Instantly Indexed Multimedia Databases of Real World Events

Gopal S. Pingali, *Senior Member, IEEE*, Agata Opalach, Yves D. Jean, and Ingrid B. Carlbom, *Member, IEEE*

***Abstract*—We introduce a new paradigm for real-time conversion of a real world event into a rich multimedia database by processing data from multiple sensors observing the event. Real-time analysis of the sensor data, tightly coupled with domain knowledge, results in instant indexing of multimedia data *at capture time*. This yields semantic information to answer complex queries about the content and the ability to extract portions of data that correspond to complex actions performed in the real world. The power of such an *instantly indexed multimedia database system*, in content-based retrieval of multimedia data or in semantic analysis and visualization of the data, far exceeds that of systems that index multimedia data only after it is produced.**

We present LucentVision, an instantly indexed multimedia database system developed for the sport of tennis. This system analyzes video from multiple cameras in real time and captures the activity of the players and the ball in the form of motion trajectories. The system stores these trajectories in a database along with video, three-dimensional (3-D) models of the environment, scores, and other domain-specific information. LucentVision has been used to enhance live television and Internet broadcasts with game analysis and virtual replays in more than 250 international tennis matches.

***Index Terms*—Broadcast, content-based retrieval, Internet, television, tracking, video, vision, visual data mining, visualization.**

I. Introduction

IN THIS paper, we address the following question: "Can a real world event be converted in real time into a multimedia database that enables rich interactive experiences of the real world event?" This question is motivated by several technology trends such as proliferation of sensors that can observe real world activity; ubiquitous availability of powerful computing devices and advances in computer vision and other sensor data processing techniques; digital convergence in networking (Internet, telephone, and broadcast networks) and access devices (PC, telephone, and TV) which facilitates remote interactive access to multimedia databases; and availability of computer graphics hardware that enables rich visualization of multimedia data. We propose that the time is ripe for the pursuit of a new class of databases that *instantly index* real world events.

Manuscript received April 21, 2001; revised February 28, 2002. This work was performed while G. Pingali, A. Opalach and Y. D. Jean were at Bell Labs.The associate editor coordinating the review of this paper and approving it for publication was Dr. Hong-Jiang Zhang.

G. S. Pingali is with the IBM T. J. Watson Research Center, Hawthorne, NY 10532 USA (e-mail: gpingali@us.ibm.com).

A. Opalach is at 505 chemin Rabiac Estagnol, 06600 Antibes, France (e-mail: agata.opalach@wanadoo.fr).

Y. D. Jean is with the Avaya Labs, New York, NY 10103 USA (e-mail: yves-jean@avaya.com).

I. B. Carlbom is with the Bell Labs, Murray Hill, NJ 07974 USA (e-mail: carlbom@lucent.com).

Publisher Item Identifier S 1520-9210(02)04860-5.

We define an instantly indexed multimedia database (IIMD) system of a real world event to be one that a) is created in real time as the real world event takes place; b) has a rich set of indices derived from disparate sources; and c) allows domain-specific retrieval and visualization of multimedia data. Notice that our definition of instant indexing emphasizes both real-time or online indexing, as well as capture of data and indices that support a user's domain-specific queries. Most multimedia database systems presented in the literature have been limited to offline indexing on a single stream of post-production material, and to low-level, feature-based indices rather than a user's semantic criteria. While many important methods have been developed in this context, the utility of these systems in real world applications is limited. We suggest that there is both significant opportunity and challenge in developing systems that index multiple streams in real time or *during production*, rather than post-production.

To further develop and exemplify this paradigm, we present LucentVision, an IIMD system developed for sports broadcasts, specifically for tennis. Sporting events are the most popular form of live entertainment in the world, attracting millions of viewers on television, personal computers, and a variety of other endpoint devices. Sports have an established and sophisticated broadcast production process involving producers, directors, commentators, analysts, and video and audio technicians using numerous cameras and microphones. Thus, there is significant opportunity for IIMD systems to become part of this production process and further engage and immerse viewers in the action, suspense, and drama of the remote live event.

LucentVision analyzes video from multiple cameras in real time, storing the activity of the players and the ball as motion trajectories. The LucentVision database also stores three-dimensional (3-D) models of the environment, broadcast video, scores, and other domain-specific information. The system allows various clients, such as TV broadcasters and Internet users, to query the database and experience a live or archived tennis match in multiple forms—for example, as 3-D virtual replays, visualizations of player strategy and performance, or as video clips showing customized highlights from the match. This system has been used in live TV and Internet broadcasts of more than 250 international tennis matches since 1998.

The rest of this paper is organized as follows. Section II discusses related work in video indexing and retrieval, real-time tracking, graphics, visualization, and spatio-temporal databases. Section III presents a generic architecture for an instantly indexed multimedia database system. Section IV discusses the instantiation of the architecture in the LucentVision system. We describe the real-time player tracking component of the Lu-

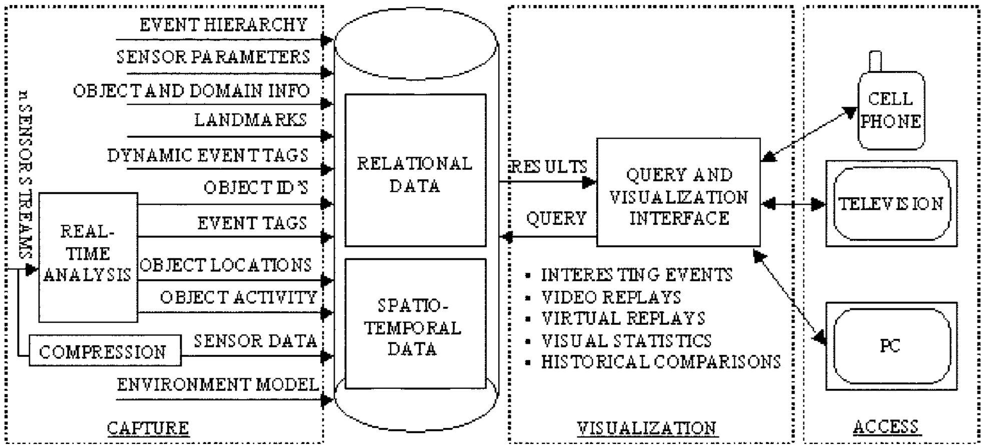


Fig. 1. Architecture of an IIMD system.

centVision system in Section V and the ball tracking component in Section VI. Section VII presents the query and visualization interface used in LucentVision. Section VIII presents a variety of visualizations made possible by LucentVision. We give special attention to the content based retrieval aspects of the system in Section IX. We end with a discussion on how the LucentVision system can be generalized to other sports and nonsport applications, and other challenges in realizing instantly indexed multimedia databases.

II. Related Research

Video indexing and retrieval are very active areas of research (see, for example, [1]–[4]). Much of this work deals with segmenting video and defining indices for efficient retrieval of video segments. Many popular techniques detect scene changes in broadcast or production video, thereby breaking video into shots, and representing each shot by a key frame. The characteristics of a key frame are then used for indexing and subsequent retrieval. A significant challenge is to translate a user's semantic indexing criteria into such low-level indices. We propose a different approach—instantly indexing multimedia data during capture to convert a real world event into a digital library in real time.

Tracking and analyzing the activity of people has a rich history (see, for example, [5]–[12]), a review of work in this area is provided in [13]. Real-time people tracking systems emerged only in recent years [14]–[17]. LucentVision uses dynamic clustering of local feature paths to derive people trajectories in real time. LucentVision is perhaps the most extensively tested people tracking system today.

In recent years, computer-generated visualizations are increasingly used in sports production to further enhance the viewer experience. Interactive visualization is becoming even more important with the ongoing convergence of television and Internet broadcasting. These trends have led to a significant amount of activity on sports analysis, visualization, and interactive video browsing [4], [18]–[25]. Most work related to sports visualization falls into two categories:

- overlay of virtual objects over video using augmented reality techniques—examples of these include the virtual first-down line in football or the virtual offside line in soccer ([23], [24]) and virtual superposition of video of two competitors taken at different times ([21]);
- generation of 3-D virtual replays which allow viewing of the action from any viewpoint—these include semi-automatic rendering of the action in a virtual environment ([18], [22]) and reconstruction of a dynamic three-dimensional model of the environment using numerous cameras (e.g., [26]–[28]).

However, the emphasis of these visualization techniques has primarily been on resynthesizing the sport, and not on deeper analysis of the sport. The sports viewer, commentator, analyst, player, or coach is often trying to obtain further *insight* into performance, style, and ultimately strategy. Our work addresses these issues.

LucentVision uses motion trajectories of objects to represent the spatio-temporal activity in a scene. The database community is recognizing the importance of motion trajectories of objects and spatio-temporal querying, especially in the context of geographic information systems and wireless localization based services. This is prompting extensions of databases to support motion trajectory based queries [29]–[31].

III. Architecture of an IIMD System

Fig. 1 illustrates the generic architecture of an IIMD system. Both static and dynamic information is captured into a database system and organized as relational and spatio-temporal data. While much of the data fits into a relational model, sensor streams, object activity data, and the environment model are not amenable to the relational model.

Dynamic information is derived mostly by real-time analysis of data from multiple disparate sensors observing real world activity. Sensor data streams are also stored in the database. Results of real-time analysis typically include identification of interesting objects (e.g., who is in the environment), their location, and activity (e.g., what are they doing, how are they moving). Real-time analysis can also result in detection of events that are interesting in a domain (e.g., someone reached their peak speed). However, the architecture does not limit generation of dynamic event tags to real-time analysis alone. Event tags may

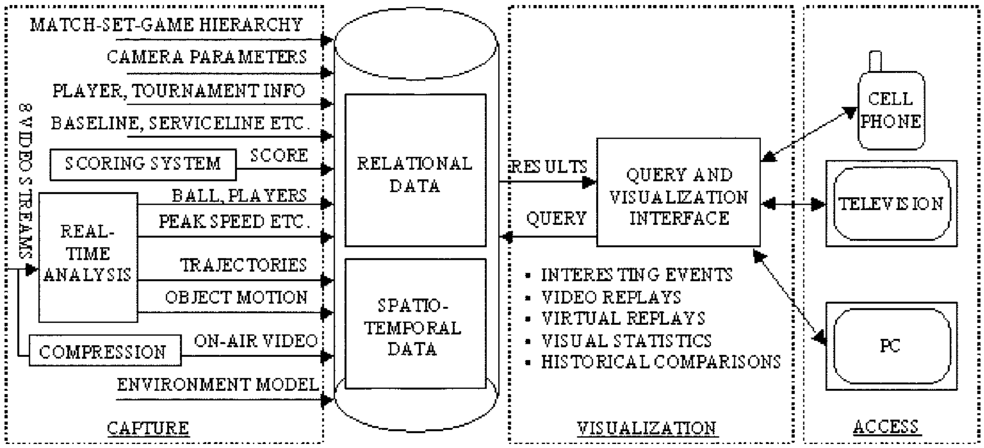


Fig. 2. Instantiation of the IIMD Architecture in the LucentVision system.

come even from semi-automated or manual sources that are available in an application domain, such as dynamic score data in a sports production setting.

An IIMD system incorporates domain knowledge in a variety of ways. First, design of tables in the relational database is based on the known event hierarchy, and known objects of interest. Second, the system maintains a geometric model of the environment, as well as the location of all sensors in relation to this model. Third, the system takes advantage of all available sources of information such as scores in the sporting domain. Fourth, design of the real-time analysis system is based on knowledge of the objects of interest in the domain. Sensor placement can also be based on domain knowledge. Finally, design of the visualization interface is based on knowledge of queries of interest in the domain.

The IIMD approach offers the following crucial advantages in data access and storage over traditional content-based media retrieval systems:

- real-time cross-indexing of disparate data (e.g., object position, speed, score, and video);
- storage of relevant data alone (e.g., only video when a person appears in a surveillance application, or only video when play occurs in a sports application).

IV. INSTANTIATION OF IIMD ARCHITECTURE IN LUCENTVISION

Fig. 2 shows the instantiation of the IIMD architecture in LucentVision. Synchronized video streams from eight cameras observing a tennis match feed into a domain-specific, real-time tracking subsystem. Two cameras are used for player tracking and six for ball tracking. The tracking subsystem outputs player and ball motion trajectories to a database, as discussed in Sections V and VI. In addition dynamic score data is obtained from a scoring system available in a tennis production. The tracking system assigns a player trajectory to the appropriate player by taking advantage of domain knowledge. It uses the rules of tennis and the current score to figure out which player is on which side of the court and seen by which camera.

LucentVision uses a relational database to organize data by the hierarchical structure of events in tennis [32]. A tennis "match" consists of "sets" which consist of "games" which, in turn, consist of "points." Each of these events has an associated identifier, temporal extent, and score. We associate trajectories $\mathbf{X}_{p1}(t)$, $\mathbf{X}_{p2}(t)$, $\mathbf{X}_{b}(t)$ corresponding to the two players and the ball with every "point," as "points" represent the shortest playtime in the event hierarchy. Each "point" also has pointers to video clips from the broadcast production. The relational database structure, with its SQL query language, provides a powerful mechanism for retrieving trajectory and video data corresponding to any part of a tennis match, as discussed in Section VII. However, the relational structure does not support spatio-temporal queries based on analysis of trajectory data. LucentVision has a spatio-temporal analysis structure built on top of the relational structure.

A visualization interface, which resides in a client device, performs queries on the database and offers the user a variety of reconstructions of the event as discussed in Section VIII. This interface is tailored to the computational and bandwidth resources of different devices such as a PC with a broadband or narrowband Internet connection, a TV broadcast system, or a next generation cellphone. LucentVision's greatest value comes from combining spatio-temporal queries with score based queries as discussed in Sections VII–IX. The system also has a video retrieval structure built on top of the relational database and performs compelling content based video retrieval as discussed in Section IX.

Reconstructions of the real world event range from high fidelity representations (e.g., high quality video) to a compact summary of the event (e.g., a map of players' coverage of the court). LucentVision produces broadcast grade graphics. It generates VRML models of the environment and its changes throughout an event. It also supports integrated media forms (i.e., video streams of event activity, VRML environments, and audio) using standards such as MPEG-4. Finally, the system produces low-bandwidth output such as scoring or event icons for cellphones and other devices.

V. TRACKING PLAYER MOTION

LucentVision uses visual tracking to identify and follow the players using two cameras, each covering one half of the court. Fig. 3 shows the typical view of a player tracking camera. The desired outputs of the player tracking system are trajectories, one per player, that represent the movement of the player. It

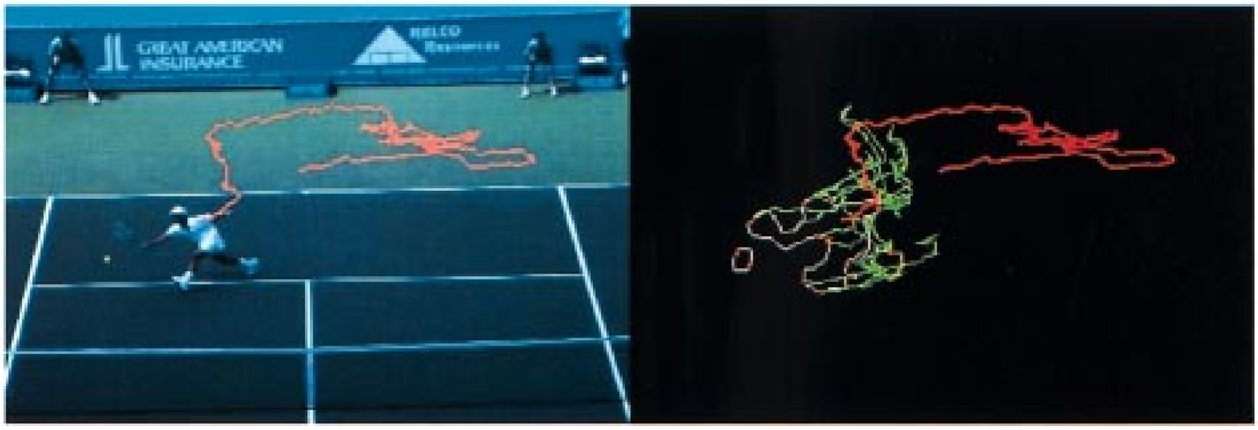


Fig. 3. (Left) Example player trajectory and (right) contours, features, and feature paths used to derive the final trajectory.

is challenging to obtain a clean segmentation of the player at all times. Differentiating the player from the background, especially in real time, is complicated by changing lighting conditions, wide variations in clothing worn by players, differences in visual characteristics of different courts, and the fast and nonrigid motion of the player. The central problem is that real-time segmentation does not yield a single region or a consistent set of regions as the player moves across the court. In addition, the overall motion of the body of the player has to be obtained in spite of the nonrigid articulated motion of the limbs.

In order to robustly obtain player trajectories, we track local features and derive the player trajectory by dynamically clustering the paths of local features over a large number of frames based on consistency of velocity and bounds on player dimensions. Fig. 4 summarizes the steps in the player tracking system. We extract the regions of motion by differencing consecutive frames followed by thresholding. This is a fast operation and works across varying lighting conditions. We use a morphological closing operation [33] to fill small gaps in the extracted motion regions. Thus,

$$B_t = (H_T(I_t - I_{t-1}) \oplus g) \ominus g \tag{1}$$

where B_t is a binary image consisting of regions of interest at time t, I_t is the input image at time t, H_T is a thresholding operation with threshold T, g is a small circular structuring element, and $\oplus$, $\ominus$ indicate morphological dilation and erosion operations. We do not have consistent segmentation of a moving player even after this operation. The number of regions per player change in shape, size, and number across frames.

Next, we determine local features on the extracted regions in each frame. The local features are the extrema of curvature on the bounding contours of the regions. The right portion of Fig. 3 shows an example of bounding contours of extracted regions (in white). The figure also shows the features as small red circles on the bounding contours. In the next step, we match features detected in the current frame with the features detected in the previous frame. This involves minimizing a distance measure D_f given by

$$D_f = k_r \delta r^2 + k_\theta \delta\theta^2 + k_\kappa \delta\kappa^2 \tag{2}$$

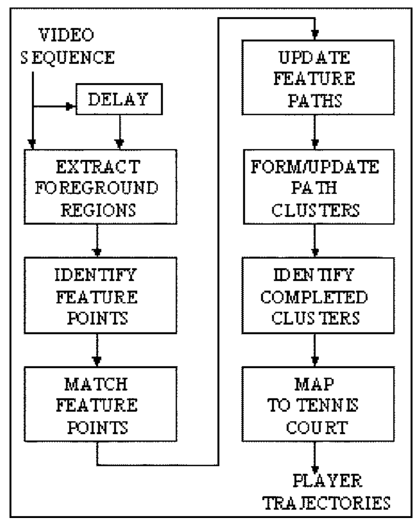


Fig. 4. Steps to track player motion in each frame of video.

where δr is the Euclidean distance between feature positions, $\delta\theta$ is the difference in orientation of the contours at the feature locations, $\delta\kappa$ is the difference in curvature of the contours at the feature locations and k_r, k_θ, k_κ are weighting factors. A feature path consists of a sequence of feature matches and indicates the motion of a feature over time. The parameters of a path $\mathbf{\Phi}$ include $\{\mathbf{x}, \mathbf{y}, \mathbf{t}, l, \mu_x, \mu_y, \sigma_x, \sigma_y\}$ where $\mathbf{x}, \mathbf{y}, \mathbf{t}$ are vectors giving the spatio-temporal coordinates at each sampling instant, l is the temporal length of the path, and μ_x, μ_y are the mean x and y values over the path and σ_x, σ_y are the variances in x and y values over the path. The right portion of Fig. 3 also shows a number of feature paths (in green) corresponding to the frame shown in the left portion of Fig. 3. There are numerous feature paths of varying lengths. These paths are typically short-lived and partially overlapping. In order to obtain the player trajectory, we dynamically group these paths into clusters as explained below.

At each time instant, we group feature paths with sufficient temporal overlap to form clusters. Multiple clusters are also grouped into a single cluster in a similar fashion. The parameters of a cluster $\mathbf{\Gamma}$ include $\{\mathbf{x}, \mathbf{y}, \mathbf{t}, \mathbf{f}, l, p, \mu_x, \mu_y, \sigma_x, \sigma_y\}$ where $\mathbf{f}$ is a vector that gives the number of features contributing to a cluster at each instant, p is the total number of paths contributing

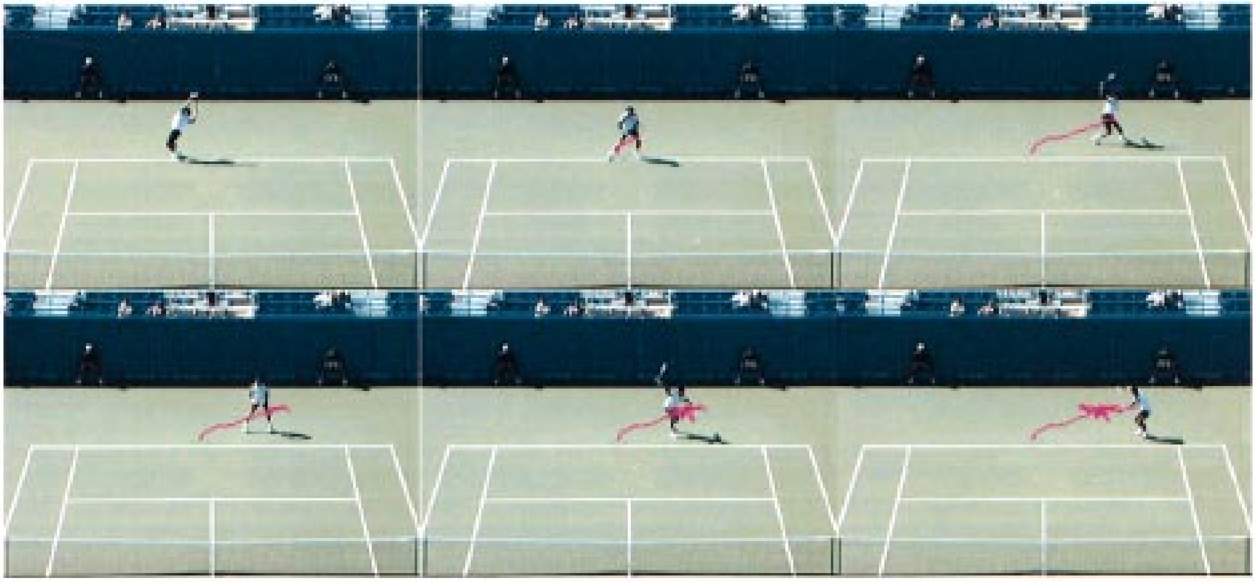


Fig. 5. Six frames showing the development of a player trajectory over the course of a point in a tennis match.

Fig. 6. Example of tracking multiple people in a different application.

to the cluster, (μ_x, μ_y) indicate the mean displacement of contributing features from the cluster coordinates and (σ_x, σ_y) indicate the variance in displacements. We group two clusters or a path and a cluster when they are close enough according to a distance measure D_Γ given by

$$D_\Gamma = \lambda_x \Delta\sigma_x + \lambda_y \Delta\sigma_y + \lambda_\tau \Delta\tau \tag{3}$$

where $\Delta\sigma_x$, $\Delta\sigma_y$ are the maximum change in variances of x and y displacements of features resulting from merging the clusters, $\Delta\tau$ is the normalized squared sum of the difference in orientations of the velocity vectors along the trajectories corresponding to the two clusters, and λ_x, λ_y, λ_τ are weighting factors based on bounds on the size of a player.

The clustering algorithm is capable of tracking several objects in real time (see Fig. 6). The motion of the body of the player results in a single dominant cluster in the tennis application. Motion of individual limbs of the player results in short-lived clusters that are distinguished from the main cluster. While it is not the focus of this paper, the smaller clusters can be analyzed to derive more information on the motion of individual limbs of a player or the motion of the racket. Sometimes, a player is not the only individual moving in the scene, even with the restricted view used in Fig. 3. Line judges also move, sometimes more than the players. We use domain knowledge on relative positions to distinguish player trajectories from those of line judges. We map player trajectories from the image plane to the court ground plane using camera calibration parameters [34].

We have extensively tested and used this tracking system [35] in numerous international tournaments played under a variety of lighting conditions (both outdoors and indoors) on different surfaces (concrete, synthetic, clay etc.). Tracking runs at 30 frames a second on a single processor such as an SGI MIPS R10000 or a Pentium III. Fig. 5 shows six frames from real-time tracking over the course of a point.

VI. Ball Tracking

Ball tracking is challenging because of the small size of the ball (67 mm in diameter), the relatively long distances it travels (over 26 m), the high speeds at which it travels (the fastest serves are over 225 kmph), changing lighting conditions, especially in outdoor events, and varying contrast between the ball and the background across the scene.

A. System Design and Configuration

The ball tracking system [36] uses six monochrome progressive scan (60 Hz) cameras connected to a quad-pentium workstation with dual PCI bus. Experiments on image resolution determined that a ball has to appear with a diameter of at least 10 pixels for reliable detection. As a result, we chose six progressive scan cameras with 640×480 pixels. The cameras cover the volume of the court and capture images with temporal resolution good enough for ball tracking and spatial resolution sufficient for identifying the ball. While cameras with an even higher speed and resolution could be used, we are limited by real-time processing and cost constraints. Monochrome cameras make the bandwidth of a dual PCI bus sufficient for concurrent full-frame capture at 60 Hz from all six cameras. Thus color, which is a strong cue for ball tracking, is sacrificed to meet other system constraints, making ball segmentation even more challenging.

The six cameras are placed around a stadium with four cameras on the side and two at the ends of the court. Each of the four side cameras is paired with one of the end cameras to form a set of four stereo pairs that track the ball in 3-D. Auto-iris lenses adjust to large lighting changes in the course of a day. Additionally, tracking parameters are dynamically updated, as explained in Section VI-C.

B. Multithreaded Tracking

With multithreaded tracking we achieve an efficient, scalable solution that works with distributed computing resources. Each camera pair has an associated processing thread. Fig. 7 gives an overview of the processing steps in each thread. A thread waits for a trigger signal to start frame capture and processing. Each thread has the following set of parameters: a trigger to start processing, pointers to a pair of associated cameras, calibration parameters of each camera, difference image thresholds, ball size parameters, expected intensity range for the ball, expected ball position in each image, size of the search window in each image, a trigger signal for the subsequent processing thread, and a pointer to the parameters of the subsequent thread. Prior to a match, we calibrate the cameras [34], taking advantage of the calibration grid provided by the court itself.

Upon receiving its trigger, a thread executes a loop of capturing frames from the camera pair, detecting the ball in the captured frames, stereo matching, and updating the 3-D trajectory and tracking parameters, until the ball goes out of view of any one of its associated cameras. At that time, the current thread initializes the parameters for the thread corresponding to the subsequent camera pair and triggers that thread.

This multithreaded approach scales in a straightforward manner to any number of cameras tracking an object over a large area. With a few modifications, the approach also scales to tracking multiple objects with multiple cameras. In this case, a thread associated with a camera pair (or set of cameras) has triggers associated with each object. The thread tracks an object when it receives a trigger signal corresponding to the object. Different tracking schemes can be used by a thread for different types of objects.

C. Ball Segmentation and Detection

We detect and segment the ball in an image by frame differencing the current and previous images and thresholding the result, finding the regions in the current image that lie in the expected intensity range for the ball, performing a logical AND operation of the regions obtained from the preceding two steps, subjecting the resulting regions to size and shape (circularity) checks, and choosing the detection closest to the expected position in the (rare) case of multiple detections. All these operations are performed only in a window defined by the expected ball position and search size parameters. Most parameters, such as the range of intensity values, expected size of the ball, size of the search window, and the differencing threshold, are dynamically updated during the course of tracking. The expected ball position is continually updated based on the current velocity of the ball.

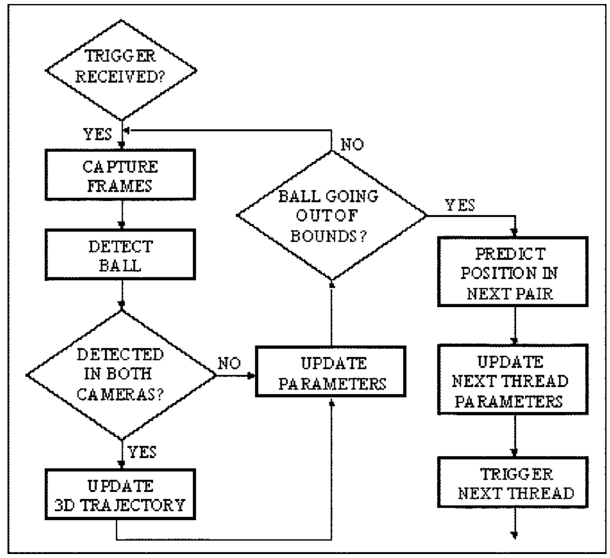


Fig. 7. Overview of processing in each ball-tracking thread associated with a camera pair.

Parameters such as the search size and range of intensity values are initially set to conservative values. The direction of the serve identifies and triggers the first thread. This thread initially has no expected ball position but a relatively large search window. We search for the ball in only one of the two camera feeds to ensure efficiency. Once the ball is detected in one camera, epipolar constraints determine the search region in the other camera. Once tracking commences, the search regions become much smaller and we use images from both cameras to detect the ball. When the current velocity of the ball indicates that the ball will be out of bounds of the current camera pair by the next frame, the current 3-D ball velocity and world to image mapping determine the positions of the ball in the next camera pair. Thus, once the initial thread starts tracking, subsequent threads look for the ball in well-defined search windows. The dynamic update of segmentation and tracking parameters are key to the success of this system.

D. Landing Spot Determination

Analysis of the 3-D ball trajectory, with appropriate interpolation, yields the ball landing spot for each serve. If the 3-D trajectory of length n has time samples $(t_1, t_2, \ldots, t_n)$, and the time sample t_c represents the last sample with a negative z velocity (computed from time t_{c-1} to t_c), then the landing spot is at a time t_l which is either between t_c and t_{c+1} or between t_{c-1} and t_c. In the first case, forward projection from the 3-D velocity and acceleration parameters at time t_c determine when the ball reaches the ground. In the second case, backward projection from the velocity and acceleration parameters at time t_{c+1} determine the landing location and time. We choose one depending on how well the velocity at the interpolated position matches the velocity at the tracked positions. Our experiments show that the choice is unambiguous.

E. Results and Accuracy Issues

Our system successfully tracked the ball on hundreds of serves at the 1999 World Championship in Hannover, the 1999

Fig. 8. Example of tracking the ball in a pair of cameras.

Paris Open, and the 2000 US Open. Fig. 8 shows an example of ball tracking in one pair of cameras. We achieve real-time tracking and the hand-off between cameras, or threads, is smooth. The trajectories are stored in a database along with other information as explained in Section IV.

We verify the accuracy of the system by several means.

- Analysis in the 2D image plane on a number of recorded sequences shows ball tracking accuracy to be within a pixel.
- Verification of the image to world mapping using a set of known points shows object space errors to be under 15 mm.
- Comparison of trajectories and landing spots with broadcast video sequences shows landing position error within 20 mm.
- Comparison of the service speed "at the racket" from tracking with the speed recorded by a radar gun shows that the difference is within 10 kmph. The tracking system estimates speed at racket as the speed of the ball at the earliest point in its trajectory. This point in the trajectory is first identified in a side camera view which then drives the search for the corresponding position in the end-camera view. Hence, there is lack of a precise match in the times at which the two speeds are measured.

Even better measures of accuracy can be derived using an independent high speed modality for capturing ground truth data.

VII. Query and Visualization Interface

A. Data Selection

Once a tennis match is stored in a database in the form of motion trajectories and domain-specific labels, the viewer can explore a virtual version of the event. This can be done even during a live match. To cope with the sheer volume of captured data, a powerful mechanism of data selection allows the user to choose only the subset of interest.

Fig. 9 shows examples of two interfaces for visualizations, one related to player tracking (left part), and the other to ball tracking (right part). For both player and ball tracking, the user selects a time window (e.g., a set or game), player(s), type of points (e.g., points won, aces), and spatial criteria (e.g., points at the baseline). LucentVision translates these selections into SQL queries and retrieves the corresponding trajectories. A preview of the retrieved trajectories appears in a window on top of the interface.

This selection procedure allows the user to formulate a wide variety of queries, including **score-based queries** (e.g., *all points won by a player against opponent's serve*), **statistics-based queries** (e.g., *the points a player ran fastest*), or **space-based queries** (e.g., *all points when the player was within five feet of the net*), or **hybrid spatio-temporal queries** (e.g., *all points in the first 20 minutes of the second set when Agassi won against Sampras' serve from behind the baseline*). In addition, LucentVision supports **historical queries** (e.g., *all matches between Sampras and Agassi in 1999 won by Agassi*). This has become a particularly important feature for tennis viewers, broadcasters, players, and coaches, as LucentVision has already captured over 250 international tennis matches.

B. Virtual Mixing Console

After selecting a data subset, the user has a set of tools for viewing and analysis. The concept of a *virtual mixing console* facilitates visualization selection, smooth transition between different visualizations, and combination of several visualizations. Selected visualizations share space in a visualization window. Some of the visualizations currently offered include: player motion maps, virtual replays of the ball, service landing positions, and serve statistics, each described in more detail in the following section. A new type of visualization can be easily added to this scheme.

VIII. Analysis and Visualization

The purpose of LucentVision is to provide the end-user with a variety of ways of experiencing and analyzing an event. In addition to retrieval of interesting video, discussed in Section IX, there are numerous ways in which the data in the LucentVision database can be visualized [37]. Although we have implemented

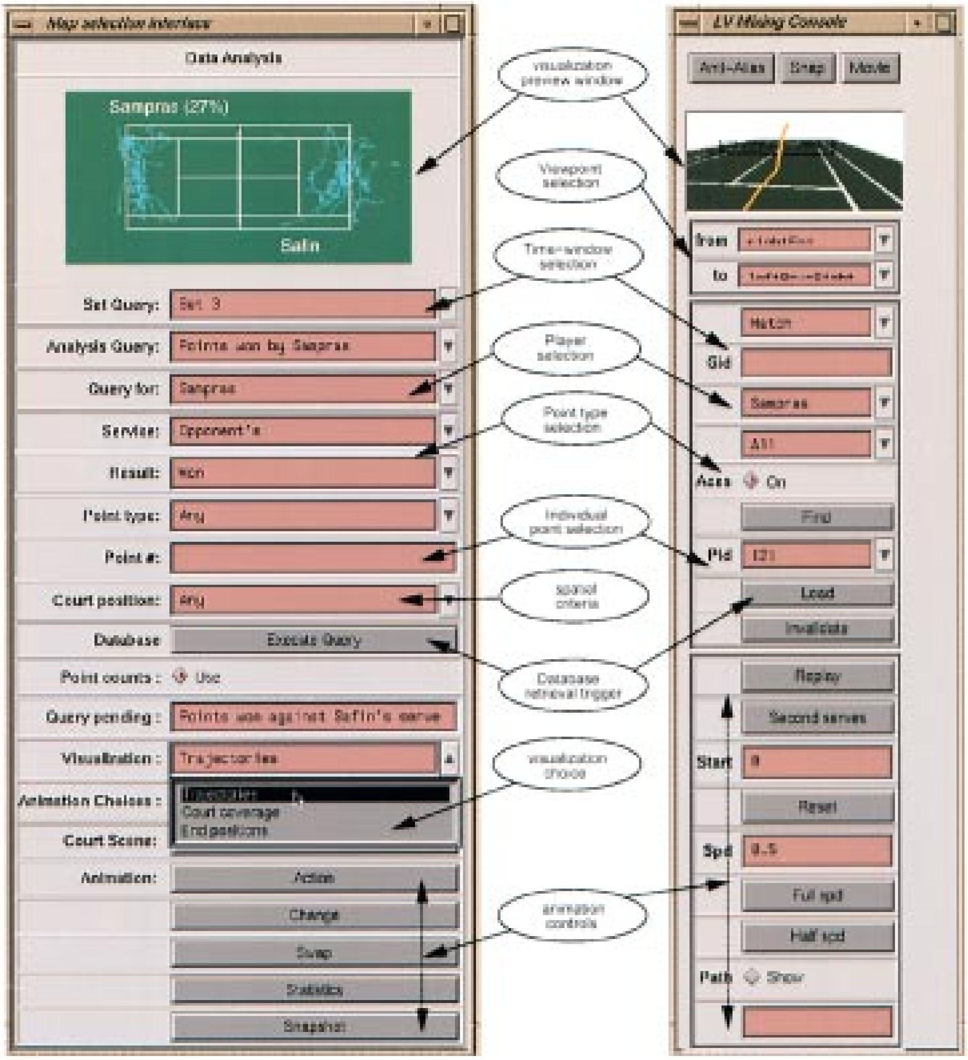


Fig. 9. The data selection and visualization interface.

many interesting visualizations, we frequently formulate new visualization schemes based on end-user feedback. The visualizations discussed below have been used hundreds of times in television broadcasts by more than 20 broadcast networks around the world. These analyses and visualizations have also been integrated into atptour.com, the website of the ATP (Association of Tennis Professionals).

A. LucentVision Maps and Statistics

The left portion of Fig. 10 shows all player motion trajectories, retrieved from the database and mapped onto the virtual court surface, for the 1999 semifinal match between Rafter and Kafelnikov in Cincinnati. The right portion of Fig. 10 shows a visualization derived from the trajectory data. In this *LucentVision Map* colors illustrate court coverage, red indicating the most frequently visited areas, followed by yellow, green and blue. The map immediately shows the general style and strategy of the players. Both players are mostly at the baseline but Rafter plays from behind the baseline while Kafelnikov plays more from inside the baseline. Rafter approaches the net more often than Kafelnikov.

Fig. 11 shows a more detailed analysis by dividing data from a match into two subsets. The left portion of Fig. 11 shows the map for points in a match (the World Championship in 1999) that are won by Sampras (and lost by Agassi), while the right portion shows points won by Agassi. A significant amount of information can be derived from these maps. Sampras wins 56% of the points while Agassi wins 44%. Sampras stays in the center of the court and spends very little time on the sides when winning. When losing, he spends a lot of time behind the baseline on the backhand side. We also see that Agassi is better able to advance from the baseline when winning than when losing.

In Fig. 11, all points in each trajectory have the same importance. More complex mapping algorithms are used to better illustrate players' strategies. Fig. 12 shows a mapping of the position of each player at the end of a point, at the deciding moment when the point is won or lost. The two player end positions for each point are linked by yellow lines. In Fig. 12, the selection is for the points won by Agassi against Sampras' serve in the second set of the 1999 World Championship match. The figure indicates that Agassi won 33% of the points in the second set for which Sampras served. The figure shows that five out of the

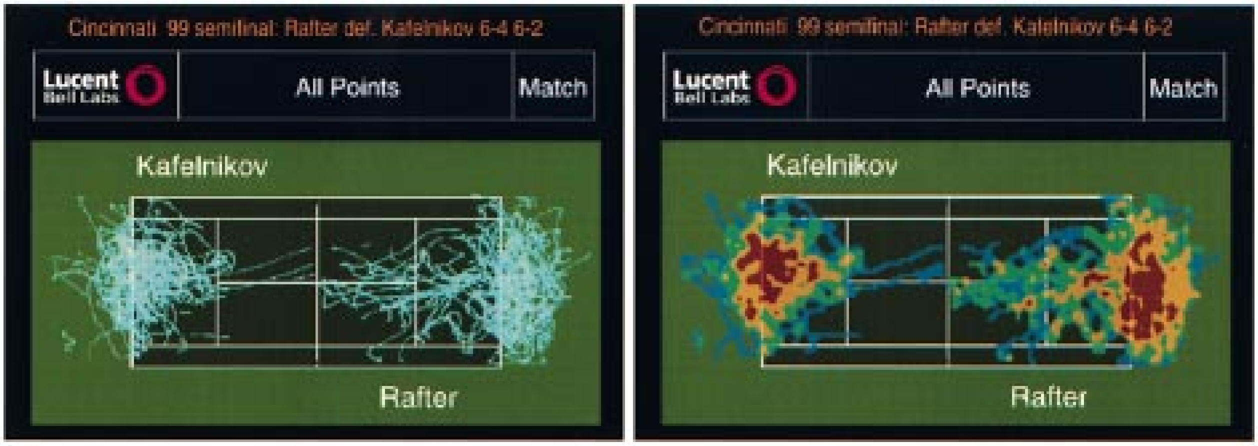


Fig. 10. Player trajectory data for a match and corresponding color-coded coverage map.

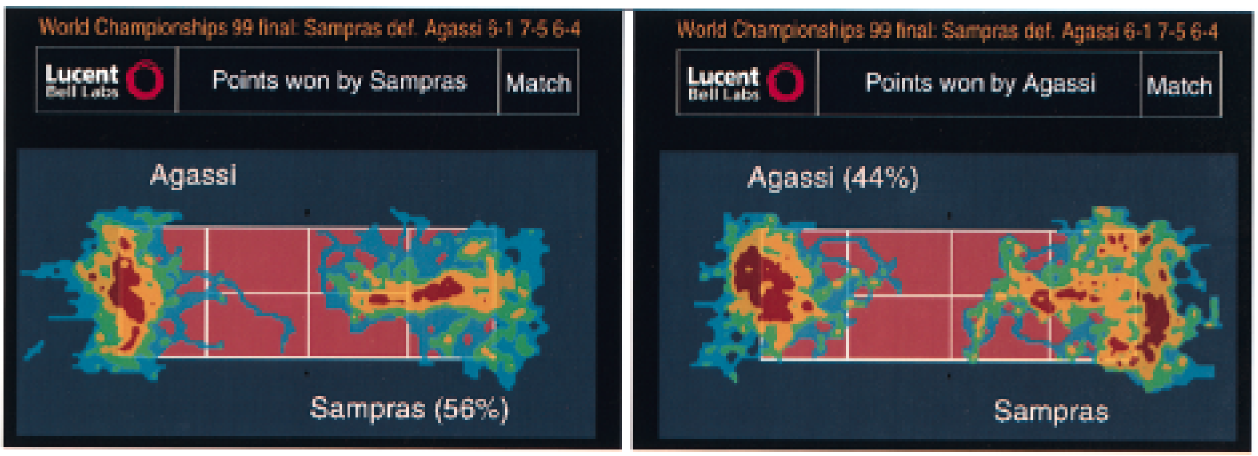


Fig. 11. Maps showing court coverage of each player during points they won.

Fig. 12. (Left) End positions of players and (right) the corresponding coverage map.

eight times that Sampras lost on serve, he was "caught" at the "T" in the middle of the court. The remaining three times, he was close to the net. The figure also indicates that Agassi hit most of his winning shots from the backhand side from close to the baseline by hitting passing shots as Sampras approached the net. More information is revealed by studying each pair of corresponding end positions in Fig. 12. The coverage map corresponding to this end position map is also seen in the same figure. The end position map reveals a significant amount of additional information that cannot be derived from the coverage map.

In this manner, the data selection power of LucentVision is used to reveal, in great detail, a player's strategy, strengths, weaknesses, and variations in strategy against different players and in different match situations. LucentVision also calculates numerical statistics such as distance travelled by players, and their average and peak running speeds.

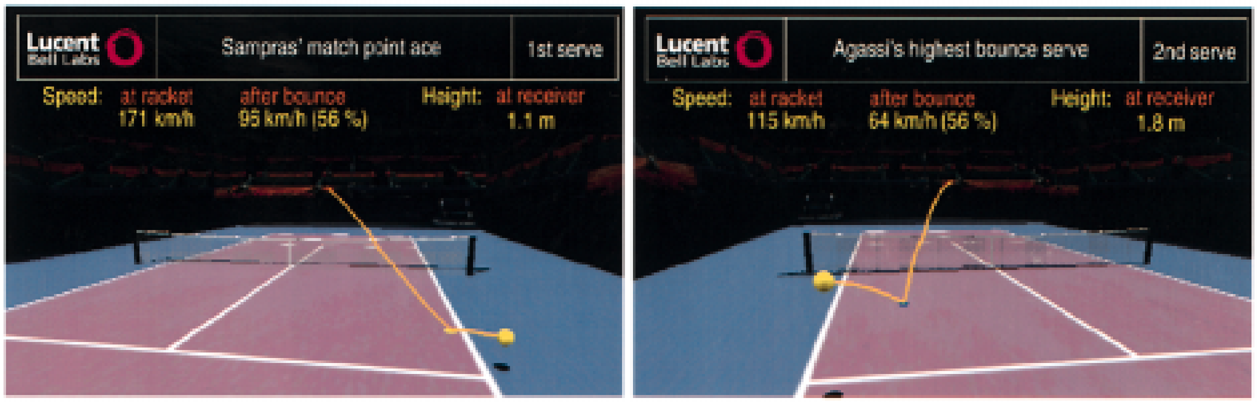


Fig. 13. (Left) Virtual replay of the fastest ace and (right) the serve with the highest bounce.

Fig. 14. Consecutive virtual replays of a sequence of serves.

Fig. 15. Maps showing serve landing spots of two players in a match.

B. Virtual Replays

Ball tracking enables other exciting visualizations. Any serve can be replayed from any point of view at any speed. For instance, a spectator can become the receiver of a serve and appreciate the dynamics of the game. The left portion of Fig. 13 shows a match-point ace served by Sampras in the final against Agassi during the 1999 ATP World Championships. As with player tracking, a large variety of statistics are available for each serve, including the ball's speed at the racket, its speed after bounce, and the height at which it reaches the receiver. The right portion of Fig. 13 shows the serve by Agassi which reached the greatest height at the receiver, in the same match. Comparing even these two examples shows the wide variety of serves in a tennis match.

C. Multiple Serve Visualizations

A sequence of superimposed serves reveal the serving style of a player or the differences in service styles among players. Fig. 14 shows superimposed trajectories from virtual replays of all aces served by Marat Safin during the final of the 2000 US Open against Pete Sampras. The viewpoint in this figure is that of the chair umpire. The viewpoint also helps the viewer to get an idea of how a chair umpire makes "linecalls" on whether the ball is in or out.

Another way of analyzing multiple serves is a *Service Landing Position Map* which gathers spots on the court where the players direct their serves. Fig. 15 shows landing position maps for the 1999 Paris Open finalists, Andre Agassi and Marat Safin. Yellow is used for first serves and blue for second serves. Agassi serves very precisely and consistently into the corners of the serving box while Safin's serves are more spread out with all his second serves going into the center.

The service patterns can be analyzed in more detail. Fig. 16 shows two maps for Safin for the same match, showing his serves won and lost. A careful viewer will notice that he often lost points when serving into the center of the serving box and far from the service line.

IX. CONTENT BASED VIDEO RETRIEVAL

LucentVision instantly indexes video from a tennis match with semantic information about the match, as discussed in Sections IV–VI. Hence, all the data selection power described in Section VII-A (including score-based, statistics-based, space-based, and hybrid spatio-temporal queries), applies to retrieval of video sequences of interest. Another major advantage of the LucentVision system is real-time operation: content-based retrieval of video is possible even during a live event. The system uses Universal Coordinated Time for temporal coordinates in all stored motion trajectories. Hence, in a television broadcasting

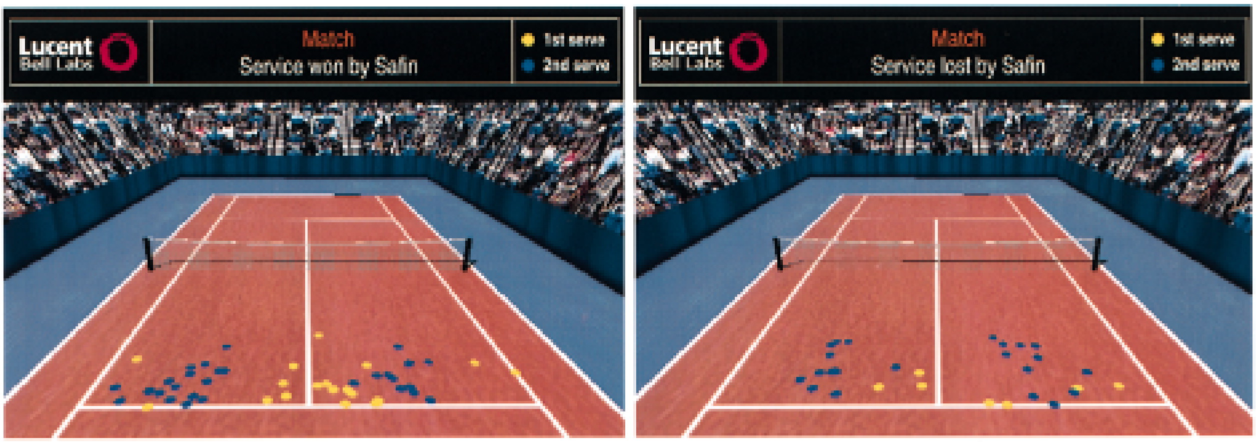


Fig. 16. Service landing positions maps for service points won and lost.

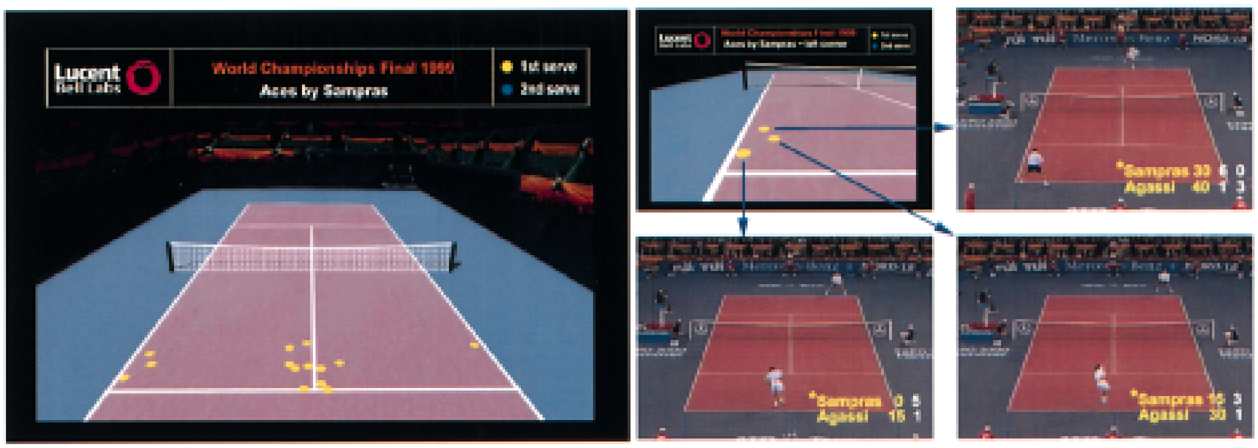


Fig. 17. Retrieving video for three aces. Left: map of landing spots of all aces served by Sampras. Second column top: selected region of the map. Frames from each of the retrieved video clips corresponding to the landing spots in the selected region are shown with corresponding scores.

context, LucentVision can use time codes to retrieve relevant video or highlights from recordings of any of the broadcast cameras. LucentVision can also drive video editing systems by automatically identifying sequences of interest.

A. Activity Map Based Indexing

LucentVision introduces a new concept of "activity map based indexing" [38] of video by combining the data selection power discussed in Section VII-A with the visualization power discussed in Section VIII. Spatio-temporal "activity maps" enable a user to view summaries of activity and *discover* interesting patterns. The user can then retrieve interesting video clips by using the *activity maps as a graphical user interface to the video*. As discussed in Section VIII, a variety of mapping choices are available to the user such as the court coverage of players, end positions of players, speed of the players, three-dimensional ball trajectories, landing positions, and speed of the ball at the receiver.

To enable activity map based indexing, LucentVision provides a media browser in conjunction with the map interface. The activity maps are temporal or spatio-temporal overlays on a model of the tennis court. Users may select specific regions of the court corresponding to activity of interest and may also modify their choices for events and mapping scheme to further refine their selection. Simultaneously, the media browser gives the user access to the corresponding video.

Fig. 17 shows an example where the user has selected the landing positions of serves as the activity mapping criterion and "all aces served by Sampras during the match" as the event based filtering criterion. As seen in the left portion of the figure, the map interface displays landing spots corresponding to the 14 aces served by Sampras in this match. The user selects the region corresponding to the left corner of the left service box, resulting in the selection of three landing spots shown in the top picture in the second column of Fig. 17. Frames from each of the corresponding three video clips are displayed by the media browser in the second and third columns. The video clips also show the corresponding game score. The three selected landing spots (top to bottom) correspond to a point in the fourth game of the second set, fifth game in the first set, and seventh game in the first set, respectively. Immediate and nonlinear access to these three different video clips is made possible by the map-based indexing. This example demonstrates the power of combining event selection and spatio-temporal activity display. Clearly, extracting such information directly from video would be very cumbersome.

Fig. 18. Retrieving video for fastest action. Left: speed map of two players and user's selection of a portion of the map. (Speed charts were derived by video analysis.) Right: four frames from the retrieved video clip.

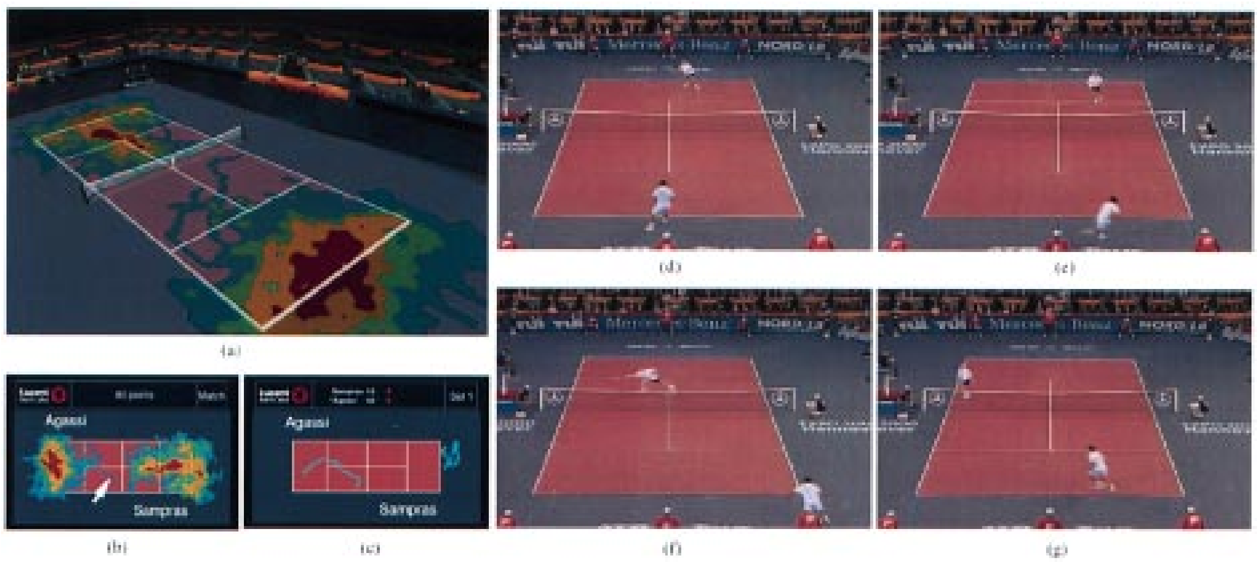


Fig. 19. Retrieving video for a particular trajectory. (a) Spatial coverage map for both players for the entire match. (b) Map viewed top down, with arrow indicating the user's selection of a portion of the map. (c) Trajectories of each player score corresponding to user's selection. (d)–(g) Four frames from the retrieved video clip corresponding to user's selection, shown clockwise.

Fig. 18 shows another example where "player foot speed" is the activity criterion, and the time window is the entire match. This results in an activity map of player speed over the course of the match as shown in the left side of Fig. 18. The user then selects a portion corresponding to a peak in the speed of one of the players. This results in the display of the score corresponding to that portion of the graph and display of the corresponding video in the media browser. Four of the video frames from this video clip are shown in the right side of Fig. 18. Once again, this is a query which would not have been possible without a visual activity map based indexing mechanism.

Fig. 19 shows a third example of activity map based indexing. In this case, "court coverage" is the mapping criterion, and the time window is the entire match. This results in the coverage map in Fig. 19(a), where the activity values are coded using four colors. The user then changes the view to look at a flat display of the court as in Fig. 19(b). In this case, the user notices that Agassi has approached the net only once in the match, and selects the corresponding region in the map [as indicated by the arrow in Fig. 19(b)]. This region selection results in a new activity map [Fig. 19(c)] showing the trajectories of both Agassi and Sampras for that one point in the match when Agassi approached the net. The score on this image indicates the fourth game in the first set, with Agassi serving, and the score in the game was 40–15 and the set score was 0–3. The media browser displays the video clip corresponding to this point. Four frames from this video clip are shown clockwise in Figs. 19(d)–(g). Each player traces the trajectories indicated in the image in Fig. 19(c). The user views this video clip and sees that Agassi lost the point that only time he approached the net!

X. Discussion and Conclusions

LucentVision exemplifies an emerging paradigm of instantly indexed multimedia databases that convert real world events in real time into a form that enables a new multimedia experience for remote users. Components of the experience include 1) immersion in a virtual environment where the viewer can choose to view any part of the event from any desired viewpoint and at any desired speed; 2) the ability to visualize statistics and implicit information that is hidden in media data; 3) the ability to search for, retrieve, compare and analyze content including video sequences, virtual replays and a variety of new visualizations; and 4) the ability to access this information in real time over diverse networks. LucentVision achieves this by following the architecture and design principles of an IIMD system outlined in Section III, especially incorporating domain knowledge such as event hierarchy, rules of the game, environment model, and sensor parameters.

How does this approach extend to other sports and nonsport applications? Moving to a different application involves a) setting up a relational database structure based on the event hierarchy for the domain; b) defining an environment model and sensor placement with respect to the model; c) developing real-time analysis modules that track dynamic activity of objects of interest; and d) designing a query and visualization interface that is tailored to the database structure and the domain. Sports applications have the advantage of a well-defined structure which makes it easier to extend this approach. For example, just as a tennis match is organized as a series of "points," baseball has a series of "pitches," basketball and American football have sequences of "possessions," and cricket has a hierarchy of "balls," "overs," "innings," etc. Thus, steps a), b), and d) above are relatively straightforward in moving to other sports, and to even less structured domains such as customer activity analysis in retail stores where the database can be organized in terms of entries and exits into different areas, time spent at different products etc.

The greatest challenge in developing IIMD systems for other applications is step c)—developing appropriate real-time analysis techniques. Hence, a good part of this paper focussed on the visual tracking techniques. Some of the issues are 1) tracking multiple people/objects; 2) dealing with occlusions; 3) identifying who is who; and 4) following more detailed actions. The player tracking approach presented here can track multiple people in real time as discussed in Section V. For example, Fig. 6 shows tracking of multiple people in a very different application [15]. A similar approach handles tracking in doubles situations in tennis matches. However, the authors did not focus on doubles due to lack of interest from broadcasters. Dealing with occlusions needs further extensions to the tracking technique in Section V, such as building appearance models during the clustering process. Person or object identification is a significant research problem. Camera placement and domain knowledge can solve the problem in several sports such as tennis, cricket and baseball. In general, identification will have to be performed based on appearance, numbers on uniforms, face recognition, or data from other sensors. Active transducers are possible in some applications, greatly simplifying the problem.

This paper presented a vision of IIMD systems and a concrete realization of the vision in LucentVision. We believe that there is great promise in this approach while a number of interesting research issues remain in realizing similar systems for a broad range of applications.

ACKNOWLEDGMENT

The authors are indebted to P. deTagyos and the Lucent Technologies Global Events Sponsorship Organization, the ATP, and numerous TV broadcast networks, who were all crucial to the conception, realization, and support of the LucentVision effort. They are grateful to management at Bell Labs and IBM Research for supporting this publication.

REFERENCES

[1] A. Hampapur, R. Jain, and T. Weymouth, "Production model based digital video segmentation," *J. Multimedia Tools Applicat.*, vol. 1, pp. 1–38, Mar. 1995.

[2] H. J. Zhang, J. Wu, D. Zhong, and S. Smoliar, "An integrated system for content-based video retrieval and browsing," *Pattern Recognit.*, vol. 30, Apr. 1997.

[3] M. Yeung and B.-L. Yeo, "Video content characterization and compaction for digital library applications," *Proc. SPIE*, vol. 3022, 1997.

[4] G. Sudhir, J. C. M. Lee, and A. K. Jain, "Automatic classification of tennis video for high-level content-based retrieval," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Databases (CAIVD'98)*, 1998.

[5] R. F. Rashid, "Toward a system for the interpretation of moving light displays," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 574–581, Feb. 1980.

[6] D. Hogg, "Model based vision: A program to see a walking person," *Image Vis. Comput.*, vol. 1, no. 1, pp. 5–20, 1983.

[7] K. Akita, "Image sequence analysis of real world human motion," *Pattern Recognit.*, vol. 17, pp. 73–83, 1984.

[8] V. T. Tsukuyama and Y. Shirai, "Detection of the movements of persons from a sparse sequence of TV images," *Pattern Recognit.*, vol. 18, pp. 207–213, 1985.

[9] K. Fukui, H. Nakai, and Y. Kuno, "Multiple object tracking system with three level continuous processes," in *IEEE Workshop on Applications of Computer Vision*, 1992, pp. 19–27.

[10] K. Rohr, "Toward model based recognition of human movements in image sequences," *Comput. Vis. Graph. Image Process.: Image Understand.*, vol. 59, no. 1, pp. 94–115, Jan. 1994.

[11] Q. Cai and J. K. Aggarwal, "Tracking human motion using multiple cameras," in *Int. Conf. Pattern Recognition*, 1996, pp. 68–72.

[12] D. M. Gavrila and L. S. Davis, "Toward 3-D model-based tracking of humans in action," in *Advances in Image Understanding, a Fetschrift for Azriel Rosenfeld*. Los Alamitos, CA: IEEE Computer Society Press, 1996, pp. 264–279.

[13] J. K. Aggarwal and Q. Cai, "Human motion analysis: A review," in *IEEE Nonrigid and Articulated Motion Workshop*, 1997, pp. 90–102.

[14] J. Segen and G. Pingali, "A camera based system for tracking people in real time," in *IAPR Int. Conf. Pattern Recognition*, 1996, pp. 63–67.

[15] G. Pingali and J. Segen, "Performance evaluation of people tracking systems," in *Third IEEE Workshop on Applications of Computer Vision*, 1996, pp. 33–38.

[16] C. Wren, A. Azerbayejani, T. Darrell, and A. Pentland, "Pfinder: Real time tracking of the human body," *Proc. SPIE*, vol. 2615, pp. 89–98, 1996.

[17] I. Haritaoglu, D. Harwood, and L. Davis, "W4 – A real time system for detecting and tracking people and their parts," in *Proc. Eur. Conf. Computer Vision*, 1998.

[18] T. Bebie and H. Bieri, "Soccerman – Reconstructing soccer games from video sequences," in *Proc. Int. Conf. Image Processing*, 1998.

[19] Y. Gong, L. T. Sin, C. H. Chuan, H. Zhang, and M. Sakauchi, "Automatic parsing of TV soccer programs," in *Proc. Int. Conf.. Multimedia Computing and Systems*, 1995, pp. 167–174.

[20] S. Intille and A. Bobick, "Visual tracking using closed worlds," in *Proc. Fifth Int. Conf. Computer Vision*, 1995, pp. 672–678.

[21] D. Kato, A. Ishikawa, M. Yamada, K. Abe, T. Suzuki, and H. Okamoto, "Development of a high-performance camera-control device and its application to tv programs," *SMPTE Journal*, pp. 718–723, October 1999.

[22] (2001) Virtual replay. Orad Hi-Tec Systems. [Online]. Available: http://www.orad.co.il/sport/index.htm

[23] (2001) Virtual game enhancements. PVI. [Online]. Available: http://www.pvi.com

[24] (2001) Sports enhancements. Sportvision. [Online]. Available: http://www.sportvision.com

[25] T. Taki, J. Hasegawa, and T. Fukumura, “Development of motion analysis system for quantitative evaluation of teamwork in soccer games,” in *Proc. Int. Conf. Image Processing*, vol. 3, 1996, pp. 815–818.
[26] T. Kanade, P. J. Narayanan, and P. Rander, “Virtualized reality: Concepts and early results,” in *Proc. IEEE Workshop on Representation of Visual Scenes*, 1995.
[27] P. Rander, P. J. Narayanan, and T. Kanade, “Virtualized reality: Constructing time-varying virtual worlds from real events,” in *Proc. IEEE Visualization*, 1997, pp. 277–283 552.
[28] P. Kelly, Katkere, Kuramura, Moezzi, Chatterjee, and R. Jain, “An architecture for multiple perspective interactive video,” in *Proc. ACM Multimedia*, 1995, pp. 201–212.
[29] M. Erwig and M. Schneider, “Developments in spatio-temporal query languages,” in *Proc. 10th Int. Workshop on Database and Expert Systems Applications*, 1999.
[30] D. Pfoser, C. Jensen, and Y. Theodoridis, “Novel approaches to the indexing of moving trajectories,” in *Proc. 26th Int. Conf. Very Large Databases*, 2000, pp. 395–406.
[31] Y. Theodoridis, M. Vazirigiannis, and T. Sellis, “Spatio-temporal indexing for large multimedia applications,” in *Proc. 3rd IEEE Conf. Multimedia Computing and Systems*, 1996, pp. 441–448.
[32] P. Douglas, *The Handbook of Tennis*. New York: Knopf, 1996.
[33] C. R. Giardina and E. R. Dougherty, *Morphological Methods in Image and Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[34] R. Y. Tsai, “An efficient and accurate camera calibration technique for 3D machine vision,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1986, pp. 364–374.
[35] G. Pingali, Y. Jean, and I. Carlbom, “Real-time tracking for enhanced tennis broadcasts,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1998, pp. 260–265.
[36] G. Pingali, A. Opalach, and Y. Jean, “Real-time ball tracking for innovative tennis broadcasts,” in *Proc. Int. Conf Pattern Recognition*, vol. 4, 2000, pp. 152–156.
[37] G. Pingali, A. Opalach, Y. Jean, and I. Carlbom, “Visualization of sports using motion trajectories: Providing insights into performance, style, and strategy,” in *Proc. IEEE Visualization*, vol. 4, 2001, pp. 75–82, 544.
[38] G. Pingali, A. Opalach, and I. Carlbom, “Multimedia retrieval through spatio-temporal activity maps,” in *Proc. ACM Multimedia*, vol. 4, 2001, pp. 129–136.

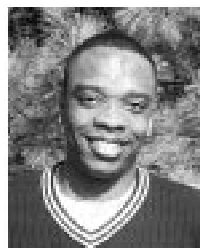
Gopal S. Pingali (S’92–M’95–SM’01) received the B.E. degree in electronics and communication engineering from Osmania University, Hyderabad, India, in 1989, and the M.S. and Ph.D. degrees in 1991 and 1993, respectively, both in computer science and engineering, from The University of Michigan, Ann Arbor.

He is currently a Research Staff Member in the Pervasive Computing Solutions group at the IBM T. J. Watson Research Center, Hawthorne, NY. Before joining IBM, he was Member of Technical Staff in the Machine Perception Research group, AT&T Bell Labs, Holmdel, NJ, from 1994 to 1996, and in the Visual Communications Research group at Bell Labs, Lucent Technologies, in Murray Hill, NJ, from 1996 to 2001. He has written over 30 research papers in the fields of computer vision, multimedia, computer graphics, microscopy, and robotics. He holds several patents in these areas. He has been a pioneer in developing real-time visual and audio-visual techniques for object tracking. He built a real-time camera-based people tracking system in 1994–1995, which was used by NCR, Inc., for customer activity analysis solutions in retail and financial environments. He invented player and ball tracking systems for LucentVision, the first real-time visual information system for sports broadcasts. His current research interests are in steerable interfaces and telepresence.

Dr. Pingali is an associate editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is a member of the ACM. Along with A. Opalach and Y. D. Jean, he received the biennial Best Industry Related Paper Award from the International Association for Pattern Recognition in 2000.

Agata Opalach received the B.Sc. degree in computer science from the University of Warsaw, Poland, in 1992, and the M.Sc. and Ph.D. degrees in computer science from the University of Sheffield, U.K., in 1993 and 1996, respectively.

For two years, she was a postdoctoral researcher at the iMAGIS research group, Grenoble, France, continuing her research into modeling and animation using implicit surfaces. She then spent a year at the European Synchrotron Facility, Grenoble, creating scientific software for parallel 3-D tomographic reconstruction, 3-D data visualization and manipulation with quantitative measures, and iterative data segmentation using a deformable implicit snake model. From 1999 to 2001, she was with the Visual Communications Research Group at Bell Labs, Lucent Technologies. She worked on visualization and data mining for LucentVision, the first real-time networked multimedia information system for sports action capture, virtual experience, and analysis. She was also part of the team developing error resilient video coding for the wireless channel. She recently moved to France, where she is exploring new career opportunities pursuing here research interests in 3-D object modeling, computer animation and visualization, immersive environments, and real-time multimedia systems.



Yves D. Jean received the B.S. degree in electrical engineering from Columbia University, New York, in 1988 and the Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1996.

From 1996 to 2000, he was a Research Scientist in the Visual Communications Research Group, Lucent Bell Laboratories, Murray Hill, NJ. He is currently a Research Scientist in the Collaborative Applications Group, AvayaLabs, Avaya, Inc., Basking Ridge, NJ. He has been a researcher in the area of computer graphics, multimedia, scientific visualization, and signal processing. He has published work dealing with the representation and control of dynamic data and models. His work includes real-time medical imaging and very large dataset visualization. He has made contributions in 3-D graphical models for steering and visualizing computational models. He was the creator of the LucentVision project at Bell Labs and is currently working in the area of multimedia systems within IP-telephony networks.

Ingrid B. Carlbom (M’88) received the Ph.D. degree in computer science from Brown University, Providence, RI, the M.S. degree in computer science from Cornell University, Ithaca, NY, and the Fil.Kand. degree from the University of Stockholm, Sweden.

She is Director of Visual Communications Research in the Multimedia Communications Research Laboratory at Bell Labs, Murray Hill, NJ. Formerly, she was Manager of Visualization Research at Digital Equipment Corporation’s Cambridge Research Lab. From 1980 to 1986, she was a Member of Professional Staff at Schlumberger-Doll Research, Ridgefield, CT, where she also served as Program Leader for the Graphical Modeling Program. Her research interests include telepresence and virtual environments, acoustical modeling, video compression, scientific visualization, and biomedical imaging.

Dr. Carlbom was a Director of SIGGRAPH from 1982 to 1986 and the chair of the SIGGRAPH Advisory Board from 1986 to 1988. She served on the *IEEE Computer Graphics and Applications*’ editorial board from 1985 to 1989, on the IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS editorial board from 1995 to 1998, and is currently a member of the editorial board for *Computers & Graphics*, and is Co-Editor-in-Chief of *Graphical Models*. She was a member of the Advisory Board for the NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization from 1991 to 2001, and serves on the Board of Directors of the NYU Center for Graphic Communications Management and Technology. She has been awarded a Doctor of Philosophy Honoris Causa by Uppsala University, Sweden, and a Distinguished Graduate School Alumna Award by Brown University.