Deterministic Approach to Content Structure Analysis of Tennis Video

Viachaslau Parshyn, Liming Chen

A Research Report, Lab. LIRIS, Ecole Centrale de Lyon

LYON 2006

Abstract. An approach to automatic tennis video segmentation is proposed. The aim is temporal decomposition of a tennis match according to its hierarchical semantic content structure which could be used to organize an efficient content based access. The approach relies on some particular characteristics and production rules that are typically employed to convey semantic information to a viewer, such as specific views and score boards. We propose quite a general framework which can be considered as a kind of a final state machine whose states relate to content units. It allows us to directly encode our notion of a tennis content structure through selection of intermediate event patterns governing transitions from one semantic segment to another. Advantage of our approach is in its expressiveness and low computational complexity. Experimental evaluations on ground-truth video were made that showed quite high segmentation accuracy.

1 Introduction

Sports video is chosen as being one of the most popular types of the TV broadcasting that appeals large audience. Nowadays, however, we often cannot permit ourselves to spend hours on watching full-time long games such as tennis matches. Moreover, some people might find it boring to watch all the video and they are interested only in the most impressive scenes. This is especially the case if one just wants to refresh in memory some episodes of an already seen game record. As it is difficult to quickly localize an interesting scene in a long video using ordinary media playing tools which provide simple functions like a forward/backward rewind, there is an evident need to provide convenient means of effective navigation. Sports video has usually a well-defined temporal content structure which could be used to efficiently organize a content-based access that allows for such functions as browsing and searching, as well as filtering interesting segments to make compact summaries. As for a tennis match, it can be represented, for example, according to its logic structure as a sequence of sets that in their turn are decomposed into games etc. In this report we propose an approach to automatic tennis video parsing that yields a temporal decomposition of a given video into such a hierarchical content structure.

To detect regular content units of video we rely on some particular characteristics and production rules that are typically employed to convey semantic information to a viewer. A tennis match, like a lot of other sports games, is usually shot by a number of fixed cameras that yield unique views during each segment. For example, a serve typically begins with switching of the camera into a global court view (see Figure 1). Since a tennis match occurs in a specific playground, this view can be detected based on its unique characteristics (we employ its color homogeneity property). In order to constantly keep the audience informed about the current

game state, score or statistics boards are regularly inserted into the broadcast according to the rules of the game. In our content parsing technique these inserts are detected and used as indicators of transitions between semantic segments. We propose quite a general framework which can be considered as a kind of a final state machine whose states relate to content units. It receives at the input a time-ordered sequence of instantaneous events like the beginning of a global view shot and processes it recursively according to pre-defined grammar rules. Some of these events, such as score board appearances are used as transition indicators while others allows for exact positioning of segment boundaries.

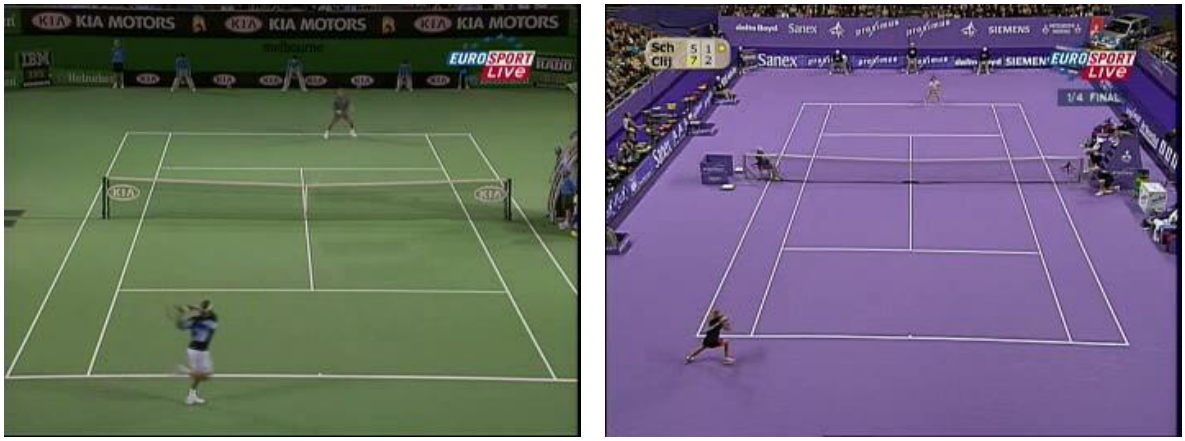

Figure 1. Global court views in tennis match

The report is organized as follows. In the next section we present a general scheme of our parsing system, define tennis content structure and give a detailed description of the parsing technique. After this we describe algorithms developed for automatic detection of the relevant events. In the next section the results of experimental evaluations are presented and discussed. In the section “Application: Tennis Analyzer” we describe our software realization of the proposed segmentation approach for the purpose of automatic content table generation and browsing of tennis video. Final conclusions then finish up the report.

2 Segmentation Framework

2.1 Semantic Structure of Video

We define a content table of video hierarchically as a sequence of nested temporal segments which are contiguous at each semantic level. Different content structures can be usually proposed depending on the needs of a user. An example of two configurations for tennis video is presented in Figure 2. It shows segment types allowed at each semantic level; segments of a higher level can comprise segments of several types in the lower level. The first configuration corresponds to the logical structure of a tennis match. According to this structure the match is decomposed into sets separated by breaks at the second semantic level; each set is divided into games and breaks at the third level etc. The second configuration

just separates the scenes of tennis rallies (“play”) from the rest parts of the video (“break”). Such more simple decomposition allows for building compact summaries consisting only of playing parts and can be used to reduce the duration of the video and the bandwidth for resource limited devices [CHA 01].

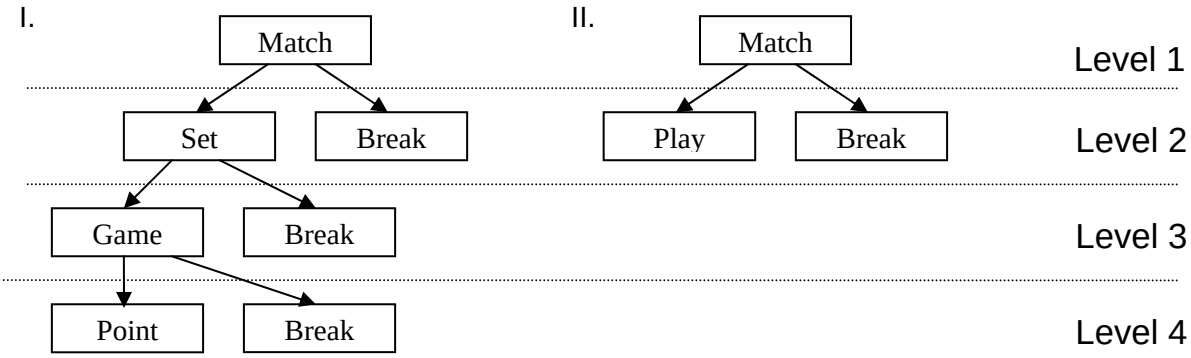

Figure 2. Two samples of a tennis video content structure.

2.2 Segmentation Principles

If we asked a person to segment video records of the same type, he would need to make up a decision concerning two interrelated problems. First, a desirable semantic structure has to be defined: how many levels of details and what segments can be included at each level. Two possible semantic structures for the tennis video are described above (see Figure 2). Second, a set of rules has to be clearly stated that are to be followed in segmenting. If the segmentation is performed only intuitively, without clear understanding of underlying principles, it will be subjective and unstable. The segmentation rules can be usually formulated as events or their combinations which signify transition between semantic segments. It is often the case when these events are suggested by the production principles, which is not surprising as these principles are based on the predefined semantic intention of the producer. For example, the beginning of a game in a tennis match could be recognized by a corresponding score board appearance or by switching to the court view after a pause and change of the serving player.

In order to segment video automatically we state the rules of transition between semantic segments explicitly at each semantic level as combinations or templates of primitive events that can be detected automatically. These templates are defined as sets of events satisfying some temporal constraints. As it was shown by Allen [ALL 83], thirteen relationships are sufficient to describe the relationship between any two intervals: *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, *equals* and their inverses. Additionally we determine relationship “*precedes*” between two point events $s1$ and $s2$ belonging to detectable classes of events $c1$ and $c2$, saying that $s1$ *precedes* $s2$ if $s1$ occurs before $s2$ and there is no other events of type $c1$ and $c2$ between them. Templates can be defined hierarchically so that templates of a higher level are composed from

templates of the lower level or primitive events. The templates that determine the transition between semantic segments are referenced hereafter as transition templates. In the general case these templates depend on segment types. Hereafter we suppose that they are dependent only on the type of the segment to which the transition occurs; in the other words each transitional template determines the beginning of the corresponding semantic segment.

To decompose tennis video according to the semantic structure presented in Figure 2.I, we propose the following definitions of the semantic segments and the corresponding event templates. Let's suppose that the set of detectable primitive events consists of global court view (denoted as *GCV*) shots (see Figure 1), rocket hits (*RH*) sounds and specific score or statistics boards of three types inserted by the producer between tennis points, games and sets respectively. At first we determine the template for the event of tennis serve or rally. When a serve/rally begins, a switch occurs to the camera providing global court view. When it finishes, the view is change so as to show, for instance, players' close-ups or the audience. So, in the simplest case a serve/rally can be defined as a global court view shot. In order to distinguish a serve/rally from replay shots which correspond sometimes to the same or similar view, rocket hits event can be additionally used. In this case a serve/rally (*SR*) event is defined as a template of two primitive events *GCV* and *RH* related as *RH during GCV*, since rocket hits are not heard during replays. Let's consider the segmentation at semantic level 2 (level of sets) in Figure 2.I. We imply that a tennis set begins with its first serve/rally. Therefore the corresponding transition template is the beginning of event *SR*, denoted as *SR.begin* (the beginning of a template is defined as the earliest beginning of its constituent events/templates; the end is defined similarly). A unique score/statistics board is usually inserted a little time after the end of a set which is defined as the end of the last rally. Hence, we detect the beginning of a break as the end of a serve/rally event which *precedes* the beginning of the corresponding score board for sets (*SBS*), i.e. the template is written as {*SR.end precedes SBS.begin*} (the first event in this case is used to precise the beginning time of the break segment). Sometimes score boards stands on the screen all the playing time. In this case transitions to break segments could correspond to the changes of the printed score. The semantic segments and the corresponding templates for semantic level 3 and 4 (level of games and points) are defined in a similar way.

Note that the defined above templates are easily detectable with a computationally effective procedure. If the beginning and the end of detected events or lower-level templates are ordered in time and thus form an input sequence of instantaneous events, these templates can be recognized in one path using state variables for event tracking. For example, the *during* relation of score/rally event is easily checked at the end of a global court view by verifying that the

beginning and the end of the rocket hits segment (if they exist) are between the beginning and the end of the global court view.

The general scheme of our parsing system is shown in Figure 3. First, relevant semantic events are detected from visual and audio sequences of an input video: score boards, global court views and rocket hits segments. These events are then looked for to distinguish transitional templates that are fed as the input to the content parser. Generally there are some constraints on possible chains of segments at each semantic level that are given by the corresponding grammar. In our case bi-grammars are employed that are sets of allowable transitions between two contiguous segments. A content table is finally generated by the content parser governed by the sequence of transition templates and by predefined grammar constraints.

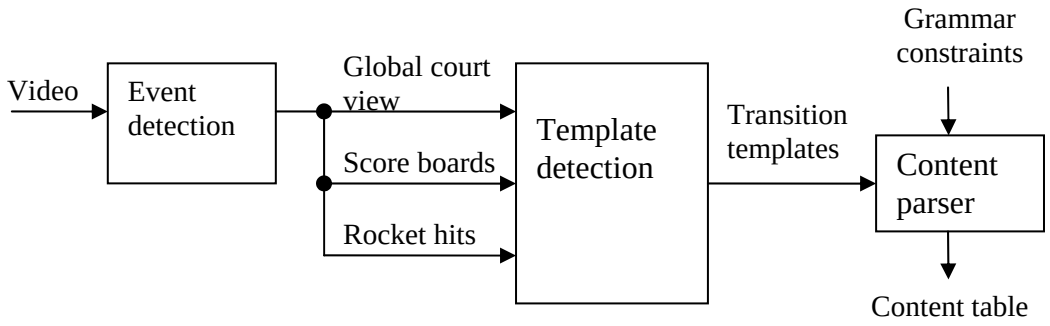

Figure 3. Parsing chain.

2.3 Segmentation Algorithm

The output content table is generated by a state machine whose states correspond to the appropriate semantic segments. The multilevel content structure of video is generated recursively, beginning at the highest semantic level. At each semantic level the parsing is driven by its grammar that imposes state transition constraints and transition template detectors that control the transition from one state to another. The corresponding parsing rules developed for the content structure of Figure 2.I are given in Table 1, Table 2 and Table 3. Column "Transition template" corresponds to the beginning of a state listed in the first column of the tables. The transition time specifies the precise transition moment for the corresponding template. In the general case it is supposed that the initial segment of a given video is unknown. That is why the state machine starts from initial undefined state at the second semantic level. For the lower semantic levels the initial machine state is chosen according to column "Initial state of the sublevel". Our recursive parsing algorithm for a given semantic level is the following:

- Detect transition templates from primary events.
- For each transition template extracted in the time order do:

- Check whether the template corresponds to an allowed next machine state. If so, do:
  - If the semantic segment corresponding to the current machine state has to be further decomposed into the segments of the lower level, initialize the current state for that level accordingly and perform the parsing recursion for that segment.
  - Go to the next machine state according to the detected pattern.

- For the remaining semantic segment corresponding to the current machine state: if it has to be further decomposed into the segments of the lower level, perform the parsing recursion for this segment.

As it was mentioned above, transition templates can be detected from an input sequence of time ordered point events in one pass. Therefore the two first steps of the algorithm can be merged into one step performed in one pass as well.

State	Allowable next states	Initial state of the sublevel	Transition template	Transition time
Initial undefined	Set	-	-	-
Set	Break	Game	*SR.begin*	*SR.begin*
Break	Set	-	*SR.end precedes SBS.begin*	*SR.end*

Table 1. Parsing rules for semantic level 2 (of tennis sets).

State	Allowable next states	Initial state of the sublevel	Transition template	Time adjustment event
Game	Break	Point	*SR.begin*	*SR.begin*
Break	Game	-	*SR.end precedes SBG.begin*	*SR.end*

Table 2. Parsing rules for semantic level 3 (of tennis games).

State	Allowable next states	Initial state of the sublevel	Transition template	Time adjustment event
Point	Break	-	*SR.begin*	*SR.begin*
Break	Point	-	*SR.end precedes SBP.begin*	*SR.end*

Table 3. Parsing rules for semantic level 4 (of tennis points).

3 Event Detection

Our scheme of the automatic tennis video parsing requires a proper choose of events detected in the raw visual and audio streams at the preprocessing stage. The following is a description of algorithms developed for automatic detection of global court views and score boards.

3.1 Global Court View

Tennis video like a lot of other types of sport video is usually shot by a fixed number of cameras that give unique views for game segments. A transition from one such view to another is sometimes an important indicator of semantic scene change. In tennis video a transition to a global court view that shows the whole field area with the players commonly signifies that a point starts and a rally begins. When the rally finishes, a transition to another view such as a player close-up or the audience usually happens. Thus, court view recognition is important for rallies scenes detection.

The first step in the detection of a specific view is segmentation of the video into views taken by a single camera or, in the other words, segmentation into shots. Color histogram difference between consecutive frames is applied in order to detect shot transitions. We use 64-bins histograms for each 3 components of the RGB-color space and concatenate them into one 192-dimensional vector. The difference between histograms of two consecutive frames is given by the dissimilarity analogue of the cosine measure:

$$D(H_i, H_j) = 1 - \frac{\sum_k H_i(k) * H_j(k)}{\sqrt{\sum_k [H_i(k)]^2 * \sum_k [H_j(k)]^2}}, \quad (1)$$

where $H_i(k)$ indicates *k*-th bin of the color histogram of frame *i*.

A simple shot detection algorithm puts a shot boundary at a frame for which the difference climbs above some threshold value. It is suitable for abrupt shot transitions that yield strong maxima of the difference value. However, in order to detect gradual transitions we need to set a low threshold value that would lead to unacceptable level of false alarms caused by fast camera motion or a change in lighting conditions. That is why we use a twin-threshold algorithm capable to reliably detect both type of shot transition [DON 01]. Abrupt shot transitions (hard cuts) are detected using a higher threshold $T1$ applied to the histogram difference between two consecutive frames. In order to find a gradual shot boundary, a lower threshold $T2$ is used. If this threshold is exceeded, the cumulative difference is calculated and compared with the threshold $T1$.

In order to exclude false positives of the shot detection algorithm caused by flashlights, additional check is made for abrupt transitions. A flashlight usually changes the color histogram considerably for one or several frames, while the frames that follow right after the flashlight resemble the frames that are before it. We compare the frames lying to the left and to the right of a potential abrupt shot transition within a window T by computing the following value:

$$D_{flash}(t) = \min_{t-T \le i < t, t < j \le t+T} D(H_i, H_j), \tag{2}$$

where t – the time index of the potential shot transition, inter-frame difference D is defined according to expression (1). If this value is below a threshold, the shot transition is rejected. We also merge the shot boundaries that are too close to each other (they are usually generated when a gradual shot transition occurs) in order to exclude very short or false shots.

Color distribution of global court view shots does not change much during the tennis match. This allows us to detect them based on their comparison with sample frames of the court view that are selected manually at the learning stage. A shots is recognized as a court view if it is close enough (in the sense of the color histogram difference defined by the expression (1)) to the appropriate sample view. Only homogeneous regions of the tennis field are taken from the learning frames in order to exclude players' figures and outliers. Several court samples and the corresponding rectangular tennis field areas selected at the learning stage of experimental evaluations are shown in Figure 4. Each learning sample is selected only once for a game or a series of games played at the same court (e.g. during the same championship).

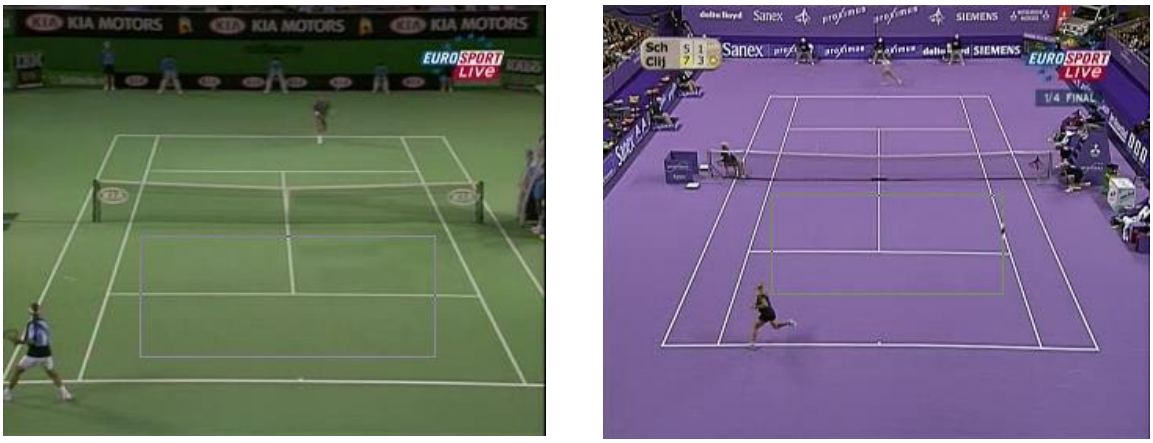

Figure 4. Global court view samples where the rectangular regions bounds learning areas.

In tennis video there are usually several types of shots that contain a big part of the tennis field at the background and, thus, resemble much the global court views. An example of such shots is players' close-up views; one such a view is shown in Figure 5 along with a court view sample. However, the court views usually take a longer part of the tennis video. Hence, we can enhance the robustness of the court view detection by grouping the shots into similarity clusters

and, then, rejecting rare clusters. Let each cluster *i* be represented by its color histogram (which is an average histogram for all the shots of the cluster) H_i and the number of its shots M_i. In order to describe our clustering algorithm, denote the set of all the clusters as *C* and the total number of clusters - as *N*. Then the algorithm can be written as the following.

- Initialize *C* as an empty set.
- For each shot of the given tennis video do:
  - Calculate a mean histogram of the shot H_{shot}.
  - Find the number *k* of the cluster closest to the shot as $k = \arg\min_{i=1,\ldots,N} D(H_{shot}, H_i)$, where *D(.)* is the difference measure between the histograms defined by (1).
  - If the distance $D(H_{shot},H_k)$ is less than the threshold *t1*, then set $M_k = M_k + 1$ and $H_k = \frac{M_k - 1}{M_k} H_k + \frac{1}{M_k} H_{shot}$. Else create a new cluster *N+1* that contains one shot and has the histogram H_{shot}, set *N=N+1*.
- Merge clusters that are close enough to each other.

  So, we can resume the global court view detection algorithm as the following.
- Segment the tennis video into the shots.
- Combine visually similar shots into the clusters.
- Calculate the time duration of each cluster for the whole video; exclude from the further consideration the clusters that last less then a predefined fraction (0.2 in this report) of the maximally long cluster.
- Recognize as court views the shots that belong to the cluster closest to the learning court view frames.

Figure 5. Player's close-up and court view sample frames that have similar color distributions.

3.2 Score Board Detection

As reflecting the state of the game, score boards could provide useful information for tennis video parsing into its logical structure (shown in Figure 2.I). Since these boards are inserted regularly according to the game rules, the mere facts of their appearance/disappearance can be used as reliable indicators of the semantic segment boundaries. Moreover, they present important information about the game and, hence, we can choose the appropriate frames as the key frames of the corresponding semantic units and thus provide convenient visual interface for browsing through the content table.

The same tennis video usually has several types of score boards that can be used to separate the segments at different levels of the semantic hierarchy. Score boards of the same type have the fixed positions on the screen and similar color bitmaps near their boundaries. The only difference between them lies in their textual content, the horizontal size (which is changed so as to hold all required data) and somewhat in their color (caused by the partial transparency). Several sample frames which contain score boards along with their bounding rectangle are shown in Figure 6 and Figure 7. We detect score boards, if we find horizontal lines of enough length placed near their upper and bottom borders. The Hough transform [HOU 59] is applied to edge points in order to detect the lines. The positions of the score boards borders are given manually during the learning – a user selects from sample tennis video the frames that contain required score tables and picks out their bounding rectangle (see Figure 6 and Figure 7). In order to enhance the robustness of detection results, smoothing is used – score boards scenes are pronounced only when the corresponding boards are detected in several frames during a period of time.

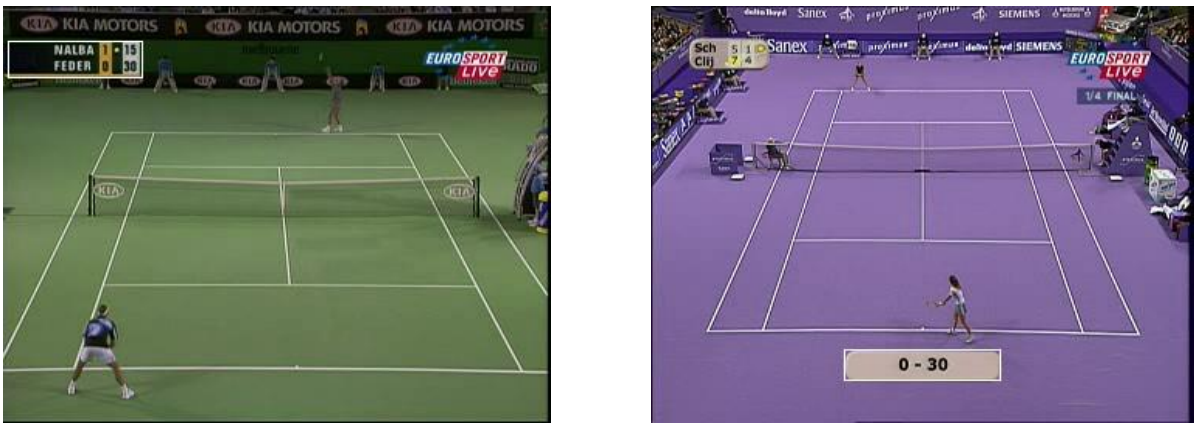

Figure 6. Samples of score boards inserted between tennis points and their bounding rectangle.

Figure 7. Samples of score boards inserted between tennis games and their bounding rectangle.

4 Experimental Evaluations

The performance of our parsing system was experimentally evaluated on three tennis video records captured from Eurosport satellite channel. One of them shows an excerpt of a tennis match of Australia Open (AO) 2003 championship, two others represent fragments of two matches of WTA tournament. The former lasts about 51 minutes, the rest two – 8.5 and 10 minutes. The two tournaments have different score board configuration and color distribution of the court which can be seen from Figure 6 and Figure 7 representing these tournaments. So, we extracted two sets of learning samples for the events detectors.

In the parsing accuracy evaluations we used the content structure presented in Figure 2.I and parsing rules of Table 1, Table 2 and Table 3. Rocket hits detectors were not used in these evaluations, so a template for a score/rally event was represented by a single general court view. Automatically parsed videos were compared with manually labeled data where the segments were defined in the same way as those used to derive the transition templates above in this report: the segments “set”, “game” and “point” begin with the first serve and end when the last rallies are over (we relate these moments to the beginning and the end of corresponding general court views). The results of segmentation performance evaluations are presented in Table 4. Semantic levels 3 and 4 (see Figure 2.I) were treated separately; level 2 was not considered as there are few set segments in the ground-truth. The values of recall, precision and F1 are calculated as

$$recall = \frac{N_c}{N_c + N_{miss}}, \tag{3}$$

$$precision = \frac{N_c}{N_c + N_{f.a.}}, \tag{4}$$

$$F1 = \frac{2 * recall * precision}{recall + precision}, \tag{5}$$

where N_c, N_{miss} and $N_{f.a.}$ are the number of correct, missed and false alarm boundaries respectively. A manually labeled boundary was considered as detected correctly if it coincided with an automatically obtained one within an ambiguity time window of 1 second. The value N_b in Table 4 stands for the number of tested boundaries in manually labeled video. In order to reduce the influence of "edge effects" on the segmentation evaluations results, the first and the last segments of the lowest semantic level were cut off by half from comparison intervals for each video record. The results of classification accuracy evaluations are given in Table 5. The value of recall and precision are computed in a similar way as expressions (4) and (5), where instead of the number of boundaries the time duration of the segments should be used. The "total duration" of segments in Table 5 is measured in seconds.

Tournament	Semantic level	Recall	Precision	F1	N_b
AO	3	0.84	0.62	0.71	19
	4	0.82	0.91	0.86	153
WTA	3	1	0.83	0.91	10
	4	0.94	0.98	0.96	63
AO+WTA	3	0.90	0.68	0.78	29
	4	0.86	0.93	0.89	216

Table 4. Segmentation results.

Semantic Level	Segment	Recall	Precision	F1	Total duration
3	Game	0.97	0.99	0.98	3320
	Break	0.97	0.91	0.94	778
4	Point	0.83	0.98	0.90	1670
	Break	0.97	0.89	0.93	1650

Table 5. Classification results total for both the tournaments.

As for processing time, our parsing technique is quite fast provided that the events are already extracted and takes less than 1 second for a usual tennis match on modern personal computers. This is because the computational complexity is approximately proportional to the number of events and the number of semantic levels. The major computational power is required to decompress the video and detect the relevant events. On our Intel Pentium 4 1.8 GHz computer this task is performed nearly in real time for MPEG1 coded video, though we did not make a lot of optimizations.

The most of the segmentation errors are caused by unreliability of event detectors. High rate of false score boards result in relatively low precision of segmentation on games and breaks

for AO tournament. It is caused by resemblance of the score board, which is a true indicator of the segment transitions, to a statistics board which was inserted in any place during games (sample frames are shown in Figure 8). One of the sources of the errors at semantic level 4 is a high false alarm rate for global court views which is caused by confusions with replay shots (they shift the transition between a point and a break). So, there is a need to improve the events detector or use additional ones. For instance, game and set score boards are often shown together with wide views (see the left frame of Figure 8). This allow us expect that their combining into a pattern would give a more reliable transition indicator.

Figure 8. Game score board (at the left) and its false counterpart.

In order to estimate the accuracy of our parsing engine without the influence of event detection errors, segmentation performance was evaluated on manually corrected events. We considered shots as global court views only if they were not replayed episodes. The evaluation results are given in Table 6. There are only few segmentation errors at the semantic level 4 for AO tournament that steam from the parsing rules. They are caused by the fact that sometimes the producer forget to show a score board or insert it after the first serve of a point.

Tournament	Semantic level	Recall	Precision	F1
AO	3	1	1	1
	4	0.91	0.95	0.93
WTA	3	1	1	1
	4	1	1	1
AO+WTA	3	1	1	1
	4	0.94	0.96	0.95

Table 6. Segmentation results for manually detected events.

5 Application: Tennis Analyzer

A computer program called “Tennis Analyzer” was developed and realized in C++ programming language using MS Visual C++ development environment. It is aimed at completely automatic generation of a content table for tennis video and provides a graphical user interface (GUI) for browsing. The block scheme of the program is depicted in Figure 9. Tennis video is given in the form of AVI or MPEG-code file. In order to extract visual and audio features that are to be used for content parsing and browsing through them, tennis video at first is split into a frame sequence and an audio samples stream. The frame sequence is segmented into shots using the twin-threshold method described above. For each shot it is calculated a key frame – the frame that has the color histogram closest to the mean histogram of the shot. Key frames are used to visually represent the corresponding shots and to classify them into court views. Score boards are detected using the learning board samples extracted from the database which is prepared with the help of the learning module. The learning interface allows a user to select a sample frame with the score board of interest and to define its bounding rectangle. The audio stream is used to detect applauses segments. The applauses are used to generate an importance mark of semantic segments, so that the longer are the applauses, the higher is the mark. At first the audio classifier produces the applauses class probabilities for every sound chunk of one second length. Then, in order to reduce the rate of the false alarms, the smoothing module detects as applauses segments only the groups of several contiguous sound chunks with high probability. As the feature extraction is slow enough, all the features are computed only once and saved to the corresponding data files, whereupon they can be used for fast browsing.

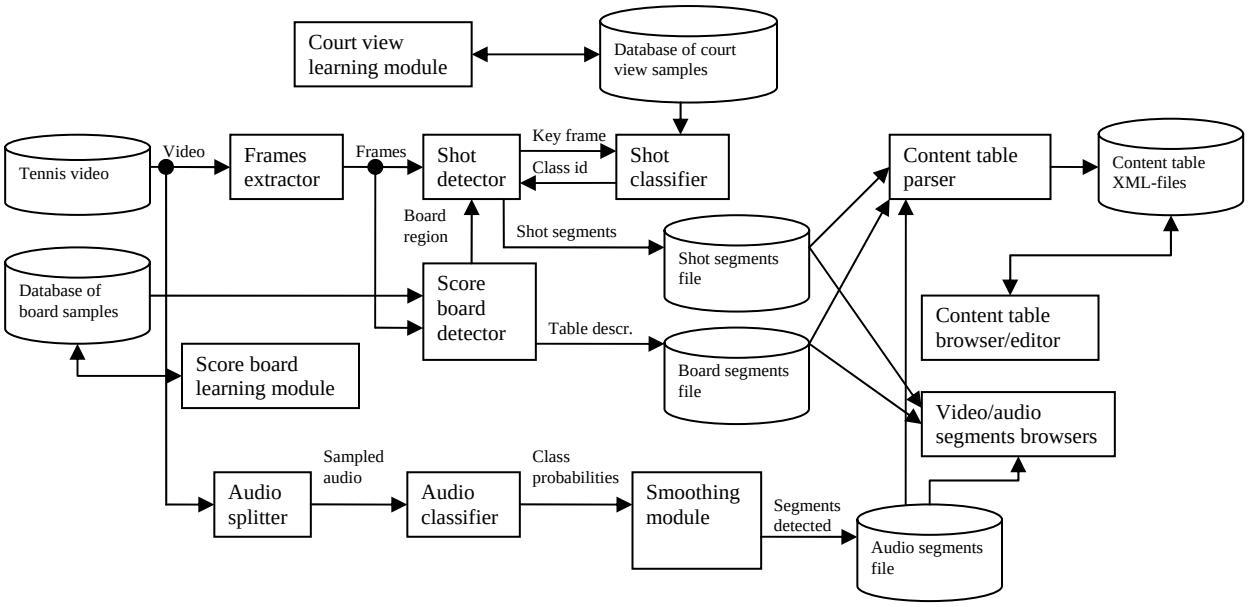

Figure 9. Block scheme of the Tennis Analyzer.

The Tennis Analyzer provides several views for tennis video browsing and analyses, as shown in Figure 9. The player window (shown at the upper right corner) allows for playing of the video using standard controls: play/stop and rewind buttons and a scrolling slider. The content view (shown at the upper left corner) represents the content table as a tree structure and allows for browsing through the content synchronously with the player window. For each selected semantic segment it represents a list of the nested segments with their attributes. The most interesting segments of the video can be filtered out based on the desirable range for the importance mark. In addition, the content view provides interface for entering the textual description for segments and for manual editing of the content structure that allows a user to correct automatically parsed structures and save them to persistent memory. The view shown at the bottom of Figure 9 represents the key frames of the shots and the frames that contain a score table. It allows for synchronous browsing with the content view and the player window as well – for the content view it can represent only the segment selected in the content tree; the player window can be rewind to any selected key frame by a simple mouse click.

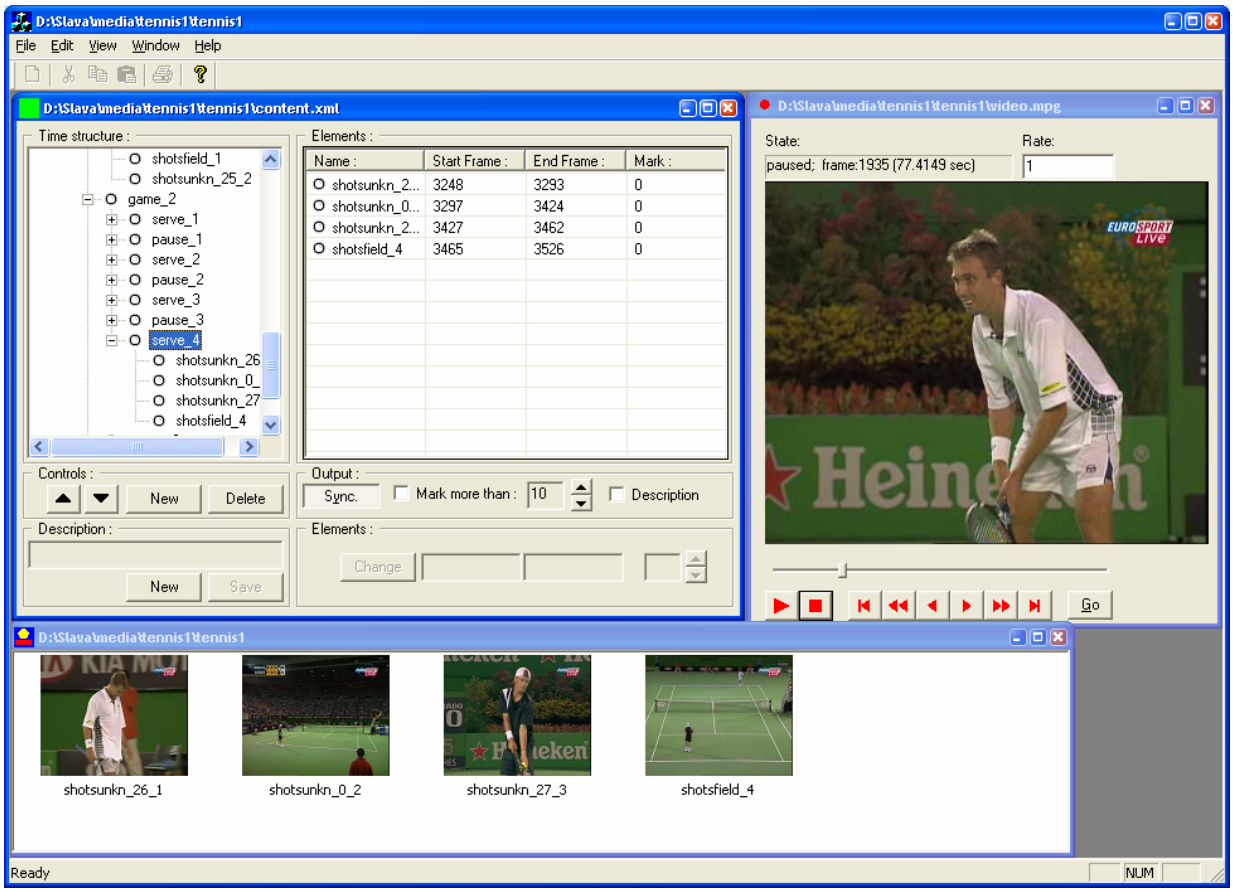

Figure 10. Tennis analyzer GUI.

6 Conclusions

A deterministic approach is proposed for hierarchical content parsing of video. It is adopted and tested for tennis video. The approach is based on some particular characteristics and production rules that are typically employed to convey semantic information to a viewer, such as specific views and score boards in tennis broadcasts. We use our notion of a tennis content structure to select unique template of events that indicate transitions to semantic segments of each type. These events along with grammar restrictions drive the parsing process.

The advantage of our approach is in its expressiveness and low computational complexity. Moreover, the experimental evaluations showed quite high segmentation accuracy, especially when high reliability of event detectors is provided. Further improvements of the proposed technique could be done in several directions. First, more robust event detectors could be elaborated, as the experimental evaluations showed that such an improvement would enhance significantly the segmentation accuracy. Second, parsing rules could be extended to include additional informational sources such as rocket hits detection, time constraints, speech recognition. Third, the currently used semantic structure could be extended so as to contain a larger variety of semantics which could provide additional possibilities for content based navigation. For instance, the points could be split into several classes such as rallies, missed first serve, ace or replay.

References

[ALL 83] Allen J.F., “Maintaining Knowledge about Temporal Intervals”, Communications of the ACM, Vol. 26, No. 11, pp. 832-843, 1983.

[CHA 01] Chang S.-F., Zhong D, Kumar R, “Real-Time Content-Based Adaptive Streaming of Sports Videos”, *Proc. IEEE CBAIVL*, Hawaii, pp.139-146, December 2001.

[DON 01] Dong Zhang, Wei Qi, Hong Jiang Zhang, “A New Shot Boundary Detection Algorithm”, *IEEE Pacific Rim Conference on Multimedia*, pp.63-70, 2001.

[HOU 59] Hough P.V.C., “Machine Analysis of Bubble Chamber Pictures”, *Int. Conference on High Energy Accelerators and Instrumentation*, CERN, pp. 554-556, 1959.