Live Sports Event Detection Based on Broadcast Video and Web-casting Text

Changsheng Xu, Jinjun Wang, Kongwah Wan, Yiqun Li and Lingyu Duan

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
{xucs, stuwj2, kongwah, yqli, lingyu }@i2r.a-star.edu.sg

ABSTRACT

Event detection is essential for sports video summarization, indexing and retrieval and extensive research efforts have been devoted to this area. However, the previous approaches are heavily relying on video content itself and require the whole video content for event detection. Due to the semantic gap between low-level features and high-level events, it is difficult to come up with a generic framework to achieve a high accuracy of event detection. In addition, the dynamic structures from different sports domains further complicate the analysis and impede the implementation of live event detection systems. In this paper, we present a novel approach for event detection from the live sports game using web-casting text and broadcast video. Web-casting text is a text broadcast source for sports game and can be live captured from the web. Incorporating web-casting text into sports video analysis significantly improves the event detection accuracy. Compared with previous approaches, the proposed approach is able to: (1) detect live event only based on the partial content captured from the web and TV; (2) extract detailed event semantics and detect exact event boundary, which are very difficult or impossible to be handled by previous approaches; and (3) create personalized summary related to certain event, player or team according to user's preference. We present the framework of our approach and details of text analysis, video analysis and text/video alignment. We conducted experiments on both live games and recorded games. The results are encouraging and comparable to the manually detected events. We also give scenarios to illustrate how to apply the proposed solution to professional and consumer services.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *abstract methods, indexing methods.*

General Terms

Algorithms, Measurement, Performance, Experimentation.

Keywords

Event Detection, Broadcast Video, Web-casting Text.

MM'06, October 23–27, 2006, Santa Barbara, California, USA.

1. INTRODUCTION

With the proliferation of sports content broadcasting, sports fans often find themselves not being able to watch live games due to reasons such as region and time difference. Usually, only a small portion in a sports game is exciting and highlight-worthy for most audience. Therefore, the ability to access (especially in a live manner) events/highlights from lengthy and voluminous sports video programs and to skip less interesting parts of the videos is of great value and highly demanded by the audience. However, current event/highlight generation from sports video is very labor-intensive and inflexible. One limitation is that the events/highlights are determined and generated manually by studio professionals in a traditional one-to-many broadcast mode, which may not meet some audience's appetites who may only be interested in the events related to certain player or team. Another limitation is that these events/highlights are usually only accessible to the audience during the section breaks (e.g. the half-time in a soccer game). Clearly, with the advent of mobile devices and need for instant gratification, it would be helpful for sports fans who are unable to watch the live broadcast, to nonetheless be kept updated on the live proceedings of the game, through personalized events/highlights. Therefore, the availability of automatic tools to **live** detect and generate personalized events from broadcast sports videos and **live** send the generated events to users' mobile devices will not only improve the production efficiency for the broadcast professionals but also provide better game viewer-ship for the sports fans. This trend necessitates the development of automatic event detection from live sports games.

In this paper, we present a generic framework and methodology to automatically detect events from live sports videos. In particular, we use soccer video as our initial target because it is not only a globally popular sport but also presents many challenges for video analysis due to its loose and dynamic structure compared with other sports such as tennis. Our framework is generic and can be extended to other sports domains. We also discuss scenarios how to deploy the proposed solution into various devices.

1.1 Related Work

Extensive research efforts have been devoted to sports video event detection in recent years. The existing approaches can be classified into event detection based on video content only and event detection based on external sources.

1.1.1 Event detection based on video content only

Most of the previous work of event detection in sports video is based on audio/visual/textual features directly extracted from video

content itself. The basic idea of these approaches is to use low-level or mid-level audio/visual/textual features and rule-based or statistical learning algorithms to detect events in sports video. These approaches can be further classified into single-modality based approach and multi-modality based approach. Single-modality based approaches only use single stream in sports video for event detection. For example, audio features were used for baseball highlight detection [1] and soccer event detection [2]; visual features were used for soccer event detection [3][4]; and textual features (caption text overlaid on the video) were utilized for event detection in baseball [5] and soccer [6] videos. The single modality based approaches have low computational load thus can achieve real-time performance in event detection [7][8], but the accuracy of event detection is low as broadcast sports video is the integration of different multi-modal information and only using single modality is not able to fully characterize the events in sports video. In order to improve the robustness of event detection, multi-modality based approaches were utilized for sports video analysis. For example, audio/visual features were utilized for event detection in tennis [9], soccer [10] and basketball [11]; and audio/visual/textual features were utilized for event detection in baseball [12], basketball [13], and soccer [14]. Compared with the single-modality based approaches, multi-modality based approaches are able to obtain better event detection accuracy but have high computational cost, thus it is difficult to achieve real-time performance.

Both single-modality based approaches and multi-modality based approaches are heavily replying on audio/visual/textual features directly extracted from the video itself. Due to the semantic gap between low-level features and high-level events as well as dynamic structures of different sports games, it is difficult to use these approaches to address following challenges: (1) ideal event detection accuracy (~100%); (2) extraction of the event semantics, e.g. who scores the goal and how the goal is scored for a "goal" event in soccer video; (3) detection of the exact event boundaries; (4) generation of personalized summary based on certain event, player or team; (5) a generic event detection framework for different sports games; and (6) robust performance with the increase of the test dataset and live videos. In order to address these challenges, we have to seek available external sources to help.

1.1.2 Event detection based on external sources

Currently there are two external sources that can be used for sports video analysis: closed caption and web. Both are text sources. Incorporation of text into sports video analysis is able to help bridge the semantic gap between low-level features and high-level events and thus facilitate the sports video semantic analysis.

Closed caption is a manually tagged transcript from speech to text and is encoded into video signals. It can be used separately to identify semantic event segments in sports video [15] or combined with other features (audio/visual) for sports video semantic analysis [16][17]. Since closed caption is a direct transcript from speech to text, it contains a lot of information irrelevant to the games and lacks of a well-defined structure. On the other hand, currently closed caption is only available for certain sports videos and in certain countries.

In addition to closed caption, some researchers attempt to use information in the web to assist sports video analysis. Xu *et.al.* [18][19] proposed an approach to utilize match report and game log obtained from web to assist event detection in soccer video. They still used audio/visual features extracted from the video itself to detect some events (e.g. goal), while for the events which are very difficult or impossible to be detected using audio/visual features, they used text from match report and game log to detect such events. The text events and video events were fused based on sports video structure using rule-based, aggregation and Bayesian inference schemes. Since some events were still detected using audio/visual features, the accuracy is much lower than event detection using text and the proposed event detection model is also difficult to be applied to other sports domains. On the other hand, the proposed framework has to structure the whole video into phases (Break, Draw, Attack) before event detection, hence it is not able to achieve live event detection.

1.2 Our Contribution

In this paper, we present a novel approach for semantic event detection from live sports game based on analysis and alignment of web-casting text and broadcast sports video. Compared with previous approaches, the contributions of our approach include:

(1) We propose a generic framework by combining analysis and alignment of web-casting text and broadcast sports video for event detection from live sports games. Particularly, the incorporation of web-casting text significantly improves the event detection accuracy and helps extracting event semantics.

(2) We propose novel approaches for game start detection and game time recognition from live broadcast sports video, which enables the exact matching between the time tag of an event in web-casting text and the event moment in the broadcast video. The previous methods [18][19] assumed that the time tag of an event in web-casting text corresponded to the video time, which is not true in the broadcast video and will cause bias for text/video alignment.

(3) We propose a robust approach based on finite state machine to align the web-casting text and broadcast sports video to detect the exact event boundaries in the video.

(4) We propose several scenarios to illustrate how to deploy the proposed solution into current professional services and consumer services.

The rest of the paper is organized as follows. The framework of the proposed approach is described in Section 2. The technical details of text analysis, video analysis and text/video alignment are presented in Section 3, 4, and 5 respectively. Experimental results are reported in Section 6. The potential applications of the proposed solution are discussed in Section 7. We conclude the paper with future work in Section 8.

2. FRAMEWORK

The framework of our proposed approach is illustrated in Figure 1. The framework contains four live modules: live text/video capturing, live text analysis, live video analysis, and live text/video alignment. The live text/video capturing module captures the web-casting text from the web and the broadcast video from TV. Then for the captured text and video, the live text analysis module detects the text event and formulates the detected event with proper semantics; the live video analysis module detects the game start point and game time by recognizing the clock digit overlaid on the video. Based on detected text event and recognized game time, the

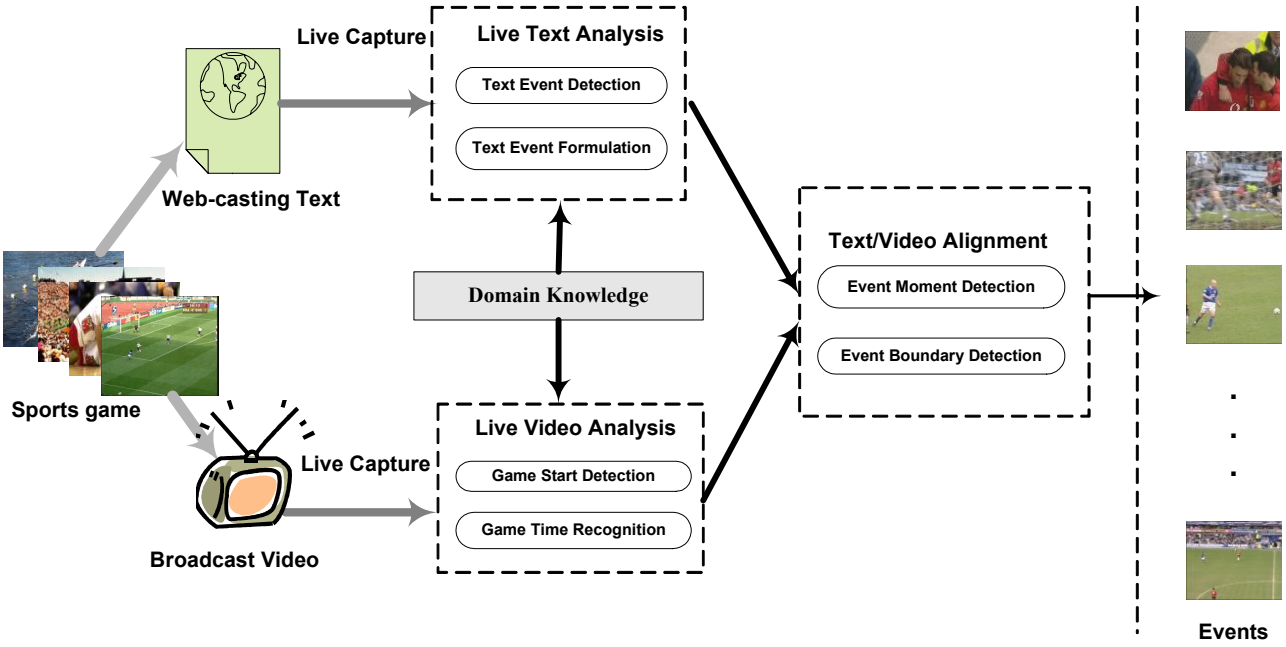

Figure 1. Framework of proposed approach

live text/video alignment module detects the event with exact boundaries in the video. This can be done by defining a video segment that contains the event moment, structuring the video segment, and detecting the start and end boundary of the event in the video. The detected video events with text semantics can be sent to different devices based on users' preferences. The proposed framework is generic and can be used for different sports domains. The technical detail of each module in the framework will be described in the following sections.

3. TEXT ANALYSIS

There are two external text sources that can be used to assist sports video analysis: closed caption (Figure 2) and web-casting text (Figure 3 and Figure 4). Compared with the closed caption which is a transcript from speech to text and only available for certain sports games and in certain countries, the content of web-casting text is more focused on events of sports games with a well-defined structure and is available in many sports websites [20,21]. Therefore we choose web-casting text as an external source in our approach. Since an event in the web-casting text contains information such as time of the event moment, the development of the event, players and team involved in the event, etc., which is very difficult to be obtained directly from the broadcast video using previous approaches, it will greatly help event and event semantics detection from a live sports game. The live capturing and analysis of web-casting text is discussed in the following subsections.

```
049095 HE MAKES A GOOD QUICK MOVE TO
049143 THE POST, RUNS RIGHT BY HIM.
049193 BRETT FAVRE THROWS A PERFECT
049245 PASS.
049255 THAT WAS MAXIMUM PASS
049318 PROTECTION.
```

Figure 2. Closed caption [16]

Figure 3. Web-casting text (Flash) [21]

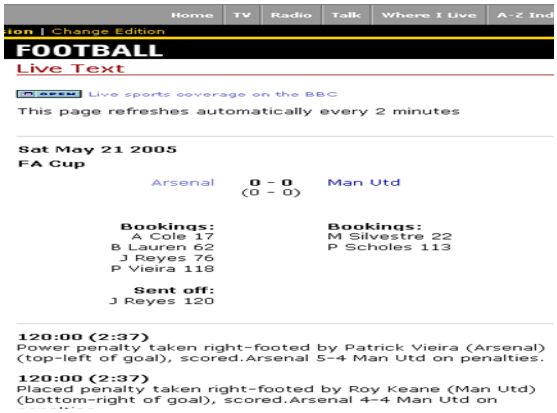

Figure 4. Web-casting text (HTML) [20]

3.1 Text Capturing

The web-casting text [20,21] serves as a text broadcasting for live sports games. The live text describes the event happened in a game with a time stamp and is updated every a few minutes. The first

step for text analysis is to live capture the text from the web in either HTML [20] or flash [21] format. Figure 5 illustrates the process of text capturing and extraction, which can be summarized as following steps: (1) our program keeps sending request to the website server regularly to get the HTML/flash file regularly; (2) the text describing the game event is extracted using rule-based keyword matching method; and (3) the program checks for the difference or update of the text event between the current file and the previous one, extracts the new text event and adds it to the event database.

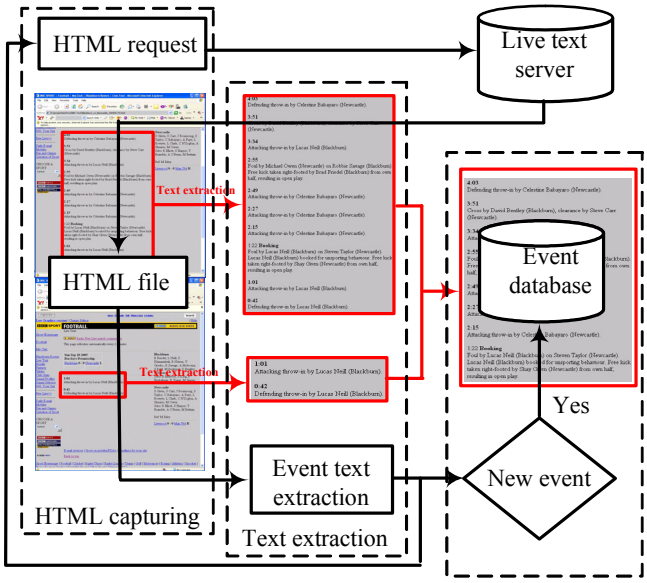

Figure 5. Text capturing and event extraction

3.2 Text Event Detection

The types of the event are different for different sports games, but the number of the event types for each sports game is limited. Therefore, in order to detect the events for certain sports game, we need to construct a database that contains all the event types for that sports game. For example, in soccer game, we select the event types listed in Table 1 for event detection. This may not cover all the event types in a soccer game, but the database is extensible and the selected events are interesting to most soccer fans.

To detect these events from the captured web-casting text, we observed from our database that each type of the sports event features one or several unique nouns, which are defined as keywords related to this event. This is because the web-casting text is tagged by sports professionals and has fixed structures. Hence by detecting these keywords, the relevant event can be recognized. We have also observed that some keywords may correspond to different events, for example, goal and goal kick are two different events if goal is defined as a keyword. In our approach, we also conduct context analysis before or after the keywords to eliminate the false alarms of event detection. Usually, the sources that provide the web-casting text can be classified into two groups: one with well-defined syntax structure [20] and the other with freestyle text [24]. Our approach uses the well-structured web-casting text, here we also present the freestyle web-casting text for comparison. To achieve accurate text event detection performance, we give different sets of keyword definitions for the well-structured web-casting text (Table 1) and the freestyle web-casting text (Table 2). Such definition is extendable.

Table 1 Keyword definition for well-structured web text.

Event	Keyword	Event	Keyword
Goal	goal, scored	Red card	dismissed, sent off
Shot	shot, header	Yellow card	booked, booking
Save	save, blocked	Foul	foul,
Offside	offside	Free kick	free kick, free-kick
Corner	corner kick	Sub.	substitution, replaced

Table 2 Keyword definition for freestyle web text

Event	Keyword
Goal	g-o-a-l or scores or goal or equalize – kick
Card	“yellow card" or “red card" or “yellowcard" or “redcard" or “yellow-card" or “red-card"
Foul	(commits or by or booked or ruled or yellow) w/5 foul
Offside	(flag or adjudge or rule) w/4 (offside or “off side" or “off-side")
Save	(make or produce or bring or dash or “pull off") w/5 save
Injury	injury and not “injury time" It is not included in the defined keywords.
Free kick	(take or save or concede or deliver or fire or curl) w/6 (“free-kick" or “free kick" or freekick)
Sub.	substitution

Once proper keywords are defined, an event can be detected by finding sentences that contain the relevant keyword and analyzing context information before and after the keyword. A simple keyword based text search technique is enough for our task.

For a detected text event, we record the type of event and players/team involved in the event for personalized summarization. The time stamp of each detected event is also logged which is used by our text/video alignment module for video event boundary detection as discussed in subsection 5.2.

4. VIDEO ANALYSIS

From the detected text event, we can obtain the time stamp which indicates when the event occurs in the game. To detect the same event in the video, we have to know the event moment in the video. An intuitive way is to directly link the time stamp in the text event to the video time, but this is not true for live broadcast video where the game start time is different from the video start (broadcasting) time. Therefore, in order to detect the event moment in the video, we should know the start point of the game in the video. We propose to combine two approaches to detect the game time: game start detection and game time recognition from the digital clock overlaid on the video. Game start detection is to detect the start point of the game in the video and use it as a reference point to infer the game time. Game time recognition is to detect the digital clock overlaid on the video and recognize the time from the video clock. The reason why we combine these two schemes instead of just using one of them is as follows: (1) game start detection is suitable for the games without clock stopping like soccer but cannot work for those games with clock stopping during the game like basketball, while game time recognition can work for both kind of games; (2) sometimes the appearance of the digital clock in the video (especially for soccer) delays for 20 seconds to a minute after the game is started, thus game start detection will help identify those events occurred before the digital clock appears; and (3) both schemes can verify each other to improve the accuracy. Therefore,

the combination of the complementary strength of both approaches is able to come up with a generic and robust framework for game time detection for various sports videos.

4.1 Game Start Detection

The Game Start Detection (GSD) is to detect the physical frame number which indexes the start of the sports game in a broadcast video. GSD is a crucial task for sports event detection systems because it links the time-stamp of a text event (e.g. 10:30 Goal by …) to an exact frame (e.g. frame 6214). For a live system, GSD can also help to reduce the computational cost by suppressing the event detection processing before the game start.

To detect the game start in a soccer game, we first extract the view-type and camera pan motion features of each video frame to obtain a feature sequence. Then we apply a finite state machine [22-23] to model the transition patterns of the sequence to find a frame that indexes to game start. Our method to extract the frame view-type and camera pan motion is elaborated as follows:

- The frame view type

$$V_f \in \{far-view, non-far-view\},\ f=1,...,N \quad (1)$$

where N is the total number of frames in the video segment. To compute V_f, we first identify the region of the soccer field by detecting the green area and then detect the maximal size of non-green blobs inside the green area. If the detected green area takes more than 33% of the whole frame, and the maximal blob size is smaller than 0.1% of the frame size, then V_f="far-view". Otherwise V_f="non-far-view". The thresholds are empirically set and validated robustly by our experiments in section 6.3.

- The camera pan motion is denoted as P_f, f=1,…, N. P_f is computed using the motion vector field from MPEG I/II video and the algorithm in [25]. To improve the accuracy, the motion information is only extracted from P frames, the camera pan motion factor for I and B frames is set to the last computed P frame value.

The extracted frame view-type and camera pan motion features are synchronized to generate an R^2 vector sequence $\{V_f, P_f\}$, f=1,…, N, which is sent to a finite state machine (FSM) to detect the game start.

FSM has been proved to be robust in modeling temporal transition patterns. For example, Bimbo *et. al.* [22] used FSM to model playfield zone change to detect soccer highlights. Leonardi *et. al.* [23] applied the FSM to exploit the sequencing in time of the low-level visual descriptors to detect semantic events.

The structure of our FSM for game start detection is illustrated in Figure 6. The FSM starts from f=1 and jumps to other states if the transition conditions illustrated in Figure 6 are met. We give an example of how the FSM operates to find the game start frame in Figure 7.

GSD is only used to help detect the event moment from those games where the clock has not appeared when an event occurs. Base on our observation, this is common for broadcast soccer video, while for other sports video such as basketball the clock always appears with the game start. Therefore, our GSD is only used for soccer video, while for other video we will directly recognize the game time from the clock overlaid on the video. Even for soccer video, if we can recognize the game time from the clock later on, we will use the recognized time to synchronize the game time. This is because game time recognition is more accurate and reliable compared with GSD. Game time recognition is described in the following subsection.

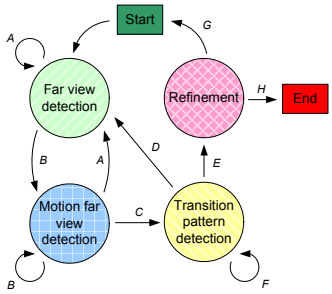

Figure 6. Finite state machine for game start detection

Transition condition A*: a far view frame is detected;* B*: a non far view frame is detected before condition C;* C*: a far view frame with high camera pan is detected;* D*: undesired frame transition pattern is detected;* E*: the desired far-view->non-far view-> far-view pattern is detected;* F*: restart to detect the start of the second half;* G*: required detection is done.*

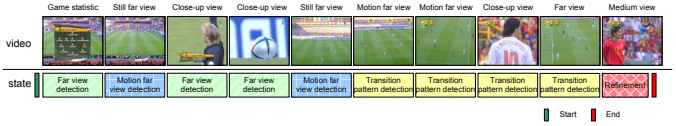

Figure 7. Temporal pattern modeling for game start detection

4.2 Game Time Recognition

In many sports games such as soccer and basketball, a video clock is used to indicate the game lapsed time. Since the time stamp in the text event is associated with the game time, knowing the clock time will help us to locate the event moment in the video. Referring to the event moment, the event boundary can be detected.

We propose a novel approach to read the video clock in real time by recognizing the clock digits using a few techniques related to the transition patterns of the clock. The most critical feature we use to locate the clock digits is a Temporal Neighboring Pattern Similarity (TNPS) measure. For the clock digits, their TNPS changes periodically. This is a very unique feature of the clock digits. With this feature, the clock digits are located without recognizing all the text characters overlaid together with the clock on the video, and the sample digit patterns are automatically extracted and labeled in real time. The clock digits are recognized while the video is playing back without any offline training. Compared with the traditional methods such as OCR, our approach is able to achieve the real time performance and the result is more reliable due to the reliable location of the clock digits in real time during the video broadcasting.

4.2.1 Clock digits location

The digital clock is overlaid on the video with other texts such as the team names and the scores as shown in Figure 8. Our algorithm first locates the static overlaid region by static region detection. The region of interest (ROI) for characters is then detected using Connected Component Analysis. A normal method to locate the clock digits is to recognize all the texts and then look for the pattern of "xx:xx", which x is a numeric character. Such a method would be complicated because we have to train the algorithm to recognize all the alphanumeric characters. Another issue we must take note is that the text characters may not be in the same font, size and color

with the numeric characters on the clock. Obviously we cannot expect good reliability and accuracy by using such a method. In our approach, we first locate the clock digits so that only numeric characters on the video clock are required to be recognized. These numeric characters are uniform in font, size and color. Hence, the recognition accuracy is improved and the whole recognition process is also simplified.

Figure 8. Overlaid video clock

Since the clock digits are changing periodically, for each character ROI, we observe its TNPS sequence, which is defined as follows:

$$S(n) = \sum_{(x,y)\in I} B_{n-1}(x,y) \otimes B_n(x,y) \qquad (2)$$

where $B(x, y)$ is the binarized image pixel value in position (x, y), n is the frame sequence number, I is the character region, and $\otimes$ is XOR operation. $S(n)$ shows the pattern change. If the pattern change is regular, the character is considered as a clock digit character. E.g., if the pattern changes once per second, it should be the SECOND digit pattern. In this way, the SECOND digit position is located. The location of other digits can be located referring to the SECOND digit position.

4.2.2 Clock digits recognition

After the clock digits are located, we observe the TEN-SECOND digit pattern change using the TNPS. At the time when the pattern change happens, we extract the pattern of "0" from the SECOND digit ROI. At the next second we extract the pattern of "1". And next is the "2", "3", "4", and so on. Therefore, all the numeric digits from 0 to 9 are extracted automatically. Since the extracted digits may vary along time due to the low quality of the video, we may extract a few patterns for the same digit character. These sample digit patterns are used to recognize all the 4 digits on the clock. Some of these sample digit patterns are shown in Figure 9.

Figure 9. Sample digit patterns

After the templates for each digit character from "0" to "9" are collected, for every frame of decoded images, every clock digit is matched against the templates. The matching score of numeric character i is calculated as follows:

$$S(i) = \min_j \{ \sum_{(x,y)\in I} T_{ij}(x,y) \otimes D(x,y) \} \qquad (3)$$

$$i=0, 1, \ldots, 9, 10$$

where $T_{ij}(x, y)$ is the binarized image pixel value in position (x, y) for the j^{th} template of numeric character i, $D(x, y)$ is the binarized image pixel value in position (x, y) for the digit character to be recognized, and I is the ROI of the digit character. When $i=10$, $T_{10j}(x, y)$ is the template for a flat region without any character. The clock digits on every frame are recognized when a best match is found. The detail of game time recognition can be found in [26].

5. TEXT/VIDEO ALIGNMENT

After we know the game time, we can detect the event moment in the video by linking the time stamp in the text event to the game time in the video. However, an event should be a video segment which exhibits the whole process of the event (e.g. how the event is developed, players involved in the event, reaction of the players to the event, etc.) rather than just a moment. Therefore, in addition to event moment, we also need to detect the start and end boundaries of the event to formulate a complete event. In this section, we present a novel approach to live detect the event boundary using video structure analysis and finite state machine.

5.1 Feature Extraction

Based on the detected event moment in the video, we define a temporal range containing the event moment and detect event boundary within this range. This is due to the following considerations: (1) since we are dealing with live video, we are only able to analyze the video close to the event moment; and (2) event structure follows certain temporal patterns due to the production rules in sports game broadcasting. Thus, we first extract some generic features from the video and use these features to model the event structure. The feature extraction is conducted in real time.

5.1.1 Shot boundary detection

The video broadcasting of a sports game generally adheres to a set of studio production rules. For example, hard-cut shot transitions are used during play to depict the fast pace game action, while gradual transitions such as dissolves are used during game breaks or lull. Most broadcasters also use a flying logo wipes during game replays. The logo may be an emblem image of the tournament or the teams/clubs/national-flag. Game replays are important visual cues for significant game moments. Typically a few replay shots are shown in slow motion between the flying logo wipe transitions. The shot transitions between these slow motion shots are usually dissolves. This leads us to perform a rudimentary structure analysis of the video to detect these shot boundaries and their transition types. The basic idea is to locate the clusters of successive gradual transitions in the video as candidate segment boundaries for significant game moments.

Detecting hard-cut shot changes is relatively easier than detecting gradual shot changes. For hard-cut detection, we compute the mean absolute differences (MAD) of successive frame gray level pixels, and use an adaptive threshold to decide the frame boundaries of abrupt shot changes. To handle gradual shot change, we additionally compute multiple pair-wise MAD. Specifically, for each frame k, we calculate its pair-wise MAD with frame k-s, where s=7, 14, 20 is set empirically. Hence, we buffer about 1 second worth of video frames, and maintain 3 MAD profiles. Figure 10 shows an example of the MAD profiles. Shot changes are usually areas where all MAD values show significant changes (deltas) in the same direction. That is, they either increase or decrease simultaneously.

In spite of this, we still observe a fair amount of false positives. These usually occur during a close-up shot of a moving player amidst a complex background clutter. Other causes of false positives include foreground occlusion and fast camera movement. This can be reduced by applying rules to compare the detected shot with its adjacent shots.

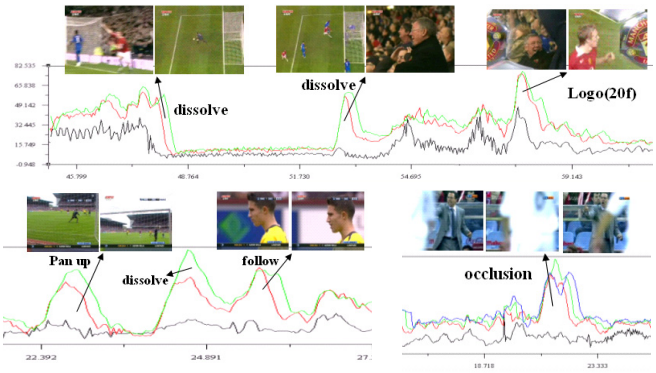

Figure 10. Simultaneous deltas in the MAD

5.1.2 Shot classification

With the obtained shot boundary, shot classification is conducted using a majority voting of frame view types (subsection 4.1) identified within a single shot. Since we have two frame view types, we can accordingly produce two types of shot, specifically the far-view shot and non-far-view shot. We also log the start and end boundary type of each shot, i.e. a hard cut boundary or a dissolve boundary, to generate an R^3 shot classification sequence S as

$$S=\{S_i\}=\{[sbt_i, st_i, ebt_i]^T\} \qquad i=1,\ldots,N \tag{4}$$

where the start boundary type $sbt_i \in$ {hard cut, dissolve}, the shot type $st_i \in$ {far-view, non-far-view}, and end boundary type $ebt_i \in$ {hard cut, dissolve}. N is the total number of shot in the sequence. Once the shot classification sequence S is generated, our system will proceed to text/video alignment to detect event boundary.

5.2 Event Boundary Detection

In a typical event detected from the web text source, the related time stamp (denoted as T_t) usually logs a most representative moment of the event (Figure 4). For example, in a goal-scoring event, the time stamp records the extract time instance when the ball goes across the bottom-line [20]. Starting from T_t, our event boundary detection module finds a suitable event boundary $[T_t\text{-}D_s, T_t\text{+}D_e]$ where the available scenes of the stated event in the original broadcast recording are all encapsulated. Here D_s and D_e indicate the time duration between event start/end boundary and event moment respectively.

To compute D_s and D_e, we observed from our database that the extracted S sequence features one of the two patterns for event boundary as illustrated in Figure 11. We have additionally observed the following rules:

Rule 1: Any far-view shot that is too long is not inside an event.

Rule 2: Most events last longer than 20 seconds.

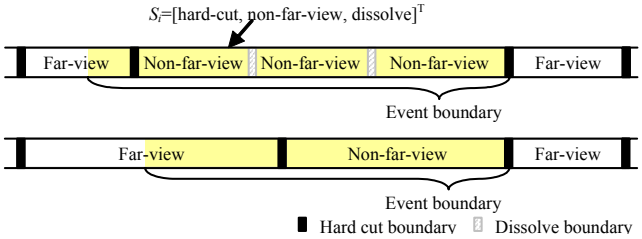

Figure 11. Event boundary pattern modeling

Hence our strategy to find D_s and D_e is as follows. When a text event is identified, our event boundary detection module first extracts a candidate segment from the video where the true event duration is included. In our current setup, the duration between the start of the candidate segment to T_t is empirically set to 1 minute, and the end of the candidate segment to T_t is set to 2 minutes. Then feature extraction (Subsection 5.1) is carried out to obtain the sequence S (Eq.4) from the candidate segment. Finally S is sent to another finite state machine (FSM) to compute D_s and D_e.

The FSM first detects the event start boundary then the event end boundary. To detect the start boundary, the FSM first identifies the shot in the S sequence (the "Start" state in Figure 12) into which the event time-stamp falls and names this shot as reference shot S_r. Starting from S_r, the FSM performs a backward search along $\{S_i\}_{i=r\text{-}1,\ldots,1}$ to find a suitable event start shot S_s. In this backward search, the FSM changes states with given conditions listed in Figure 12. The FSM sometimes jumps into the "Hypothesis" state if it cannot verify whether a far-view shot is inside the event boundary (e.g. it is a replay) or outside the event. In the "Hypothesis" state, the FSM assumes the far-view shot to be inside the event boundary and checks whether such an assumption violates any rules (e.g. it results in too long an event boundary). Note in Figure 11 that, as the event start boundary is not aligned with the start boundary of S_s, a "Start boundary refine" state is adopted to find a suitable starting frame as the exact event start boundary. This is achieved by thresholding the duration between the desired event start boundary frame to the ending frame of S_s. After the start boundary is detected, the FSM performs a forward search along $\{S_i\}_{i=s+1,\ldots,N}$ to find the event end boundary. The algorithm in the forward search is similar to the backward search except that it is working in a forward direction.

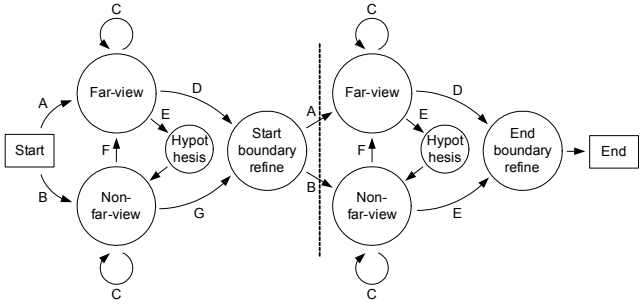

Figure 12. Finite state machine for event boundary detection

Transition condition A*: far- view shot;* B*: non-far-view shot; C shot type unchanged; D: Rule 1 satisfied or Hypothesis failed; E: A far-view shot but does not satisfy Rule 1; F: same as A; G: Hypothesis failed.*

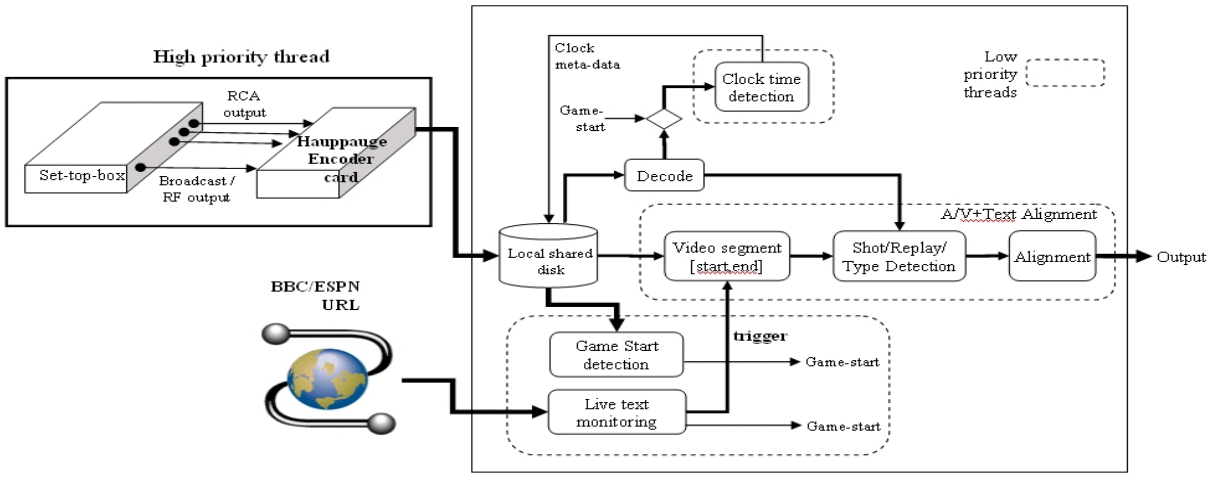

Figure 13. Schematic Diagram of our live set-up

6. EXPERIMENTAL RESULTS

We conducted our experiments on both live games and recorded games. The recorded games are used to evaluate individual modules and the live games are used to evaluate the whole system. The dataset, experimental setup and results are described and reported in the following subsections.

6.1 Text Analysis

We conducted text event detection experiment on 8 games (6 EPL and 2 UEFL). To give a comparison, we used two types of web text: well-structured web text [20] and freestyle web text [24]. The former presents a well-defined syntax structure which significantly facilities our keyword based text event detection method. The freestyle web text lacks of decent structure for event description. Due to its dynamic structure and diverse presenting style, freestyle web text is more difficult for event detection. Table 3 and Table 4 list the event detection performance from well-structured web text and freestyle web text respectively. The relatively lower precision/recall in Table 4 validates the advantage of using well-structured web text for text event detection.

6.2 Video Analysis

6.2.1 Game start detection

The performance of Game Start Detection (GSD) is tested using 8 EPL games, 2 UEFA games and 15 International friendship games. The starts (both the start of the first half and the second half) of 12 games are detected within 5 seconds, 6 games within 15 seconds, 5 games within 30 seconds, and 2 above 30 seconds. Some results are listed in column "GSD" of Table 5. It can be seen that some of the detected game starts are delayed due to two reasons: 1) The presence of captions which causes incorrect frame view type classification, and 2) The occurrence of early events which lead to the game pause and interferes the game start detection.

6.2.2 Game time recognition

The performance using our Game Time Recognition (GTR) for game start detection is tested using 8 EPL games, 4 UEFA 2005 games, 4 Euro-Cup 2004 games and 10 World-Cup 2002 games. For most game videos the clock digits can be located without any false location and the recognition accuracy after correct location of clock digits can achieve above 99%. Some detailed results are listed in column "GTR" of Table 5. The inaccurate recognition for 2 EPL games are due to the small game clock in MPEG I video recording which leads to incorrect clock digits location identification. It also can be seen that the result of GTR are more accurate and reliable than the result of GSD.

Table 3. Text event detection based on well-structured web text

Event	Precision/Recall	Event	Precision/Recall
Goal	100%/100%	Red card	100%/100%
Shot	97.1%/87.2%	Yellow card	100%/100%
Save	94.4%/100%	Foul	100%/100%
Free kick	100%/100%	Offside	100%/100%
Corner	100%/100%	Substitution	100%/100%

Table 4. Text event detection based on freestyle web text

Event	Precision/Recall	Event	Precision/Recall
Card	97.6%/95.2%	Free kick	96.7%/100%
Foul	96.9%/93.9%	Save	97.5%/79.6%
Goal	81.8%/93.1%	Injury	100%/100%
Offside	100%/93.1%	Substitution	86.7%/100%

6.3 Text/Video Alignment

To assess the suitableness of the automatically selected event boundary, we use the Boundary Detection Accuracy (BDA) [14] to measure the detected event boundary compared with the manually labeled boundary.

$$BDA = \frac{\tau_{db} \cap \tau_{mb}}{max(\tau_{db}, \tau_{mb})} \quad (5)$$

where τ_{db} and τ_{mb} are the automatically detected event boundary and the manually labeled event boundary, respectively. The higher the BDA score, the better the performance. Table 6 lists the BDA scores for 4 EPL games. It is observed that the boundary detection performance of free kick events is lower than other events. This is because our selected web casting text source usually includes other event (e.g. foul) before the free kick event, and hence the extracted time stamp is not accurate, which affects the alignment accuracy.

Table 5. GSD and GTR results on 8 games (6 EPL, 2 UEFA)

Game	Actual start	GSD	GTR
ManU-Sunderland	17:25	17:24	18:23
Portsmouth-Arsenal	8:05	8:50	8:05
Arsenal-WestBrom	1:55	2:35	1:55
Bolton-Chelsea	08:01	8:02	7:59
Aston-VillaBolton	07:38	7:48	07:38
Blackburn-Newcastle	03:26	03:29	03:26
Chelsea-BayernMunich	12:08	12:11	12:08
Liverpool-Chelsea	11:35	11:40	11:35

Table 6. Event boundary detection

Event	BDA	Event	BDA
Goal	90%	Red card	77.5%
Shot	86.9%	Yellow card	77.5%
Save	97.5%	Foul	77.7%
Free kick	43.3%	Offside	80%
Corner	40%	Substitution	75%

6.4 Live Performance

6.4.1 Live setup

Figure 13 shows the schematic workflow of our live experimental setup. The system uses Dell Optiplex GX620 PC (3.4G dual-core cpu, 1G memory) with Hauppauge PCI-150 TV capture card. Our main reason for selecting the Hauppauge PCI-150 encoder card is because we were able to simultaneously read the output MPEG file while it is being written. Another key consideration is maintaining a balance of CPU resource to sustain both the live video capture and our intention for highlight detection with minimum delay. The main delay comes from the live text availability on the target URL. Once the event is identified from the web-text, the average processing delay for event boundary detection is around 10 seconds.

6.4.2 Live event boundary detection

Our system went "live" over the April 13th-15th 2006 weekend EPL broadcast. Some integration oversight restricted the system to only complete its run for the 1st half of 4 games: Bolton vs. Chelsea, Arsenal vs. WestBrom, Portsmouth vs. Arsenal and ManUnited vs. Sunderland. We improved our system and a second live trial was conducted from the June 10th to July 10th for all the 64 World-Cup 2006 games. All processing modules were able to execute seamlessly for the whole match of 61 games where 3 games were missed due to erroneous system configuration. The live event boundary detection performance for live EPL games and World-Cup games is listed in Table 7 and Table 8 respectively.

Since we only dealt with the 1st half of 4 live EPL games, some events had not occurred in the games, which were listed as NA in Table 7.

7. APPLICATIONS

The proposed solution for live event detection will benefit both professional service providers and consumers. We give two scenarios below to illustrate how to deploy the proposed solution into professional and consumer services.

Table 7. Event boundary detection for 4 EPL games

Event	BDA	Event	BDA
Goal	75%	Red card	NA
Shot	82.5%	Yellow card	83%
Save	90%	Foul	77.7%
Free kick	40%	Offside	85.3%
Corner	66.7%	Substitution	NA

Table 8. Event boundary detection for 61 World-Cup games

Event	BDA	Event	BDA
Goal	76.7%	Red card	82%
Shot	76.1%	Yellow card	84%
Save	60%	Foul	77.7%
Free kick	43.3%	Offside	70.5%
Corner	75%	Substitution	78.1%

7.1 Professional Services

The delivery of sports video highlights over new media channels such as 3G is an attractive option for both service-providers and consumers. Most of the existing commercial offerings are of three types: (1) live SMS updates, (2) live video streaming of the game over 3G, and (3) post-game 3G short highlight video clips. There is clearly a market gap for live 3G short highlight video updates. The main reasons for the gap are (a) concerns for dilution of TV rights, (b) accuracy and acceptability of video, and (c) cost. As regards to rights dilution, this is a business issue and we would not dwell too much onto it, apart from mentioning that there is an increasing concern amongst the EU regulatory bodies that overly-restrictive business contracts in premium sports content are hindering the development of new media markets such as 3G. As for the accuracy and acceptability of video, we argue that this is a mindset issue. Traditionally, video highlights creation is part of post-production. Extensive video editing is required to put together different interesting segments and an automatic system would not be as good. This point is valid but we argue that our system is not trying to do that in the first place. It attempts to segment a short, continuous portion from the running broadcast which would hopefully encompass the key highlights of the sports event. This would suffice for an instant video alert market. The traditional way of crafting post-production content can still continue. As for cost concerns, our system uses low-cost off-the-shelf equipment and is automatic. In occasions where it may require operator assistance, we expect this inspection effort to be minimal and expedited.

7.2 Consumer Services

We foresee a significant market in consumer client-based applications. With the advent of pervasive broadband/UMTS connectivity, IPTV, home media centers and plummeting cost of set-top OEM, we envision a great demand for both time-shifting and place-shifting video services. In the latter, especially, relevant video can be detected from a broadcast, segmented and transcoded over an IP channel to a mobile device. IP rights may not be that big an issue as for professional service-providers. We believe the computed footprint of our system can be further reduced to fit into these scenarios.

8. CONCLUSION

Event detection from live sports games is a challenge task. We have presented a novel framework for live sports event detection by

combining the live analysis and alignment of web-casting text and broadcast video. Within this framework, we have developed live event detection system for soccer game and conducted live trials on various soccer games. The experimental results are promising and validate the proposed framework.

We believe that the incorporation of web-casting text into sports video analysis, which combines the complementary strength of low-level features and high-level semantics, will open up a new possibility for personalized sports video event detection and summarization and create a new business model for professional and consumer services. In this paper, our focus is on event detection from live sports games. After we have the events and event semantics, it is not difficult to create personalized summary related to certain event, player, team or their combination according to user's preference.

Web-casting texts for various sports games are accessible from many sports websites. They are generated by professionals or amateurs using various styles (well-structured or freestyle) and different languages. Our future work will focus on exploiting more web-casting text sources, investigating more advanced text mining approach to deal with web-casting text (e.g. automatic detect event keywords) with different styles and languages, and conducting live trials on more sports domains.

9. REFERENCES

[1] Y. Rui, A. Gupta, and A. Acero, "Automatically extracting highlights for TV baseball programs", In *Proc. of ACM Multimedia*, Los Angeles, CA, pp. 105-115, 2000.

[2] M. Xu, N.C. Maddage, C. Xu, M.S. Kakanhalli, and Q. Tian, "Creating audio keywords for event detection in soccer video", In Proc. of *IEEE International Conference on Multimedia and Expo*, Baltimore, USA, Vol.2, pp.281-284, 2003.

[3] Y. Gong, L.T. Sin, C.H. Chuan, H.J. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs", In *Proc. of International Conference on Multimedia Computing and Systems*, pp. 167-174, 1995.

[4] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization", *IEEE Trans. on Image Processing*, vol. 12:7, no. 5, pp. 796–807, 2003.

[5] D. Zhang, and S.F. Chang, "Event detection in baseball video using superimposed caption recognition", In *Proc. of ACM Multimedia*, pp. 315-318, 2002.

[6] J. Assfalg, M. Bertini, C. Colombo, A. Bimbo, and W. Nunziati, "Semantic annotation of soccer videos: automatic highlights identification," *Computer Vision and Image Understanding (CVIU)*, Vol. 92, pp. 285–305, November 2003.

[7] R. Radhakrishan, Z. Xiong, A. Divakaran, Y. Ishikawa, "Generation of sports highlights using a combination of supervised & unsupervised learning in audio domain", In *Proc. of International Conference on Pacific Rim Conference on Multimedia*, Vol. 2, pp. 935-939, December 2003.

[8] K. Wan, and C. Xu, "Robust soccer highlight generation with a novel dominant-speech feature extractor", In *Proc. of IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, pp.591-594, 27-30 Jun. 2004.

[9] M. Xu, L. Duan, C. Xu, and Q. Tian, "A fusion scheme of visual and auditory modalities for event detection in sports video", In *Proc. of IEEE International Conference on Acoustics, Speech, & Signal Processing*, Hong Kong, China, Vol.3, pp.189-192, 2003.

[10] K. Wan, C. Xu, "Efficient multimodal features for automatic soccer highlight generation", In *Proc. of International Conference on Pattern Recognition*, Cambridge, UK, Vol.3, pp.973-976, 23-26 Aug. 2004.

[11] M. Xu, L. Duan, C. Xu, M.S. Kankanhalli, and Q. Tian, "Event detection in basketball video using multi-modalities", In *Proc. of IEEE Pacific Rim Conference on Multimedia*, Singapore, Vol.3, pp.1526-1530, 15-18 Dec, 2003.

[12] M. Han, W. Hua, W. Xu, and Y. Gong, "An integrated baseball digest system using maximum entropy method", In *Proc. of ACM Multimedia*, pp.347-350, 2002.

[13] S. Nepal, U. Srinivasan, and G. Reynolds, "Automatic detection of goal segments in basketball videos, In *Proc. of ACM Multimedia*, Ottawa, Canada, pp.261-269, 2001.

[14] J. Wang, C. Xu, E.S. Chng,, K. Wan, and Q. Tian, "Automatic generation of personalized music sports video", In *Proc. of ACM International Conference on Multimedia*, Singapore, pp.735-744, 6-11 Nov. 2005.

[15] N. Nitta and N. Babaguchi, "Automatic story segmentation of closed-caption text for semantic content analysis of broadcasted sports video," In *Proc. of 8th International Workshop on Multimedia Information Systems '02*, pp. 110–116, 2002.

[16] N. Babaguchi, Y. Kawai, and T. Kitahashi, "Event based indexing of broadcasted sports video by intermodal collaboration," *IEEE Trans. on Multimedia*, Vol. 4, pp. 68–75, March 2002.

[17] N. Nitta, N. Babaguchi, and T. Kitahashi, "Generating semantic descriptions of broadcasted sports video based on structure of sports game," *Multimedia Tools and Applications*, Vol. 25, pp. 59–83, January 2005.

[18] H. Xu and T. Chua, "The fusion of audio-visual features and external knowledge for event detection in team sports video," In *Proc. of Workshop on Multimedia Information Retrieval (MIR'04)*, Oct 2004.

[19] H. Xu and T. Chua, "Fusion of multiple asynchronous information sources for event detection in soccer video", In *Proc. of IEEE ICME'05,* Amsterdam, Netherlands, pp.1242-1245, 2005.

[20] http://news.bbc.co.uk/sport2/hi/football/teams/

[21] http://sports.espn.go.com/

[22] M. Bertini, R. Cucchiara, A. D. Bimbo, and A. Prati, "Object andevent detection for semantic annotation and transcoding," in Proc.IEEE Int. Conf. Multimedia and Expo, Baltimore, MD, Jul. 2003, pp.421–424.

[23] R. Leonardi and P. Migliorati, "Semantic indexing of multimedia documents," *IEEE Multimedia*, Vol. 9, pp. 44–51, Apr.–June 2002.

[24] http://soccernet.espn.go.com/

[25] Y. Tan and et al, "Rapid estimation of camera motion from compressed video with application to video annotation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 10-1, pp. 133–146, 2000.

[26] Y. Li, C. Xu, K. Wan, X. Yan, and X. Yu, Reliable video clock time recognition, In *Proc. of Intl. Conf. Pattern Recognition*, Hong Kong, 20-24, Aug. 2006.