Real-Time View Recognition and Event Detection for Sports Video

Authors:

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

For special issue on Multimedia Database Management Systems at Journal of Visual Communication and Image Representation, 2004.

Correspondence Contact:

Prof. Shih-Fu Chang

Department of Electrical Engineering

500 W. 120th St., Rm. 1312

Columbia University

New York, NY 10027

sfchang@ee.columbia.edu

tel: 212-8546894

fax: 212-8540300

url: http://www.ctr.columbia.edu/~sfchang

Real-Time View Recognition and Event Detection for Sports Video

Keywords:

Video Indexing, Content-Based Video Retrieval, Sports Video Analysis, MPEG-7

Abstract

In this paper, we present a general framework and new effective algorithms to detect the syntactic structures that are at a level higher than shots. In sports video, such high-level structures are often characterized by the specific views (e.g., pitching or serve) and the subsequent temporal transition patterns within each temporal structural segment. We have developed robust statistical models for detecting the domain-specific views with real-time performance and high accuracy. The models combine domain-independent global color filtering method and domain-specific constraints on the spatio-temporal properties of the segmented regions (e.g., locations, shapes, and motion of the objects). The real-time performance was accomplished by using efficient compressed-domain processing at the front end and computational expensive object-level processing on filtered candidates only. High-level events (e.g., strokes, net plays, baseline plays) are also detected after the view recognition. Results of such structure and event detection allow for efficient browsing and summarization of long sports video programs.

1. Introduction

In a typical content-based video indexing system, videos are decomposed into shots which are then processed to obtain constituent objects and features [1]. As shown in Figure 1, these extracted entities form a comprehensive feature library and a useful framework for describing video content. While these features are useful in video retrieval such as similarity searching, they lack information at the semantic level. To address this problem, research incorporating domain specific knowledge provides a promising direction.

In this paper, we present a high-level scene and structure analysis system for sports videos. This system is built on top of object segmentation, feature extraction and matching techniques we have developed in our prior works [10], [11], [12]. Specifically, we present model-based and rule-based methods for detecting important, recurrent scenes in sports videos (e.g., pitching scenes in baseball and serving scenes in tennis). Our methods use off-line supervised learning and on-line adaptive updating to obtain effective scene models and the associated spatio-temporal rules. To achieve high accuracy, we also adopt a multi-stage process in which complex object-level features are analyzed in the later stage to verify candidate videos detected by less complex models in the initial stage.

Real-time performance is emphasized in this work in order to satisfy the requirement of live video filtering and dynamic content-adaptive encoding. In [16], we combine the real-time structure detection tool presented here with adaptive rate encoding to improve the overall quality of the video – video segments of important events (e.g., pitching) are allocated more bandwidth than other segments. We achieve the real-time performance by extracting and analyzing most of the features in the compressed domain (i.e., MPEG-1 or -2 compressed format). Given the detected scenes and structures in the video, we further develop an effective summarization and browsing interface allowing users to easily access content structure and event index of videos.

Combination of domain knowledge and low-level features has been explored in some works. In [9], an edge model is used to match anchorperson views in specific news programs. The story structure is then reconstructed by finding anchorperson views as well as commercials. In [3], logical story units (LSU's) are extracted from movies according to temporal consistency of frame-level features (e.g., color histogram). The consistency is based on the assumption that an event related to a specific location and certain characters will result in consistent features in a period of time. In [4], a multi-modality memory model was used to detect audio-visual scene boundaries in films by measuring the consistency of visual features, audio features, and their synchronization.

Compared with other types of videos, sports videos have different characteristics. A sports game usually occurs in one specific location, contains rich motion information and has well-defined syntactic and semantic structures. Event detection in basketball and tennis videos has been studied in [7], [8], [13] respectively, but without parsing high-level temporal structures. View and event classification has been proposed in [14] for soccer video indexing, but without using object-based verification framework and compressed-domain approaches for real-time performance. Detailed comparison between the proposed methods and the prior works will be presented in Section 2.

In this paper, we present a real-time structure parsing and event detection system for sports videos. Compared to existing work, our system has the following unique features.

- A general framework for video structure parsing and event detection.

- Combination of domain-independent global model-based filtering methods with domain-specific object-level spatio-temporal constraints.

- Multi-stage semantic scene detection and verification algorithms using our unique moving object segmentation and feature analysis methods.

- Real time processing performance by processing most features in the compressed domain.

- Higher accuracy demonstrated in specific sports domains such as tennis and baseball.

In the rest of the paper, we will first discuss content structures in sports videos. The scene detection and structural analysis system is described in Section 3, using tennis as an example. Experiment results in tennis and baseball videos are shown. In Section 4, we present methods for detecting high-level events (e.g., net plays and strokes in tennis) occurring within each scene. Finally, in Section 5, we describe an effective summarization and browsing application, which allows users to easily access content structure and event index in the video. Conclusions and future works are summarized in Section 6.

2. Content Structures in Sports Video

Sports video represents a popular type of broadcast TV programs. Compared to other videos such as news and movies, sports videos have well-defined content structures and domain-specific rules. A long sports game is often divided into a few segments. Each segment may again contain several sub-segments. For example, in football programs, a game contains two halves, and each half has two quarters. Within each quarter, there are many plays, and each play usually starts with the formation in which players line up on two sides of the ball. In tennis, a game is divided first into sets, then games and serves (as shown in Figure 2).

In addition, in a sports video, there are a fixed number of cameras in the field that result in recurrent distinctive scenes throughout the video. In tennis, when a serve starts, the scene is usually shown with the full-court views (Figure 3). In baseball, typically each pitch starts with a pitching view taken from behind the pitcher. These views usually have consistent visual-audio attributes that do not vary greatly from game to game. For example, in baseball, the pitching scenes typically show consistent features and spatio-temporal relationships of constituent objects of the scene (e.g., grass, ground, pitcher, catcher, and batter in the pitching scene). This property has been explored in our prior work in learning effective detectors for baseball pitching based on a structured scene model [15]. In tennis, the full-court scenes typically show consistent ground colors, court lines, and the players (e.g., as shown in Figure 3). Although there will certainly be variations in colors (different court surfaces), lighting, and view angles, we argue the consistence is significant and effective pre-filters can be developed to detect candidates while more complex rules (such as rules associated with the lines and players) can be used to improve the detection performance. These characteristics allow us to develop effective video indexing methods that combine domain-specific syntax and rules, generic machine-learning tools, and automatic tools for video object analysis.

Given a sports video, our primary goal is to identify important structures and events that are interesting to users, and automatically or semi-automatically build a content index allowing for efficient search and retrieval of such events and structures. A few prior works have been conducted to analyze sports videos. In [7], a basketball video annotation system is presented. Using specific knowledge of basketball games, the system detects wide-angle shots versus close-up shots. Camera motions are further analyzed within wide-angle shots to detect possible fast break, steal and probable shots. When users want to retrieve one type of event (e.g., fast break), only sequences satisfying corresponding motion criteria are shown. Tennis videos are studied in [8]. The approach is based on the extraction of a model for the tennis court-lines from videos. A player tracking algorithms is developed to track players, and then a reasoning module is used to map low-level positional information to high-level tennis play events. In [13], a sequence of ad hoc audio features (e.g., ball hitting sound followed by crowd shouting sound) are used to detect highlight segments in baseball videos. In [14], frame-level features and some domain-specific features are used to detect various views and events in soccer videos. In [15], our prior work uses interactive learning tools to develop the optimal features and classifiers in a hierarchical scene model for detecting views in baseball. Constraints associated with the objects (e.g., color and shape of ground or grass regions) were automatically discovered by analyzing user annotated video samples. Although a different detection framework is used in this paper, such discovered object-level constraints are very useful and have been incorporated in the object-level verification stage of the system described in this paper.

Our work in this paper shares similar goals as the above works but emphasizes more on parsing the temporal structure by detecting the recurrent content units (e.g., play in baseball and serve in tennis). We adopt a distinctive framework, which incorporates a systematic learning approach and an adaptive model selection method. It also combines domain-independent filtering methods with special refinement methods using domain-specific rules.

Another challenging problem that has not been addressed in prior works is the real time performance issue. Live sports video broadcasting has important applications. Interest of audience goes down greatly after the game results are known. The real-time capability in generating event indices is critical in practical applications for live sports videos. Even for archived sports videos, such real-time indexing system still provides significant value in cutting the potentially prohibitive cost involved in manual annotation.

In this paper, we present a general framework and new effective algorithms to analyze structures, views, and events of broadcasted sports videos in the real time. Our approach consists of two phases – training and operation, as shown in Figure 4. First, in the training phase, feature models and object rules are learned automatically or semi-automatically. In the operation phase, optimal models are selected to adapt to new video programs from live sources and used to detect target scenes and events in new videos. A scene verification model is also applied to reduce false positives. Finally, high-level events within detected scenes are detected based on constraint models on spatio-temporal properties of the segmented objects.

3. Multi-Stage View Recognition

To automatically analyze the temporal structure of a sport game, we need to detect visual cues that indicate the beginning and/or ending of each structural element. Some common features occurring between top-level sections are commercials, embedded texts and special logos. Many methods have been proposed for commercial and text detection [5], [6]. Here we will study the detection of basic units within a game, such as serves in tennis and pitches in baseball. These units usually start with a special view. Simple color based approaches have been suggested in [8]; but based on our experiments to be described later, such approaches do not achieve adequate performance. Furthermore, as color information varies from game to game, adaptive methods need to be exploited to handle such variations. In our real-time framework, we first use a fast adaptive color filtering method to identify possible candidates, then apply complex verification methods using spatio-temporal properties of segmented regions. In the following we present the components of the system, and evaluation results using tennis video as an example.

3.1 Color Based Adaptive Filtering

Color based filtering is applied to key frames of video shots. First, the global color models are built through training process based on clustering. The manually selected training data contain sample segments from different games with diverse conditions. Assume h_i, *i=1...,N* are color histograms of serve scenes in the training set. A k-means clustering is used to generate *K* models (i.e., clusters), $M_1,\ldots,M_K$, such that,

$$h_i \in M_j\ , \quad if\ \ D(h_i, M_j) = \min_{k=1}^{K}(D(h_i, M_k)) \qquad (1)$$

where $D(h_i, M_k)$ is the distance between h_i and the mean vector of M_k, i.e. $H_k = \frac{1}{|M_k|}\sum_{h_i \in M_k} h_i$ and $|M_k|$ is the number of training scenes belonging to model M_k. This means that for each model M_k, H_k is used as the representative feature vector. In order to accommodate diverse conditions (e.g., lighting, court surface, and viewing angles), we deliberately include a large number of color models in the training process, and then narrow down to a small set of models by adapting the model pool to the initial portion of the video from test data.

This raises a typical chicken-and-egg problem, as we need to know serve scenes in the new data in order to train a correct model. To solve the problem, we detect the first *L* (e.g., a small number like 8) serve scenes using all models (i.e. clusters), $M_1,\ldots,M_K$. That is to say, all models are used in the initial filtering process. If one shot is close enough to any model, the shot will be considered as a likely candidate and passed to subsequent verification processes (see 3.2 and 3.3).

$$h_i^{'} \in M_j, \quad if\ D(h_i^{'}, M_j) = \min_{k=1}^{K}(D(h_i^{'}, M_k)) \ \ and\ \ D(h_i^{'}, M_j) < TH \qquad (2)$$

where $h_i^{'}$ is the color histogram of the *i*th shot in the new video, and *TH* is a given filtering threshold to accept shots with sufficient color similarity. In our experiments (Section 3.4), the TH value is chosen based on the empirical distribution modes obtained from the training data. Shot *i* is detected as a serve scene if the subsequent object-level verification process (described in 3.2) is also successful. In this case, we mark this serve scene as being detected by model M_j (i.e., classify the scene into the model M_j). If the verification fails, $h_i^{'}$ is removed from the set of M_j.

After L serve scenes are detected, we find a matched model M_o, by searching for the model with the most serve scenes.

$$|M_o| = \max_{k=1}^{K}(|M_k|) \tag{3}$$

where $|M_k|$ is the number of incoming scenes being classified into the model M_k. We consider model M_o and a few (e.g., 2) of its closest neighboring models as representative models for the particular video programs under consideration.

The adaptive model selection process described above allows us to avoid the burden of developing customized models for different video sources (e.g., channels, courts). The same pool of models obtained from a single training process are applied in the initial stage when given a new video source. A matched model plus several of its minor variations are identified in the initial process and are then used for detecting target scenes in any future data from the same source.

3.2 Object Segmentation Based Verification

Color histograms are global features that can be computed and compared faster than real time. However, with the color feature only, our experiments showed a detection accuracy less than 80%. Many close-up scenes of the play field and replay scenes are likely to be detected as false positives. To improve detection accuracy, we extend our previous work [10],[11] on salient region and object segmentation to compute localized spatial-temporal features. Compared with global features, such as color histograms, spatial-temporal features are more reliable and effective in detecting given scene models. Especially in sports videos, special scenes often consist of multiple objects subject to spatio-temporal constraints (e.g., players in the server scene in tennis).

In [11], we have developed a foreground object segmentation system based on region merging, tracking, and background layer modeling. Here we extend such prior technique to detect and track objects in sports.

In regular unconstrained video, as depth information is not well preserved, many approaches have been proposed to use multiple 2D parametric models to estimate multiple motion layers. Many existing methods rely only on motion information in grouping image pixels or blocks into motion layers, and thus usually result in inaccurate segmentation on motion boundaries. As there is a strong inter-dependence between motion estimation and layer segmentation, poor segmentation in turn will degrade the accuracy of motion estimation.

To overcome these problems, applying our prior results in region segmentation and tracking, we have developed a two-stage moving objects detection method for sports video. This method uses regions with accurate boundaries to effectively improve motion estimation results. Furthermore, we explore the temporal constraint in a video shot to achieve more reliable object detection results. The two general stages of our algorithm are shown in Figure 5.

In the first stage, we apply an iterative motion layer detection process based on the estimation and merging of affine motion models. Each iteration generates one motion layer. The difference from existing methods is that motion models are estimated from spatially segmented color regions instead of individual pixels or blocks. In the second stage, temporal constraints are applied to detect moving objects in spatial and temporal space. Layers in individual frames are linked together based on consistency and motion projection of their underlying regions. One or more layers are declared as motion objects according to spatio-temporal consistency rules. More detailed descriptions are available in [11].

To achieve real-time performance, here segmentation is performed on the down-sampled images of the representative frames of each shot (e.g., first I frame in each shot). The down-sampling rate used in our experiment is 4, both horizontally and vertically, which results in images with size 88x60. Motion fields are estimated using the hierarchical approach as proposed in [2]. An example of segmentation and detection results is shown in Figure 6.

Figure 6 (b) shows the region segmentation result. The court is segmented as one large region, while the player closer to the camera is also extracted. The court lines are not preserved due to down-sampling. Black areas shown are tiny regions that are dropped at the end of segmentation process. Figure 6 (c) shows the final moving object detection result. In this example, the result is quite satisfactory, and only the desired player is detected. Sometimes a few background regions may also be detected as foreground moving object. We will address this issue by using the tracking algorithm discussed in Section 4. Here for verification purpose, as we will describe below, the important requirement is not to miss the player.

Given the segmented regions and motion foreground objects, we apply the following verification rules derived manually based on domain knowledge. First, there must be a large region (e.g. larger than two-thirds of the frame size) with consistent color (or intensity for simplicity). This large region corresponds to the tennis court. The uniformity of a region is measured by the intensity variance of all pixels within the region (Eq 4).

$$Var(p) = \frac{1}{N}\sum_{i=1}^{N}[I(p_i) - \bar{I}(p)]^2 \tag{4}$$

where N is the number of pixels within a region *p*. $I(p_i)$ is the intensity of pixel *i and* $\bar{I}(p)$ is the average intensity of region *p*. If *Var(p)* is less than a given threshold, the size of region *p* is examined to decide if it corresponds to the tennis court.

Secondly, the size and position of the foreground moving player are examined. The condition is satisfied if a moving object with certain size conditions is detected within the lower half part of the previously detected large region corresponding to "court". In a downsized 88x60 image, the size of a player is usually between 50 to 200 pixels. As our detection method is applied at the beginning of a serve, players usually stay near the bottom court line. Thus the position of a detected player has to be within the lower half part of the court.

3.3 Edge Based Verification

One unique characteristic of serving scenes in the tennis game is that there are horizontal and vertical court lines. Ideally if a camera is positioned at the rear-top point of the court and all court lines are captured (Figure 2), rules corresponding to a complete court can be used to verify serve scenes with high precision. However, in practice, due to camera panning, zooming, or object occlusion, usually not all court lines are viewable. Use of full-court conditions will result in a low recall rate of serve scene detection.

Since we already use color filtering and region-level verification processes, matching conditions of court lines are made relatively loose. An example of edge detection using the 5x5 Sobel operator is given in Figure 7.

Note that the edge detection is performed on a down-sampled image and inside the detected court region only. Hough transforms are conducted in four local windows (shown in Figure 8) to detection straight lines. Windows 1 and 2 are used to detect vertical court lines, while windows 3 and 4 are used to detect horizontal lines. It greatly increases the accuracy in detecting straight lines to use local windows instead of a whole frame. As shown in the figure, each pair of windows roughly cover a little bit more than half of a frame, and are positioned somewhat close to the bottom border. This is based on the observation about the typical court line locations within the court view.

The verifying condition is that there are at least two vertical court lines and two horizontal court lines being detected. Note these lines have to be apart from each other by a certain distance, as noisy edge detection and Hough transform may produce duplicate lines. This is based on the assumption that despite of camera panning, there is at least one side of the court, which has two vertical lines, being captured in the video. On the other hand, camera zooming will always keep two of three horizontal lines, i.e., the bottom line, middle court line and net line, in the view.

3.4 Experiments and Discussion

We applied the above filtering and verification scheme to tennis and baseball videos. Two different color model pools and sets of verification rules are trained and constructed respectively. The color models used in the first filtering stage include clusters extracted from short clips (a few minutes) from different sources (e.g., different channels or games). We keep the training samples diverse in order to build a comprehensive color model pool. When given a test video source, the best color model is identified based on the initial detection results. While view variation is a critical issue for color only approaches, our approach has incorporated region-based and edge based verification rules and thus is flexible in handling such situations. Considering there are only limited types of play fields, we can lower our filtering threshold to allow more false alarms in the first stage, and rely on the region/edge based verification process to improve overall precision.

We conducted two separate experiments. The first one use 1-hour test video, part of which is also present in the training set. Table 1 shows the results of this test. The overall recall and precision rates are very good – about 95%. In the second experiment, we use test data that does not have overlap with the training set. Table 2 includes the results, which are comparable to the training performance discussed above. This shows that our scheme is not sensitive to view variations caused by changes in colors, slight camera angles, lighting, and fields. Slightly more false alarms are detected in the baseball video, while no more misses are observed. This means our verification rules for baseball video are rather relaxed. It is possible to add more restrictions on object size, shape, position and other features. On the other hand, for the tennis video, there is no increase in the number of false positives. This is because the verification model matches serve scenes very well.

In a separate test, we test serve view detection using the object-feature verification model only, without the first stage of color filtering. Interestingly we obtained accuracies similar to those described above. This indicates that our object-feature verification model matches the tennis serve views very well, at least for the test sequences we have used. However, the global color filtering stage is needed for real-time processing purpose – it is capable of removing unlikely candidates quickly without needing to apply the fine verification procedures on every key frame.

These above results are very good compared to existing approaches using color matching only. Based on our experiments, previously proposed approaches using color histogram filtering can only achieve about 80% precision in order to obtain near 100% recall. Furthermore, despite of using advanced segmentation and feature extraction, our overall detection process is performed in real time.

4. Event Detection

Techniques described above detect canonical views in sports video. Such canonical views typically correspond to start of recurrent semantic units (e.g., serve in tennis) in sports video. In this section, we focus on detecting and summarizing what happened in a scene. Especially, we have adapted our moving object detection algorithm to track a tennis player in real-time, analyze the player's trajectory, and infer the actions and events associated with the player.

4.1 Player Tracking

In [11], we presented an automatic moving object detection method that consists of two stages: (1) an iterative motion layer detection step that is performed in individual frames, and (2) a temporal tracking process linking layers across frames and classify the linked layers into foreground moving objects vs. background. Here we adapt this approach to track tennis players within court view in real time. In our present system, we focus on the player who is close to the camera. The player at the far side is smaller and may not be inside the view. In order to achieve real-time performance, we apply the object segmentation methods on the down-sized images, thus make detection of small objects difficult.

For local motion layer detection, regions in down-sampled I- and P-frames are segmented and matched to extract motion layers. The reason to skip B-frames is because bi-direction predicted frames require more computation to decode. To ensure real-time performance, only one pair of anchor frames are processed every half second. For a MPEG stream with a GOP size of 15 frames, the I-frame and its immediate following P-frame are used. Motion layer detection is not performed in later P frames in the GOP. This change requires a different temporal detection process for detecting moving objects. We describe the enhancement below.

As half second is a rather large gap for the estimation of motion fields – motion-based region projection and tracking from I frame to another I frame are not reliable, especially when there are fast motions in a scene. Thus, a different process is required to match moving layers detected in individual I-frames. We use the following temporal filtering process to select and match objects that are detected in I frames.

Assume O_i^k is the k^{th} object (*k=1,…,K)* at the i^{th} I-frame in a video shot, $\vec{p}_i^k$, $\vec{c}_i^k$ and s_i^k are the center position, mean color and size of the object respectively. We define the distance between O_i^k and another object at j^{th} I-frame, O_j^l, as weighted sum of spatial, color and size differences.

$$D(O_i^k, O_j^l) = w_p \left\| \vec{p}_i^k - \vec{p}_j^l \right\| + w_c \left\| \vec{c}_i^k - \vec{c}_j^l \right\| + w_s \left| s_i^k - s_j^l \right| \tag{5}$$

where w_p, w_c and w_s are weights on spatial, color and size differences respectively. If $D(O_i^k, O_j^l)$ is smaller than a given threshold, *O_TH*, objects O_i^k and O_j^l match with each other. We then define the match between an object with its neighboring I-frame $i+\delta$ as follows,

$$F(O_i^k, i+\delta) = \begin{cases} 1 & \exists\, O_{i+\delta}^l,\, D(O_i^k, O_{i+\delta}^l) < O\_TH \\ 0 & otherwise \end{cases} \tag{6}$$

$where\ \delta = \pm 1, ..., n$. Let $M_i^k = \sum_{\delta=\pm 1,...,n} F(O_i^k, i+\delta)$ be the total number of frames that consist matches of object O_i^k *(k=1,...,K)* within the period $i-\delta$ to $i+\delta$, we select the object with maximum M_i^k. This means that if $M_i^r = \max_{k=1,...,K}(M_i^k)$, the r^{th} object is kept at the i^{th} I-frame. The other objects are dropped. The above process can be considered as a general temporal median filtering operation. Ideally, the preserved object corresponds to the most salient moving object, namely the tennis player in the near side.

After the above selection, we obtain the trajectory of the lower player by measuring the center coordinates of the selected moving objects in each I frame. Here some clarifications are due. First, if no object is found in a frame, linear interpolation is used to fill the missing point. When there are more than one objects being selected in a frame (in the situation when more than one objects have the same maximum number as defined above), the one that is spatially close to the object in the previous I frame is used. In addition, for speed reason, instead of using the affine motion model to compensate camera motion, here we use the detected net lines to roughly align different frames.

Experiment results of player tracking in one serve scene are shown in Figure 9. The first row shows down sampled frames. The second row contains player tracking results. The body of the player is well tracked and detected. Successful tracking of tennis players provides a foundation for high-level semantic event analysis. Compared with the tracking algorithm in [8], which computes residual errors to find moving objects and then searches players in pre-defined windows, our approach provides higher accuracy as well as real time performance.

4.2 Trajectory Analysis

The extracted trajectory is analyzed to obtain more detailed information about each play. Presently, we focus on two aspects. The first one is the position of the player. As players usually play at bottom lines, we want to find cases when a player moves to the net zone. The second one is to estimate the number of strikes by the player within a serve. Users who want to learn stroke skills or play strategies may be interested such information.

Given a trajectory containing *K* coordinates, $\vec{p}_k$ *(k=1,...,K),* at *K* successive I-frames, we first detect "still points" and "turning points". $\vec{p}_k$ is called a still point if,

$$\min(\|\vec{p}_k - \vec{p}_{k-1}\|, \|\vec{p}_k - \vec{p}_{k+1}\|) < TH \tag{7}$$

where *TH* is a pre-defined threshold. Furthermore, two consecutive still points are merged into one. If point $\vec{p}_k$ is not a still point, the angle at the point is computed. $\vec{p}_k$ is a turning point if

$$\angle(p_k p_{k-1}, p_k p_{k+1}) < 90^o \tag{8}$$

An example of object trajectory is shown in Figure 10. After detecting still and turning points, we use them to judge the player's positions. If there is a position close to the net line (vertically), the serve is classified as net-zone play. The estimated number of strokes is the sum of the numbers of turning and still points.

Experiment results of the one-hour test video are given in Table 3. In the video, the ground truth includes 12 serves with net play within about 90 serve scenes (see Table 1), and totally 221 strokes in all serves. Most net plays are correctly detected. False detection of net plays is mainly caused by incorrect extraction of player trajectories or court lines. Stroke detection has a precision rate about 72%. Beside the reason of incorrect player tracking, some errors are caused by limitations of our estimation model. First, at the end of a serve, a player may or may not strike the ball in his or her last move. Many serve scenes also show players walking in the field after the play. In addition, a sever scene sometimes contains two serves if the first serve failed. These may cause problems since currently we detect strokes based on the movement information of the player. To solve these issues, more detailed analysis of motion such as speed, direction, repeating patterns in combination with audio analysis (e.g., hitting sound) or ball tracking will be useful.

5. Summarization and Browsing

As we have observed, video data contain large amount of visual and semantic information. Even after content indices are generated, how to show these indices in a limited-size display is a challenging issue. In this section, we present a system for video browsing and summarization using the structure parsing and event detection results presented earlier. The system has two unique access methods for users to find desired video shots.

First, the system provides a summarization interface (Figure 11a). It shows the statistics of video shots, divided into categories of long, intermediate and short shots. It also shows the number of canonical scenes in a specific domain. For instance, in tennis, these are serve, net-zone play, or commercial. Seeing these summaries, users may follow up with more specific request by choosing a category (e.g., serve). As shown in Figure 11b, users can choose to go to any interesting scene categories directly.

The second interface combines the sequential temporal order and the hierarchical structure among video shots within a video. As shown in Figure 12, the structure tree is listed in the left window. In this example, games are listed at the top level. There are commercial breaks between games. Under each game, there are many serves. Each serve contains the serving shots and a few follow-up shots.

In the video shown in Figure 12, the first game includes 16 serves. Each serve segment is labeled with the length of the segment, type of play in this serve and the approximate number of strokes in a serve. For example, a label "(L) S B 4" means a long segment, server, base-line play and approximately 4 strokes.

All these elements are organized as nodes in a tree. This allows users to easily navigate from top summary levels to detailed levels. When users click on any nodes, the corresponding key frame will be shown in the right window, and users can start to play the video at the corresponding moment.

6. Conclusion and Open Issues

In this paper, we present a general framework for structure parsing and high-level event detection for sports videos, using tennis and baseball as examples. Our system combines generic techniques (for feature extraction, object detection/tracking, clustering) and domain-specific methods (for view recognition and player event detection). Real time processing performance is achieved by exploring feature extraction and matching in the compressed domain. Our experiments have demonstrated high accuracy in sports domains such as tennis and baseball.

Detection of the structures and events in sports video facilitates development and deployment of new video applications, such as highlight generation for long programs, live filtering of important events etc.

Under the framework, there are many issues that can be further explored to produce a accurate comprehensive summary of sports videos. For tennis videos, we can include audio information to detect the number of strokes within each serve more accurately. Caption text showing the scores is common in broadcast sports programs. By detecting these text boxes and recognizing scores, we can recognize the status of a game. Readers are referred to [16] for an effective system for sports event summarization by video text recognition.

References

[1] S.-F. Chang, W. Chen, H. Meng, H. Sundaram, and D. Zhong, "VideoQ: An Automated Content-Based Video Search System Using Visual Cues", ACM 5th Multimedia Conference, Seattle, WA, Nov. 1997.

[2] M. Bierling, "Displacement Estimation by Hierarchical Block Matching", SPIE Vol 1001, Visual Communication & Image Processing, 1988.

[3] A. Hanjalic, R.L. Lagendijk, J. Biemond, “Automated High-Level Movie Segmentation for Advanced Video Retrieval Systems”, IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.4, June 1999.

[4] H. Sundaram and S.-F. Chang, Determining Computable Scenes in Films and their Structures using Audio Visual Memory Models, ACM Multimedia 2000, Los Angeles, CA, Oct 30-Nov 3, 2000.

[5] R. Lienhart, C. Kuhmunch and W. Effelsberg, “On the Detection and Recognition of Television Commercials”, Proc. of IEEE International Conference on Multimedia Computing and Systems, June, 1997, Ottawa, Canada.

[6] T. Sato, T Kanade, E. K. Hughes, M. A.Smith, “Video OCR for Digital News Archives”, Proc. Of the 1998 International Workshop on Content-based Access of Image and Video Database, January 3, 1998 Bombay, India.

[7] D.D. Saur, Y.-P. Tan, S.R. Kulkarni, and P.J. Ramadge, “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, San Jose, CA.

[8] 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.

[9] H. Zhang, SY Tan, SW Smoliar, and G. Yihong, “Automatic Parsing and Indexing of News Video”, ACM/Springer-Verlag Journal of Multimedia Systems, 2 (6), pp. 256-266, 1995.

[10] D. Zhong and S.-F.Chang, "Video Object Model and Segmentation for Content-Based Video Indexing", IEEE International Symposium on Circuits and Systems, Hong Kong, June 9-12, 1997.

[11] D. Zhong and S.-F. Chang, "Long-Term Moving Object Segmentation and Tracking Using Spatio-Temporal Consistency”, IEEE International Conference on Image Processing, Thessaloniki, Greece, October 7-10, 2001.

[12] D. Zhong and S.-F. Chang, Structure Analysis of Sports Video Using Domain Models, IEEE Conference on Multimedia and Exhibition, Tokyo, Japan, Aug. 22-25, 2001.

[13] Y. Rui, A. Gupta, and A. Acero, "Automatically Extracting Highlights for TV Baseball Programs," the 8th ACM International Conference on Multimedia, Oct. 2000, Marina Del Rey, CA.

[14] Y. Gong, L.T. Sin, C. Chuan, H. Zhang, and M. Sakauchi, ““Automatic parsing of TV soccer programs”, *Proc. ICMCS’95*, Washington D.C, May, 1995

[15] A. Jaimes and S.-F. Chang, “Integrating Multiple Classifiers in Visual Object Detectors Learned from User Input,” 4th Asian Conference on Computer Vision (ACCV 2000), Taipei, Taiwan, January 8-11, 2000.

[16] S.-F. Chang, D. Zhong, and R. Kumar, Real-Time Content-Based Adaptive Streaming of Sports Video, IEEE CVPR Workshop on Content-Based Access to Video/Image Library, Hawaii, Dec. 2001.

[17] D. Zhang and S.-F. Chang, “Event Detection in Baseball Video Using Superimposed Caption Recognition”, ACM Multimedia 2002, Juan Les Pins, France, December 1-6, 2002. (ACM MM 2002).

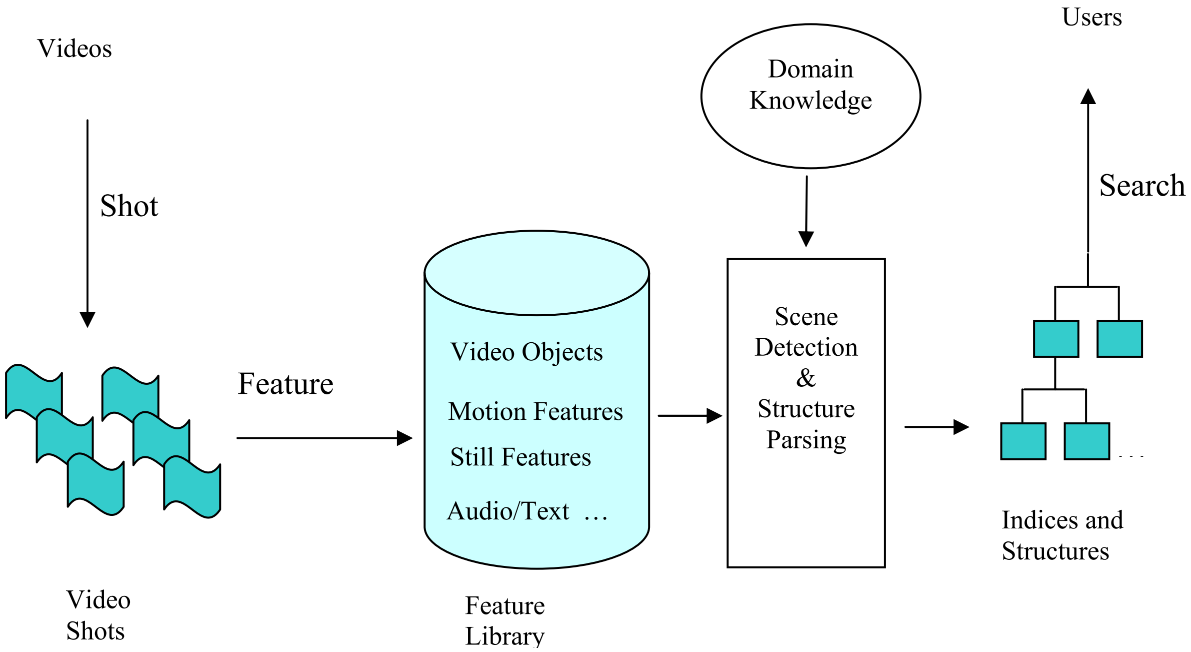


Figure 1. A typical content based video indexing and retrieval scenario

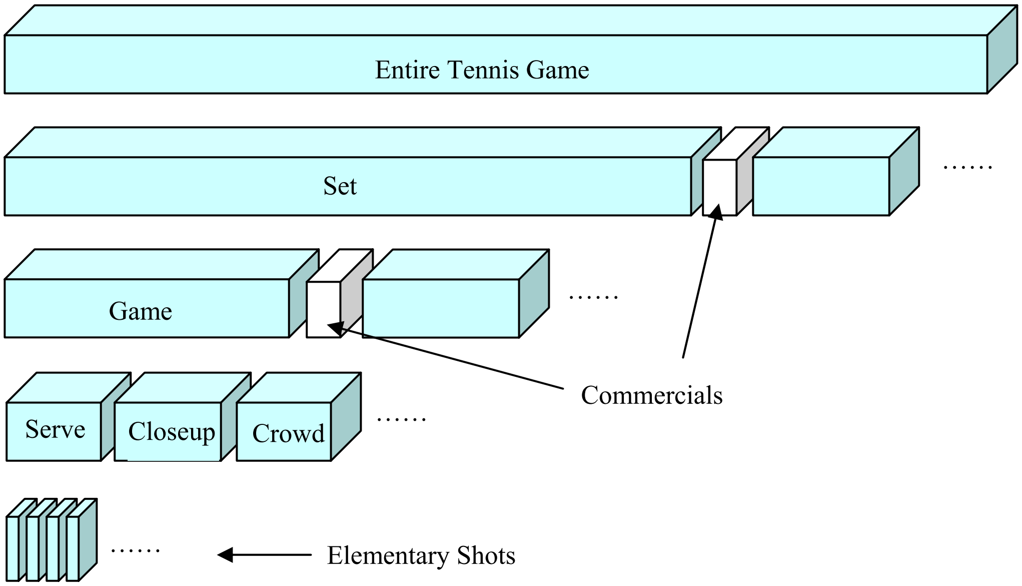


Figure 2. The content structure of sports video programs (example: tennis)

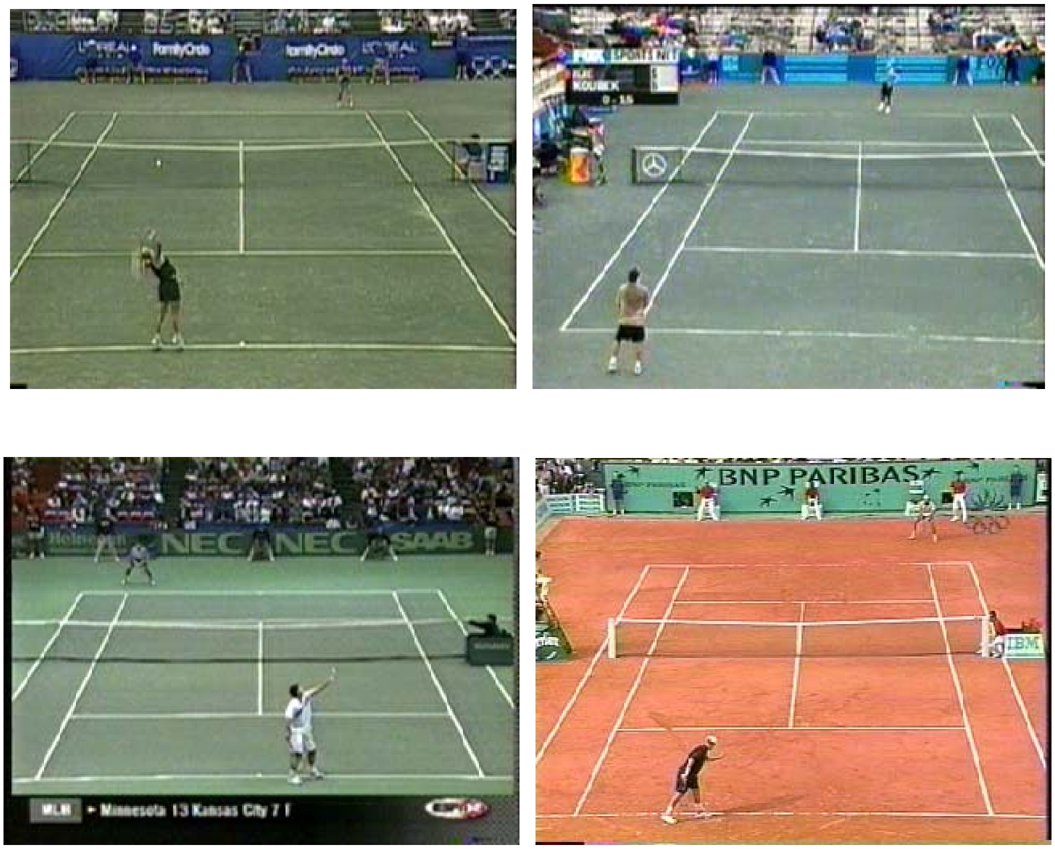


Figure 3. Serving scenes from four different tennis games

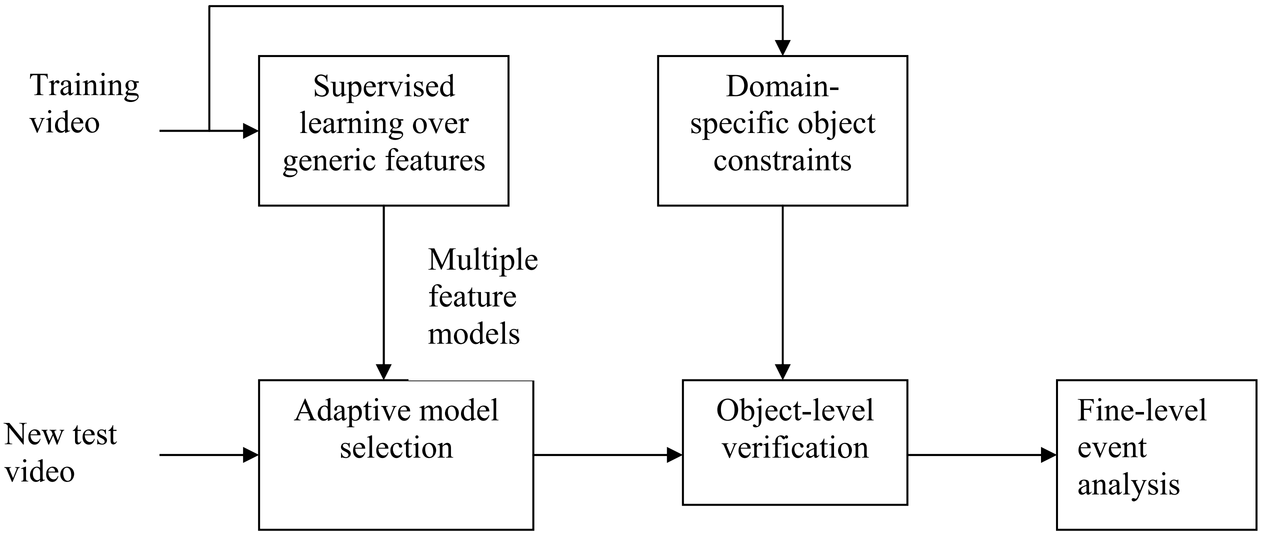

Training video
Supervised learning over generic features
Domain-specific object constraints
Multiple feature models
New test video
Adaptive model selection
Object-level verification
Fine-level event analysis

Figure 4. system architecture for sports video structure analysis and event detection

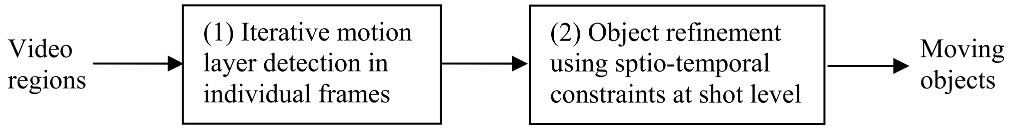
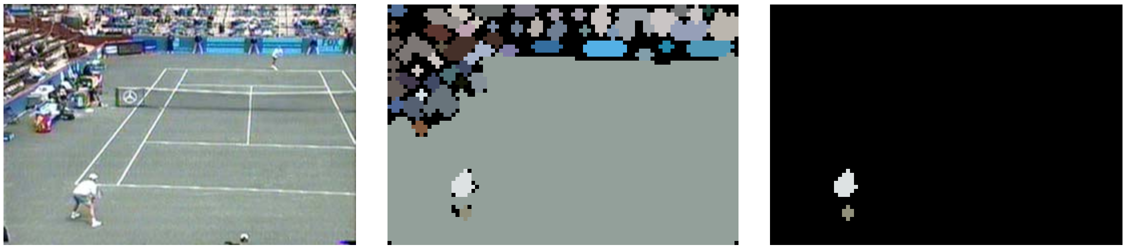


Figure 5. Two-stage moving object detection based on region segmentation and tracking results



(a) original key frame (b) region segmentation (c) moving object

Figure 6. An example of automatic region segmentation and moving object detection (note the camera may be moving)

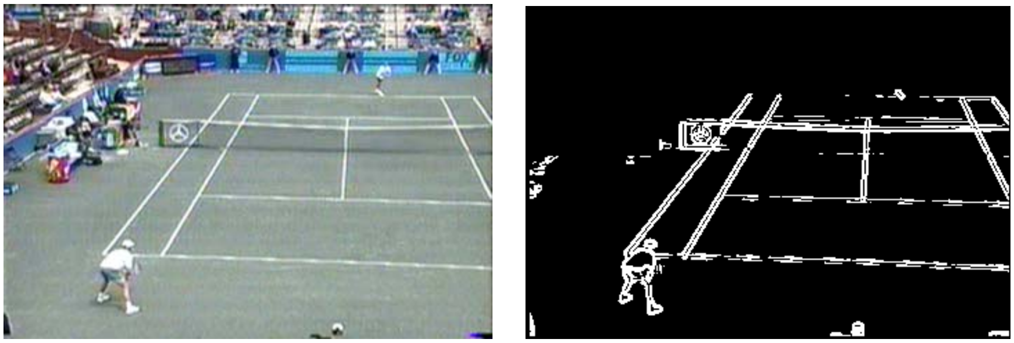


Figure 7. Edge detection within the detected court region

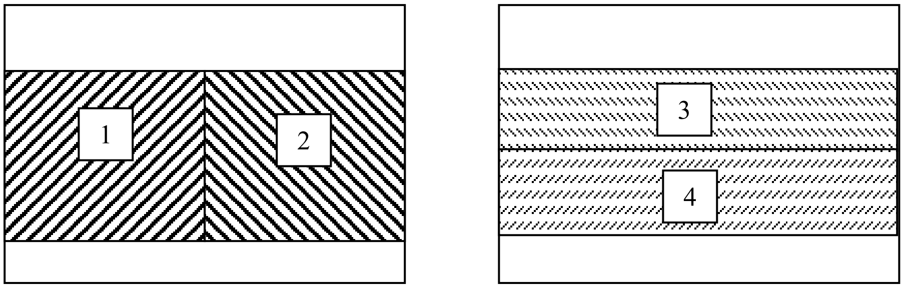


Figure 8. Applying Hough-transform based line detection within specific areas

Table 1 View detection results (test video overlaps with training set)

	Ground truth	# of Miss	# of False
Tennis (serve)	89	7	2
Baseball (pitch)	93	3	4

Table 2 View detection results (no overlap between training and test data)

	Ground truth	# of Miss	# of False
Tennis (serve)	74	6	1
Baseball (pitch)	57	1	5

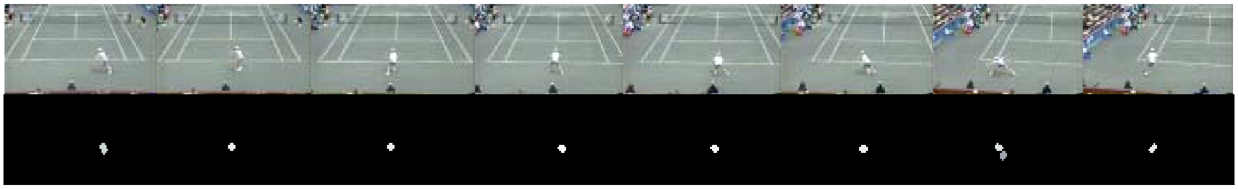


Figure 9. Tennis player tracking within a serve scene

(results are shown on 8 I-frames with 14 frames between every two I farmes)

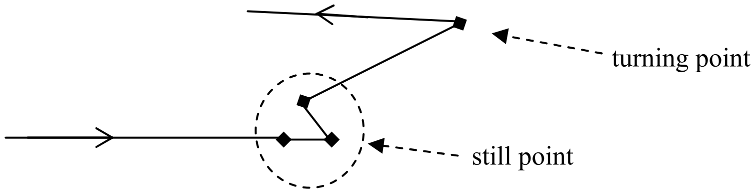
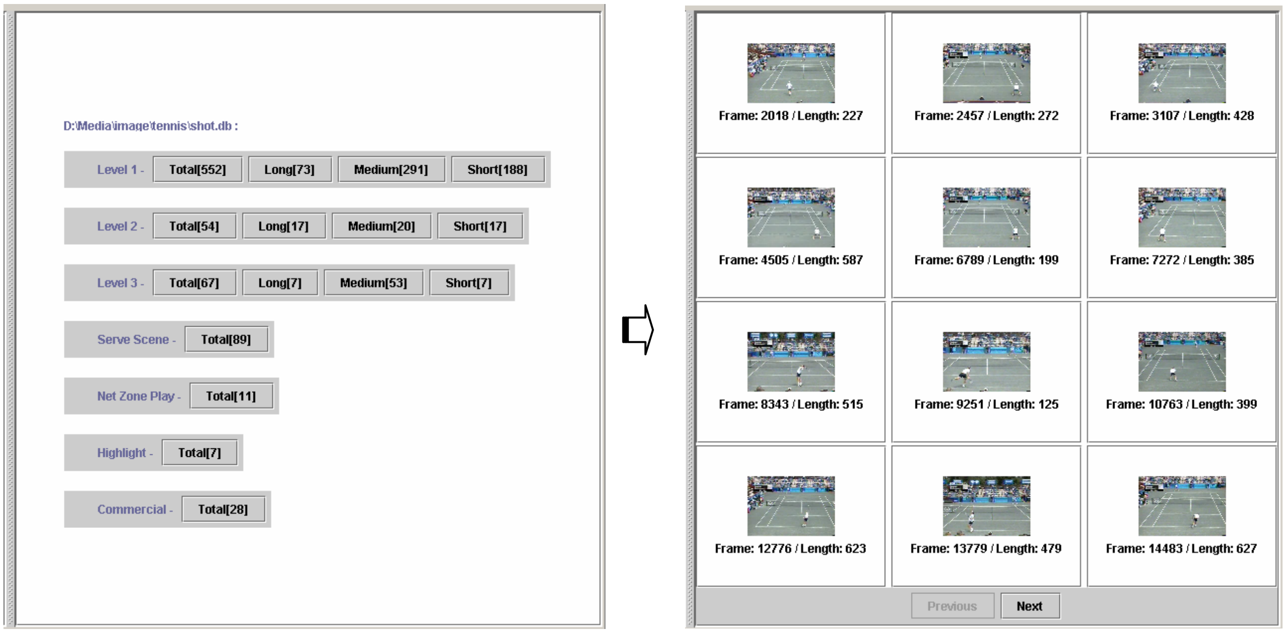


Figure 10. Detection of still and turning points in object trajectory

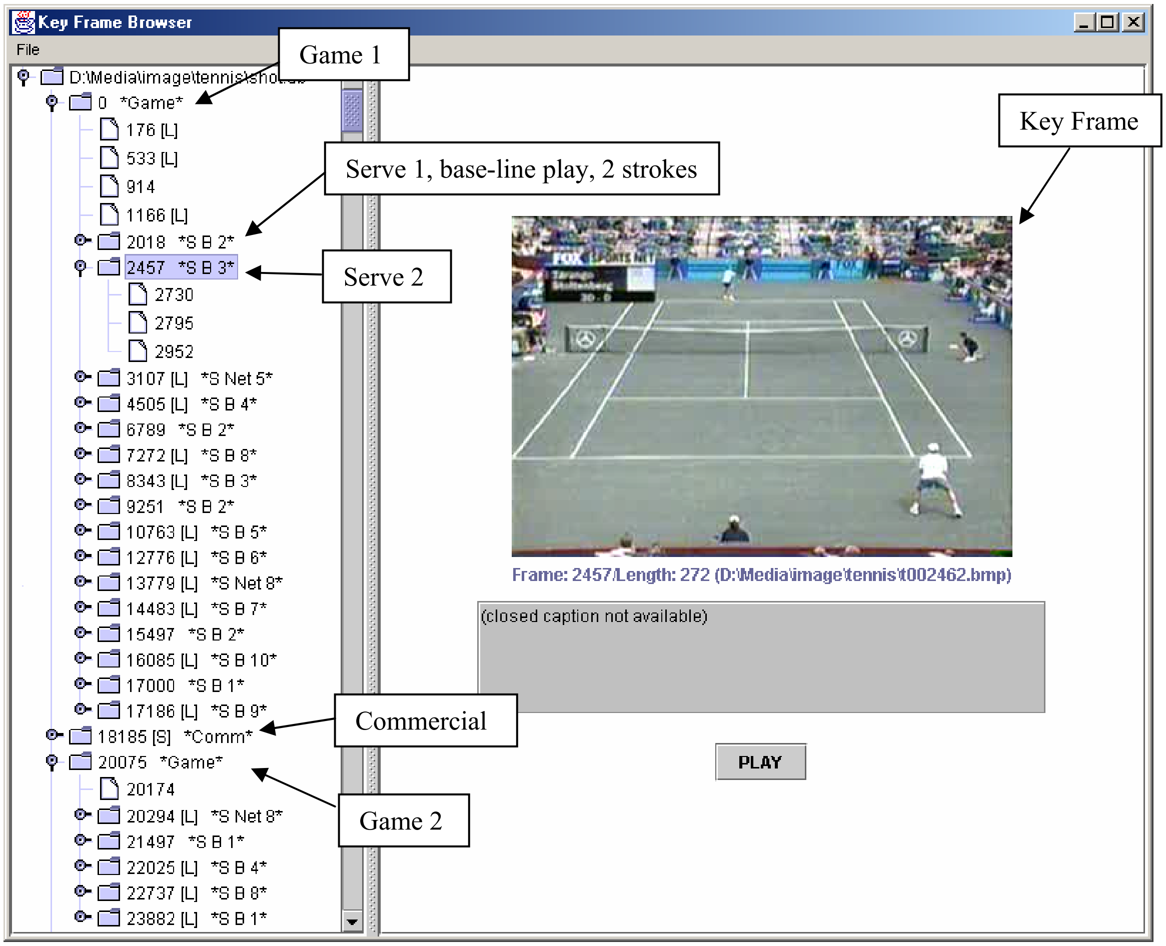
Table 3 Trajectory analysis results for one hour tennis video

	# of Net Plays	# of Strokes
Ground Truth	12	221
Correct Detection	11	216
False Detection	7	81


D:\Media\image\tennis\shot.db :
Level 1 - Total[552] Long[73] Medium[291] Short[188]
Level 2 - Total[54] Long[17] Medium[20] Short[17]
Level 3 - Total[67] Long[7] Medium[53] Short[7]
Serve Scene - Total[89]
Net Zone Play - Total[11]
Highlight - Total[7]
Commercial - Total[28]
Frame: 2018 / Length: 227
Frame: 2457 / Length: 272
Frame: 3107 / Length: 428
Frame: 4505 / Length: 587
Frame: 6789 / Length: 199
Frame: 7272 / Length: 385
Frame: 8343 / Length: 515
Frame: 9251 / Length: 125
Frame: 10763 / Length: 399
Frame: 12776 / Length: 623
Frame: 13779 / Length: 479
Frame: 14483 / Length: 627
Previous
Next

(a) (b)

Figure 11. Summarization interface providing scene index to video



```
D:\Media\image\tennis\shot
    0  *Game*
        176 [L]
        533 [L]
        914
        1166 [L]
        2018  *S B 2*
        2457  *S B 3*
            2730
            2795
            2952
        3107 [L]  *S Net 5*
        4505 [L]  *S B 4*
        6789  *S B 2*
        7272 [L]  *S B 8*
        8343 [L]  *S B 3*
        9251  *S B 2*
        10763 [L]  *S B 5*
        12776 [L]  *S B 6*
        13779 [L]  *S Net 8*
        14483 [L]  *S B 7*
        15497  *S B 2*
        16085 [L]  *S B 10*
        17000  *S B 1*
        17186 [L]  *S B 9*
    18185 [S]  *Comm*
    20075  *Game*
        20174
        20294 [L]  *S Net 8*
        21497  *S B 1*
        22025 [L]  *S B 4*
        22737 [L]  *S B 8*
        23882 [L]  *S B 1*
```

Frame: 2457/Length: 272 (D:\Media\image\tennis\t002462.bmp)

(closed caption not available)

Figure 12. Hierarchical Browsing Interface for Parsed Structures in Tennis Video