Submitted to IEEE CVPR CBAIVL Workshop, July 2001

Real-Time Content-Based Adaptive Streaming of Sports Videos

Shih-Fu Chang, Di Zhong, and Raj Kumar
{sfchang, dzhong, kumar}@ee.columbia.edu
Department of Electrical Engineering, Columbia University, NY, USA

Abstract

In this paper, we present a real-time software system for filtering live video content and then adaptively streaming the video over resource limited networks or devices. The rate adaptation is content-based and dynamically varied according to the event structure and the video content. The system includes a content analysis module for detecting important segments in the video, a variable-rate encoding module, and a buffer management module for streaming the variable rate video over low bandwidth networks. We have implemented a prototype system, which performs real-time content adaptation for tennis and baseball with promising results.

Keywords

Video filtering, content-based video, adaptive streaming, event detection, MPEG-7

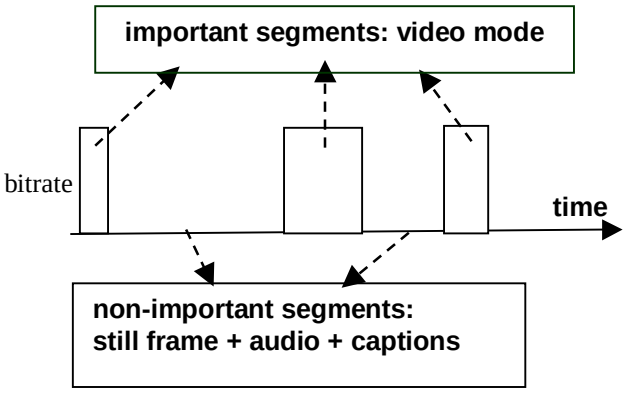

Figure 1. Content-based adaptive streaming of videos over bandwidth limited links. Important segments are transmitted with full motion audio-video, while during non-important segments only still frames, audio, and captions are transmitted.

1 Introduction

In a bandwidth limited environment, content filtering is important for reducing the program duration or the bandwidth. Real-time filtering over live content is necessary in time critical applications. Sports videos are important instances of this problem. Users' interests in watching the video is time sensitive, fading rapidly after the game is over and the result is known. Other characteristics of sports video – the large audience base, predictable program structures, and long durations make sports video an excellent domain for real-time video filtering.

Filtering selects important segments and removes the rest. Depending on application settings, filtering can be implemented in the streaming mode or the alert messaging mode. The former streams the "selected" content to the end user in the "live" mode, while the latter sends alert messages to users to indicate occurrence of events of interest. In practice, filtering criteria may include selection of video segments showing important activities (e.g., pitching, serving, plays, scores).

There has been work on video shot segmentation, content-based indexing, event detection, and structure analysis. But a systematic framework and effective algorithms for exploring the regular event structure and corresponding camera views in sports videos has not been well addressed. Applications of the content adaptation in real-time video streaming have not been shown either. More detailed discussion of the prior work and contribution of this paper will be presented later.

Given content selection criteria, the video streams can be adapted dynamically. Figure 1 shows an example scenario for *content-based adaptive streaming*. Full-motion audio-video content is displayed during important periods (e.g., pitching and follow-up plays in baseball) while during non-important periods, only key frames, audio, and text are displayed. Such adaptation is particularly useful for mobile personalized applications. It allows for efficient usage of the bandwidth and can maximize the video quality of important video segments over bandwidth limited links. During the non-important segments, low-bit-rate data (e.g., audio and text) are transmitted and displayed. Users can still monitor the activities in the program by listening to the audio or viewing the key frames with text captions.

In this paper, we present a real-time event detection and filtering system realizing the content-based adaptive streaming concept. The system achieves very good results (higher than 90% precision/recall) in the initial domains-baseball and tennis. The component for detecting important events is based on the algorithms developed in our prior

works [1, 2]. We first discuss the characteristics of the sports domain. We discuss related works and our new contributions in Section 3. Section 4 includes descriptions of the system architecture and constituent components. It also includes brief reviews of the event detection modules. We discuss the design and status of our current prototype in Section 5. Other applications of the system are also mentioned. Conclusions and our current research are given in Section 6.

2 Domain Characteristics

Sports video is a major part in most broadcasting TV programs. Compared to other videos such as news and movies, sports videos have well-defined content structure and domain rules. Each sports video program consists of regular structures, such as pitch-batter-inning sequences in baseball, serve-game-set sequences in tennis, and play-break sequences in soccer. In addition, in a sports video, there are a fixed number of cameras in the field that result in unique views during each segment. In tennis, when a serve starts, the scene is usually switched to the court view. In baseball, each pitch usually starts with a pitching view taken by the camera behind the pitcher. Detection of such domain-specific views helps determination of the fundamental content unit, such as pitch, serve, play, etc. Furthermore, for TV broadcastings, there is special information (e.g., score board, player's name) displayed on the screen to indicate the status of the game. Changes of such inserted information follow the rules of the game, e.g., the progression of the ball count and the inning number. The constraints in the progression of such information allow for development of effective recognition tools. One of our principles in automatic video filtering is to exploit all the aforementioned domain knowledge - predictable temporal structures, unique views, and constrained state transitions.

Let us define three related but distinctive terms, view, FSU (fundamental semantic unit), and event. A *view* refers to a specific angle and location of the camera when the video is captured. During a continuously captured shot of video, the view may be static or changing due to the camera control operations. In sports video, there are a finite number of views using predetermined locations and angles. For example, in baseball, typical views are "whole field", "player close-up", "ball/runner tracking", "out field", etc.

FSUs are instances of recurrent happenings of a video content unit corresponding to important semantics at a specific level, such as pitch, play, inning, etc. For example, in several types of sports (baseball, tennis, golf, basketball, soccer, etc), there is a fundamental level of content which corresponds to an intuitive cycle of activity in the game. For baseball, the FSU could be the unit corresponding to a pitch (including the follow-up activities, such as catch or run), the complete play of a batter, or an inning. For tennis, the FSU could be the unit corresponding to a serve (including follow-up activities such as strokes), or a set. For soccer, the FSU may correspond to a play (when the ball is in the field and is "alive"). A FSU may include multiple views (e.g., the pitching view plus the follow-up views.) A video program can be decomposed into a sequence of FSUs. Consecutive FSUs may be next to each other without time gaps, or may have additional content (e.g., videos showing crowd, commentator, or player transition) inserted in between. FSU's provide an intuitive level for accessing and summarizing video content. Organizing video data to these levels provides abstraction at multiple levels with flexible granularity. Figure 2 shows a multi-level FSU structure of a tennis program.

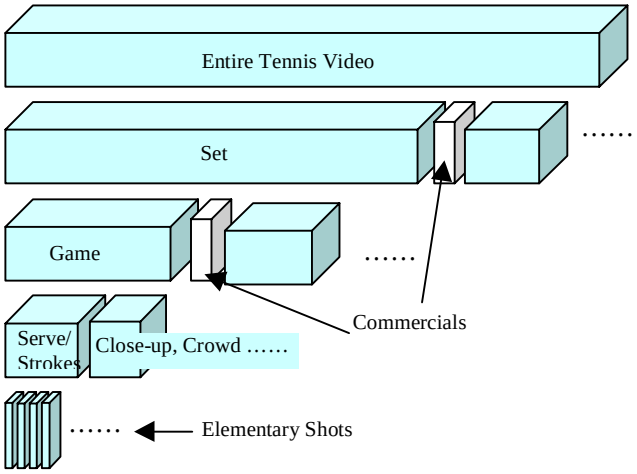

Figure 2. Typical content structure in tennis video

Events are happenings of actions in the video, such as score, hit, serve, pitch, penalty, etc. A FSU may include occurrence of one or more different categories of events.

The use of the above three terms may be interchanged sometimes due to their correspondence in specific domains. For example, a view taken from behind the pitcher typically indicates the pitching event. The pitching view plus the subsequent views showing activities (e.g., motion tracking view or the out field view) constitute a FSU at the pitch-by-pitch level. A FSU at a higher level (e.g., player-by-player, or inning-by-inning) can be recognized based on the recognition of other information such as recognition of the ball count/score by video text recognition and the domain knowledge about the rules of the game.

3 Related Work and Contribution

Content summarization and adaptation has been called for in the emerging MPEG-7 standard [4]. MPEG-7 provides several description schemes for describing summaries and adaptation mechanisms of audio-visual data, especially for

universal media access (UMA) applications over heterogeneous platforms. Such a standard framework facilitates rapid deployment of services and exchange of content. However, it does not standardize specific tools or algorithms for achieving summarization and adaptation.

There has been a lot of work on video shot segmentation and object indexing. The former decomposes video sequences into short shots by detecting discontinuity in visual and/or audio features. The latter extracts video objects and indexes the objects based on their features. These approaches typically focus on low-level structures and features, and do not provide high-level event detection capabilities. Some efforts [5, 6, 7] have been made to segment video into high-level units such as scenes. Others aim at detecting generic events in audio-visual sequences by integrating multimedia features [8]. However, applicability of such techniques to the sports domain has not been demonstrated. More importantly, the unique structures and rules in sports video production were not explored.

There are some works on sports content analysis. Gong et al [12] presented a soccer video analysis system, which classified key-frames of each video shot according to the physical location in the field (right, left, middle) or the presence/absence of the ball. A system for classifying each shot of tennis video to different events was proposed in [13]. A system for detecting events in basketball games (e.g., long pass, steals, fast field changes) was described in [14]. The specific content structures and domain knowledge described in Section 2 were not explored and integration with adaptive streaming was not proposed.

Multimedia adaptation techniques have been developed for transcoding the multimedia content in a UMA environment [9, 10], but such existing systems primarily use generic types (e.g., image file formats or generic purposes) or low-level attributes (e.g., bit rate).

In some video coding standards such as MPEG-2 and MPEG-4, profiles have been specified to support scalable coding. Video is encoded into different layers of substreams which can be transmitted and decoded adaptively. In a time-varying or heterogeneous network environment (e.g., multicast) different layers can be selected or dropped depending on the available bandwidth and/or client capabilities. In contrast, the proposed content-based adaptive streaming approach dynamically changes the bit rate or modality (e.g., video vs. key frame) based on the video content. A general adaptation scheme can be developed to combine the content-based scheme and the scalable coding schemes mentioned above.

In our prior works [1, 2], we used two different approaches to detecting event structures in sports video. We aimed at effective combination of the generic computational framework and the domain-specific knowledge. The general framework allows for rapid generalization to different domains, while the use of domain specific knowledge enables boosting of performance. In [1], we analyzed the event structures by detecting the canonical views such as pitching in baseball and serving in tennis. Such views are detected using global feature clustering and object-level verification rules. In [2], we adopted a frame-based label sequence processing framework for play-break segmentation of soccer videos. We used a simple, but effective grass area feature to map sampled frames to mid-level labels (global, zoom-in, and close-up), then developed effective rules for segmenting plays/breaks from the label sequences.

In this paper, we present a real-time adaptation system utilizing our event detection modules together with other components for rate adaptation and buffer management. The contributions of the paper include the novel system architecture integrating event filtering with dynamic video rate adaptation, and the demonstration of real-time performance in filtering/streaming live video programs.

4 System Overview

Figure 3 shows the system architecture of the content-based adaptive streaming system. The live video programs are received and then processed at the server or the filtering gateway (e.g., at mobile base station). The event detection and structure analysis module analyzes the input video and marks the important segments of video based on a preset criterion or customized user preference. One simple filtering criterion for baseball is to mark all pitching shots and subsequent shots showing continuing activities as important. For tennis, the criteria could be to select all serving shots and follow-up activity shots as important. For soccer, the important segments may include all the plays, excluding the break segments between plays. Non-important shots may include commercials, commentators, and close-up shots of crowds, players, etc.

After the event detection module, the content adaptation module manipulates the actual audio-visual sequence according to the adaptation schedule. For example, if we adopt the adaptation schedule shown in Figure 1, video of all the important segments will be kept while the non-important segments will be replaced by key frames. The adapted content will be fed to the encoding module which is compliant with industry or standard formats. The actual encoding method may vary depending on the selected encoding and streaming format (e.g., MPEG-1, Real, or Microsoft). For some encoders, we can just encode the important segments plus the initial frames of the non-important segments. This is similar to the case of a live

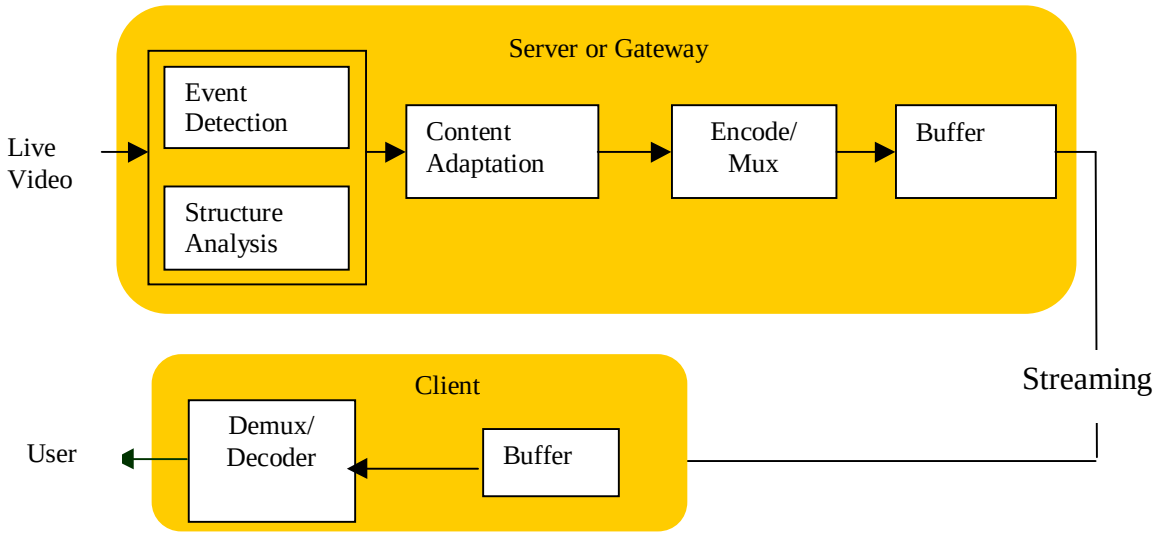

Figure 3. System Architecture of Content-Based Adaptive Video Streaming

encoding session with the encoding operation turned on and off intermittently, according to the content importance. For other encoders, we may need to use the key frames to replace all frames of the non-important segments and then encode the new sequence as a continuous stream.

We envision the client terminal to be a hand-held device or desk top PC with a low-bandwidth network link. To send the adaptive-rate video over a low-bandwidth link, buffer management modules are needed at the server and the client locations to smooth out the bursty rate. Finally, audio/video streams are demuxed, decoded, and displayed at the client according to the content adaptation schedule.

4.1 Adaptive Streaming Over a Low-Bandwidth Link

A content-based stream produced by the adaptation schedule shown in Figure 1 will produce variable bit rate. Delivery of such streams over bandwidth limited networks requires careful planning of system resources and operations. Figure 4 shows the relationships between the input video rate, the transmission rate, and the status of the buffers at the server and client. Here we assume a constant channel rate, which is between the high rate and the low rate used in the adaptive stream. Line L_2 indicates the accumulated transmission rate, which is equal to the product of the elapsed time and the channel rate. Vertical distance between L_2 and L_1 equals the buffer size at the server while vertical distance between L_3 and L_1 equals the buffer size at the client. C_1 represents the accumulated rate of the input video. C_2 represents the accumulated rate of the displayed video at the client and is simply a time shifted copy of C_1 if no buffer exceptions occur (i.e., overflow or underflow). The shift amount equals the net latency time from the arrival time at the server until the playback time of each video packet. Indicated in shaded areas, the vertical distance between C_1 and L_2 equals the buffer status at the server, while the distance between C_2 and L_2 equals the client buffer status.

We discuss the procedure for setting the required system resources in the following. In typical wireless systems, the channel rate and the client buffer size are the primary system constraints. The server does not have very tight resource constraint and thus minimizing the server buffer size is not a major concern. Let us first consider an offline situation in which we have complete information of the time intervals of all the important and unimportant segments. Given the channel rate, the slope of L_2 is determined. Given the client buffer size, the vertical gap between lines L_2 and L_3 is fixed. Given the time intervals of the important segments (high rate) and unimportant segments (low rate), we can determine the permissible high and low bit rates. Using different high/low rates will change the slopes of segments on curve C_2. The rates should be set to contain curve C_2 between lines L_2 and L_3. Once the high/low rates are determined, a suitable server buffer size can be set, which in turn determines the playback latency.

For an online situation in which the information of time intervals is not deterministically given, we need to collect the statistical information of the time intervals of the important and unimportant segments. Statistical models may be developed and statistical queuing analysis may be done to analyze the buffer status. Under some simplification conditions, a statistical bound on buffer error rates may be derived [11] and requirements of buffer resources and playback latency can be optimized.

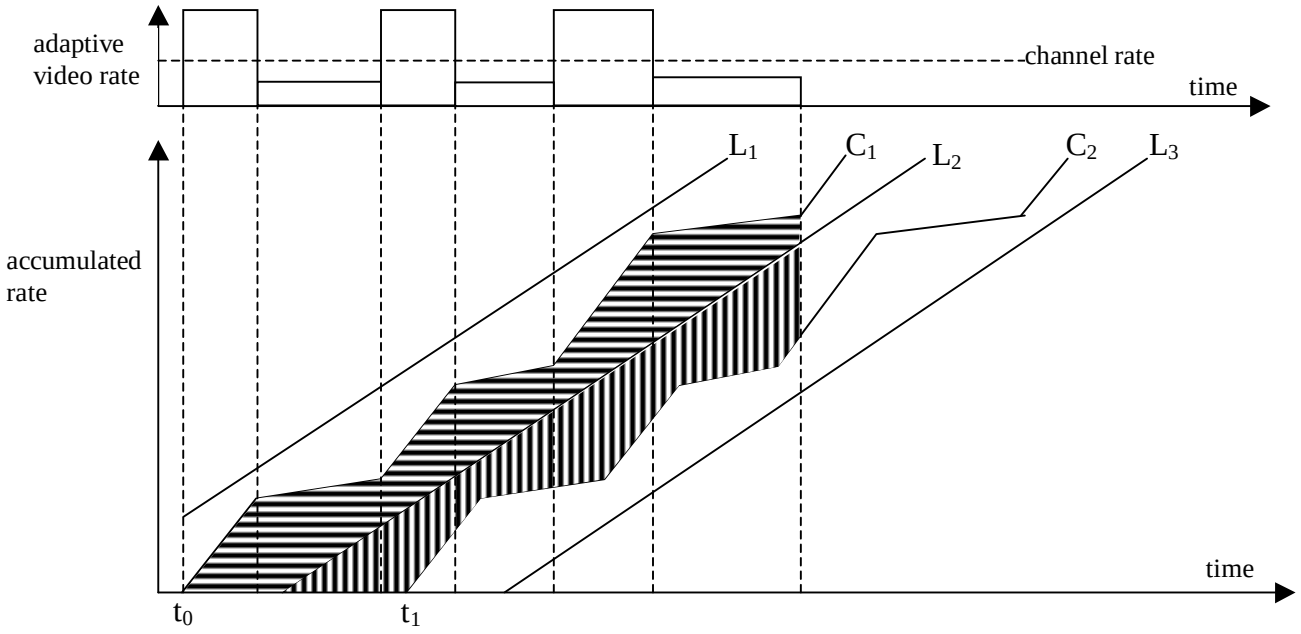

Figure 4. Content-based adaptive streaming - relationships between video bit rates, transmission rate, and buffer status. All the curves show accumulated rates. L_2: channel rate, L_1: channel rate plus server-side buffer side, L_3: channel rate less client-side buffer size, C_1: server-side input video rate, C_2: client-side video display rate.

4.2 Structure Analysis/Event Detection Module

Figure 5 includes a conceptual diagram for developing the video event filtering applications. Given a specific domain such as baseball or tennis games, fundamental semantic units (FSU) are identified by hand according to domain knowledge (program structures and game rules). As described earlier, FSU can be selected to be one of several different levels, according to the application focus or individual user interest.

Given the FSU structure of videos in specific domains, we developed automatic tools to detect the boundaries of FSU's. In [1] [2], we have demonstrated real-time detection of pitching events in baseball, serving views in tennis, and play segments in soccer. To illustrate the process, Figure 6 shows the architecture of our real-time tennis serve view detection system. It includes multiple computing stages. First, the incoming video data is decomposed into shots by using a real-time compressed-domain shot detector. In the second stage, a single key frame of each shot is extracted and frame-level features (such as color histogram) are extracted. Such frame-level features are matched against color models that have been constructed in a separate supervised clustering process. The output of this stage includes candidate shots whose global color features match those of the clusters of the unique views obtained in the training process. An adaptive process is used for each new video program to select the most relevant visual clusters corresponding to the canonical views. Candidate shots passing the global visual matching process are further examined by analyzing the constituent video objects and their spatio-temporal features (such as motion, geometry, and locations of the player and court lines).

Object-level constraints and rules applied in the verification stage can be specified by experts exploring the domain knowledge or learned in a semi-automatic way. An interactive learning system called Visual Apprentice [3] was used to obtain basic forms of the rules associated with the constituent objects in the canonical view. The Visual Apprentice system allows users to explicitly define structured scene models which consist of multiple levels of components corresponding to real-world objects in the scene. It also learns features and classification tools of individual component objects through interactive labeling at the component level.

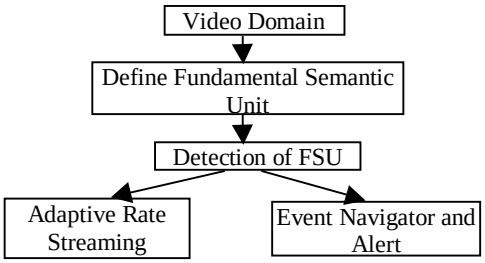

Figure 5. Conceptual Process for Defining Video Event Structure and its Applications

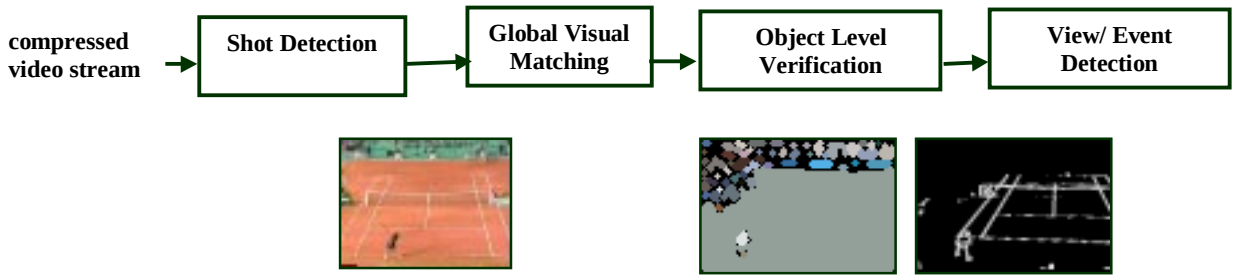

Figure 6. System Architecture for Automated Canonical View Detection in Tennis Video

In some domains, the object-level features also allow for detection of higher-level events. For example, as shown in [1], by tracking the location and moving pattern of the foreground moving object in tennis video, we were able to detect the number of strokes and the position of the player in relation to the net with reasonable accuracy. Such information is useful in the summary and navigation tools which will be described later.

Automatic segmentation of video objects and analysis of object-level features involves complex computation. However, since such computation is used as refined verification to candidate shots and is not applied in every shot, real-time processing speed is achieved for the whole video.

In [2], we have also developed event structure analysis algorithms for detecting play segments within soccer video programs. We adopted a frame-based label-sequence processing framework for play-break segmentation. To explore the domain constraints, we used a simple, but effective grass area feature to map sampled frames to mid-level labels (global, zoom-in, and close-up). We developed effective rules for segmenting plays/breaks from the label sequences. When tested over diverse programs from different sources (soccer programs from different countries), our system achieved very good performance in detecting the play events, although the boundary time precision still needs to be improved.

5 Prototype and Related Applications

We are currently developing an integrated system combining the above event detection and adaptive streaming modules. Our demo testbed consists of a Pentium III class PC with two processors serving as the gateway, connected to a Compaq IPAQ with twisted-pair wiring serving as the physical layer and UDP/TCP/IP on top.

The gateway and client software is implemented in C++ using the Microsoft Format SDK. The gateway software consisted of a real-time structure parsing/event filtering module that identifies the FSUs and classifies the content importance. It also includes an encoding module to which the content filtering result is passed. The encoding module, which is implemented using the Microsoft Format SDK, uses the filtering information to compress the incoming stream into a variable rate MPEG-4 stream. The incoming stream is compressed into two target bitrates: a predetermined high target bitrate and a predetermined low target bitrate depending on whether the segment is classified as important or less important. The high target bitrate stream contains both video and audio, while the low target bitrate stream consists of a still-image and audio. The still-image is typically a frame extracted from the corresponding segment. This stream is passed to the PDA over UDP/IP and the twisted-pair link and is played back with a media-player again using Microsoft's Format SDK. The final playback stream uses the schedule similar to the one shown in Figure 1. For baseball, all the pitching segments plus their follow-up action shots are shown with the high bitrate, while the rest of the video are shown with the low bitrate. Similarly, for tennis, all the serving segments plus their follow-up action shots are shown at the high bitrate.

The testbed currently demonstrates the real-time performance of the software system in event filtering and adaptive streaming. The wireless link with bandwidth constraints has not been included and is currently simulated by the twisted-pair wiring. The incoming video sequences are pre-recorded sports programs and are read from the local disk in real-time during the test. Adding a hardware component to capture live videos broadcasted over the air and use them as the test input should not significantly affect the real-time performance of the system.

In a test using baseball videos, we found that the minimum acceptable quality, at the QCIF resolution, for the high target bitrate (for both audio and video) was about 30 kbps and for the low target bitrate (for the still frames and audio) was about 10 kbps. The actual bitrate slightly varied from the target rate but stabilized towards the target rates. The mean lengths of the important and less-important segments, as reported by the event filtering module were approximately the same (25 seconds and 32 seconds respectively) leading to an average bitrate of approximately 20 kbps.

The event filtering module implements the view detection algorithm described in Section 4.2. The view detection process currently detects pitching in baseball and serving in

tennis. It also uses simple rules in determining the importance of the subsequent shots after each pitching or serving shot. The process achieves very good accuracy (precision and recall both higher than 90%) in our current experiments using a few hours of tennis and baseball broadcast programs as test content. The training data was 10 minutes in length in each domain and was obtained from different programs than the test video. More detailed discussion of these modules can be found in [1, 2].

5.1 Other Applications

The event filtering tools are also useful for other applications such as video navigators or browsers used in personal video recorders. Figure 7 shows one example interface. The left side shows the tree-structure indexes of video content at different levels, such as shots and serves. The summary shown on the right side indicates the statistics of events such as the number of pitches, serves, or plays. By clicking on a specific category, users can access all units the category (e.g., all the serve events or all the pitching events).

The above browsing/navigation interface can be used together with content alert messaging applications. For example, an alert message can be sent to the user's personal devices when an important video event is detected in the live or stored content. Such video segments can be cached to the personal devices and viewed by the user offline. The above interface as shown in Figure 7 can be used to browse through the detected events efficiently.

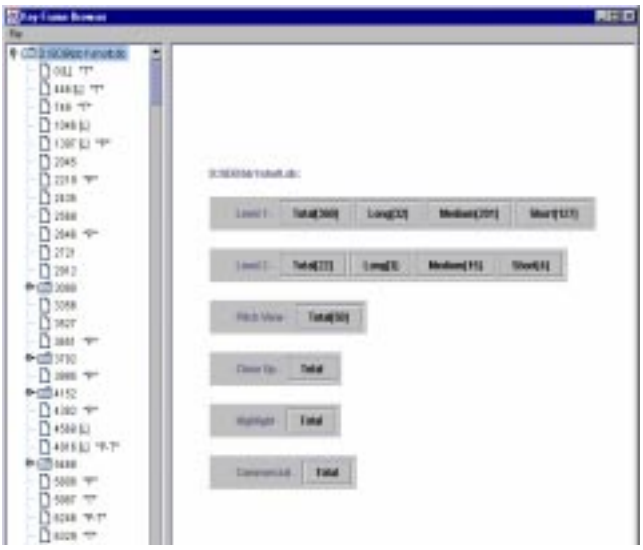

Figure 7. Event summary and navigation interface. The left side shows a tree-structure index to shots and events. P indicates "pitches" and T indicates "strokes". The right side shows the statistics of events and views.

6 Conclusions

We have presented a working real-time video filtering system that realizes the unique *content-based adaptive streaming* concept. We described the overall system architecture combining the event detection module, the rate adaptation module, and buffer management schemes. We also discussed procedures for determining suitable system parameters such as the buffer size, the permissible bit rates, and the playback latency.

The system detected and filtered the important events in live broadcast sports videos, and adapted the bit rate of the incoming streams according to the content importance. The importance can be defined based on the event structure or user preferences. The filtering results can also be utilized to build a summarization or navigation tool, in which users can navigate through the videos at multiple levels of abstraction efficiently. We have developed detection modules and a basic working system for adaptive streaming for baseball and tennis. We achieved the real-time performance in software by combining effective tools in compressed-domain processing, multi-feature fusion, and domain knowledge exploitation.

Our current work focuses on detection of higher-level events (e.g., scores, players) in the sports domain and application in other domains, such as soccer and film. Although less structured than other sports domains, soccer videos can be automatically parsed into fundamental semantic units of plays and breaks with promising results [2]. We are testing the quality of the adaptive streaming method over soccer videos. In the film domain, the content is even less structured and the production rules are less constrained. We are developing techniques to extend the concept of fundamental semantic units and new methods for generating "skims" of individual content units (e.g., scenes) based on the production syntax and viewer's psychological models. In addition, we are investigating new implementations over actual wireless links and the feasibility of using new standard coding tools such as MPEG-4 scalable encoding.

7 References

1. D. Zhong and S.-F. Chang, "Structure Analysis of Sports Video Using Domain Models," IEEE Conference on Multimedia and Exhibition, Japan, Aug. 2001.
2. P. Xu, L. Xie, S.-F. Chang, A. Divakaran, A. Vetro, and H. Sun, "Algorithms and System for High-Level Structure Analysis and Event Detection in Soccer Video," IEEE Conference on Multimedia and Exhibition, Japan, Aug. 2001.
3. A. Jaimes and S.-F. Chang, "Model Based Image Classification for Content-Based Retrieval," SPIE

Conference on Storage and Retrieval for Image and Video Database, Jan. 1999, San Jose, CA.
4. S.-F. Chang, T. Sikora, and A. Puri, "Overview of the MPEG-7 Standard," IEEE Transactions on Circuits and Systems for Video Technology, special issue on MPEG-7, June 2001.
5. H. Sundaram and S.F. Chang. “Determining computable scenes in films and their structures using audio visual memory models”, ACM Multimedia 2000, Oct 30 - Nov 3, Los Angeles, CA.
6. M. Yeung B.L. Yeo Time-Constrained Clustering for Segmentation of Video into Story Units, Proc. Int. Conf. on Pattern Recognition, ICPR ′96, Vol. C pp. 375-380, Vienna Austria, Aug. 1996.
7. R. Lienhart et. al. Automatic Movie *Abstracting*, Technical Report TR-97-003, Praktische Informatik IV, University of Mannheim, Jul. 1997.
8. M. R. Naphade and T. S. Huang, "Semantic Video Indexing using a probabilistic framework," International Conference on Pattern Recognition, Barcelona, Spain, Sept. 2000.
9. J. R. Smith, R. Mohan and C. Li, "Scalable Multimedia Delivery for Pervasive Computing", ACM Multimedia Conference (Multimedia 99), Oct. -Nov., 1999, Orlando, Fl.
10. A. Fox and E. A. Brewer, "Reducing WWW Latency and Bandwidth Requirements by Real timer Distillation", in Proc. Intl. WWW Conf., Paris, France, May 1996.
11. P. R. Jelenkovic, "Network Multiplexer with Truncated Heavy-Tailed Arrival Streams", INFOCOM'99, New York, NY, March 1999.
12. Y. Gong et al “Automatic parsing of TV soccer programs”, In Proc. IEEE Multimedia Computing and Systems, May, 1995, Washington D.C.
13. G. Sudhir, J. C.M. Lee and A.K. Jain, “Automatic Classification of Tennis Video for High-level Content-based Retrieval”, Proc. Of the 1998 International Workshop on Content-based Access of Image and Video Database, January 3, 1998 Bombay, India.
14. D. D. Saur, T.-P. Tan et al. “Automated Analysis and Annotation of basketball Video”, Proceedings of SPIE's Electronic Imaging conference on Storage and Retrieval for Image and Video Databases V, Feb 1997.