The Fusion of Audio-Visual Features and External Knowledge for Event Detection in Team Sports Video

Paper #:1568934849

ABSTRACT

Most existing systems detect events in broadcast team sports video using only internal audio-visual features with limited success. We found that there are many sources of external knowledge such as the match reports and real time game log that are widely available on newspapers and the web that can help in detecting events. This paper proposes a scalable framework that utilizes both internal AV features and various external knowledge sources to detect events and identify their boundaries from full-length match videos. Besides detecting events, the framework is also capable of discovering detailed semantics and answer questions regarding these semantics. In this way, the framework supports diverse questions, including personalized summarization. We demonstrate the effectiveness of the framework using 3 full-length soccer matches.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: abstracting methods, indexing methods.

General Terms

Algorithms, Design, Experimentation.

Keywords

Sports video, semantics, event detection, question answering.

1. INTRODUCTION

Sports video, a popular video genre having a large audience base and good commercial potential, has been an active research area for semantic analysis. Various sports have been studied, including baseball [5], soccer [17], American football [16], basketball [13], swimming [11], tennis [3], cricket [4], and others. As a basic semantic entity, event means a series of actions or a state with a distinct meaning. Event detection is to locate the video segments that depict events, and preferably, to identify the boundaries as well. A lot of research has been done on event detection [7], which serves as the basis for annotation [14], indexing [13], highlight generation [9] and summarization [4].

Most current research focuses on analyzing the internal audio-visual (AV) features of video to detect events. Early efforts attempted to infer events from patterns of physical features [4] or fixed scenes [18]. Bertini et al.[11], Assfalg [14] extended the AV analysis approach by adopting the more rigorous "model-based" approach that uses handcrafted domain-specific event models to infer the occurrences of events. Similar deterministic approaches were also applied in systems with features from multi-modalities of: audio [12], videotext [7], and automatic speech-to-text transcripts [9]. Model-based approaches have been very successful in practice, as they are precise, easy to implement and computationally efficient. However, models are laborious to construct and are seldom reusable, and they are unable to handle subtle events that do not have a distinct physical appearance, such as *yellow/red card* events in soccer. Because of these limitations, only a subset of events in a domain can be detected using the model-based approach.

As we incorporate more features, model-based approach becomes more inefficient and difficult to hand-craft. Machine learning approach is naturally evolved to support fully automated event detection. Zhou et al. [13] used decision tree to model the domain knowledge of basketball in rules; while Han et al. [5] utilized maximum entropy classifiers to find the most suitable features and models for event detection. These methods analyzed each shot independently, without taking into account their sequential relationships. Assfalg et al. [15] recognized the role of sequential relations, and modeled the event using HMMs.

To boost robustness against variation in low-level features and adaptability of event detection schemes, mid-level representation were used in event detection. For example, Duan et al. [3] used shot classes and audio keywords as mid-level features.

In a broader interpretation, event detection can also be viewed as the task of detecting different event structures together with boundaries. Many works on structure analysis for soccer video recognized two structures of the match: break-play. The methods basically fell into three groups. Xu et al. [2] identified structures by scene types (long, medium, and close-up) in a deterministic way. Xie et al. [17] modeled each structure using HMMs, and classified which model the structure belongs to. Though single layer HMMs are successful in classifying a candidate event, they cannot identify the structure's boundaries. Xu et al. [1] explored the use of hierarchical HMM (HHMM) to detect events and reconstruct the temporal relations between them; while Xie et al. [20] utilized HHMM to find the break-play structures and their boundaries simultaneously in an unsupervised manner.

A major trend in event detection is the intermodal collaboration of video, audio and textual information. Sources for textual information are videotext [7], ASR [9], closed caption text [10, 16, 19], and the web [10]. Most of these systems analyzed closed caption text, with the help of audio and shot boundary detection, to detect general events [16] and segment "live" play of American football. Babaguchi et al. [10] analyzed several modes of inter-modal collaborations, combining two or more features of textual, visual, auditory, videotext and/or external metadata from the web. With heavy reliance on closed caption text, these systems cannot work in situations where text is not available.

The above reviews show that most event detection systems based on internal AV features have inevitable drawbacks. Deterministic model-based approaches have problem with high dimensionality in feature space; while learning-based methods require a large number of training samples to cover all possible situations. Due to their weaknesses, neither method could give really satisfactory results over large test dataset. On the other hand, Babaguchi et al. [10] suggested that the game logs (or gamestats) given on the web can be utilized as cues to identify events. In fact, besides game logs, there is a lot more information available from diverse sources. It has different levels of details about a particular event. All these information sources should be leveraged to support semantic analysis of sports video. To differentiate this kind of information from the audio-visual signals contained inside the video sequence, we call it external knowledge. We expect the external knowledge to enhance the performance of event detection as some of the sources, like game log, are both accurate and complete, although they often lack boundary information.

The aim of this paper is to develop a scalable framework that fuses internal AV features and multiple external knowledge sources to detect events and identify their boundaries in broadcast team sports video. In the ideal case when all external knowledge sources are available, the framework should be able to detect the full range of events and their boundaries with high accuracy. As less external knowledge is available, the framework should gracefully scale down its event detection capabilities and accuracy. When no external knowledge is available, the framework should be able to detect major events using only internal AV features. In addition to event detection, the framework also aims to support event question answering and summarization over a wide range of event types and player contributions. This is similar to video question answering discussed in Yang et al. [21].

The main contribution of this paper is in developing and demonstrating the effectiveness of a scalable event detection framework incorporating both internal AV features and external knowledge sources.

The rest of the paper is organized as follows: Section 2 explains the event detection problem for soccer video. Section 3 describes the use of AV features and external knowledge to detect events. Section 4 presents the experiment results. Finally, Section 5 concludes the paper.

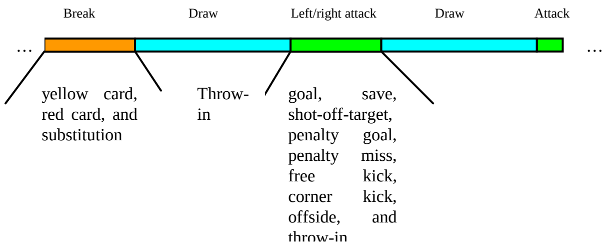
2. Event Detection in Soccer Videos

2.1 Events and Event Models

In this paper we focus on soccer video, although our framework is applicable to other team sports. The common characteristic in all team sports is in manipulating a ball towards a target or goal post, around where most interesting events take place. Therefore activities near the goal post and far from it should be differentiated with respect to semantic importance. In soccer, the game can be grouped into 3 phases, representing different stages of play towards scoring goals. The phases are: *break*, *draw*, and *attack*, where

- *Break* – The match is not going on. It could be prior to start, pause, or finish of match.
- *Draw* – No team is on the offense. It is also called "*plain play*" in some papers.
- *Attack* – One team is on the offense, while the other team defends.

Within the above three phases, a range of events take place. The full range of important event types in soccer are: *goal*, *save*, *shot-off-target*, *penalty goal*, *penalty miss*, *free kick*, *corner kick*, *yellow card*, *red card*, *throw-in*, *substitution*, and *offside*. This list is derived from game logs, web commentaries, match reports, and studio reviews. Most event types are associated with certain phases. Figure 1 depicts a hierarchical model for event detection, by first structuring the whole match into phases then detecting the events within particular phases. As dictated by the property of the game, the majority of events are associated with *attack*. Still, there is exception, like *throw-in* may happen in multiple phases.



Break
Draw
Left/right attack
Draw
Attack
...
...
yellow card, red card, and substitution
Throw-in
goal, save, shot-off-target, penalty goal, penalty miss, free kick, corner kick, offside, and throw-in

Figure 1: Event types associated with each phase

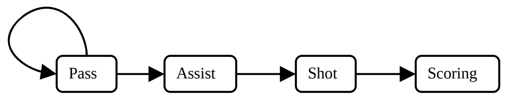


Figure 2: Event model for goal

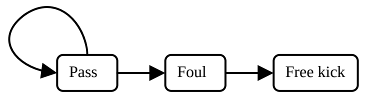


Figure 3: Event model for free kick

Each event is in turn decomposed into a series of temporal actions. Most actions have predicable patterns in the AV space, like the action of scoring a goal is detected by distinct visual sequence accompanied by excitement in commentator's speech. Most events can be modeled as a state-transition graph comprising actions. For example, the event models for *goal* and *free-kick* are shown in Figures 2 and 3 respectively.

2.2 External Knowledge for Event Detection

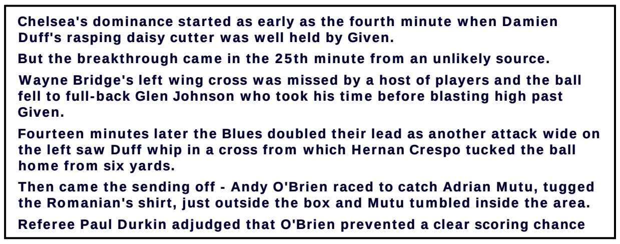
Most existing systems are based on AV features, which are effective only in detecting events that have distinct physical characteristics. Examples of such events are *goals* and *attempt-on-goals*. Hence most existing works focused on detecting such events [11] [18].

AV features, however, are not effective for subtle events such as *yellow card*, *red card* and *offside*, or events involving semantic entities such as players. In addition, it is unable to differentiate the importance of events within each event category. For example, a foul event leading to a *yellow card* may be considered as pivotal that changes the course of the whole match. Such events can only be detected with the help of additional knowledge.

Some typical sources of external knowledge are:

- match reports on the newspaper and web;
- reviews made in the studio between the halves and after the whole match;
- live commentary on the web;
- real-time game logs on the web, like what ESPN provides for the English Premier League (EPL).

Here we focus on two such external knowledge sources that are widely available and provide vastly different levels of details to the match. They are the match reports and real-time game logs on the web. An example of match report (partial) is given in Figure 4. By performing linguistic analysis on the match report, we expect to extract information on: (a) *goals* with timing and scorers' names; (b) important *saves*; and (c) important *yellow/red card* events and *substitutions* that are considered by the newspaper reporters to be critical to the course of the match. Such knowledge may be used to help AV analysis in detecting major events more accurately and in assigning higher importance to events mentioned in the reports.



Chelsea's dominance started as early as the fourth minute when Damien Duff's rasping daisy cutter was well held by Given.

But the breakthrough came in the 25th minute from an unlikely source.

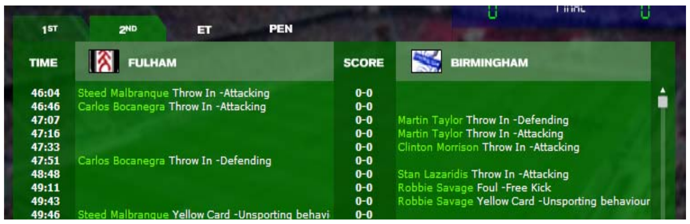
Wayne Bridge's left wing cross was missed by a host of players and the ball fell to full-back Glen Johnson who took his time before blasting high past Given.

Fourteen minutes later the Blues doubled their lead as another attack wide on the left saw Duff whip in a cross from which Hernan Crespo tucked the ball home from six yards.

Then came the sending off - Andy O'Brien raced to catch Adrian Mutu, tugged the Romanian's shirt, just outside the box and Mutu tumbled inside the area.

Referee Paul Durkin adjudged that O'Brien prevented a clear scoring chance

Figure 4: Example of Match Report
(Source - http://www.chelseafc.com/)



TIME	FULHAM	SCORE	BIRMINGHAM
46:04	Steed Malbranque Throw In -Attacking	0-0	
46:46	Carlos Bocanegra Throw In -Attacking	0-0	
47:07		0-0	Martin Taylor Throw In -Defending
47:16		0-0	Martin Taylor Throw In -Attacking
47:33		0-0	Clinton Morrison Throw In -Attacking
47:51	Carlos Bocanegra Throw In -Defending	0-0	
48:48		0-0	Stan Lazaridis Throw In -Attacking
49:11		0-0	Robbie Savage Foul -Free Kick
49:43		0-0	Robbie Savage Yellow Card -Unsporting behaviour
49:46	Steed Malbranque Yellow Card -Unsporting behavi	0-0	

Figure 5: Example of game log
(Source - http://www.soccernet.com/)

An example of game log (partial) is given in Figure 5. The information resolution of game log is action. Each entry of game log corresponds to an action, giving information on the time, player, action type, and outcome of the action. The whole game log provides a complete description of the flow of actions. It is safe to postulate that we can build all types of events with boundaries by merging appropriate entries.

With the fusion of AV features and external knowledge, the details of events that can be extracted are: <event start time, event end time, event type 1, event type 2, names of players involved>. The dual event type is to cater to events that have more than one event type such as the *goal* scored by *penalty*, *corner kick* or *free kick*.

3. Proposed Approach

3.1 Overall Framework

From the semantics point of view, the framework comprises four layers, namely: the feature layer, event layer, semantic unit layer, and user layer, as shown in Figure 6.

The lowest layer is the feature layer, which models the internal audio-visual features. It does not contain much semantic meaning.

Next is the event layer, which models the events with boundaries. Most events are detectable by analyzing internal AV features only. However, the full set of events requires supplementary analysis of external knowledge.

Above the event layer is the semantic unit layer that models semantic units in the granularity of actions. Each unit corresponds to a particular action in an event, and is modeled by the 3-D space of: event ID, action type and player who performed the action. For ranking purposes, each semantic unit has an importance value, which is a function of the three identifying variables in the 3-D space. The function reflects the user's interest in different events, action types or players. The importance value could be the default ones to suit general users' interests, or specific to a user based on a user profile that bias towards certain event types, action types or players. The use of semantic units permits great flexibility in forming semantic entities and in supporting diverse user profiles.

Finally at the User Layer, we support question-answering on a wide range of semantic entities and user profiles. A semantic entity is the subject of the question, e.g. the contribution of *Bergkamp* (the player). Each semantic entity is modeled by a set of semantic units. Note that the constituent semantic units may describe different events, action types or players. For example, the contribution of *Bergkamp* comprises all semantic units involving him, which may span multiple events and multiple action types. A question may involve multiple semantic entities, for example, the question "who makes the greatest contribution" involves the contribution of every player. For each question, the system identifies all semantic entities involved, calculates the importance value of each of them, ranks them (if necessary), and returns the suitable ones as the answer. It should be noted that the personalized summarization of the whole match is actually just a particular query in the system.

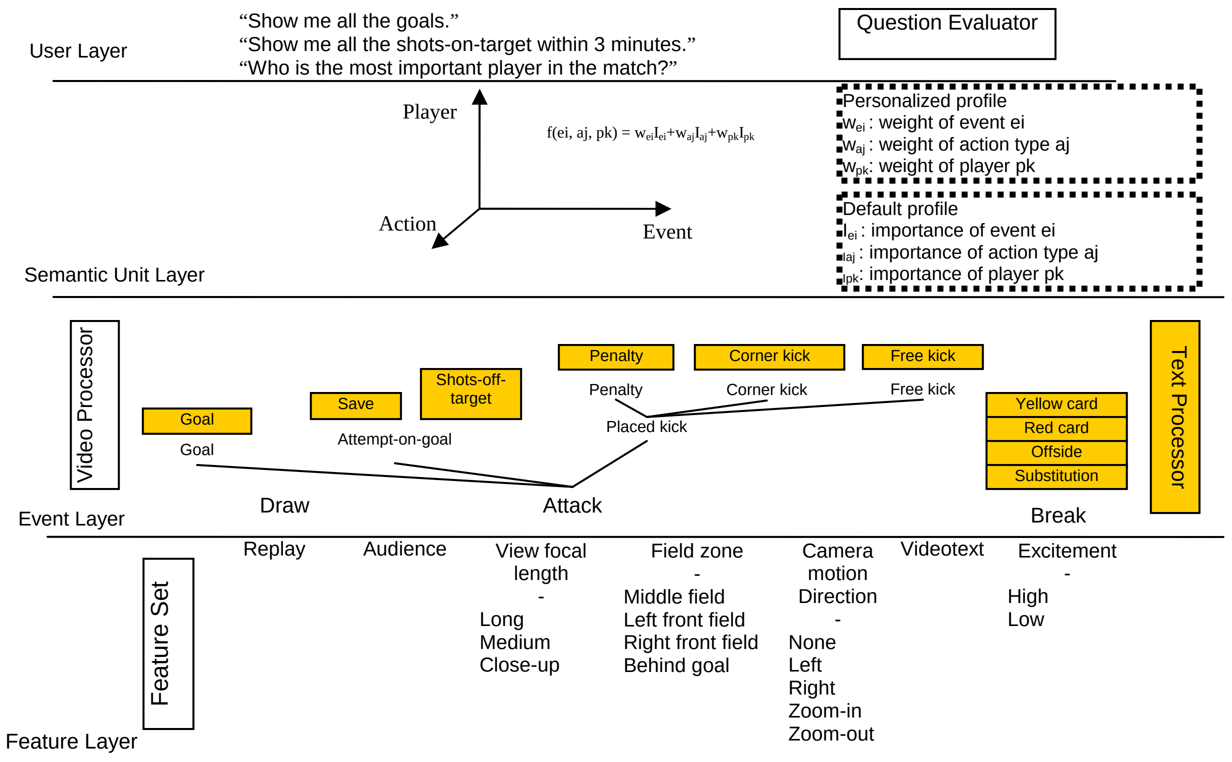

User Layer
"Show me all the goals."
"Show me all the shots-on-target within 3 minutes."
"Who is the most important player in the match?"
Question Evaluator
Player
Action
Event
f(ei, aj, pk) = $w_{ei}I_{ei}+w_{aj}I_{aj}+w_{pk}I_{pk}$
Personalized profile
w_{ei} : weight of event ei
w_{aj} : weight of action type aj
w_{pk}: weight of player pk
Default profile
I_{ei} : importance of event ei
I_{aj} : importance of action type aj
I_{pk}: importance of player pk
Semantic Unit Layer
Video Processor
Text Processor
Goal
Save
Shots-off-target
Penalty
Corner kick
Free kick
Attempt-on-goal
Placed kick
Yellow card
Red card
Offside
Substitution
Draw
Attack
Break
Event Layer
Feature Set
Replay
Audience
View focal length - Long Medium Close-up
Field zone - Middle field Left front field Right front field Behind goal
Camera motion Direction - None Left Right Zoom-in Zoom-out
Videotext
Excitement - High Low
Feature Layer

Figure 6: The framework

From the systems point of view, the framework consists of two major modules: video processor, which analyzes internal AV features, and text processor, which extracts events from external knowledge. The system exhibits different detection capabilities when utilizing different amount and type of external knowledge.

- With the use of internal AV features only, the system can detect major goal-related events and their boundaries.
- With combinatory use of internal AV features and external knowledge derived from match reports, the system can detect major goal-related events and their boundaries, plus important *red/yellow card* events and player *substitutions*, as mentioned in the reports. The reports also provide clues for assigning importance to events, especially among the same type.
- With the combinatory use of internal AV features and external knowledge derived from game log, the system can detect the full range of events and their boundaries. The system also supports queries based on semantic units.

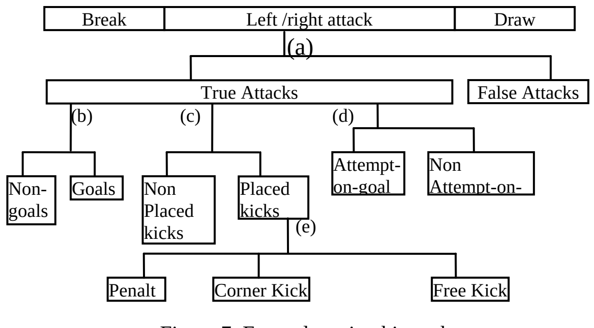


Figure 7: Event detection hierarchy

3.2 Video Processor

The video processor aims to generate a list of entries, each representing an event in terms of <event start time, event end time, event type>. By analyzing only the AV contents, it provides the lowest level of capabilities in the framework when no external knowledge is available. Limited by the characteristics of AV features and the capabilities of techniques to process them, the video processor only aims to detect goal-related events that have distinct patterns in AV feature space. The target events are: *goal*, *attempt-on-goal* (collective term for *save* and *shot-off-target*), *penalty goal*, *penalty miss*, *free kick*, and *corner kick*.

As depicted in Figure 7, the video processor adopts a two-step event detection strategy by performing: (a) global structural analysis using probabilistic methods; and (b) local event detection/classification in a deterministic way. Deterministic methods are more desirable than probabilistic methods in local setting when the variation of patterns is small and when training samples are insufficient. On the other hand, probabilistic methods are more robust globally. It is therefore a good compromise in adopting the two-step strategy.

3.2.1 AV Features to be used

Guided by the analysis of the characteristics of each event type, features that have the best distinguishing power are identified in each detection task. For example, to detect *break* events in the match, we need *view type* as feature to differentiate different views such as replay, audience, medium shot, and close-up. In order to distinguish *attacks* from *plain plays*, we need features like field zone and camera motion direction. To further distinguish *back passes* and *trivial attacks* from *true attacks*, we also need to know the *motion activity*. Also, we distinguish *attempt-on-goals* from among all *true attacks* by looking for audio feature that signifies high excitement level in commentator's speech. Finally, to differentiate *free kicks* from *corner kicks*, we need to know the *cumulative camera motion distance* and *density of field lines*.

The above (partial) analysis suggests that different features should be used for different stages of classification/detection tasks. Table 1 summarizes the features used. The list of features cover most AV features used in most recent reported systems, including a number of mid-level features used in [3].

We employ established algorithms to automatically extract the following features: audience view [12], replay view [6], view focal distance [2], field zone [11], camera motion direction [22], excitement level in commentator's speech [12], videotext [7], and motion activity [17]. We compute cumulative camera motion distance by calculating the sum of the motion distance in the camera motion direction in each P frame over the whole camera motion sequence; and density of field lines as the ratio of the number of field line pixels over the grass area in a frame before the kick.

3.2.2 *The Event Detection Steps*

We use a two-layer hierarchical HMM to segment and classify the phases of the match simultaneously [20]. The features used in this step are *replay view*, *audience view*, *view focal distance*, *field zone* and *camera motion direction*. As shown in Figure 8, four structure elements at the top layer q^1_j (j=1..4) are designed to correspond to the phases of: *break*, *draw*, *left attack*, and *right attack*; while each phase has three states $q^0_{j,k}$ (j=1..4, k=1..3) representing variations that may occur within the phase, plus an exit state e^0_j, through which the corresponding phase transits to another. The states at the bottom layer produce symbols that are observed as the sequence of feature vectors. Note that we split the *attacks* into *left* and *right attacks* to facilitate better subsequent event detection steps and understanding of the progress of the match.

Each individual HMM at the bottom layer is trained with the segmented samples of *break*, *draw*, *left attack*, and *right attack* using the EM algorithm [24] to obtain the parameters $A^0_{k_1,k_2}(k_1,k_2=1,2,3)$ between states. The across-layer and top layer transition parameters $\tilde{A}_{j,k}(j=1,2,3,4;k=1,2,3)$ and $A^1_{k_1,k_2}(k_1,k_2=1,2,3)$ are trained using the labeled sequence of phases, such as *break – plain play – left attack – plain play – right attack – break*… Converged HHMM generates a state in terms of $q^0_{j,k}$ for each unit of test video, and units which are mapped to the same phase are joined.

Figure 7 shows that after *attacks* are identified, a hierarchy of event detection steps is carried out. Table 2 gives the purpose, input, outcome, features, and algorithm of each event detection step. These steps are self-explanatory and will not be elaborated here due to limited space.

Table 1 Features used in video processing

Feature item	**Values**
(1) Replay View	True, false
(2) Audience View	True, false
(3)View focal length	Long shot, medium shot, and close-up
(4) Field zone	Middle field, left front field, right front field
(5) Camera motion direction	None, left pan, right pan, zoom-in, zoom-out
(6)Excitement level in commentator's speech	High, low
(7) Videotext	The text appearing on videotext
(8) Unit duration (the unit is defi below)	Long, short
(9) Motion activity	High, low
(10) Cumulative camera motion distance	Numeric value
(11) Density of field lines	Numeric value

Table 2 Steps after phase segmentation

Step	Purpose	Input	Outcome	Algorithm	Features
(a)	To screen false attacks	All attacks generated by phase segmentation	Classified true attacks and false attacks (back passes, trivial attacks, etc)	HMM classification	(i) Unit duration, (ii) Motion activity
(b)	To detect goal	All attacks generated by structure analysis	Goals	Heuristics: No attack structure is between this attack structure and subsequent presence of goal videotext.	(i) Presence of goal videotext
(c)	To find placed kicks (penalty, or corner kick, or free kick)	Segments of true attacks preceded by a medium shot	Placed kicks	HMM classification	(i) Presence of medium shot, (ii) Unit duration, (iii) Motion activity, (iv) Presence of camera panning
(d)	To detect attempt-on-goal (save or shot-off-target)	All true attacks	Attempt-on-goals	Heuristics: Excitement level = high & followed by close-up	(i) Excitement level, (ii)Presence of subsequent close-up

(e)	To differentiate penalty, corner kick and free kick	All placed kicks	Penalty, corner kick, and free kick	SVM	(i) Cumulative camera motion distance (ii) Density of field lines

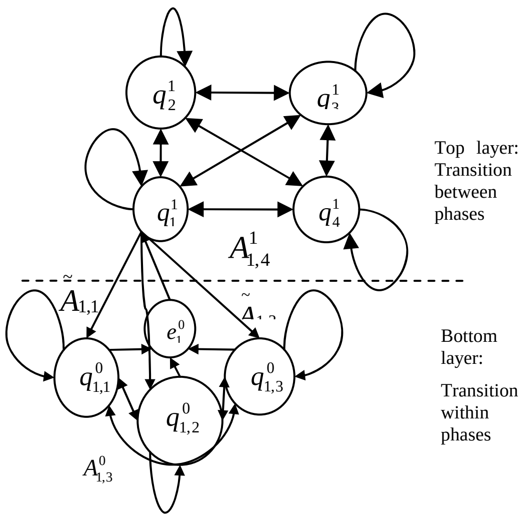


Figure 8: Hierarchical HMM for phase segmentation

3.3 Text Processor

The aim of text processor is to generate a list of entries, each representing an event in terms of: <Event start time, event end time, event type 1, event type 2, names of players involved>. In cases when the boundaries are not available, event time is used. For example, a *goal* scored by a *penalty* may be described as *<70:00, 72:00, goal, penalty, Koumas>*, and an *offside* may be described as *<40:50, offside, Desailly>*.

Although there are many sources of external knowledge as discussed in Section 2.2, this paper considers only the use of match reports and game logs to demonstrate how vastly different types of descriptions are supported by the text processor.

3.3.1 Processing of Match Reports

Match reports from the news sources tend to be short, descriptive in free text form, and cover only important events of interests to general readers. Thus the list of detected events is most probably incomplete, with missing attributes. This is either because accurate extraction is hard for free text, like player identification, or the information is not there at all, like the boundary information. An entry of event with partial information is acceptable, provided both time and event type are known.

When extracting event entries from match reports, keywords or synonyms of certain event types play a vital role. They signal where the event may occur, and what the event type is. Babaguchi et al. [19] described how indicative keywords found in closed caption text helped to detect the actual occurrence of certain events in American football. Because exact event type is required, we adopt rule-based information extraction techniques to locate the event and identify its type. This is done in four steps as follows:

1. We compile list of player names, synonyms of all event types and synonyms of all outcomes.
2. We detect terms denoting times which we call time entities.
3. For each time entity found, we analyze a window of terms centered around the time entity. If we detect a synonym of a certain event type within the term window, we consider an occurrence of event type has been found. An player's name found within the term window is considered as the performing player;
4. In case of ambiguity in detecting event type for certain events, such as the *save* and *shot-off-target* events, we develop additional rules to differentiate such events.

Sometimes, it may be ambiguous regarding the event type or player. This is usually caused by the appearance of multiple event types or players within the term window. In this case, a conflict resolution scheme is invoked, for example, to identify the closest event type or select the player in the attacking team as the correct one. Alternatively, the ambiguity in event type can be kept till video processor makes the decision.

3.3.2 Processing of Game Log

Game log (see Figure 5) provides a complete description of the flow of actions, with each action being represented as:

Action :=<time, player's name, action, outcome of the action>

It is thus reasonable to build events by merging appropriate actions. To extract individual events, there are two problems that need to be addressed: (a) where is the occurrence of the event type of interest; and (b) which surrounding actions should be included in the extracted event? As each action has an outcome, the action that has the event name as its outcome is likely to be the one where the event happens. Thus looking for terms corresponding to event type in the field of action outcome is accurate in locating events. This solves the first problem. The second problem is essentially a text segmentation task as we are looking for a segment of actions that is about the same event. Existing text segmentation systems [16], [19] are mainly statistical-based, designed to identify large body of coherent text as event unit. However, being a field-delimited text, game log contains too few words in each action unit to exhibit statistical patterns. This is also true for closed caption text for sports matches, which have a small number of words and incomplete grammar structures.

The basic method we use to detect and segment events from game log is by utilizing the event model. An event model describes how actions transit from one to another until it reaches the concluding action – usually with outcome of the event type. The words of event type signify the end of the action chain, and the system traces the chain backward until it reaches the first action in the event model, or when a *break* is found. The list of actions traversed is then merged into an event. In case a long action chain is found, an upper limit of traversal length is imposed, which is found by empirical studies of soccer matches. The problem with events derived from game log is that the event boundaries are usually approximate, partly because of misalignment in timing during the logging of actions by human operators.

3.4 Fusion of Video and Text Events

The output of video processor is a list of events, which we call video events, with boundaries in terms of video frames. The output of text processor is another list of events which we call text events. Video processor is accurate in the classification and segmentation of *break*, *plain play* or *attack* phases of the match. This provides the basis to align each pair of corresponding video and text events at the global game level. The text events extracted from match reports are accurate, though incomplete. They include the *goal, penalty goal/miss, save, free/corner kick, red/yellow card*, and *substitution*. Most such events only have the (approximate) time that the event occurs but not the boundaries. The events extracted from game log are assumed to be accurate and complete. However, the game log events do not have accurate boundaries.

An event binder is designed to merge video events with text events derived from match reports and/or game log. The merging process uses the phases of *break - plain play - attach* derived from the video processor as the guide, and is carried out as follows:

a) As depicted in Figure 1, certain events only occur in certain phases. Thus if a text event found falls into a wrong phase, such as a *save* falls into a *plain play* phase, it alerts the event binder to perform re-alignment. The problem is usually caused by large and varying time lag in the game log records entered by human operators.

b) When a text event is found in the correct phase but does not have complete boundary information such as what happens in most match report events, the binder finds the closest video event of the right type, if any, and uses that to fix the boundaries. However, if no video event can be found, the video processor will be invoked to look for such event in the temporal range suggested by the text event.

c) For text events derived from game log that have only approximate boundaries, the binder finds the corresponding video events and aligns the two segments of events at a common reference points. The binder essentially crops the video event to text event boundary, and rounds the boundaries to the nearest video unit boundaries. This is because while text event boundary may be semantically correct, they may be slightly off because of time alignment errors. On the other hand, video event boundaries are more appropriate for viewing purpose.

d) Some text events may be missed by video processor. This is often caused by ambiguous results of classifiers, such as *free kick* vs. *corner kick* and *attempt-on-goal* vs. *ordinary true attack* etc. The event binder then invokes the video processor to perform re-analysis within the temporal range suggested by the text event. The temporal range for the re-analysis is set to [-1 minute, +1 minute] around the reported time in the case of match report events, and [starting boundary - 30 seconds, ending boundary + 30 seconds] in the case of game log events. If an event of the desired type is found, then the event is recovered. If the desired event type has no distinct AV characteristics and cannot be detected by the video processor, such as the *substitution* event, then the whole segment of the relevant phase, such as the whole *break* phase, will be assigned to the event type. In this way, the more accurate text events are used to help disambiguate the results of AV classifiers to detect events beyond their capabilities.

e) After time alignment and re-analysis of AV contents by the video processor, if a video event is found that is not part of the game log text events, it is discarded. If, however, a video event is found but is not part of the match report event, it is kept.

4. Testing and Results

We used two EPL matches as training data: Southampton vs. Charlton (December 7, 2003), and Manchester United vs. Manchester City (December 13, 2003); and three EPL matches as testing data: Chelsea vs. Newcastle United (November 9, 2003), Manchester United vs. Arsenal (September 21, 03), Wolves vs. Chelsea (September 20, 03). Each match is about 95 minutes in duration. For each test match, we downloaded 5-7 match reports from bbc.co.uk, skyports.com, and www.premierleague.com. The game log is downloaded from www.soccernet.com. Table 3 summarizes the statistics of the training and testing data.

Table 3 Basic information of experiment data

	Training data	**Testing data**
Number of matches	2	3
Total duration	190 minutes	265 minutes
Number of goals	9	10
Number of attempt-on-goals	37	46
Number of penalty goals	0	1
Number of penalty misses	0	0
Number of corner kicks	10	13
Number of free kicks	7	11
Number of offsides	10	24
Number of substitutions	4	10
Number of yellow cards	2	8
Number of red cards	0	1

We apply heuristics to manually determine the ground truth on boundaries of events. For *goal, save* and *shot-off-target* events, the boundaries start from the frame when the view enters the front field, and end with the hard cut before the celebration or replay scenes. For *penalty, free kick* and *corner kick* events, the boundaries are defined to be the start of the camera motion sequence immediately after the medium shot, and end with the hard cut (like the *goal*) or the frame when the view leaves the front field (like a futile attack). For *red/yellow card and substitution* events, the boundaries are defined to be the start and end of the scene which depicts a card being signaled, or the substitution going on.

We perform tests based on three event detection configurations:

- Configuration (a): use only internal AV features.
- Configuration (b): fusion of internal AV features and match reports.
- Configuration (c): fusion of internal AV features with game log.

Each configuration has different capabilities of detecting the full range of events. Configuration (a) does not support the detection of *offside, substitution, yellow-card*, and *red-card* events. The *save* and *shot-off-target* events are measured collectively for configurations (a) and (b) as they do not support differentiation of *save* and *shot-off-target*.

The accuracy of event detection is measured based on the correctness of event types as well as the boundaries. The boundary is considered to be correct if the start and end of the boundary is within 2 seconds (or 50 frames) of the ground truth. The results of the tests are presented in Table 4.

From the Table, we can draw the following observations:

- The precision and recall of the *attempt-on-goal* (*saves* and *shot-off-targets*) for configurations (a) and (b) are poor. This is because the algorithm we used to detect excitement portions in commentator's speech using short time energy is naïve and ineffective. If we were to adopt more features and using the more sophisticated algorithms [3], we expect to have significant improvement In the results. Also, for configuration (a), not all *goal* events are detected because the supporting videotext detection is not very robust; as some videotexts for *goal* are mixed up with those of other types, such as the *substitution*.
- The precision and recall of some event types for configuration (c) are not 100%, although game log is supposed to be accurate and complete in recording all actions. This is because: (a) some records are missed from the original game log; and (b) there are large and varying off-sets in timings of certain recorded actions from the actual times.
- Overall, Configuration (c) performs the best with over 90% accuracy in both recall and precision for all event types. Configuration (b) has slightly better performance than Configuration (a). In addition, through the use of match reports, Configuration (b) is able to detect some *yellow/red card* and *substitution* events, and achieves 100% accuracy in detecting *goal* events.
- We note that Configurations (a) and (b) could achieve over 83% in boundary correctness based on the sub-set of events tested. On the other hand, Configuration (c) could achieve only 76% in event boundary detection among all the correctly detected events of appropriate types. One reason for this is that the event boundary ground truth for those additional events *(substitution, yellow/red card*) included in Configuration (c) are quite subjective and not precise.

The experimental results confirm our conjecture that with detailed external knowledge (game log), we could improve the accuracy of event detection. Further, it also has added advantage of being able to handle the full range of events. On the other hand, partial external knowledge from match reports does not improve the precision or recall of the overall results significantly. This is because only very few events that are missed by the video processor can be accurately extracted from match reports. The match report, however, improves the quality of the results in terms of being able to detect all major events precisely, such as *goal* and *yellow/red card*.

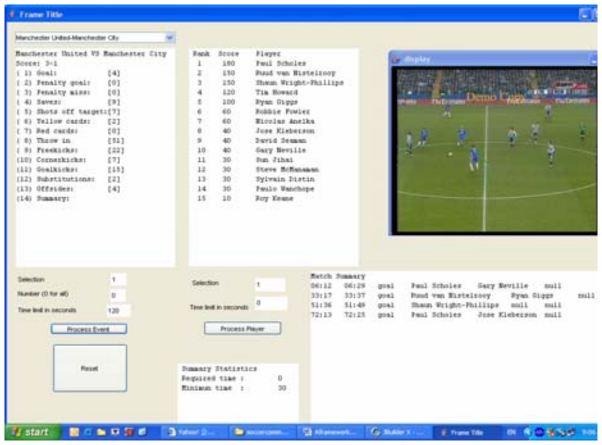
Based on the events detected and semantic units, the system supports versatile question-answering. Table 5 gives some example questions that require different semantics and with time constraints. Such questions can be answered by our system. Details of our question-answering evaluations are beyond the scope of this paper. Figure 9 shows the system interface.

Table 4 Results of event detection and boundary identification

	(a)		(b)		(c)	
	Recall	Precision	Recall	Precision	Recall	Precision
Goal	7/10	7/10	10/10	10/10	10/10	10/10
Save	24/46	24/47	25/46	25/48	19/21	19/20
Shot-off-target					23/25	23/25
Penalty goal	1/1	1/1	1/1	1/1	1/1	1/1
Penalty Miss	-	-	-	-	-	-
Corner kick	8/13	8/11	9/13	9/12	12/13	12/13
Free kick	7/11	7/12	7/11	7/12	10/11	10/11
Offside	0/24	-	0/24	-	22/24	22/22
Substitution	0/10	-	8/10	8/8	9/10	9/10
Yellow card	0/8	-	5/8	5/5	7/8	7/7
Red card	0/1	-	1/1	1/1	1/1	1/1
Boundary correctness	82.4%		83.3%		76.5%	

Table 5 Example questions

"Show me all the goals."
"Show me all the saves with assists."
"Show me all the shot-off-targets by Mutu."
"Show me all assists."
"Show me all actions Mutu performs."
"Show me all assists by Mutu."
"Show me all assists by Mutu in shot-off-targets."
"Show me Mutu's best 5 actions."
"Show me all the shot-off-targets within 120 seconds."



```
Manchester United VS Manchester City
Score: 3-1
( 1) Goal:              [4]
( 2) Penalty goal:      [0]
( 3) Penalty miss:      [0]
( 4) Saves:             [9]
( 5) Shots off target:[7]
( 6) Yellow cards:      [2]
( 7) Red cards:         [0]
( 8) Throw in           [51]
( 9) Freekicks:         [22]
(10) Cornerkicks:       [7]
(11) Goalkicks:         [15]
(12) Substitutions:     [2]
(13) Offsides:          [4]
(14) Summary:
```

```
Rank  Score   Player
 1     180    Paul Scholes
 2     150    Ruud van Nistelrooy
 3     150    Shaun Wright-Phillips
 4     120    Tim Howard
 5     100    Ryan Giggs
 6     60     Robbie Fowler
 7     60     Nicolas Anelka
 8     40     Jose Kleberson
 9     40     David Seaman
 10    40     Gary Neville
 11    30     Sun Jihai
 12    30     Steve McManaman
 13    30     Sylvain Distin
 14    30     Paulo Wanchope
 15    10     Roy Keane
```

Selection 1
Number (0 for all) 0
Time limit in seconds 120

Selection 1
Time limit in seconds 0

```
Match Summary
06:12   06:29   goal   Paul Scholes    Gary Neville   null
33:17   33:37   goal   Ruud van Nistelrooy    Ryan Giggs    null
51:36   51:49   goal   Shaun Wright-Phillips  null   null
72:13   72:25   goal   Paul Scholes    Jose Kleberson  null
```

```
Summary Statistics
Required time :      0
Minimum time  :      30
```

Figure 9 System interface

5. Discussion and Conclusion

This paper presented a scalable framework that fuses internal AV features and multiple external knowledge sources to detect events and identify their boundaries in broadcast team sports video. Tests of three full length matches in EPL soccer demonstrate that with the use of external knowledge sources, the framework improves its event detection accuracy and quality. In addition, it also supports question answering based on semantic units.

This research is only beginning. We will carry further research in the following directions. First, we will refine the techniques for the fusion of events from AV and external knowledge sources. Second, we will investigate better techniques to assign importance to events based on different aspects of the match. Third, we will extend the framework to other broadcast team sports.

6. REFERENCES

[1] G. Xu, Y.-F. Ma, H.-J. Zhang, and S. Yang, "A HMM Based Semantic Analysis Framework for Sports Game Event Detection", In Proc. of IEEE International Conference on Image Processing (ICIP), 2003

[2] P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and System for Segmentation and Structure Analysis in Soccer Video", In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2001

[3] L.-Y. Duan, M. Xu, T.-S. Chua, Q. Tian, and C.-S. Xu, "A Mid-Level Representation Framework for Semantic Sports Video Analysis", In Proc. of ACM Multimedia, 2003

[4] B. Li and M. I. Sezan, "Event Detection and Summarization in Sports Video", In Proc. of IEEE workshop on Content-Based Access of Image and Video Libraries, 2001

[5] M. Han, W. Hua, W. Xu, and Y. Gong, "An Integrated Baseball Digest System Using Maximum Entropy Method", In Proc. of ACM Multimedia, 2002

[6] H. Pan, B. Li, and M. I. Sezan, "Automatic Detection of Replay Segments in Broadcast Sports Programs by Detection of Logos in Scene Transitions", In Proc. of International Conference on Acoustic Speech and Signal Processing (ICASSP), 2002

[7] D. Zhang, and S.-F. Chang, "Event Detection in Baseball Video Using Superimposed Caption Recognition", In Proc. of ACM Multimedia, 2002

[8] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. W. Wan, "Tracjectory-Based Ball Detection and Tracking with Applications to Semantic Analysis of Broadcast Soccer Video", In Proc. of ACM Multimedia, 2003

[9] Y. Ariki, M. Kumano, and K. Tsukada, "Highlights Scene Extraction in Real Time from Baseball Live Video", In Proc. of 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR), 2003

[10] N. Babaguchi, and N. Nitta, "Intermodal Collaboration: A Strategy for Semantic Content Analysis for Broadcasted Sports Video", In Proc. of IEEE International Conference on Image Processing, 2003

[11] M. Bertini, A. Del Bimbo, and W. Nunziati, "Model Checking for Detection of Sport Highlights", In Proc. of 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, 2003

[12] D. Sadlier, S. Marlow, N. O'connor, and N. Murphy, "MPEG Audio Bitstream Processing Towards the Automatic Generation of Sports Programme Summaries", In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2002

[13] W. Zhou, A. Vellaikal, C.-C. Jay Kuo, "Rule-based Video Classification System for Basketball Video Indexing", In Proc. of ACM Multimedia, 2000

[14] Delte J. Assfalg, M. Bertini, C. Colombo, A. D. Bimbo, and W. Nunziati, "Semantic Annotation of Soccer Videos: Automatic Highlights Identification", In Proc. of Intenational Conference on Computer Vision and Image Understanding, 2003

[15] J. Assfalg, M. Bertini, A. D. Bimbo, W. Nunziati, and P. Pala, "Soccer Highlights Detection and Recognition Using HMMs", In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2002

[16] N. Nitta, N. Babaguchi, and T. Kitahashi, "Story Based Representation for Broadcasted Sports Video and Automatic Story Segmentation", In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2002

[17] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Structure Analysis of Soccer Video with Hidden Markov Models", In Proc. of International Conference on Acoustic Speech and Signal Processing (ICASSP), 2002

[18] D. Zhong and S.-F. Chang, "Structure Analysis of Sports Video Using Domain Models", In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2001

[19] N. Babaguchi, “Towards Abstracting Sports Video by Highlights”, In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2000

[20] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, “Unsupervised Discovery of Multilevel Statistical Video Structures Using Hierarchical Hidden Markov Models”, In Proc. of IEEE International Conference on Multimedia and Expo (ICME), 2003

[21] H. Yang, L. Chaisorn, Y. Zhao, S.-Y. Neo, T.-S. Chua, “VideoQA: Question Answering on News Video”, In Proc. of ACM Multimedia, 2003

[22] H. Zhang, C. Y. Low, and S. W. Smoliar, “Video Parsing and Browsing Using Compressed Data”, Multimedia Tools and Applications, No. 1, pp. 89-111, 1995

[23] M. Galley, K. R. McKeown, E. Fosler-Lussier and H. Jing, “Discourse Segmentation of Multi-Party Conversation”, In Proc. of 41st Annual Meeting of the Association for Computational Linguistics (ACL), 2003

[24] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, “Learning Hierarchical Hidden Markov Models for Video Structure Discovery”, Technical Report, 2002-006, ADVENT Group, Columbia University, 2002.