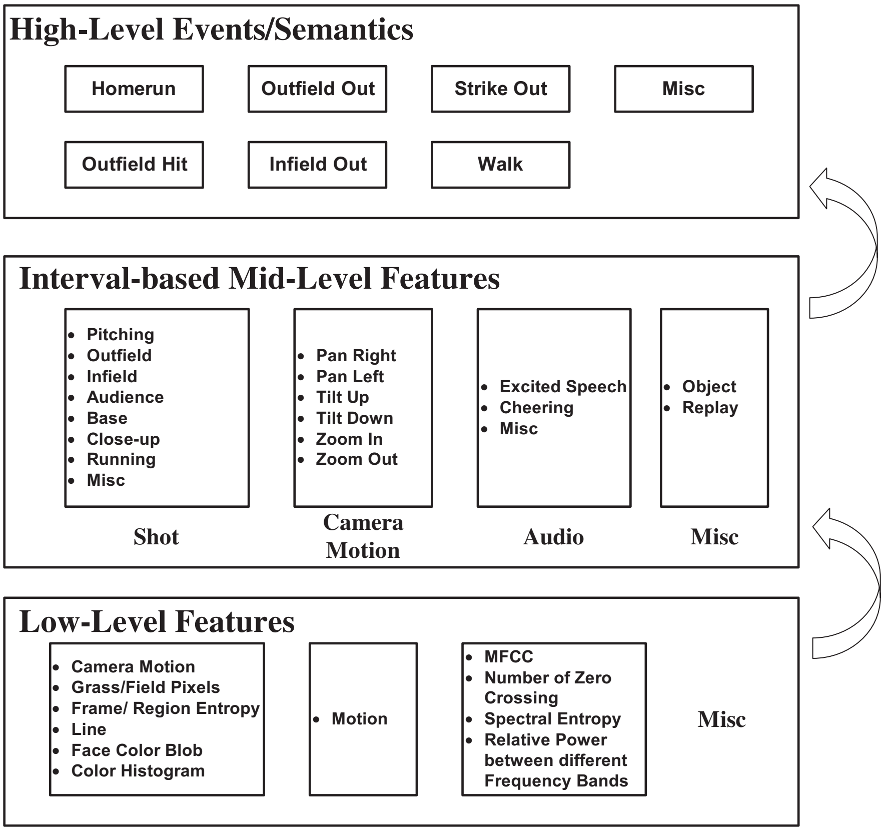
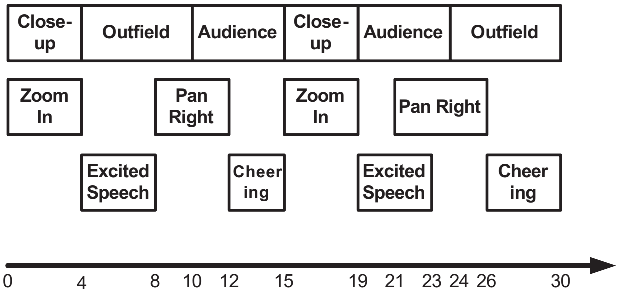


Fig. 4. Interval-based mid-level feature detection from different source models in baseball.



(a) A video consisting of seven feature types from three modalities

Feature Type	Time Period	Feature Type	Time Period	Feature Type	Time Period
Close-up	{0, 4}	Zoom In	{0, 4}	Excited Speech	{4, 8}
Outfield	{4, 10}	Pan Right	{8, 12}	Cheering	{12, 15}
Audience	{10, 15}	Zoom In	{15, 19}	Excited Speech	{19, 23}
Close-up	{15, 19}	Pan Right	{21, 26}	Cheering	{26, 30}
Audience	{19, 24}				
Outfield	{24, 30}				

(b) An example of temporal database

Fig. 5. Interval-based representation of multimodal features.

2.2. Co-occurrence symbol transformation

As described, we adopt a probabilistic classification method, HMM, to solve the multiple-recognition problem in event classification. HMM requires that the symbols fed into it are sequential temporally without symbol durations overlapped in time. Namely, when a temporal database is constructed from multimodal features, it cannot be utilized by HMM directly because the features selected from different modalities may occurs simultaneously in video, resulting in overlapped symbol durations within a temporal sequence. To apply multimodal features in HMM-based classification, a symbol transformation method is proposed to convert a temporal database into co-occurrence symbol representation.

Co-occurrence symbol transformation method includes two steps: *temporal segmentation* and *symbol coding*. In temporal segmentation, a time period is segmented whenever a new feature(s) starts or the feature which occurred previously ends. Features from each modality should be considered in order to convey the context of multimodal features. There is a buffer called BUF to record the occurring feature type(s) in the current time period. In symbol coding, each segmented time period is coded by a single symbol which represents all the feature-type(s) occurring during it and the coding process is achieved by looking up a codebook which records all the generated symbols for the mapping that have occurred. The mapping could be one-to-one (one symbol to one feature) or one-to-many (one symbol to many co-occurring features). If the mapping is found in the codebook, the corresponding symbol is output. Once a mapping cannot be found, a new symbol is generated to represent the co-occurrence feature type(s) in BUF and then this mapping is added to the codebook and the symbol is output. After a series of symbol outputs, a sequential symbol sequence, called a *co-occurrence symbol sequence*, can be obtained for used in HMM. The detailed algorithm is described in Fig. 6.

The visualization of symbol transformation process using the proposed algorithm is shown in Fig. 7, where the example temporal database in Fig. 5 is used as input data. The vertical dashed lines show how the input temporal database is segmented according to the start times and end times of features; and each symbol on the bottom of the figure represents the co-occurring feature(s) in the corresponding segmented time period. In this example, the output symbol string (called the *co-occurrence symbol sequence*) is 'abcdeaefdcb'. A step by step explanation of process in Fig. 7 is shown in Table 1. At $t = 0$, features *Close-up & Zoom In* are inserted into BUF because of their occurrence. At $t = 4$, the codebook is looked up using the content of BUF, *Close-up & Zoom In*, but the symbol mapping is not found. Therefore, the mapping between features *Close-up & Zoom In* and a new codeword symbol 'a' is added to codebook and the symbol 'a' is outputted to represent the co-occurring features *Close-up & Zoom In* in time period [0,4]. Then, *Close-up & Zoom In* are removed from BUF because of their end; and *Outfield & Excited Speech* are added into BUF because of their occurrence. Similar process goes till $t = 15$ and four more new symbols, b, c, d, and e, are produced to represent the co-occurring features, *Outfield & Excited Speech*, *Outfield & Pan Right*, *Audience & Pan Right*, *Audience & Cheering*, respectively. At $t = 19$, the codebook is looked up for the content of BUF, *Close-up & Zoom In*. Since *Close-up & Zoom In* can be found in codebook, the corresponding mapping symbol 'a' is outputted to represent it in time period [15,19]. Then, features *Close-up & Zoom In* are removed from BUF because they end; and *Audience & Cheering* are added into BUF because of their occurrence. The process goes for each time period of the input temporal database and finally, the co-occurrence symbol sequence is 'abcdeaefdcb'. It is observed that, with this representation, full temporal relations (including overlapping) among multimodal features are captured by using co-occurring symbols in a sequential manner that HMM can use.

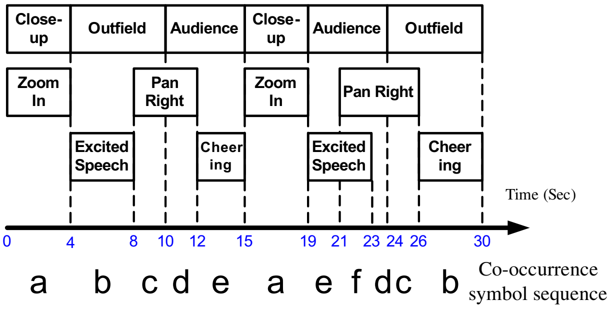


Fig. 7. Co-occurrence symbol coding of the example temporal database in Fig. 5, including temporal segmentation (dashed line) and output co-occurrence symbol sequence.

2.3. HMM classification for event detection

Multimodal features in a video can be represented by co-occurrence symbol sequences after data representation and symbol transformation; then event class is obtained by HMM classification which consists of training and testing. In training, separate HMM of each event class is trained by training co-occurrence symbol sequences of each category using the Baum-Welch method and trained HMMs are used for reference during testing. In testing, the event class of an un-annotated test sequence O is obtained

```
Multimodal Co-occurrence symbol coding algorithm:
Functionality: Transform temporal database of multimodal features into co-occurrence symbol sequences
Input: Temporal database of multimodal features
Output: Sequential co-occurrence symbol sequence

1. Sort multimodal features by start time in ascending order
2. While(there is still start of feature or end of feature) { // occurring features change
      if( BUF not empty ) {
         // co-occurrence symbol lookup
         if( symbol = codebook_look_up(features in BUF) is not NULL )
               output the symbol;
        else { generate a new symbol to represent the features in BUF;
               add the new symbol and its corresponding features to the codebook;
               output the new symbol; }
      }
      // update BUF
      if( end of feature(s) occurs ) remove the vanishing feature from BUF ;
      if( start of feature(s) occurs ) put the new feature(s) into BUF ;
  }
```

Fig. 6. Multimodal co-occurrence symbol coding algorithm.

Table 1
Step by step operations of co-occurrence symbol coding algorithm.

Time	BUF	Codebook	Output symbols
$t = 0$	Empty	Empty	
$t = 4$	Close-up & Zoom In	a: Close-up & Zoom In	a
$t = 8$	Outfield & Excited Speech	a: Close-up & Zoom In b: Outfield & Excited Speech	b
$t = 10$	Outfield & Pan Right	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right	c
$t = 12$	Audience & Pan Right	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right	d
$t = 15$	Audience & Cheering	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering	e
$t = 19$	Close-up & Zoom In	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering	a
$t = 21$	Audience & Cheering	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering	e
$t = 23$	Audience & Pan Right & Cheering	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering f: Audience & Pan Right & Cheering	f
$t = 24$	Audience & Pan Right	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering f: Audience & Pan Right & Cheering	d
$t = 26$	Outfield & Pan Right	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering; f: Audience & Pan Right & Cheering	c
$t = 30$	Outfield & Excited Speech	a: Close-up & Zoom In b: Outfield & Excited Speech c: Outfield & Pan Right d: Audience & Pan Right e: Audience & Cheering f: Audience & Pan Right & Cheering	b

by choosing the maximum likelihood between test sequence O and each trained HMM.

3. Experiments

3.1. Experiment setup

Baseball event recognition is used to evaluate the performance of the proposed framework. Six major event classes in typical baseball video are chosen: home run (HR), outfield hit (OutHit), outfield out (OutOut), infield out (InOut), strike out (SO), and walk. The mid-level features adopted come from three modalities: visual, camera motion and audio. For visual modality, a game is segmented into video shots based on changes in the visual scene due to editing and video shots are classified into eight categories: pitching, infield, outfield, audience, base, close-up, running, misc. For camera motion modality, they are clustered into six kinds: pan right, pan left, tilt up, tilt down, zoom in and zoom out. For audio modality, we extract audience cheering and exciting commentator speech. Manually labeled ground truth features are used since this study focuses on the event recognition performance using temporal context of multimodal mid-level features. However, labeling of these mid-level features can be potentially automated with methods in [12,33,34]. Fig. 8 shows an example temporal database of a homerun event, using selected features from three modalities.

In our experiments, baseball videos from five baseball games of MLB (Major League Baseball/USA) and NPB (Nippon Professional

Baseball/Japan) totalling 18.5 h are evaluated. All the videos are compressed in MPEG-1 format with frame size of 352 × 240 and frame rate of 30 fps. The event numbers of experimental videos are shown in Table 2. One-third videos of each event class were used as training set while the other two-thirds were used for testing set. The manually labeled events in the testing data were used as the ground truth to evaluate the event classification of our system. Confusion matrix is used to give the full picture of the classification performance. The precision/recall rates of each event class are calculated from the confusion matrix. Accuracy is the overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications. The number of original vocabularies in our experiments is 16, including 8 visual scenes, 6 camera motions, and 2 audio interval features.

3.2. Experiment results

The performance of the proposed method is evaluated with respect to six parts: event recognition results, effect of multimodal feature integration, effect of dataset selection, recognition of more event classes, testing on type of sequences not learned before, and comparisons of computational complexity and execution time.

3.2.1. Event recognition results

Three different methods integrating temporal information of multimodal features are compared to evaluate the event recognition performances: (1) *proposed co-occurrence HMM*, (2) *Product-HMMs*, (3) *FP-match* methods, where Product-HMM and FP-match are the approaches described in introduction.

Table 3 shows the performance comparison of the three methods. It is observed that the proposed method outperformed the other two methods and the accuracy of *FP-match* method is apparently lower than the two HMM-based methods. To detail the classification results for each type of event, Tables 4–6 show the confusion matrix of the three methods separately. For the proposed method, OutHit and OutOut are classified better than the other four baseball events. Homeruns, compared to other event classes, are much easier to be misclassified by all three methods. Since our experiments consist of much fewer Homerun samples than others, the misclassification problem may be due to insufficient training samples. For learning-based methods such as the proposed method and the product-HMM, the classification performance could be improved if more training samples are provided. Although InOut samples are the largest set among all categories, there are still some misclassification. More samples need to be collected for each category so that HMM could learn different transition patterns among the event class. The classification of Walk is apparently low. The SO and Walk events are different in semantics (the batter is out or safe), however, they are quite similar in feature context: a short close-up shot transitions of the pitcher, batter, or coach. Therefore, it is very difficult to tell SO from Walk by selected mid-level features. More distinguishable mid-level features need to be investigated to solve this problem. For *Product-HMMs* method, the overall classification is somewhat lower than the performance of proposed method, and it performs poorly for InOut classification, obtaining relatively low ratio in recall. For *FP-match* method, we impute the poor performance to the multiple-matching problem. During the experiments of total 255 test sequences, 182 test sequences (about 71%) are classified to multiple events, resulting in overall poor performance. As for the non-integer values in the confusion matrix of *FP-match* method, this phenomenon is due to different classification results of multiple candidates of maximum length. Suppose three candidates with maximum length L are matched for one test sequence, two candidates are classified

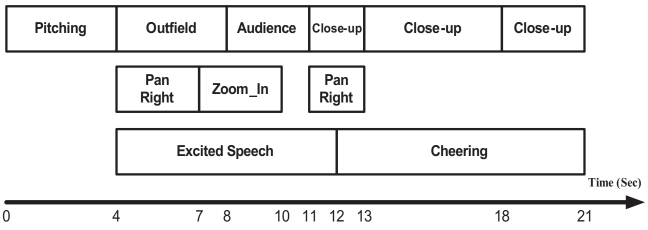


Fig. 8. Temporal database of a homerun event.

Table 2
Event statistics of experimental video dataset.

	HR	OutHit	OutOut	InOut	SO	Walk	Total
NPB	7	40	33	65	31	17	193
MLB	8	30	47	48	43	15	191
NPB + MLB	15	70	80	113	74	32	384

Table 3
Performance comparison of three methods.

	Proposed method	Product-HMMs	FP-match
Accuracy	64.3%	54.5%	34.2%

Table 4
Confusion matrix of proposed Co-occurrence HMM method.

	Homerun	OutHit	OutOut	InOut	SO	Walk	Recall (%)
Homerun	5	5	0	0	0	0	50.0
OutHit	0	30	16	0	1	0	63.8
OutOut	2	0	44	0	7	0	83.0
InOut	0	1	0	37	36	1	49.3
SO	1	0	0	0	45	3	91.8
Walk	0	1	0	0	17	3	14.3
Precision	62.5%	81.1%	73.3%	100%	42.5%	42.9%	

Table 5
Confusion matrix for Product-HMMs method.

	Homerun	OutHit	OutOut	InOut	SO	Walk	Recall (%)
Homerun	2	5	3	0	0	0	20.0
OutHit	4	31	12	0	0	0	66.0
OutOut	2	7	42	0	2	0	79.2
InOut	1	5	48	13	7	1	17.3
SO	0	0	2	0	47	0	96.0
Walk	0	0	0	0	17	4	19.0
Precision	22.2%	64.6%	39.3%	100%	64.4%	80.0%	

Table 6
Confusion matrix for FP-match method.

	Homerun	OutHit	OutOut	InOut	SO	Walk	Recall (%)
Homerun	10	10	0	0	0	0	50.0
OutHit	22.8	41.6	3	0.2	0	0	61.5
OutOut	12	29.5	20.3	1.9	1.2	0	31.3
InOut	10.9	30.1	16.4	27.3	1.3	0.3	31.6
SO	26	26	0	0	4.5	18.5	6.0
Walk	5	6.3	0	0	2.1	12.6	48.5
Precision	11.5%	29.0%	51.1%	92.9%	49.5%	40.1%	

into homerun and one is classified into OutHit. The test sequence is recognized as 0.66 HR and 0.33 OutHit.

Number of states is an important design parameter in the context of HMM. To know how it affects recognition performance, experiments were conducted for using different numbers of HMM states in the proposed method, where mixed dataset (MLB + NPB) with multimodal features (Visual/Motion/Audio) were adopted and the result is shown in Fig. 9. It is observed that the maximum recognition accuracy occurred at state number = 8, which is the value the proposed method used in our experiments. Since the optimal number of state is not universal for all HMM methods, we also conducted similar experiments for *Product-HMMs*. By comparing Fig. 9 and Table 3, it is worth noticing that, even with non-optimal state number (e.g., 10), the proposed method still outperformed *Product-HMMs* with optimal state number, indicating the superiority of the proposed method.

3.2.2. Effect of multimodal feature integration

To explore the effect of multimodal features integration, different combinations of multimodal features were employed for the three methods and the performance comparisons are listed in Table 7. It is worth noticing that if only single-modality feature is employed, both proposed co-occurrence HMM and Product-HMM become conventional HMM and therefore, obtain the same performance as shown in the last three rows of the table. As the number of adopted feature types increases, however, the performance of the proposed method improves significantly, but that of Product-HMM does not have obvious improvement. The difference of accuracy rates can be up to 11.0% when three modalities are employed. The reason is that Product-HMM which assumes independent relation among different feature-types utilizes only *before*, *after*, and *meet* temporal relationships among single features. Besides *before*, *after*, and *meet* relationships, the proposed co-occurrence coding can describe temporal overlapping relationships among multimodal features such as the *overlap* among outfield shot, pan right, and excited speech or the *overlap* between close-up shot and cheering in homerun example of Fig. 7. Therefore, Product-HMM cannot obtain much benefit from the increase of adopted feature types, while the proposed method which can utilize full temporal relationship among multimodal features can improve the performance much more. Moreover, the proposed method performed better for visual + motion than for visual + audio. The reason is that, compared to the integration of visual and audio features, the integration of visual and motion makes the event classes more distinguishable to each other by the proposal method. However, this may not be always true for all the test data and all the applications. As for FP-match method, it did not have obviously improvement when the number of adopted feature types increases. The reason is that it suffers from multiple-matching problems. As the simple example shown in Fig. 10, the temporal sequence of SO event is also contained by that of OutHit event, therefore the SO test data will be recognized as both SO and OutHit events. Such multiple-matching problems make FP-match method unable to take advantages from multimodal features.

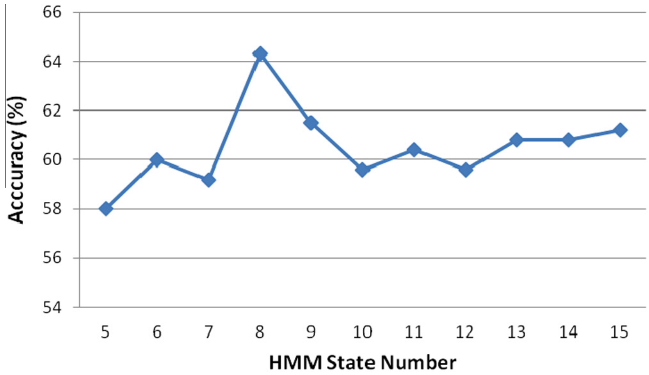


Fig. 9. Variation of recognition accuracy over number of HMM states.

3.2.3. Effect of dataset selection

To examine how dataset selection affects event recognition performance, baseball videos are divided into three datasets: MLB, NPB and MLB + NPB (mixed), according to the countries that broadcasted them. The three methods are performed on each dataset and the event recognition performances are compared. Since the shooting habits of the cameramen and the reactions of the sports anchor and the audience could be different, mixed dataset combined different temporal feature transition patterns from different broadcasts. As shown in Table 8, the performances of product-HMMs and FP-match methods on single dataset alone were better than on the mixed dataset. On the contrary, the proposed method performed better on mixed dataset, indicating that the more training data we used, the more benefits the proposed method can take from the different feature-patterns of different datasets and therefore, improve the recognition performance. Since HMM is essentially a statistical method, large training set is needed to obtain the statistical characteristics. The more and different the training samples are adopted, the better classification performances the HMM is probably to perform unless the training set has been large enough to cover all the characteristics to distinguish all the event classes. Because of the statistical property of HMM, the proposed method benefits from more training datasets.

3.2.4. Recognition of more event classes

This section shows event recognition performances of recognizing more event types. In addition to event recognition on six-type events (described in the experiment setup section), experiments were also conducted for four more event types: *double play* (DP), *sacrifice hit* (SH), *infield hit* (InHit), and *hit by pitch* (HBP). We collected 15 DP, 13 SH, 7 InHit, and 4 HBP events from 8 baseball games totaling 28 h. The sample set is not large because the four event types are not frequent events in baseball games. The occurrence rate of the four event types in 8 games is about 9% which is approximated by dividing (15 + 13 + 7 + 4) to 432 (8 games $*$ 9 innings in a game $*$ 6 events in an inning). The results of recognizing 10 events were shown in Tables 9 and 10 and it is observed that all three methods achieved a lower recognition rate than on six-type events. This is due to that the more event types we selected for recognition, the more similarities there may exist among them and therefore, it is more difficult for recognition methods to distinguish them correctly. Noticing that, compared to 6-type event classification, even though the performance of all the methods dropped on 10-type event classification, the proposed method still performed the best, with 52.1% accuracy rate. For the proposed method, recognition of Homerun, OutHit, OutOut, SO and Walk events are nearly not affected by the new event types. However, many InOut events are misclassified into DP, SH and InHit because they are all infield events. The multiple matching problem of FP-match method became more serious for 10-type event recognition with a poor accuracy rate of 21.6%.

3.2.5. Testing on type of sequences not learned before

This section examines the performance of the proposed method by using the kind of test sequences that are not learned before. Towards this goal, a CPBL (Chinese Professional Baseball League/Taiwan) baseball dataset consisting of 70 events as shown in Table 11 is adopted for event testing on the HMMs trained from MLB and NPB training datasets. Note that these trained HMMs were not trained by any CPBL sequence. The classification results of the

Table 7
Performance comparisons for different multimodal feature integration.

	Proposed method (%)	Product-HMMs (%)	FP-match (%)
Visual + motion + audio	64.3	54.5	34.2
Visual + motion	58.0	53.3	33.9
Visual + audio	54.5	53.3	40.3
Motion + audio	48.6	51.8	32.5
Visual	53.3		40.9
Motion	47.1		38.1
Audio	40.0		27.0

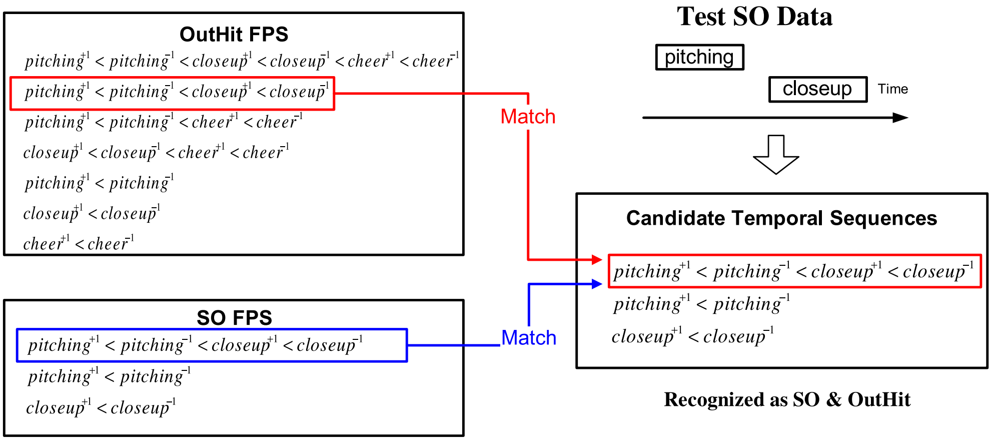

OutHit FPS
$pitching^{+1} < pitching^{-1} < closeup^{+1} < closeup^{-1} < cheer^{+1} < cheer^{-1}$
$pitching^{+1} < pitching^{-1} < closeup^{+1} < closeup^{-1}$
$pitching^{+1} < pitching^{-1} < cheer^{+1} < cheer^{-1}$
$closeup^{+1} < closeup^{-1} < cheer^{+1} < cheer^{-1}$
$pitching^{+1} < pitching^{-1}$
$closeup^{+1} < closeup^{-1}$
$cheer^{+1} < cheer^{-1}$
Match
Test SO Data
pitching
closeup
Time
Candidate Temporal Sequences
$pitching^{+1} < pitching^{-1} < closeup^{+1} < closeup^{-1}$
$pitching^{+1} < pitching^{-1}$
$closeup^{+1} < closeup^{-1}$
SO FPS
$pitching^{+1} < pitching^{-1} < closeup^{+1} < closeup^{-1}$
$pitching^{+1} < pitching^{-1}$
$closeup^{+1} < closeup^{-1}$
Match
Recognized as SO & OutHit

Fig. 10. A simple example of multiple recognition problem of FP-match method.

Table 8
Performance comparisons for different dataset selection.

	Proposed method (%)	Product-HMMs (%)	FP-match (%)
MLB dataset	54.1	60.3	35.7
NPB dataset	61.7	67.2	40.3
MLB + NPB dataset	64.3	54.5	34.2

Table 9
Performance comparison of three methods for 10 event recognition.

	Proposed method	Product-HMMs	FP-match
Accuracy	52.1%	48.6%	21.6%

Table 11
The CPBL dataset.

	HR	OutHit	OutOut	InOut	SO	Walk	Total
CPBL	3	13	14	21	13	6	70

Table 12
Performance comparison of three methods.

	Proposed method	Product-HMMs	FP-match
Accuracy	51.4%	48.6%	32.7%

Table 10
Precision/Recall of the three methods for 10 event recognition.

	Proposed method		Product-HMMs		FP-match	
	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)
Homerun	50	40	25	20	13.5	48.8
OutHit	73.7	59.6	73.7	59.6	37.2	52.4
OutOut	70.7	77.4	55.3	79.2	74.5	25
InOut	93.3	18.7	100	8	89	3.2
SO	55.6	91.8	73.4	95.9	0	0
Walk	100	14.3	100	19	39.1	33.8
DP	17.4	40	12.5	20	4.7	20.5
SH	13.3	44.4	15.4	44.4	4.6	29.7
InHit	5.9	25	2.4	25	2.5	3.6
HBP	28.6	100	0	0	2.5	50

three methods are shown in the tables below where Table 12 shows the accuracy rate and Table 13 shows the precision and recall rates. It is observed that the accuracy of all methods drop since this kind of sequences were not learned before. However, the

Table 13
Precision/recall of the three methods CPBL dataset.

	Proposed method		Product-HMMs		FP-match	
	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)	Prec. (%)	Recall (%)
Homerun	50	33.3	0	0	11.1	50
OutHit	55.6	38.5	46.7	58.3	28.4	59.4
OutOut	58.8	71.4	35.7	71.4	53.1	33.7
InOut	100	38.1	100	9.5	92.3	22.7
SO	33.3	76.9	65	100	71.4	11.8
Walk	50	33.3	66.7	33.3	29.1	33.3

Table 14
Execution time comparison of three methods.

	Proposed method (s)	Product-HMMs (s)	FP-match (s)
Training Time	0.197	0.188	0.016
Testing Time	0.0002	0.0002	9.34

decrease is not very large (about 5.9% on average). The results imply that the interval feature structures of different baseball video dataset are similar in some extent although they might have different shot transitions and editing habits.

3.2.6. Comparisons of computational complexity and execution time

The computational complexity of HMM consists of time complexity and memory space complexity. Given a HMM which has N states, M symbols, average length of training sequences as T, and average length of testing sequences as L, the time complexity of HMM training is $O(N^2T)$ using Baum-Welch method and the time complexity of HMM testing is $O(L)$.

Therefore, the time complexity bound of HMM training and testing is determined by HMM training. The space complexity of HMM is $O(N^2 + NM)$. Let L_v, L_m, and L_a be the average lengths of training sequences from visual, motion, and audio sources, respectively. The training time complexity of product HMM method is $O(N^2 \cdot \max(L_v, L_m, L_a))$. The average sequence length of proposed co-occurrence HMM method is determined by endpoint number of multiple source features, therefore it is bounded by $O(L_v + L_m + L_a)$. The average sequence length of the two methods are the same order, therefore the training time complexities of proposed co-occurrence HMM and product-HMMs methods are both $O(N^2 \cdot \max(L_v, L_m, L_a))$, and the testing time complexities are both $O(L)$. The total time complexities of both methods are $O(N^2 \cdot \max(L_v, L_m, L_a))$. On the other hand, the proposed co-occurrence HMM method have higher space complexity than product-HMMs method since combined features generates more symbols. The space complexity of product-HMMs method is $O(N \cdot \max(L_v, L_m, L_a))$ and the space complexity of the proposed co-occurrence HMM method is $O(N \cdot (L_v + 1) \cdot (L_m + 1) \cdot (L_a + 1))$. For the actual case in the experiments, the maximum symbol number of proposed co-occurrence HMM method is $(8 + 1)(2 + 1)(2 + 1)(2 + 1)(2 + 1) = 729$ which is obtained by multiplying feature-type number of visual, pan, tilt, zoom, and audio. After incrementally adding new symbols for co-occurrence features during the coding process, the total number of produced co-occurrence symbols is 159.

The FP-match method has two main steps: (1) frequent pattern mining, and (2) classification by TS containing verification. Therefore, the time depends totally on these two steps. Let T_M denote the training time which is the time to mine the frequent pattern sets of each event class. Let L denote the average length of testing temporal sequences, and W denote the sliding window size. The classification by TS containing verification consists of (L-W + 1) times of window slides and generates $(L - W + 1)2^W$ candidate temporal sequences for verification. Let T_C denote the containment verification time between one temporal sequence and the frequent pattern sets. Then the testing complexity is $O((L - W + 1)2^W T_C$, and the total complexity of FP-match method will be $O(2^W)$.

Besides computational analysis, execution time comparison has also been made by running the three methods on an Intel Duo CPU T7300, 2.00 GHz with 2 GB RAM for our baseball experimental dataset. The execution times of the three methods are compared and listed in Table. 14. It is observed that the two HMM-based methods have similar training times, while the FP-match method has a relatively short training time. However, the two HMM-based methods have much shorter testing time, compared to FP-match method. This is due to that, once HMMs have been trained, the classification using HMMs can be done very fast by computing probabilities. The testing time of FP-match method is much slower because of containment verification of candidate temporal sequences.

4. Conclusions and future work

In this paper, we have developed a novel system that is able to automatically detect and classify high-level events by using temporal context of multimodal mid-level features. A new event classification approach aiming at taking advantages of full temporal relationship of multimodal features and recognizing events by HMM probabilistic classification is proposed. By representing a video using a temporal database of mid-level features, a co-occurrence symbol transformation method is proposed, which encodes multimodal features as a sequence of symbols with full temporal relationship kept among them. Then, the resulting co-occurrence symbol sequence is fed into HMM directly for maximum likelihood event classification. The experimental results have demonstrated the efficiency, effectiveness, and robustness of the proposed method. In the future, more features will be studied to explore the feature selection problem. The following questions need to be answered: (1) How to find the optimal set of mid-level features; (2) How to decide whether a mid-level feature should be selected; (3) What kind of measurement could be used for feature selection. Automatic detected features would be tested to demonstrate the fault tolerance of proposed method. Moreover, more application domains will be investigated thoroughly, not only in baseball domain but other sports domain and stocks etc., to support the generality of the proposed method.

References

[1] X. Lexing, H. Sundaram, M. Campbell, Event mining in multimedia streams, Proc. IEEE 96 (4) (2008) 623–647.

[2] C. Poppe, S.D. Bruyne, R.V.d. Walle, Generic architecture for event detection in broadcast sports video, in: Paper presented at the Proceedings of the 3rd International Workshop on Automated Information Extraction in Media Production, Firenze, Italy.

[3] D.W. Tjondronegoro, Y.P.P. Chen, Knowledge-discounted event detection in sports video, IEEE Trans. Syst. Man Cybernet. A: Syst. Hum. 40 (5) (2010) 1009–1024.

[4] T. Zhang, C. Xu, G. Zhu, S. Liu, H. Lu, A generic framework for event detection in various video domains, in: Paper presented at the Proceedings of the International Conference on Multimedia, Firenze, Italy.

[5] M. Fleischman, D. Roy, Grounded language modeling for automatic speech recognition of sports video, in: Proceedings of ACL-08: HLT, 2008, pp. 121–129.

[6] X.-Q. Zhu, X.-D. Wu, A.K. Elmagarmid, Z. Feng, L. Wu, Video data mining: semantic indexing and event detection from the association perspective, IEEE Trans. Knowl. Data Eng. 17 (5) (2005) 665–677.

[7] M. Chen, S.-C. Chen., M.-L. Shyu, Hierarchical temporal association mining for video event detection in video databases, in: Data Engineering Workshop, IEEE 23rd International Conference on, 17–20 April 2007, pp. 137–145.

[8] M. Chen, S.-C. Chen, M.-L. Shyu, K. Wickramaratna, Semantic event detection via multimodal data mining, IEEE Sig. Proc. Mag. 23 (2) (2006) 38–46.

[9] J.F. Allen, Maintaining knowledge about temporal intervals, Commun. ACM 26 (11) (1983) 832–843.

[10] C.G.M. Snoek, M. Worring, Multimedia event-based video indexing using time intervals, IEEE Trans. Multimed. 7 (4) (2005) 638–647.

[11] M. Fleischman, P. Decamp, D. Roy, Mining temporal patterns of movement for video content classification, in: Paper presented at the Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, Santa Barbara, California, USA.

[12] M. Fleischman, D. Roy, Unsupervised content-based indexing of sports video, in: Paper Presented at the Proceedings of the International Workshop on Workshop on Multimedia Information Retrieval, Augsburg, Bavaria, Germany.

[13] S.-Y. Wu, Y.-L. Chen, Mining nonambiguous temporal patterns for interval-based events, IEEE Trans. Knowl. Data Eng. 19 (6) (2007) 742–758.

[14] M.-S. Dao, N. Babaguchi, A new spatio-temporal method for event detection and personalized retrieval of sports video, Multimed. Tools Appl. 50 (1) (2010) 227–248.

[15] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[16] T. Starner, J. Weaver, A. Pentland, Real-time American sign language recognition using desk and wearable computer based video, IEEE Trans. Pattern Anal. Mach. Intell. 20 (12) (1998) 1371–1375.

[17] J. Yamato, J. Ohya, K. Ishii, Recognizing human action in time-sequential images using hidden Markov model, in: Computer Vision and Pattern Recognition, 1992, Proceedings CVPR ‘92, 1992 IEEE Computer Society Conference on, 15–18 Jun 1992, pp. 379–385.

[18] P. Baldi, Y. Chauvin, T. Hunkapiller, M.A. McClure, Hidden Markov models of biological primary sequence information, Proc. Nat. Acad. Sci. 91 (3) (1994) 1059–1063.

[19] B. Li, M. I. Sezan, Event detection and summarization in sports video, in: Content-Based Access of Image and Video Libraries, 2001 (CBAIVL 2001). IEEE Workshop on, 2001, pp. 132–138.

[20] J. Assfalg, M. Bertini, C. Colombo, A. Del Bimbo, W. Nunziati, Semantic annotation of soccer videos: automatic highlights identification, Comput. Vis. Image Underst. 92 (2–3) (2003) 285–305.

[21] P. Chang, M. Han, Y.-H. Gong, Extract highlights from baseball game video with hidden Markov models, in: Image Processing. 2002. Proceedings. 2002 International Conference on, 2002, pp. 609–612.

[22] G. Xu, Y.-F. Ma, H.-J. Zhang, S. Yang, Motion based event recognition using HMM, in: Pattern Recognition, 2002. Proceedings of the 16th International Conference on, 2002, vol. 832, pp. 831–834.

[23] N. Babaguchi, Y. Kawai, T. Kitahashi, Event based indexing of broadcasted sports video by intermodal collaboration, IEEE Trans. Multimed. 4 (1) (2002) 68–75.

[24] C.G.M. Snoek, M. Worring, A review on multimodal video indexing, in: Multimedia and Expo, 2002. ICME ‘02. Proceedings. 2002 IEEE International Conference on, 2002, vol. 22, pp. 21–24.

[25] M.-F. Weng, Y.-Y. Chuang, Multi-cue fusion for semantic video indexing, in: Paper presented at the Proceedings of the 16th ACM International Conference on Multimedia, Vancouver, British Columbia, Canada.

[26] H. Xu, T.-S. Chua, The fusion of audio-visual features and external knowledge for event detection in team sports video, in: Paper presented at the Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, New York, NY, USA.

[27] M. Brand, N. Oliver, A. Pentland, Coupled Hidden Markov Models for Complex Action Recognition, Proc. IEEE Computer Vision and Pattern Recognition (1996).

[28] A. V. Nefian, L. H. Liang, X. X. Liu, X. Pi, C. Mao, K. Murphy, A coupled HMM for audio-visual speech recognition, Proc. IEEE ICASSP'2002, Vol. 2, pp. 2013–2016, May, 2002.

[29] J. Huang, Z. Liu, Y. Wang, Y. Chen, E.K. Wong, Integration of multimodal features for video scene classification based on HMM, in: Multimedia Signal Processing, 1999 IEEE 3rd Workshop on, 1999, pp. 53–58.

[30] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, C.-S. Xu, A mid-level representation framework for semantic sports video analysis, in: Paper presented at the Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA.

[31] L.-Y. Duan, M. Xu, Q. Tian, C.-S. Xu, J.S. Jin, A unified framework for semantic shot classification in sports video, IEEE Trans. Multimed. 7 (6) (2005) 1066–1083.

[32] Y. Ding, G. Fan, Event detection in sports video based on generative-discriminative models, in: Paper presented at the Proceedings of the 1st ACM International Workshop on Events in Multimedia, Beijing, China.

[33] P. Bouthemy, M. Gelgon, F. Ganansia, A unified approach to shot change detection and camera motion characterization, IEEE Trans. Circ. Syst. Video Technol. 9 (7) (1999) 1030–1044.

[34] G. Tardini, C. Grana, R. Marchi, R. Cucchiara, Shot detection and motion analysis for automatic MPEG-7 annotation of sports videos, image analysis and processing – ICIAP 2005, in: F. Roli, S. Vitulano (Eds.), Lecture Notes in Computer Science, vol. 3617. Springer, Berlin/Heidelberg, 2005, pp. 653–660.