Automatic Classification of Tennis Video for High-level Content-based Retrieval[‡]

G. Sudhir[*], *John C. M. Lee*[*], *and Anil K. Jain*[†]

Technical Report HKUST-CS97-2
August 7, 1997

[*]Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

[†]Department of Computer Science
Michigan State University
East Lansing, MI 48824-1226, U.S.A.

[‡] This research is supported in part by the Sino Software Research Centre of The Hong Kong University of Science and Technology under grant SSRC96/97.EG03 and by RGC Direct Allocation Grant under grant DAG96/97.EG26

Email: {sudhir,cmlee}@cs.ust.hk, jain@cps.msu.edu

Abstract

This report presents our techniques and results on automatic analysis of tennis videos for the purpose of high-level classification of tennis video segments to facilitate content-based retrieval. Our approach is based on the generation of a image model, valid under perspective projection, for the tennis court-lines in a tennis video. We first derive this model by making use of knowledge about dimensions of a tennis court, typical camera geometry used when capturing tennis videos, and the connectivity information about the court lines. Based on this model, we develop a court line detection algorithm and a robust player-tracking algorithm to track the tennis players over the image sequence. In order to select only those video segments which contain tennis court from an input raw footage of tennis video, we also propose a color-based tennis court clip selection method. Automatically extracted tennis court lines and the players' location information form crucial *measurements* for the high-level reasoning module which analyzes the relative positions of the tennis players with respect to the court lines and the net, in order to map the measurements to high-level events (semantics) like *baseline-rallies*, *serve-and-volleying*, *net-games*, *passing-shots*, etc. Results on real tennis video data are presented demonstrating the validity and performance of the approach.

1 Introduction

Automatic classification and retrieval of video information based on *content* is a very challenging research area. Recently, there have been many research efforts addressing various relevant issues [1, 2, 3, 4, 5, 6, 7] (see [8] for a recent survey). The most difficult question faced at the outset is: what does *content* mean? Or, more specifically, how should one characterize *visual content* present in video, and how to extract it for the purpose of generating useful, high-level annotations to facilitate content-based indexing and retrieval of video segments from digital video libraries. It is generally accepted that *content* is too subjective to be characterized completely as it often depends on the *objects*, *background*, *context*, *domain*, etc. in the video. This is one of the main reasons why the problem of content-based access is still largely unsolved. On the other hand, advances in video and storage technology have enabled (*i*) representation of visual data in digital form for archiving and browsing, (*ii*) efficient storage using compression algorithms, and (*iii*) fast retrieval with random-access capability using storage media like CD-ROMs, Laser Discs, Video CDs and the latest DVD-Video and DVD-ROM systems.

Recent literature contains a number of approaches to characterize visual content based on color, texture, shape and motion [9, 10, 11, 12, 13, 14]. While these approaches have their merit of being applicable to generic image and video data, they also have a major limitation, that is, they can characterize only *low-level* information. The end users will almost always like to interact at *high-level* when accessing or retrieving images and video segments from a database, and the importance of useful, high-level annotations can not be over emphasized. At the same time, manual generation of high-level annotations is quite cumbersome, time consuming and expensive. Hence, there is a dire need for algorithms that are able to automatically infer high-level content using the low-level information extracted from data. While this is extremely difficult for images and video in general, limiting the analyses to specific domains can help overcome many hurdles in automatic generation of high-level (semantic) annotations as relevant to those specific domains. This report presents a case study wherein domain-specific knowledge and constraints are exploited to accomplish the difficult task of automatic annotation of video segments with high-level semantics, to enable content-based retrieval.

There have been some recent attempts to exploit *contextual* or *domain knowledge* for automatic generation of high-level annotations for video segments. Most of them attempt to *relate* or *map* the low-level information measured from the image and video data to high-level content pertinent to the specific domains [15, 16, 17]. Zhang et al. [15] present a method to parse and index *news* video, based on a syntactic, model-based approach. While they parse the news video into “ancherperson shot” and “news story shot” using some image processing techniques and succeed in automatically creating a more detailed description for a *structured* news video, they do not attempt to describe the *content* of the news video for the obvious reason that it falls into the domain of generic video and hence is far too difficult a problem. Gong et al. [16] address the problem of automatic parsing of TV soccer programs. They make use of the standard layout of a soccer field (domain knowledge) and partition it into nine categories such as “mid-field” and “penalty-area”. Then they apply various image processing techniques like edge detection and line identification on TV images for automatic determination of camera location. A similar attempt has been made by Yow et al. [18] wherein they analyze a video library of soccer games. They report special techniques for automatically detecting and extracting soccer highlights by analyzing image content, and for presenting the detected shots by panoramic reconstruction of selected events. A context-based technique for automatic extraction of embedded captions is reported by Yeo and Liu [19]. They apply their method directly to MPEG compressed video and extract visual content which are highlighted using the extracted captions. Saur et al. [17] present a method for automatic analysis and annotation of basketball video. They use the low-level information available directly from MPEG compressed video of basketball as well as the prior knowledge of basketball video structure to provide high-level content analysis like "close-up views", "fast breaks", "steals", etc.

In this report, we present our approach and results for an automatic analysis of tennis video for the purpose of high-level classification of tennis video segments to facilitate content-based retrieval. We exploit the available domain knowledge about tennis video and demonstrate that it is possible to generate meaningful semantic annotations applicable to the domain. Our goal is to automate the generation of useful and high-level annotations like *baseline-rallies*, *passing-shots*, *net-games* and *serve-and-volley games*, to pertinent segments of the video. We have chosen these annotations because they form some of the most common play events in a tennis video. Video is an important source in the sport of tennis, particularly in the areas of teaching, coaching and learning how to play the game [20, 21]. They are used to teach the rules of the game, analyze and summarize tennis matches and tournaments, demonstrate proper playing techniques, document the history of the sport, and profile current and past champions[1]. Annotating raw, unstructured

[1] Videotapes are the dominant video sources today [21]. However, with the advent of digital com-

tennis video with these high-level content can help professional tennis players and coaches to retrieve video segments in a more meaningful manner from a digital library of tennis video. For example, a player who wants to improve/develop his *serve-and-volley* abilities would only be interested in retrieving and studying a variety of *serve-and-volley* video segments. Or, a tennis coach giving a visual demonstration of *passing* techniques and/or stratagies to players is mainly interested in a collection of *passing-shot* video segments as part of his video instructional material. If the above mentioned high-level annotations are available for indexing a digital library of tennis video, then players or a coach can get content-based access to relevant video segments. In this report, we present our techniques to achieve this objective. Experimental results on real tennis video data are presented to demonstrate the merit of our approach.

The rest of the report is organized as follows. In Section 2, we present an overview of our system. We present a color-based algorithm in Section 3 to select video clips containing tennis court only from the input raw footage of tennis video, for further analysis. In Section 4, we derive a quantitative model for a tennis court which is valid on the image domain, under the assumption of perspective projection model for the camera. Based on this model, Section 5 presents our tennis court-line detection algorithm. In Section 6, we present our player-tracking algorithm to track the two tennis players over the image sequence, and in Section 7, we describe briefly our method of *mapping* low-level *positional* information about the tennis court-lines and the locations of the two tennis players to high-level tennis-play events in the video segments. Finally, in Section 8 we present some

pression and storage technology and the associated advantages of random and network-based access capabilities, we foresee that digital video will play a dominant role in future.

of the high-level annotation results we have obtained, and conclude the report in Section 9.

2 Overview of the System

Fig. 1 shows the block diagram of our system. The modules shown within the dotted lines form the core part of our system and are the main subject of this report. Our system analyses the input tennis video at different levels. First, the system processes an input raw footage of tennis video and partitions the footage into shots containing continuous sequence of tennis court, using a color-based shot selection approach. Then, the video segments containing tennis court are input to the court-line detection module which detects the tennis court-lines in the individual frames. This information, along with the raw video data present in the video segment, is then used in the detection and tracking of the two players in the video segment by the player tracking module. The high-level reasoning module analyses the outputs of these two modules for inferencing different tennis-play events in the video segment. These analyses result in meaningful high-level (semantic) annotations for the segments of the input video. After the analyses, the video segments are organized in an information management system with the high-level annotations providing indices for content-based retrieval. This information management system takes other useful textual data also as inputs like, for example, the type of the court (*clay* or *carpet* or *hard* or *grass*), the names of the players, tournament type, year, etc. The end users interact with this information system through a GUI for providing queries to the system. Typical queries are like: (*i*) Retrieve *serve-and-volley* clips containing *John McEnroe* in 1984, (*ii*) Retrieve *passing-shot* clips containing *Ivan Lendl* on a *carpet* court, (*iii*) Retrieve *baseline-rally* clips containing *Andre Agassi* on a *hard* court in 1992, etc. The system responds by searching the library and retrieves the matching video clips for display: they are shown as small icons on the icon pad of the GUI; they can be played on the display pad of the GUI by the user through a mouse-click. It should be noted that while the whole process of annotation is done off-line, the end user interactions with the system are in fact on-line with near real-time retrievals.

As mentioned above, the four modules shown within dotted lines in Fig. 1 form the core part of the system and more details about our techniques and algorithms concerning the four modules are presented in this report. In the following section, we present details about the coor-based shot partitioning approach we have used.

3 Color-based Selection of Court Segments

In this section we present our technique for the selection of video segments containing tennis court from an input raw footage of tennis video. This is necessary since the further analyses presented in the next sections are done on video clips containing tennis court only[2].

For this purpose, we propose a color-based algorithm to automatically indicate whether a particular image frame belongs to a tennis court scene or not. As mentioned in Section 2, a tennis court belongs to one of the four different classes: (*i*) *carpet*, (*ii*) *clay*, (*iii*) *hard* and (*iv*) *grass* courts. These classes can be distinguished by their different color properties.

[2] A typical tennis video contains many other scenes interleaved between the tennis court scenes like, for example, as those of spectators or a player or an umpire. This is especially so in between the points. These scenes need to be segmented out from the input raw footage of tennis video.

However, each class can have multiple representative colors as its member. Based on statistical analysis using many example frames, we summarize the color properties of the four classes in the following table. Using this information, we apply the following algorithm to select relevant clips containing tennis court.

Table 1: Standard colors of different types of tennis court.

Class of Court	Mean colors (R:G:B) of example members	Class Thereshold	Fraction Thereshold
Carpet	112 : 168 : 124 72 : 76 : 52 64 : 64 : 64 64 : 92 : 60	25	40
Clay	136 : 68 : 40 164 : 68 : 52	25	25
Hard	40:48:20	25	25
Grass	96 : 136 : 48 88 : 124 : 36	25	40

Court segment selection algorithm

For each image frame of the input tennis video, do the following:

1. Place a 100 × 100 window at the center of the image frame.

2. Compute the color with maximum frequency – *max_color* – from the color-frequency distribution in the window.

3. Compute the Euclidean distances of the *max_color* from each class of courts. (The distance of a *color* from a court class is the *minimum* of distances of the *color* from the members of the class.)

4. Let *court_class* be the class of court corresponding to the *minimum* of the Euclidean distances and let *court_color* be the corresponding member color.

5. If the Euclidean distance between the *max_color* and the *court_color* is more than the Class Threshold (see Table 1) corresponding to *court_class*, then classify the image as not a tennis court frame and stop. Otherwise, continue.

6. Initialize $court\_points = 0$ and $total\_points = 0$.

7. For each pixel in the 100×100 window:

   (*i*) Compute the Euclidean distance of the color of the pixel from *court_color*.

   (*ii*) If the distance is more than the Fraction Threshold (see Table 1) corresponding to *court_class*, then $court\_points = court\_points + 1$.

   (*iii*) $total\_points = total\_points + 1$.

8. Compute $court\_fraction = \frac{court\_points}{total\_points}$.

9. If *court_fraction* is more than 0.6, then classify the input image frame as a tennis court frame belonging to *court_class*; otherwise, not. Stop.

After the application of the algorithm, the *contiguous* segments of *similarly* classified frames are selected as tennis video clips and used for further analysis for high-level classification (see Fig. 1). Typical results of application of the above tennis court clip selection algorithm are given in Fig. 2.

Remark 3.1 *Note that the color of a* carpet *court can be widely varying in general. We have used some of the standard colors in the above table. However, for any* carpet *court with a non-standard color, an example image frame from the input tennis video needs to be given as a training set for accurate classification by using the above algorithm.*

The following section presents in detail a quantitative model which we have developed for a tennis court under the perspective projection assumption for the camera. This model is used in the development of a court-line detection algorithm in Section 5.

4 A Model for Tennis Court

In this section, we derive a quantitative model for a tennis court in the image domain under the assumption of *perspective* camera projection. For this, we exploit the form-based and camera geometry-based constraints as described below.

We make the following assumptions:

A1 The dimensions and connectivity (form) of the tennis court-lines are known (see Fig. 3 for a complete specification of the tennis court dimensions; for convenience, only the *singles* court is shown in the figure).

A2 The camera geometry is shown in Fig. 4 with the equations for imaging given by equations (1) and (2) mentioned below.

According to the camera geometry shown in Fig. 4, the tennis court can be viewed as a symmetrically located 2D object in a frontal plane (i.e., a plane perpendicular to the viewing direction of the camera; see dashed lines in Fig. 4) which is tilted away by an *unknown* angle θ about the X-axis in the object-centered coordinate system and then perspectively projected on to the image plane of the camera using the camera-centered coordinate system. Also note that the object-centered coordinate system is modeled simply as a translation of the camera-centered coordinate system by an *unknown* amount D

along the viewing direction of the camera (the Y'-axis; also the Y-axis). Mathematically, the complete imaging mechanism can be described by the following equations:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} 0 \\ D \\ 0 \end{bmatrix} \tag{1}$$

and

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\ \frac{X'}{Y'} \\ f\ \frac{Z'}{Y'} \end{bmatrix} = \begin{bmatrix} f\ \frac{X}{Y\cos\theta + Z\sin\theta + D} \\ f\ \frac{-Y\sin\theta + Z\cos\theta}{Y\cos\theta + Z\sin\theta + D} \end{bmatrix}, \tag{2}$$

where, $[X, Y, Z]^t$ is the coordinate vector of a point on the court in the object-centered coordinate system, θ is the tilt angle about the X-axis, $[X', Y', Z']^t$ is the coordinate vector of the point on the court in the camera-centered coordinate system, f is the focal length of the camera and $[x, y]^t$ is the perspective projection vector of the point in the image plane. Typical frames of a tennis video shown in Figs. 5, 9 and 12 confirm the validity of the camera geometry shown in Fig. 4.

While we use the knowledge of tennis court dimensions and assume a typical imaging geometry, we do *not* make any assumptions about the knowledge of the tilt angle θ, the distance D and the camera focal length f. However, we make the following assumption about feature extraction:

A3 It is possible to extract from an image of a tennis court the three projected segments of the court-lines corresponding to $\overline{P_0P_2}$, $\overline{P_0P_1}$ and $\overline{P_1P_3}$ (see Fig. 4).

In the rest of the report, we refer to these three segments as the basic segments. We now present the following result:

Lemma 1 *Under the assumptions* **A1**, **A2** *and* **A3**, *it is possible to reconstruct the perspective projections of all the other points on the tennis court-lines, especially for the points* P_4 *to* P_{13}. *Thus, it is possible to completely recover the tennis court-lines in the image domain.*

Proof: We prove the lemma by showing that it is possible to reconstruct the perspective projection of the point P_{13} on the image. Similar argument can be given for all the other points. Let L_1 be the length of the perspective projection of the segment $\overline{P_0P_2}$ and L_2 be the length of the perspective projection of the segment $\overline{P_1P_3}$. From assumption **A3**, it is possible to measure L_1 and L_2. Also, from assumptions **A1** and **A2**, it can be shown that (see appendix A)

$$\frac{L_1}{L_2} = \frac{-21\sin\theta + D}{-39\sin\theta + D}. \tag{3}$$

From the above equation, we have

$$F = \frac{L_1 - L_2}{39L_1 - 21L_2}, \tag{4}$$

where $F = \frac{\sin\theta}{D}$. Now, let B_1 be the image domain vector (i.e., the perspective projection) corresponding to the vector $P_0\vec{P}_1$ and let B_2 be the image domain vector corresponding to the vector $P_0\vec{P}_2$. Similarly, let B' be the image domain vector corresponding to the vector $P_0\vec{P}_{13}$. Then, from assumptions **A1** and **A2**, it can be shown that (see appendix B) B'

can be expressed as a linear combination of B_1 and B_2 as

$$B' = \alpha B_1 + \beta B_2, \tag{5}$$

where

$$\alpha = \frac{78(-21F+1)}{18(39F+1)} \quad \text{and} \quad \beta = \frac{2(-39F+1)}{(39F+1)}.$$

Since the value of the parameter F can be estimated (see Eq. (4)), B' can be recovered and hence the perspective projection of the point P_{13} can be reconstructed on the image. Expressions similar to Eq. (5) can be derived for all other points on the tennis court-lines wherein the values of α and β change depending on each point. The values of α and β can be determined from the knowledge of the measurable parameter F (see Table 2). This completes the proof of the lemma. □

From the above lemma and the information about the form (connectivity) of a tennis court, it is possible to recover the tennis court-lines completely in the video frames. The advantages of using the proposed mathematical model are two-fold:

1. It enables the recovery of the projected tennis court-lines in the video frames with the help of minimal number of measurements (and the associated heuristics) needed for making the assumption **A3** hold.

2. The complete tennis court frame can be recovered even in those parts where the court-lines are occluded and/or are very difficult to extract. This gives very important information about the location of all the segments of the tennis court-lines in a given image. This helps in an accurate determination of the relative position

of players with respect to the tennis court-lines which is crucial in the high-level reasoning steps for generating semantic annotations for the tennis video.

Table 2 lists the values of α and β for the perspective projections of the tennis court points P_0 to P_{13} (see Fig. 3) for expressing them as linear combinations of the basis vectors B_1 and B_2.

Table 2: Values of α and β for the points on the tennis court-lines shown in Fig. 3. The parameter F can be measured (see Eq. (4)).

Point	α	β	Point	α	β
P_0	0	0	P_8	$\frac{39\,(-21F+1)}{18}$	$2\,(-39F+1)$
P_1	1	0	P_9	$\frac{60\,(-21F+1)}{18\,(21F+1)}$	0
P_2	0	1	P_{10}	$\frac{60\,(-21F+1)}{18\,(21F+1)}$	$\frac{(-39F+1)}{(21F+1)}$
P_3	1	$\frac{(-39F+1)}{(-21F+1)}$	P_{11}	$\frac{60\,(-21F+1)}{18\,(21F+1)}$	$\frac{2\,(-39F+1)}{(21F+1)}$
P_4	2	0	P_{12}	$\frac{78\,(-21F+1)}{18\,(39F+1)}$	0
P_5	1	$\frac{2\,(-39F+1)}{(-21F+1)}$	P_{13}	$\frac{78\,(-21F+1)}{18\,(39F+1)}$	$\frac{2\,(-39F+1)}{(39F+1)}$
P_6	$\frac{39\,(-21F+1)}{18}$	0			
P_7	$\frac{39\,(-21F+1)}{18}$	$(-39F+1)$			

Remark 4.1 *In the above analysis, for convenience, we have chosen* $\vec{P_0P_1}$ *and* $\vec{P_0P_2}$ *for the basis vectors* B_1 *and* B_2*, respectively. It should be noted that, any other two segments of the tennis court-lines could be used for the same purpose as long as they span the 2D image domain.*

Based on the model derived in this section, we present our tennis court-line detection algorithm in the following section.

5 Tennis Court-line Detection Algorithm

The purpose of the tennis court-line detection algorithm is to extract court-lines in a given tennis video frame. This algorithm has two parts: (*i*) extraction of the three basic court-line segments in the image domain, and (*ii*) reconstruction of the complete tennis court-lines according to the mathematical model described in the previous section.

5.1 Extraction of the basic line segments

Extraction of the basic line segments is necessary to make the assumption **A3** hold. For this, we first describe briefly a straight-line detection algorithm that we have used. The following straight-line detection algorithm takes as inputs (*i*) a starting point and (*ii*) a line-growing direction. It also uses the following heuristics: (*a*) the color of the court-line segments is the brightest in the image, and (*b*) points on a court-line segment are continuous. Note that these heuristics are valid for good quality tennis video.

Straight-line detection algorithm

1. Go to the given starting point in the image.

2. Consider a 1-pixel strip of image intensities *perpendicular* to the given line growing direction (we have fixed the length of the strip to 15 pixels on each side of the line).

3. Compute the *mean* and *std* (standard deviation) of the image intensities in the strip. Note the locations of the pixels whose intensities exceed the threshold $Th = mean + 2.5\, std$. Call them *Line-Points*.

4. Find the *middle* point of the *Line-Points* in the 1-pixel strip. Update the given

starting point to the point which is 1-pixel next to the *middle* point, *along the given line growing direction.* If the updated starting point is not an 8-neighbor of the previous starting point, STOP (this is done only for the points other than the given starting point) and consider that the line segment has ended; go to step 6. Otherwise, go to step 5.

5. Using this updated starting point, repeat steps 2-4.

6. List all the *Line-Points* detected during steps 1-5. Fit a straight-line through the line points in the *least square error* sense and compute the parameters (equation) of the line segment.

We apply the above mentioned straight-line detection algorithm to detect four different straight-line segments as follows:

(a) Initialize the starting search point at slightly below the image center. (This heuristic is used only for the first frame; see the next subsection.)

(b) From this point, apply the straight-line detection algorithm to detect the *vertical* line segment downwards till the point which is the perspective projection of the point P_3 (see Fig. 3) is reached (the straight-line detection algorithm stops at this point). Consider the end point of this segment as the starting point of the next line segment.

(c) Apply the straight-line detection algorithm to detect the *horizontal* line segment rightwards till the point which is the perspective projection of the point P_1 (see

Fig. 3) is reached. Consider the end point of this segment as the starting point of the next line segment.

(d) Apply the straight-line detection algorithm to detect the *vertical* line segment downwards till the point which is the perspective projection of the point P_0 (see Fig. 3) is reached. Consider the end point of this segment as the starting point of the next line segment.

(e) Apply the straight-line detection algorithm to detect the *horizontal* line segment leftwards till the line detection algorithm stops.

(f) Using the equations of the four straight-line segments detected in the steps 2-5, compute the locations of the perspective projections of the points P_0, P_1, P_2, and P_3, by considering intersections of the appropriate straight-line segments.

Note that we compute the locations of the end points corresponding to the basic line segments using the equations of four straight-line segments. Unlike direct corner detection methods which are troubled by the localization versus detection uncertainties [22], using intersections of straight-line segments to detect corner points leads to very accurate localization also. This in turn increases the accuracy of the basis vectors B_1 and B_2 and helps reliable reconstruction of the complete tennis court-lines in the image domain using the mathematical model as described in the next subsection.

5.2 Reconstruction of the complete tennis court

After the detection of the three basic segments of the tennis court in the image, we reconstruct the points P_4 to P_{13} according to the model derived in the previous section (see Table 2). Then, we use the knowledge about the form (connectivity) of the tennis court-line segments to reconstruct the complete tennis court in the image domain. Except for the first frame, we also reinitialize the center of the tennis court-line in the image based on the reconstructed tennis court-lines in the previous image (this is used as the starting point in the step (a) above). Only for the first frame, we choose the starting point at slightly below the center of the image (during step (a) above).

Fig. 5 shows the typical result of application of our line detection algorithm to real tennis video frames. Without any loss of generality, we have reconstructed the *singles* court only in all the images. Fig. 6 shows the result of tracking the court-lines over a sequence of tennis video frames. Note that the camera sways slightly to the right and left during the play in this typical tennis video, thus creating slight deviations from the assumed camera geometry. However, the court-line detection algorithm has successfully tracked the court-line segments during the play. This shows the robustness of the proposed model-based approach for court-line detection.

6 Tennis Player Tracking Algorithm

In this section, we present our player-tracking algorithm. For the purpose of tracking the players over the image sequence, the initial locations of the two players in the image domain have to be located. For this, we use the following motion-based detection algorithm.

Player detection algorithm

1. Track the tennis court-lines over the first few frames and determine if there is any

camera motion present during the period. (The intersection points of the tennis court-line segments are used to estimate a 6-parameter affine 2D motion model in the *least square error* sense [12] and the six model parameters are analysed to check if there is any genuine camera motion.)

2. Select the first frame and the *next* frame (e.g., the 2^{nd} or 3^{rd} frame).

3. If there is a camera motion detected in step 1, then:

   (i) Estimate a 2D affine motion model for the motion between the two frames;

   (ii) Generate a motion-predicted frame from the first frame for the *next* frame (the 2^{nd} or 3^{rd} frame);

   (iii) Generate a residue frame after subtracting the motion predicted frame from the *next* frame;

   otherwise:

   (i) Generate a residue frame after subtracting the first frame from the *next* frame.

4. Threshold and binarize (0 and 255) the residue frame, and smooth the residue frame to eliminate the noise (we employ an iterative smoothing algorithm).

5. Compute the "focus of interest" regions for locating each player by using the court-line information extracted for the *next* frame, and generate two appropriate search windows (see Remark 6.1 below).

6. Extract the largest *connected* component in the smoothed residue frame inside the two window locations and compute the centroids of the two connected components;

these two centroids represent the locations of the two players in the *next* frame.

Remark 6.1 *In step 5 above, we mentioned about* appropriate *search windows. These search windows are rectangular areas used to limit the connected component analysis to within those windows only, in the smoothed residue image. Thus, they serve as* focus of interest *regions. These search windows are defined adaptively depending on the video frame being considered. For the initial video frame, the width of the search windows is determined by the reconstructed* baselines *of the tennis court in the image domain and the search windows are centrally located at the* baselines *with a fixed height value for the search windows. For all other frames, we choose the locations of the search windows based on the* latest *player-tracking results available. This way, the windows are adaptively defined and the player detection is done within the windows only.*

Fig. 7 shows the typical result of using the above algorithm for motion-based detection of the player locations. After the detection of the locations of the two players in the initial frame, we generate *templates* for the two players and call them *Top Player* (TP) and *Bottom Player* (BP), since the TP appears near the top edge and the BP near the bottom edge in an image. The templates are nothing but square windows of fixed sizes placed centrally at the locations of the two players (we use a larger window size for the BP since the physical size of BP is usually larger than that of TP), which contain the image data (pixels) from the video frame. Since the two templates are centrally placed at the detected locations of the players, they contain the image data corresponding to the two players and hence form crucial inputs to the player-tracking algorithm.

After the template generation, we use the following algorithm for tracking the two players over the rest of the image sequence. We describe the algorithm for a template T (which can be either that of BP or that of TP) extracted from a *current* frame and being tracked in the *following* frame(s).

Player tracking algorithm

1. Let T be the template of size $\omega \times \omega$ centered at location (p, q) in the *current* frame C. Let F be the *following* frame and let N be the *next* frame, i.e., the frame following F.

2. Generate a binary image H containing only the reconstructed tennis court lines of F using the court-line detection algorithm given in Section 5:

$$H(i,j) = \begin{cases} 0 & \text{if a tennis court line passes through } (i,j) \\ 1 & \text{otherwise} \end{cases} .$$

3. Set $Max\_value = 256 \times \omega^2$.

4. For each pixel location (i, j) in a $b \times b$ window B around (p, q) in F, do:

   (a) Compute $match\_value$ between T and the similar sized template in F located at $(p + i, q + j)$ using

   $$match\_value \;=\; \sum_{(u,v)\in\omega\times\omega} |T(u,v) - F(p+i+u, q+j+v)| \;\cdot H(p+u, q+v);$$

   (b) If $Max\_value > match\_value$, do:

   $$Max\_value = match\_value$$

$min\_p = p + i$ and $min\_q = q + j$.

5. The matching player location in F is (min_p, min_q). Update the location of the player using $(p, q) = (min\_p, min\_q)$. Update the contents of template T from the data in frame F at the updated location of (p, q).

6. Update the *current* frame and the *following* frame using $C = F$ and $F = N$.

7. Repeat steps 1-6 till there is no *next* frame N.

Note that the *match_value* computed in step 4.(a) above is the net absolute difference value between the windows, and the algorithm used above is basically a full-search, minimum absolute difference algorithm (MAD) – used widely in block-based motion estimation algorithms [23]. However, we have modified the basic algorithm by exploiting the available information about the tennis court-lines in the *current* frame by using a binary weighting factor H. The basic purpose of this weighting factor is to *discard* those points of T which contain pixels corresponding to the static object – the tennis court. Thus the template represents the player information more appropriately, leading to more accurate results for the template matching algorithm. Also note that after the match is found, the algorithm updates the template T so that it represents the *current* information available about the player. This is also very important for accurate performance of the tracking algorithm because the player information keeps changing dynamically as the play progresses over the image sequence and, if such updating of template is not done, the modified MAD algorithm may easily give an incorrect match, thereby leading to errors while tracking the players. The results reported in Section 8 represent the typical performance of the player tracking algorithm given above.

7 High-level Reasoning Module

In this section we describe briefly the analysis done in the high-level reasoning module. This module takes the information about the tennis court-lines extracted by the court-line detection module and the information about the player positions over the image sequences extracted by the player-tracking module. These two measurements form the crucial inputs to the high-level reasoning module. These low-level measurements are mapped to high-level (semantic) interpretations relevant to the tennis-play events in the video segment. For this, simple logical decisions are made based on the values of these measurements. More specifically, those decisions are made on the basis of the relative positions of the player locations with respect to the tennis court-lines. For example, if the positional information extracted by tracking the two players during the video segment confirms that they play essentially near their respective baselines (see Fig. 8), then the high-level reasoning module maps the set of measurements to the *baseline-rallies* play event and annotates the video segment accordingly. The following table summarizes this *mapping* of low-level measurements (positional information) to the high-level content about tennis-play events. In Table 3, we use the following abbreviations (see Fig. 8):

BL	$\Longrightarrow$	Baseline
SL	$\Longrightarrow$	Service line (horizontal)
NN	$\Longrightarrow$	Near the Net
BLC	$\Longrightarrow$	Center of Baseline
SLC	$\Longrightarrow$	Center of Service line

The following section presents some of the results we have obtained on real tennis video segments, using the proposed automatic classification approach.

Table 3: Mapping low-level positional information to high-level tennis-play events

No.	Top Player (TP)		Bottom Player (BP)		High-level Annotation
	Initial Location	Final Location	Initial Location	Final Location	
1	BL	BL	BL	BL	Baseline-rallies
2	BL	NN	BL	BL	Passing-shot
3	BL	BL	BL	NN	Passing-shot
4	BL	BL	BLC	SLC	Serve-and-Volley
5	BLC	SLC	BL	BL	Serve-and-Volley
6	SL	NN	SL	NN	Net-game

8 Results

We have implemented the proposed tennis video classification approach and have obtained excellent results. In the implementation of our approach, we partition an input tennis video segment into chunks of 30 frames and process each chunk similarly. For each chunk of video frames, we do the following:

1. Detect the tennis court-lines in all the frames of the chunk using the method given in Section 5;
2. Detect the location of the two tennis-players in the second or third frame of the chunk by applying the motion-based strategy given in Section 6;
3. Track the two players over the rest of the video frames of the chunk using the player-tracking algorithm given in Section 6.

In what follows, we present in detail the results of using our automatic tennis video analysis approach for two tennis video segments containing different high-level content. We believe these results demonstrate the merit of the model-based approach proposed here.

8.1 A *Baseline-rallies* Video Clip

In this subsection, we present the results we have obtained for a video segment containing baseline rallies between two tennis players. The video segment consists of 120 frames (4 chunks of 30 frames each) of size 320×240 pixels. In the video segment, the two players play essentially from their respective baselines and there is no camera motion detected. Fig. 9 shows some of the video frames for visual depiction of the tennis-play event in the video segment. The results of tracking players during the play are shown separately in Fig. 10. The results of applying our technique are summarized pictorially in Fig. 11 which shows both the detected tennis court-lines as well as the tracks of the two tennis players as the match progresses over the video segment.

8.2 A *Passing-shot* Video Clip

In this subsection, we present the results we have obtained for a video segment containing a passing-shot situation between two tennis players. The video segment consists of 90 frames (3 chunks of 30 frames each) of size 320×240 pixels. In the video segment, the two players play initially from their respective baselines and, as the match progresses, the bottom player (BP) rushes towards the net while the top player (TP) stays at the far-end baseline. There is no camera motion in the video segment. Fig. 12 shows some of the video frames for visual depiction of the tennis-play event in the video segment. The results of tracking players during the segment are shown in Fig. 13. A pictorial summary of the results of applying our technique is shown in Fig. 14, which shows both the detected tennis court-lines as well as the tracks of the two tennis players as the match progresses over the video segment.

9 Summary and Conclusion

In this report, we have presented a case wherein domain-specific knowledge and constraints are exploited to accomplish a very difficult task of automatic annotation of video segments with high-level semantics. Particularly, we have presented our research techniques and results regarding automatic analysis of tennis video for the purpose of high-level classification of tennis video segments to facilitate content-based retrieval. Our approach is based on a quantified model for the tennis court in the image domain under the assumption of perspective projection for the camera. We derive this model by making use of the *a priori* knowledge about actual dimensions of a tennis court as well as the typical camera geometry used while taking a tennis video (we have used only the information regarding the general camera imaging geometry and we do not use any particular information about the camera parameters or object distances). We make use of the connectivity information about the tennis court-lines also (i.e., the information about the "form" of the object), and have demonstrated the applicability of the model on real tennis video frames by developing a tennis court-line detection algorithm and by showing the results of application of the algorithm on the real tennis court images. We have also presented a robust tennis-player tracking algorithm in order to track the two tennis players over the image sequence. We use the automatically extracted tennis court and the players' location information, which form the crucial *measurements*, for the purpose of high-level reasoning where we analyze the relative positions of the two players with respect to the tennis court-lines and the tennis-net, so as to *map* the measurements to high-level events (semantics) like *baseline-rallies*, *passing-shot*, *serve-and-volleying*, *net-games*, etc. Results on real tennis video data are presented to demonstrate the merit of the approach.

We believe that similar exploitation of domain-specific constraints in other applications can be highly useful in overcoming the very difficult task of automatic annotation of real video segments with high-level content pertinent to those specific domains. We feel that a fine *blend* of domain-specific techniques and general purpose algorithms would help a long way in achieving the lofty goal of providing content-based access to video.

Acknowledgement

We thankfully acknowledge the support of Sino Software Research Center of The Hong Kong University of Science & Technology for carrying out this research under the grant SSRC96/97.EG03 and the support of Research Grants Council under the grant DAG96/97.EG26.

Appendix

A Derivation of Eq. (3)

From Fig. 4 and Eqs.(1) and (2), it follows that the perspective projections p_0, p_1, p_2 and p_3 of the points P_0, P_1, P_2 and P_3 (see Fig. 3) respectively, are given by

$$p_0 = \begin{pmatrix} f\,\frac{13.5}{(-39\sin\theta+D)} \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix}, \quad p_1 = \begin{pmatrix} f\,\frac{13.5}{(-21\sin\theta+D)} \\ \\ f\,\frac{-21\cos\theta}{(-21\sin\theta+D)} \end{pmatrix},$$

$$p_2 = \begin{pmatrix} f\,\frac{0}{(-39\sin\theta+D)} \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix}, \text{ and } p_3 = \begin{pmatrix} f\,\frac{0}{(-21\sin\theta+D)} \\ \\ f\,\frac{-21\cos\theta}{(-21\sin\theta+D)} \end{pmatrix}. \tag{6}$$

Using the above expressions, we now have the lengths L_1 and L_2 of the two segments $\overline{p_0p_2}$ and $\overline{p_1p_3}$ respectively, in the image domain given by

$$L_1 = f\,\frac{13.5}{(-39\sin\theta + D)} \text{ and } L_2 = f\,\frac{13.5}{(-21\sin\theta + D)}. \tag{7}$$

Hence Eq. (3) follows. □

B Derivation of Eq. (5)

Using the expressions for the points $p_0, p_1\,\text{and}\,p_2$ from Eq. (6), we have the two basis vectors B_1 and B_2 given by

$$B_1 = p_0\vec{p}_1 = \begin{pmatrix} f\,\frac{13.5}{(-21\sin\theta+D)} \\ \\ f\,\frac{-21\cos\theta}{(-21\sin\theta+D)} \end{pmatrix} - \begin{pmatrix} f\,\frac{13.5}{(-39\sin\theta+D)} \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix} = \begin{pmatrix} \frac{f}{D}\,\frac{(-18)(13.5)F}{(-21F+1)\,(-39F+1)} \\ \\ \frac{f}{D}\,\frac{18\cos\theta}{(-21F+1)\,(-39F+1)} \end{pmatrix}$$

$$B_2 = p_0\vec{p}_2 = \begin{pmatrix} 0 \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix} - \begin{pmatrix} f\,\frac{13.5}{(-39\sin\theta+D)} \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix} = \begin{pmatrix} \frac{f}{D}\,\frac{(13.5)}{(-39F+1)} \\ \\ 0 \end{pmatrix} \qquad (8)$$

where $F = \frac{\sin\theta}{D}$. Now consider the perspective projection p_{13} of the point P_{13} (see Fig. 3) which is given by

$$p_{13} = \begin{pmatrix} f\,\frac{-13.5}{(39\sin\theta+D)} \\ \\ f\,\frac{39\cos\theta}{(39\sin\theta+D)} \end{pmatrix}.$$

The vector $p_0\vec{p}_{13}$ is given by

$$p_0\vec{p}_{13} \;=\; \begin{pmatrix} f\,\frac{-13.5}{(39\sin\theta+D)} \\ \\ f\,\frac{39\cos\theta}{(39\sin\theta+D)} \end{pmatrix} - \begin{pmatrix} f\,\frac{13.5}{(-39\sin\theta+D)} \\ \\ f\,\frac{-39\cos\theta}{(-39\sin\theta+D)} \end{pmatrix} = \begin{pmatrix} \frac{f}{D}\,\frac{-27}{(39F+D)(-39F+D)} \\ \\ \frac{f}{D}\,\frac{78\cos\theta}{(39F+D)(-39F+D)} \end{pmatrix}.$$

Since B_1 and B_2 span the 2D image domain space, $p_0\vec{p}_{13}$ can always be expressed as a linear combination of B_1 and B_2 as follows

$$p_0\vec{p}_{13} \;=\; \alpha\,B_1 \;+\; \beta\,B_2.$$

Substituting for B_1, B_2 and $p_0\vec{p}_{13}$, we have

$$\begin{pmatrix} \frac{f}{D}\,\frac{-27}{(39F+D)(-39F+D)} \\ \\ \frac{f}{D}\,\frac{78\cos\theta}{(39F+D)(-39F+D)} \end{pmatrix} = \alpha \begin{pmatrix} \frac{f}{D}\,\frac{(-18)(13.5)F}{(-21F+1)\,(-39F+1)} \\ \\ \frac{f}{D}\,\frac{18\cos\theta}{(-21F+1)\,(-39F+1)} \end{pmatrix} + \beta \begin{pmatrix} \frac{f}{D}\,\frac{(13.5)}{(-39F+1)} \\ \\ 0 \end{pmatrix}. \tag{9}$$

Solving the two simultaneous equations of Eq. (9), we have

$$\alpha = \frac{78(-21F+1)}{18(39F+1)} \quad \text{and} \quad \beta = \frac{2(-39F+1)}{(39F+1)}$$

as desired. □

References

[1] S.W. Smoliar and H.J. Zhang. Content-based Video Indexing and Retrieval. *Multimedia*, 1(2):356–365, 1994.

[2] A.D. Bimbo, E. Vicario, and D. Zingoni. Sequence Retrieval by Contents through Spatio Temporal Indexing. In *Proc. IEEE Symposium on Visual Languages*, pages 88–92, 1993.

[3] Y. Tonomura, A. Akutsu, Y. Taniguchi, and G. Suzuki. Structured Video Computing. *IEEE Multimedia*, 1(3):34–43, Fall 1994.

[4] R. Jain and A. Hampapur. Metadata in Video Databases. *ACM SIGMOD Record*, 23(4), 1994.

[5] Hong Jiang Zhang, Atreyi Kankanhalli, and S.W. Smoliar. Automatic Partitioning of Full-motion Video. *Multimedia Systems*, 1(1):10–28, 1993.

[6] John C. M. Lee, Q. Li, and W. Xiong. VIMS: A Video Information Manipulation System. *Multimedia Tools and Applications: Special issue on Representation and Retrieval of Visual Media in Multimedia Systems*, 4(1):7–28, Jan. 1997.

[7] E. Ardizzone and M. La Cascia. Automatic Video Database Indexing and Retrieval. *Multimedia Tools and Applications: Special issue on Representation and Retrieval of Visual Media in Multimedia Systems*, 4(1):29–56, Jan. 1997.

[8] Gulrukh Ahanger and Thomas D.C. Little. A Survey of Technologies for Parsing and Indexing Digital Video. *Journal of Visual Communication and Image Representation*, 7(1):28–43, 1996.

[9] Michael J. Swain and Dana H. Ballard. Color Indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[10] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin. The QBIC Project: Query Images by Content using Color, Texture and Shape. In *SPIE Proc. Storage and Retrieval for Image and Video Databases*, volume 1908, pages 173–186, 1993.

[11] A. Pentland, R.W. Picard, and S. Scaroff. Photobook: Tools for Content-based Manipulation of Image Databases. In *SPIE Proc. Storage and Retrieval for Image and Video Databases II*, volume 2185, pages 34–46, 1994.

[12] G. Sudhir and John C. M. Lee. Video Annotation by Motion Interpretation using Optical Flow Streams. *Journal of Visual Communication and Image Representation*, 7(4):354–368, 1996.

[13] A. Akutsu *et al.* Video Indexing using Motion Vectors. In *SPIE Proc. Visual Communication and Image Processing'92*, volume 1818, pages 522–530, 1992.

[14] W. Xiong, John C. M. Lee, and Rui Hua Ma. Automatic Video Data Structuring through Shot Partitioning and Key Frame Selection. *Machine Vision and Applications: Special issue on Storage and Retrieval for Still Image and Video Databases*, 1996. to appear. (Technical Report HKUST-CS96-13).

[15] HongJiang Zhang, Shuang Yeo Tan, Stephen W. Smoliar, and Gong Yihong. Automatic Parsing and Indexing of News Video. *Multimedia Systems*, 2:256–266, 1995.

[16] Yihong Gong, Lim Teck Sin, Chua Hock Chuan, HongJiang Zhang, and Masao Sakauchi. Automatic Parsing of TV Soccer Programs. In *Proc. Intl. Conf. on Multimedia Computing and Systems*, pages 167–174, May 1995.

[17] Drew D. Saur, Yap-Peng Tan, Sanjeev R. Kulkarni, and Peter J. Ramadge. Automated Analysis and Annotation of Basketball Video. In *Storage and Retrieval for Image and Video Databases V*, volume SPIE-3022, pages 176–187, Feb. 1997.

[18] D. Yow, Boon-Lock Yeo, M. Yeung, and B. Liu. Analysis and Presentation of Soccer Highlights from Digital Video. In *Second Asian Conference on Computer Vision (ACCV'95)*, volume 2, pages 499–503, 1995.

[19] Boon-Lock Yeo and B. Liu. Visual Content Highlighting via Automatic Extraction of Embedded Captions on MPEG Compressed Video. In *Proc. of SPIE Conf. on Digital Video Compression: Algorithms and Technologies*, volume 2668, pages 38–47, Feb., 1996.

[20] A Lumpkin. *A Guide to the Literature of Tennis.* Greenwood Press, Connecticut, U.S.A., 1985.

[21] Dennis J. Phillips. *The Tennis Source Book.* The Scarecrow Press, Inc., U.S.A., 1995.

[22] H. Wang and M. Brady. Corner Detection with Sub-pixel Accuracy. Technical report, Robotics Research Group, Department of Engineering Science, University of Oxford, 1992.

[23] F. Dufaux and F. Moscheni. Motion Estimation Techniques for Digital TV: A Review and a New Contribution. *Proceedings of the IEEE*, 83(6):877–891, June 1995.

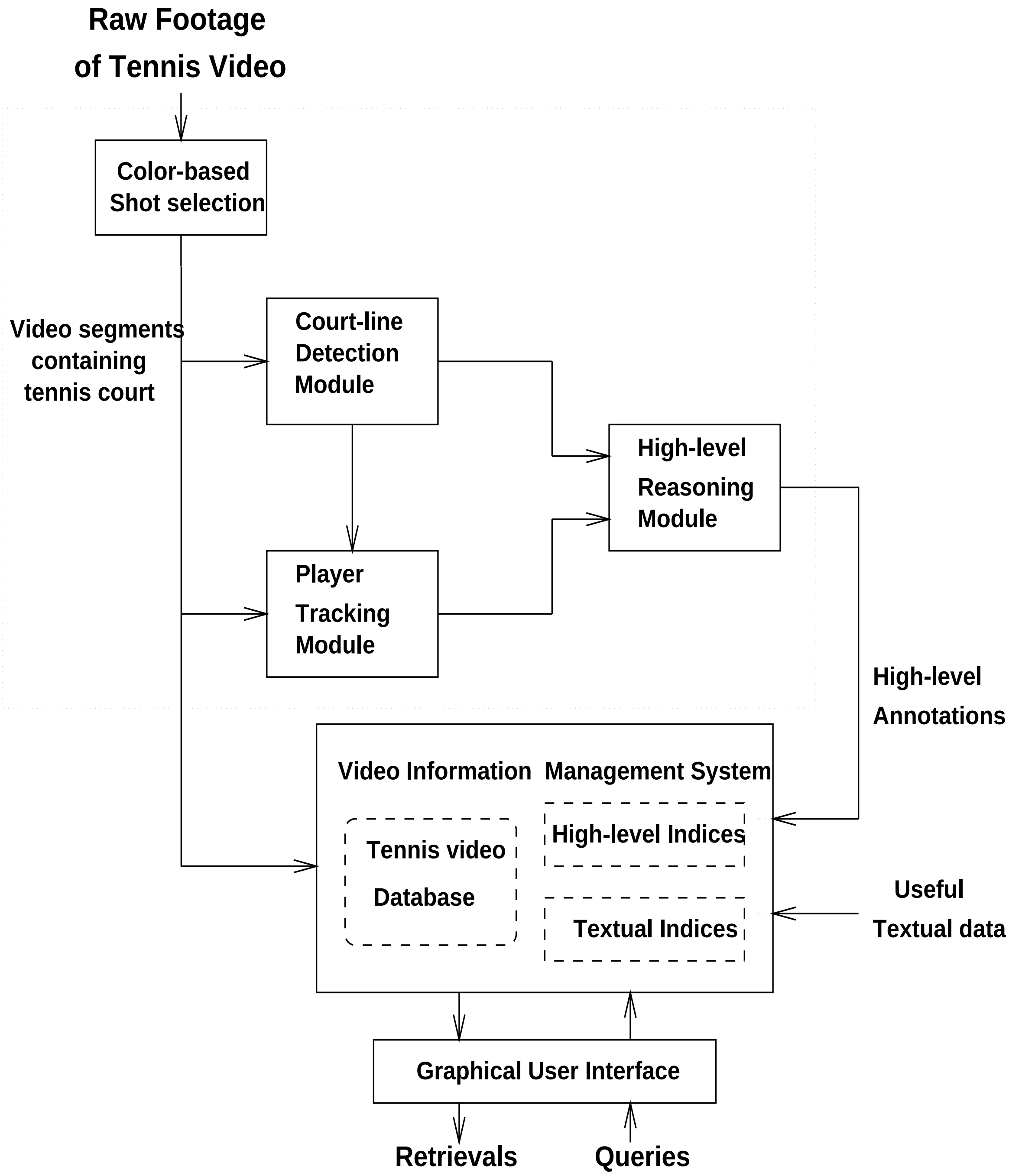

Figure 1: Block diagram of our tennis video analysis system.

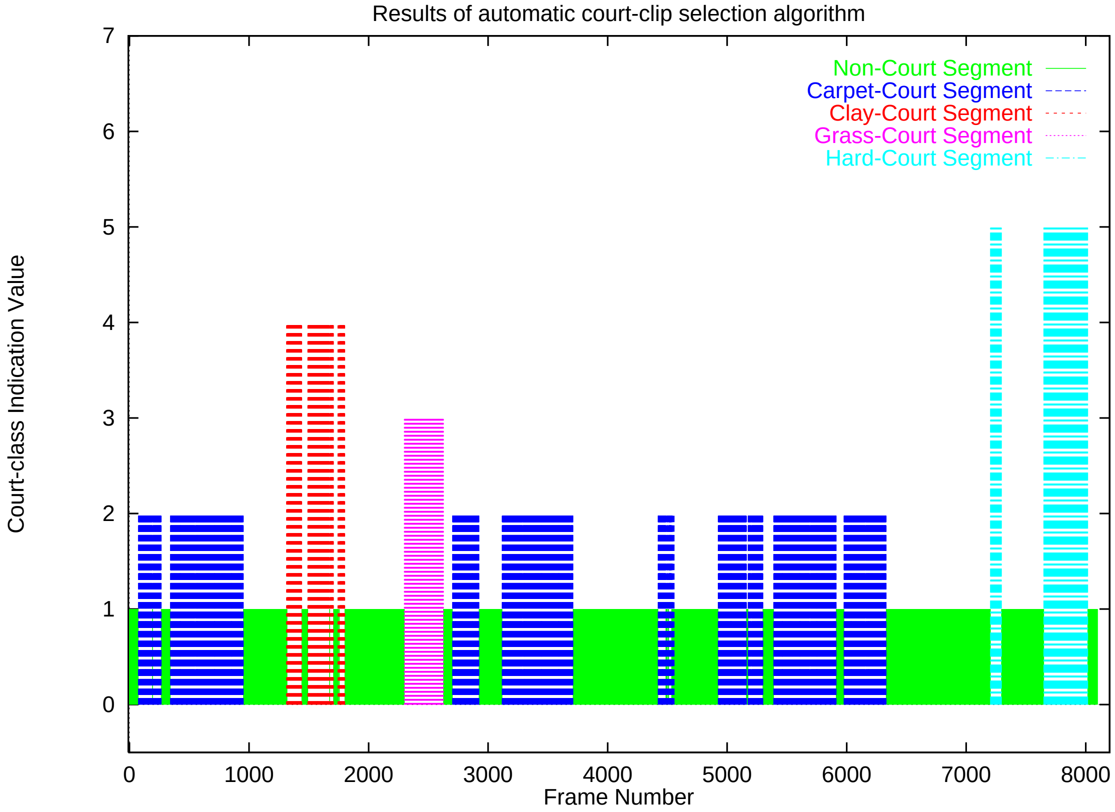

Figure 2: Results of applying our tennis court clip selection algorithm on 4.5 minutes of input video from a VHS tape titled *SuperStars of Women's Tennis* © 1994 Vestron Video. This tape contains the clips of all types of tennis courts thus making it most suitable for testing our color-based tennis court clip selection algorithm. The results have been manually verified to be correct.

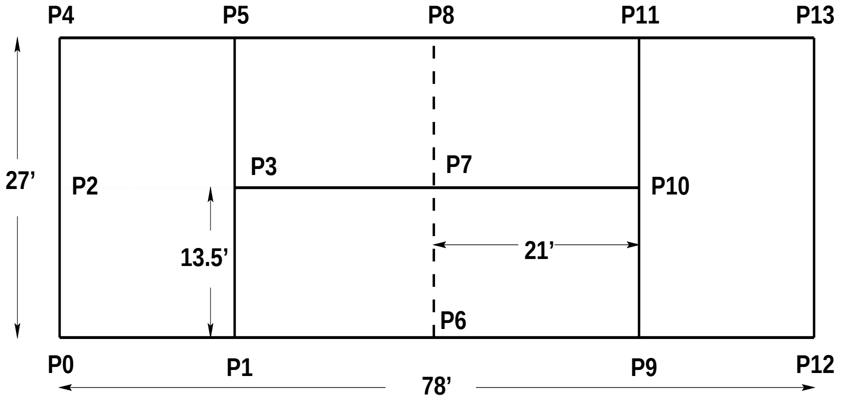

Figure 3: Tennis court dimensions. Without any loss of generality, only the *singles* court dimensions are shown. Note the corners marked P_0 to P_{13} which are referred to in the report.

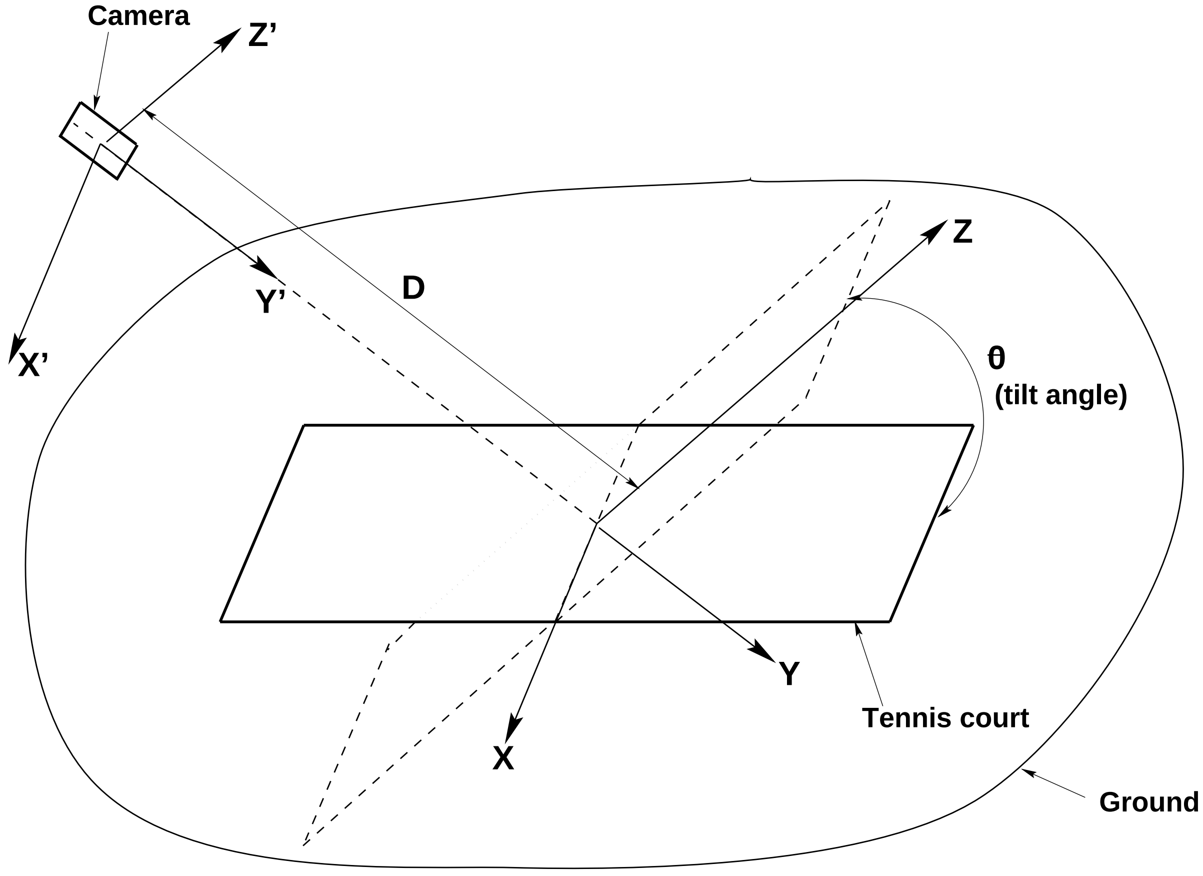

Figure 4: Typical camera geometry used while shooting tennis video.

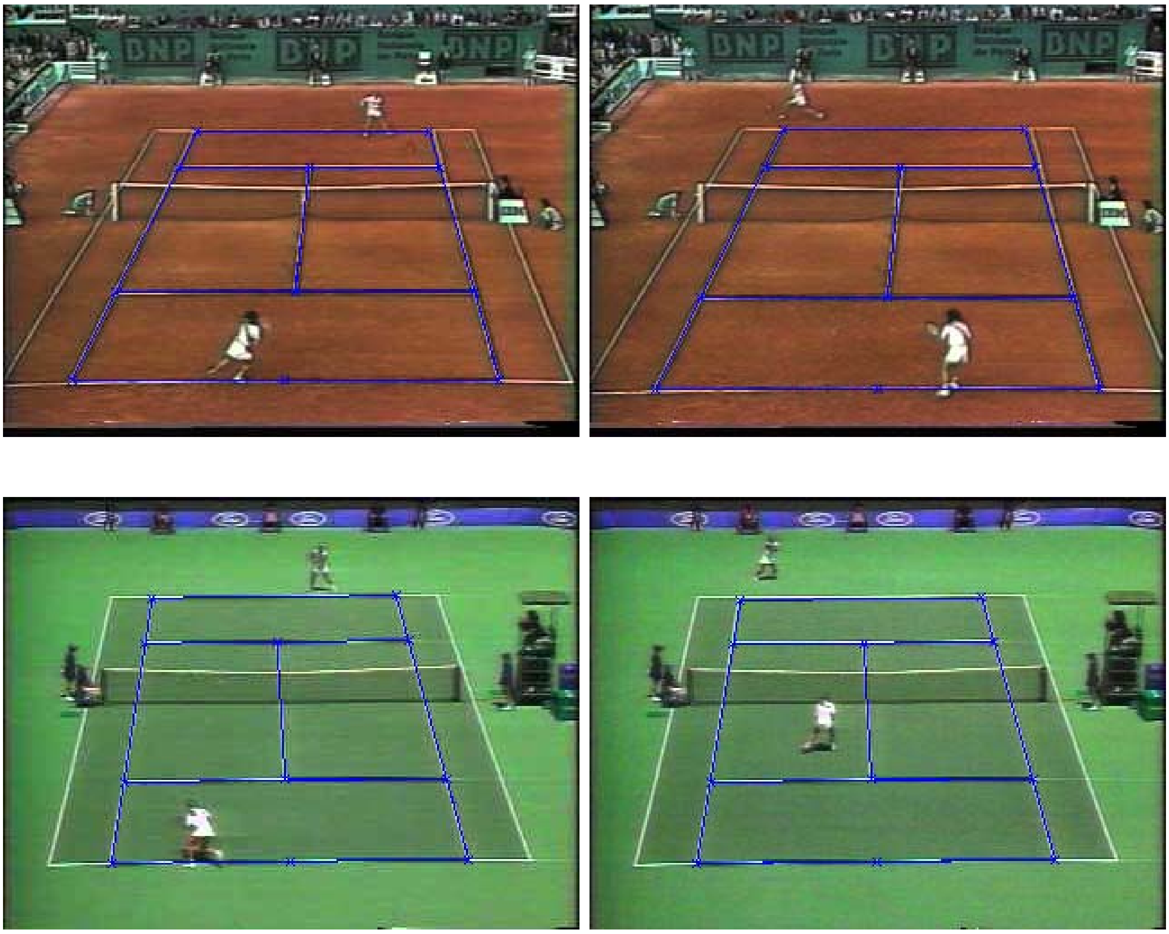

Figure 5: The tennis court-lines detected in some tennis video frames shown superimposed in blue color on the images. The images in the top row are from the video of a *clay* court match while those in the bottom row are from the video of a *carpet* court match. Note the differences in the parameters of the camera geometry. However, the same tennis court-line detection algorithm of Section 5, which does not depend on any particular parameters of the camera geometry, has been applied to all the frames. Court-line segments have been accurately detected in all the frames.

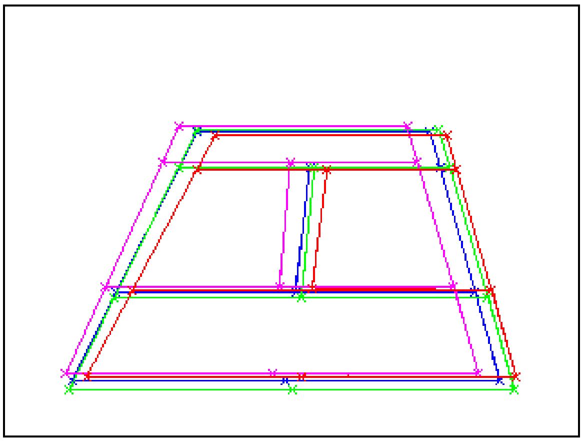

Figure 6: The tennis court-lines tracked over the image sequence of a tennis video segment are shown superimposed in different colors. The order of frames is from *blue* to *green* to *magenta* to *red*.

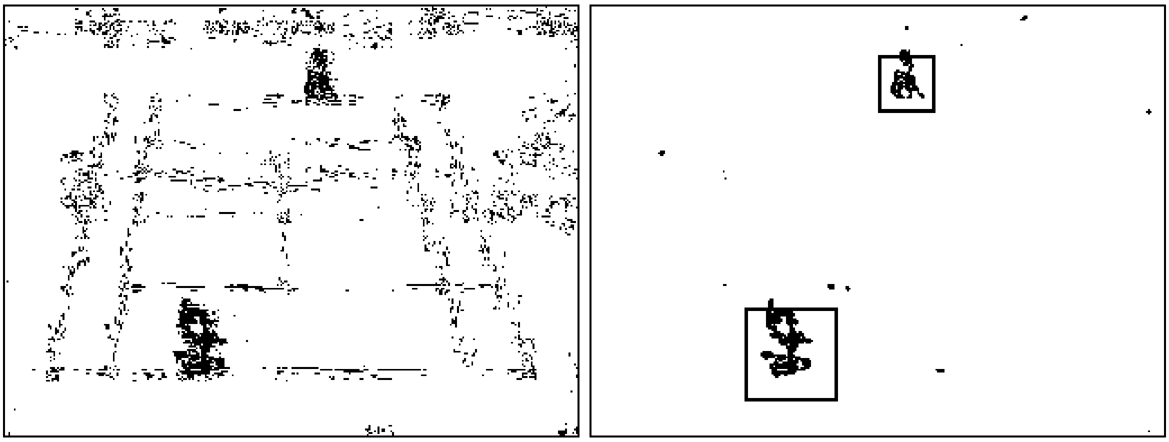

Figure 7: Typical result of applying the player detection algorithm (right) to the residue frame (left). The two square windows in the right image show the templates corresponding to the two players located centrally at their respective detected locations.

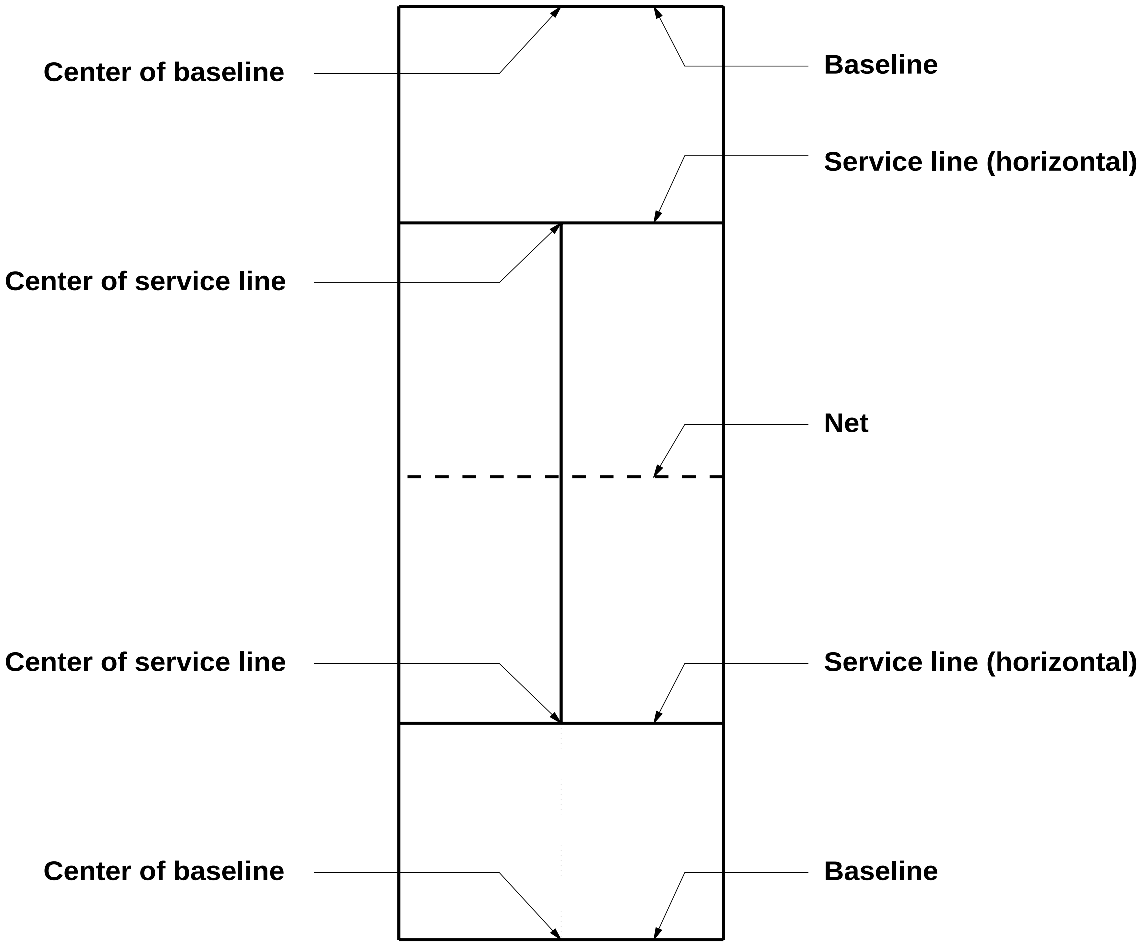

Figure 8: Names of different line segments on a tennis court.

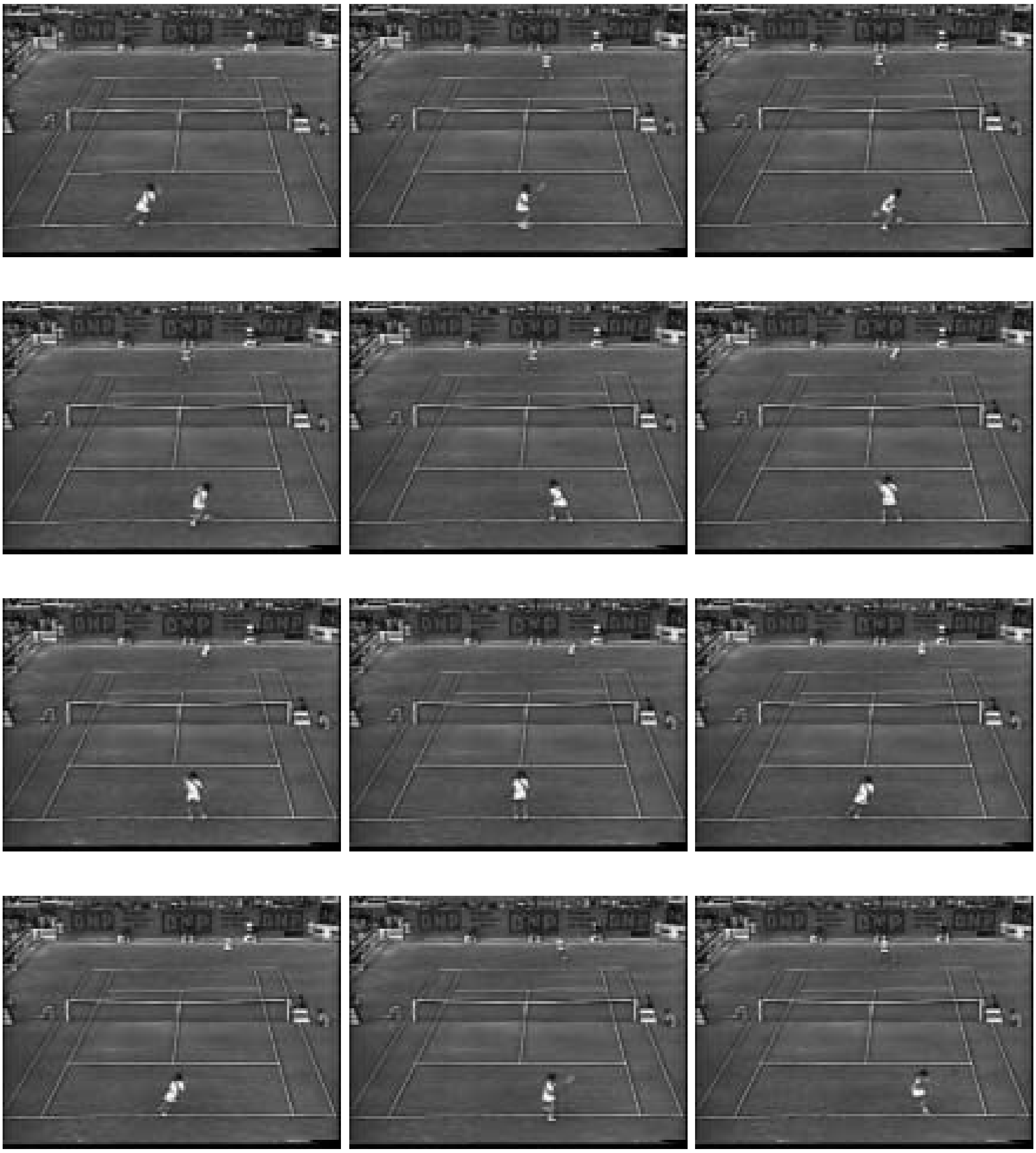

Figure 9: Selected frames from the *Baseline-rallies* video segment consisting of 120 consecutive frames to visually depict the tennis-play event. The four rows of images correspond to the 4 successive chunks (each of 30 frames) in a top-down order. Each row shows the frames numbered 1, 15 and 29 in the corresponding chunk of 30 frames. The size of the frames is 320 × 240 pixels. Here, the frames are shown to a scale slightly less than half the original size of the frames.

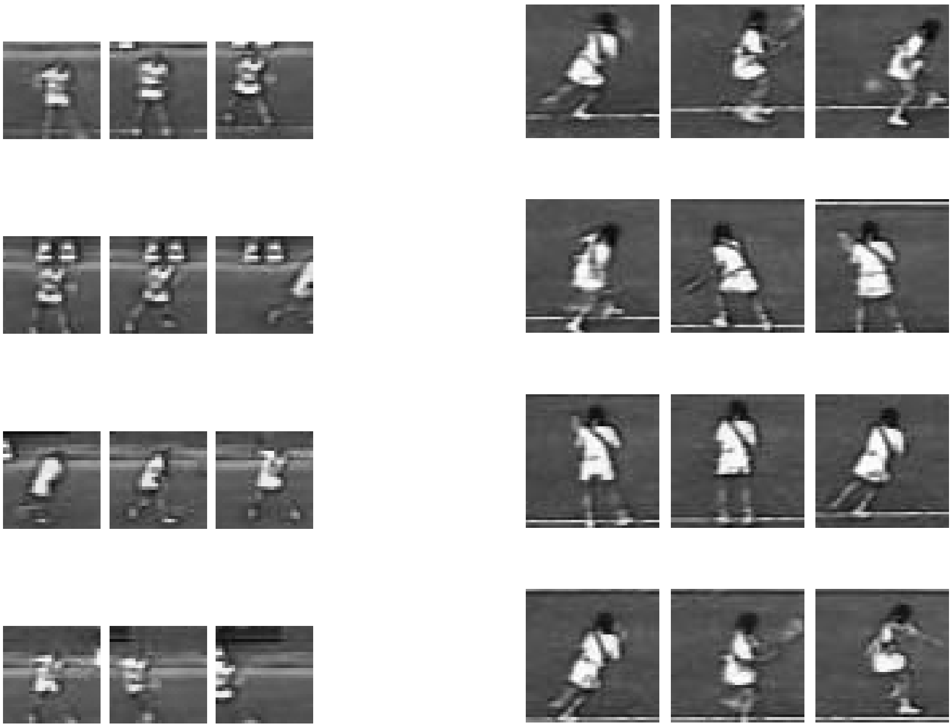

Figure 10: The templates of the players as they are tracked for 120 consecutive frames of the *Baseline-rallies* video segment. Shown on the left are the templates of the Top Player (TP), and those of the Bottom Player (BP) are shown on the right. The four rows of images correspond to the 4 successive chunks (each of 30 frames) in a top-down order. For each player, each row shows the templates etched out from the frames numbered 1, 15 and 29 in the corresponding chunk of 30 frames. The size of the TP templates is 30×30 pixels and that of BP templates is 50×50 pixels. The templates shown here are of actual scale.

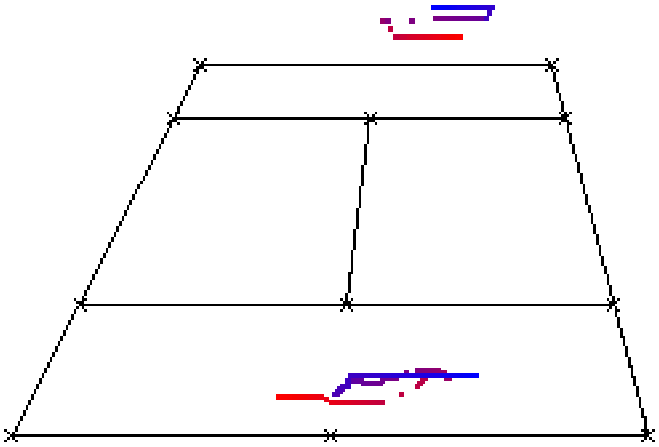

Figure 11: Pictorial presentation of the results of automatic analysis of *Baseline-rallies* video segment. The tennis court lines detected on the image domain are reconstructed and shown in black lines above. The tennis court-lines detected from only one frame are shown above since there is no camera motion in the video segment. (This is automatically detected by analysis given in Section 5.) The results of tracking the two tennis players over the image sequence are shown in color – their initial positions are shown in red and, as the play progresses and their locations change, the color changes gradually towards blue with their final location shown in blue. Note that the tracks of the players are near their respective *baselines* only. The high-level reasoning module rightly classifies this set of measurements as belonging to *Baseline-rallies* tennis-play event.

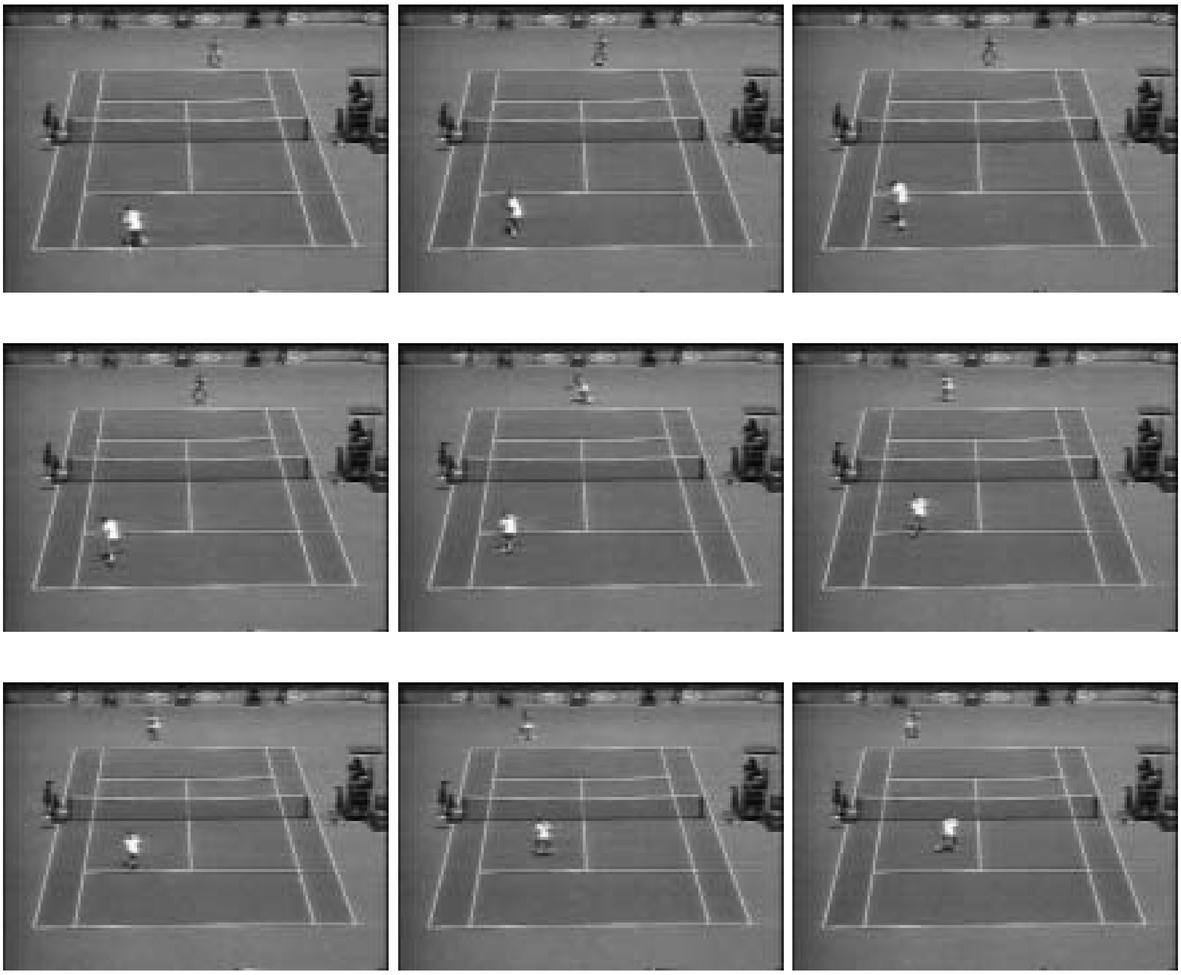

Figure 12: Some selected frames from the *Passing-shot* video segment consisting of 90 consecutive frames to visually depict the tennis-play event. The three rows of images correspond to the 3 successive chunks (each of 30 frames) in a top-down order. Each row shows the frames numbered 1, 15 and 29 in the corresponding chunk of 30 frames. The size of the frames is 320 × 240 pixels. The frames are shown at a scale slightly less than half the original size of the frames.

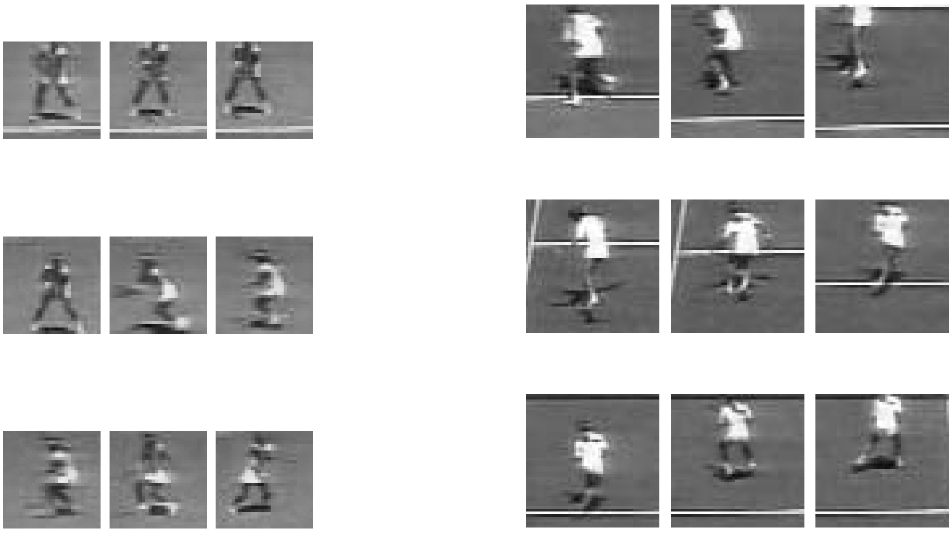

Figure 13: The templates of the players as they are tracked for 90 consecutive frames of the *Passing-shot* video segment. Shown on the left are the templates of the Top Player (TP), and those of the Bottom Player (BP) are shown on the right. The three rows of images correspond to the 3 successive chunks (each of 30 frames) in a top-down order. For each player, each row shows the templates etched out from the frames numbered 1, 15 and 29 in the corresponding chunk of 30 frames. The size of the TP templates is 30 × 30 pixels and that of BP templates is 50 × 50 pixels. The templates are shown at actual scale.

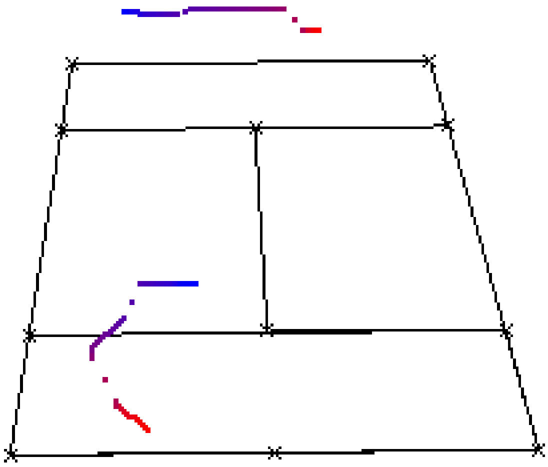

Figure 14: Pictorial presentation of the results of automatic analysis of *Passing-shot* video segment. The tennis court lines detected on the image domain are reconstructed and shown in black lines above. The tennis court-lines detected from only one frame are shown above since there is no camera motion in the video segment. The results of tracking the two players over the video segment are shown in color – their initial positions are shown in red and, as the play progresses and their locations change, the color changes gradually towards blue with their final location shown in blue. Note that the track of the bottom player (BP) moves towards the tennis net during the play while that of the top player (TP) stays essentially at the far-end baseline. The high-level reasoning module rightly classifies this set of measurements as belonging to *Passing-shot* tennis-play event.