Tennis Real Play: an Interactive Tennis Game with Models from Real Videos

Jui-Hsin Lai, Chieh-Li Chen, Po-Chen Wu, Chieh-Chi Kao, and Shao-Yi Chien

Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University, Taipei, Taiwan
{juihsin.lai, chiehli.chen, pochen.wu.tw, chiehchi.kao}@gmail.com, sychien@cc.ee.ntu.edu.tw

ABSTRACT

Tennis Real Play (TRP) is an interactive tennis game system constructed with models extracted from videos of real matches. The key techniques proposed for TRP include player modeling and video-based player/court rendering. For player model creation, we propose a database normalization process and a behavioral transition model of tennis players, which might be a good alternative for motion capture in the conventional video games. For player/court rendering, we propose a framework for rendering vivid game characters and providing the real-time ability. We can say that image-based rendering leads to a more interactive and realistic rendering. Experiments show that video games with vivid viewing effects and characteristic players can be generated from match videos without much user intervention. Because the player model can adequately record the ability and condition of a player in the real world, it can then be used to roughly predict the results of real tennis matches in the next days. The results of a user study reveal that subjects like the increased interaction, immersive experience, and enjoyment from playing TRP.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Statistical database*; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithm, Design, Experimentation

1. INTRODUCTION

In recent years, a number of interactive videos have been proposed on the Internet. For example, there are now several such videos of magic shows on YouTube. These videos are unique in that users not only watch the videos but also participate in the show to guess the answers to magic tricks. It is our opinion that such interactive videos can engage user interest, and that the concept of interactive videos can be further extended.

In this paper, we propose Tennis Real Play (TRP), an interactive tennis game constructed from videos of real matches. TRP is inspired by advances in video analysis/annotation, video-based rendering, and interactive sports games. Previous relevant work is reviewed below.

Video Analysis/Annotation, like event annotation and content segmentation, is a popular and important topic because of the dramatic increase in the number of videos. Many previous studies on this issue have been published. For extraction of sport videos, Wang et al. [25] proposed a method to analyze video structure and automatically replay highlights of soccer videos. Wang and Parameswaran [26] analyzed the ball trajectories in tennis matches to detect player tactics for video annotation, and Zhu et al. [28] proposed recognition of player actions in tennis videos for event annotation. For content segmentation of tennis videos, Lai et al. [13] proposed methods for separating and reintegrating content to enrich videos of tennis matches. By applying these previously studied methods, a large number of videos can be organized so that users can immediately find highlights in a long video.

Video-based Rendering, which can be described as an extension of image-based rendering, is a method to rearrange video frames to create a new video. In studies of video-based rendering, Schodl et al. [23] proposed Video Textures to automatically cut videos into clips and re-loop them to create continuous animations. Efros et al. [2] proposed methods to recognize and classify player actions in football videos. Using such clips of player actions, new actions can be synthesized. Inamoto and Saito [9] rendered a free-view football game from videos from multiple cameras, with the free-view synthesis yielding a fresh viewing experience. Lai et al. [12] proposed Tennis Video 2.0, which rendered multiple players in continuous motion at the same time to create a more interesting viewing experience.

Interactive Sports Games are a popular form of entertainment. In particular, the interactive tennis game in Wii Sports [1] is one of the most well-known among such games. Innovative user interfaces such as those of Wiimote and Wii Fit provide a more realistic gaming experience and changes the way people play games. Not surprisingly, Wii has successfully engaged users throughout the world. In contrast to the interactivity in Wii, Play Station 3(PS3) places more

*Area chair: Pal Halvorsene

MM'11, November 28–December 1, 2011, Scottsdale, Arizona, USA.

[1]Wii Sports. http://wiisports.nintendo.com/

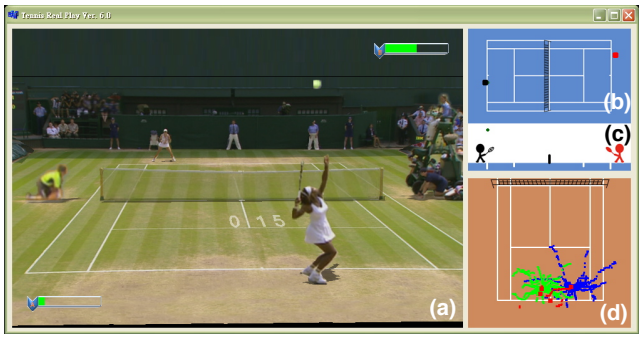

Figure 1: The user interface consists of four components: (a) the main screen of the rendering result, (b) the coordinates of the players and ball on XY-axis, (c) the coordinates of the players and ball on YZ-axis, and (d) motion paths in the player database.

emphasis on visual quality. Top Spin 3 [2] is one of the tennis games on PS3, and the realistic player postures and lighting effects offer the user a more vivid viewing experience. Inspired by both games, one of the aims of this study is to build a tennis game that offers an interactive experience similar to that of Wii along with vivid video texture.

TRP is an interactive tennis game system constructed with models extracted from videos of real tennis matches. As shown in the game frame in Figure 1(a), the textures of the players and the background court are extracted from real videos of matches, and player postures are immediately rendered according to the user's control. To implement TRP, we propose a system framework consisting of player model creation and player rendering, and a video with system overview is available on the website[3]. When creating a player model, projection to fiducial coordinates is proposed to normalize object sizes and the motion trajectories of segmented players. Next, a 4-state-transition model is proposed to model tennis players' behaviors. For player rendering, we propose methods to select suitable clips (moving/hit/standby) from the database. The most interesting and unique feature of our player rendering is that the movement abilities and hitting strength of a rendered player will depend on these clips and statistics in the videos of real matches. Subsequently, we construct a 3D model of the tennis court from the background image and build the game system on this model. As shown in Figures 1(b) and (c), the player's state is recorded in 3D coordinates and the game frame is rendered with a virtual camera. By combining the 3D model with techniques in video-based rendering, the proposed system can render game frames in different viewing angles. The contributions of this paper are listed below.

- To the best of our knowledge, this is the first work to integrate video-based rendering and interactive sports game, and all the rendering characters perform with the characteristics of the players in the real world. We can say that image-based rendering leads to a more interactive and realistic rendering.

[2]Top Spin3. http://www.topspin3thegame.com/
[3]Demo videos. https://sites.google.com/site/juihsinlai/trp

- Conventional video games utilize expensive motion capture systems to build the player database which results in rather fixed action sets. Moreover, the player model cannot be updated often. The presented approach might be a good alternative for these methods. The users can even play with new players recorded in the sport video or their friends as the player in the home-made video.

- The game results of TRP can reflect the match results in real world because the game characters can adequately record the abilities and conditions of players in the match video. This property can then be used to roughly predict the results of real tennis matches in the next days.

- We provide viewers a new way to enjoy sports videos; viewers not only enjoy the match by watching videos but also have more immersive experience from playing match videos. In other words, a new match content can be integrated to the game as soon as a new match video is available without much user intervention.

The remainder of this paper is structured as follows. Section 2 describes in detail the generation of the player database, extraction of statistics, and player modeling. Clip selection, smoothing transitions, and system integration for player rendering are described in Section 3. Section 4 presents experimental results along with discussions. Finally, the major findings of the paper are summarized in Section 5.

2. PLAYER MODEL

Player model creation is a key component in TRP because player behaviors in the game including hitting preference and movement characteristics are controlled by the model. Each player has a unique player model in different tennis games. Consequently, the essence of this study is the creation of player models from videos of real matches.

2.1 Player Database

The first step in building the player database is to segment the player from the videos. The possible camera motions in tennis videos—panning, titling, and zooming—make the segmentation process quite difficult. Several previous studies have examined this problem. Lai et al. [14] projected video frames onto a sprite plane for background image generation, which was used for the segmentation of foreground objects. Han et al. [5] proposed a camera calibration module that employed the geometric layout in the form of a court model, which could then be used for player segmentation. The player database not only includes segmentation masks but also records where a player stands in each video frame. Nevertheless, the scale of each mask and position in the player database changes as a result of camera zooming or panning. Therefore, a procedure is required to normalize the database. For this purpose, we propose a fiducial coordinate system upon which all database information is projected. The fiducial plane is the tennis court in a fiducial coordinate system from bird's eye view, as shown in Figure 2.

Next, we assume that the bottom coordinates of the player mask (P_x, P_y) specify the position of the feet on the court. The normalized coordinates are presented as $(P_x^{'}, P_y^{'})$ by pro-

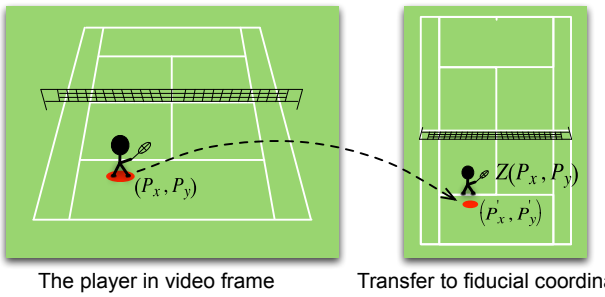

Figure 2: Illustration of player position and size normalization.

jecting (P_x, P_y) onto the fiducial coordinate system according to the following equations.

$$P_x' = G_x(P_x, P_y) = \frac{m_0 P_x + m_1 P_y + m_2}{m_6 P_x + m_7 P_y + 1}, \quad (1)$$

$$P_y' = G_y(P_x, P_y) = \frac{m_3 P_x + m_4 P_y + m_5}{m_6 P_x + m_7 P_y + 1}, \quad (2)$$

where m_0 to m_7 are the parameters of homography projection. As players are extracted from video frames, the sizes of the segmented players will be influenced by camera parameters and a player's position on the court. Therefore, two factors are considered to normalize player size: the projection parameters of the camera and the player's court position. The zooming rate parameters of the players can be estimated by calculating the deviation of $G_x(x, y)$ and $G_y(x, y)$ with respect to the horizontal and vertical zooming factors on the coordinates (P_x, P_y), respectively. Therefore, the normalized player size $Z(P_x, P_y)$ is obtained by multiplying the horizontal and vertical zooming factors, as expressed in the following equation.

$$Z(P_x, P_y) = \frac{\partial G_x(P_x, P_y)}{\partial P_x} \frac{\partial G_y(P_x, P_y)}{\partial P_y}. \quad (3)$$

As shown in Figure 2, the normalized player position and size can be calculated with perspective parameters and the player's position.

2.2 Hit Statistics

After building the database, it is necessary to extract the statistics of a player's characteristics in videos of their tennis matches. Determining a player's positions at the time of hitting is the key step in extracting this information because each hit strength and direction can be detected by the connection of positions at the time of hitting. To detect the time at which the player hits the ball, audio detection and ball trajectory analysis are applied. On the one hand, the hit sound is detected efficiently by the peak amplitude in an audio signal when the background noise is low. However, the hit sound is occasionally masked by an audience's cheer or a broadcaster's voice. To detect the hit time under loud background noise, information on ball trajectories is used instead of audio signals. According to the method of detecting the tennis ball trajectory in [13], the state of the ball can be modeled and used for hit time detection.

After their strengths and directions are detected, the hits must be further categorized. The hit categories include forehand volley, backhand volley, forehand stroke, backhand stroke, and drop shot. However, it is difficult to identify each hit category from videos with a single camera view. To simplify the identification process, two hit categories, forehand and backhand, are first labeled by the program. Several previous studies have investigated this problem. For example, Roh et al. [21] proposed a curvature scale space (CSS) to describe the characteristic shape features of the extracted players and used it to detect hit categories. However, we find that features in CSS cannot accurately identify hit categories in our experiments because players' shapes are dissimilar, and it is difficult to find common features in the same hit category. Using the methods in [28], we calculate and sum the values of optical flow on the right and left sides of the player. If the optical flow on the right side of the player is larger than that on the left side, the hit posture is regarded as a forehand. Otherwise, the hit posture is regarded as a backhand. The strength of forehand and backhand hits in various directions will be considered in Section 3.1.2.

Once forehand and backhand hits are categorized, we use the player position as a clue to identify volleys and strokes. The posture is identified as a volley if the player's position is close to the net and a stroke if player's position is far from the net. Although these assumptions are not always correct, most of the time they are true. In addition, the classification rates between stroke and volley can be improved by more information from different viewing angles of cameras. In sum, after detecting the hit time and identifying the hit category, each hit posture is classified as a forehand stroke, backhand stroke, forehand volley, or backhand volley.

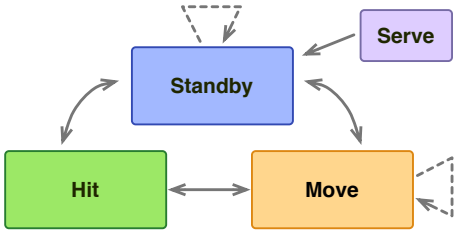

Figure 3: Proposed four-state-transition model for tennis player behaviors.

2.3 Behavior Model

Even after the player database has been constructed and the hit statistics generated, it remains difficult to vividly render a player's behavior without a behavior model. In previous studies of video-based rendering, Schodl and Essa [22] and Colqui et al. [1] synthesized new video clips by rearranging frames in the original video sequences. Both studies presented ingenious schemes to create new motions by displaying the original video clips in a different order. However, most of the objects in their test videos have simpler structures such as those of hamsters, flies, or fish. No human videos were used because human behaviors are more complex. For video-based rendering with human motions, Phillips and Watson [20] introduced template sequences and performed two forms of data-based action synthesis: "*Do as I Do*" and "*Do as I Say*". Although the results were impressive, the system could only synthesize actions based on templates of existing motion sequences. Furthermore, Flagg et al. [3] presented photo-realistic animations of hu-

man actions. By pasting numerous sensors on a human subject's body to detect motion, this study overcame many challenges encountered in synthesizing human actions and achieved vivid rendering effects. Nevertheless, this approach is not suitable for the present application.

In our opinion, a model that simulates the behavior of tennis players is needed. Therefore, we propose the four-state-transition model shown in Figure 3. The four states are *Serve*, *Standby*, *Move*, and *Hit*. The arrows in the figure represent the allowable state transitions. All the behaviors of tennis players during a match can be expressed by these state transitions. For example, a player may move right, wait to hit, and then perform a forehand stroke—a sequence which can be modeled by state transitions from *Move*, to *Standby*, to *Hit*. In the case of serving, a player may serve, run to a standby position, wait for the opponent's return, run to a hitting position, and then hit—a sequence which can be modeled as *Serve*, *Move*, *Standby*, *Move*, and *Hit*. As in the intra-state transitions for *Standby* and *Move* shown by the dotted lines in Figure 3, the same state may recur several times in rendering, the associated details of which are described in Section 3.1. Under this behavior model, the player database in Section 2.1 is further classified into four categories. As noted in Section 2.2, four different hitting postures—forehand stroke, forehand volley, backhand stroke, and backhand volley—are collected in the *Hit* category. Subsequently, actions between two hits are classified in the *Move* category. Action clips without movement in the *Move* category are then collected to form the *Standby* category.

3. PLAYER RENDERING

In this section, methods of player selection are proposed in which suitable clips are selected from the database to render the various motions and postures of players. For seamless connection between clips, the proposed approaches can smooth transitions in the player's shape, color, and motion. Subsequently, the game system is integrated with the rendered background court and foreground objects.

3.1 Database Selection

In previous studies focusing on video-based rendering, Schodl et al. [23] and Schodl and Essa [22] extracted textures of a video and synthesized new video clips by rearranging frames in the original video sequences. In both studies, the measure of similarity for video synthesis is the pixel difference between two frames. However, the methods in these studies were not suited for human rendering because the complexity of human images is higher than that of natural scenes used in previous studies. In previous studies of video-based rendering in humans, Kitamura et al. [11] used greater numbers of similarity measures such as motion vectors and contours, and both studies achieved more vivid visual effects. These studies should encourage the consideration of more similarity measures for better visual effects in human rendering. Nevertheless, the challenges are greater in player rendering because unlike the dedicated human videos from users captured in [3] and [11], the player database is constructed from videos of matches. In other words, the database may be incomplete for some postures, and thus lead to more difficult rendering. Therefore, in the proposed method for player rendering, the first step is to select appropriate player clips.

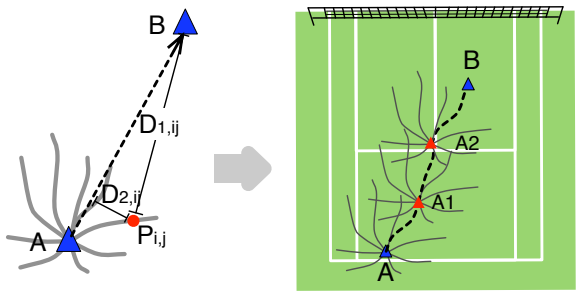

Figure 4: (a) Selecting a suitable motion clip to form a movement path from position A to position B by computing similarity. (b)Three motion clips forming a movement path.

3.1.1 Clip Selection for Player Moving

Suppose a motion path is to be rendered from position A to position B as illustrated in Figure 4(a), and that the best path is the shortest one (dotted line). Occasionally, it is difficult to find a clip from the player database that fits the dotted line, and multiple moving clips must be connected to form the moving posture. A critical problem is how to choose suitable moving clips to achieve seamless connections. Suppose there are n moving clips, and moving clip i, $i \in [1, n]$, is composed of l_i frames. $P_{i,j}$ is the player position in frame j of moving clip i in fiducial coordinates, $j \in [1, l_i]$. All player positions can form the possible movement trajectories such as those shown by the grey lines in Figure 4(a). The process of moving clip selection is to choose a motion path $[P_{i,1}, ..., P_{i,j}]$ which connects positions A and B. Three primary factors should be considered in moving clip selection: distance $\mathbf{D_{1,ij}}$, distance $\mathbf{D_{2,ij}}$, and index j.

- $\mathbf{D_{1,ij}}$ is the distance from position $P_{i,j}$ to position B. The selected moving clip should shorten the distance to the destination position B. As shown in Figure 4(a), the shorter the distance $\mathbf{D_{1,ij}}$, the better the motion path $[P_{i,1}, ..., P_{i,j}]$.
- $\mathbf{D_{2,ij}}$ is the distance from position $P_{i,j}$ to the shortest path. The selected moving clip should not be far from the shortest path. A shorter $\mathbf{D_{2,ij}}$ indicates that the rendered motion path from the selected clip is more efficient for player movement.
- A larger j is preferred because longer clips would make the rendering result smoother. In other words, a smaller j indicates that many shorter clips may be needed to render the motion path. This makes the path less smooth.

Taking these factors into consideration, the decision function at each position $P_{i,j}$ in moving clip selection is formulated as the sum of $\mathbf{D_{1,ij}}$, $\mathbf{D_{2,ij}}$, and j.

$$MD_{ij} = D_{1,ij} + D_{2,ij} - c_m j, \tag{4}$$

where c_m is a weighting coefficient to balance the effectiveness of clip length. The moving clip with the minimum value of MD_{ij} is recognized as the most suitable. Occasionally, the selection procedure is repeated several times and multiple clips are selected to form the motion path. As the

illustration in Figure 4(b) shows, three clips, $\overline{AA_1}$, $\overline{A_1A_2}$, and $\overline{A_2B}$, form the motion path. It should be noted that the time to move from position A to position B depends on the player database. In other words, a shorter running time is needed if the player moves fast in various directions in the real video, which is recorded in the database as shown in Figure 1(d). The rendering of player motion based on extracted clips containing the movement characteristics of the player in videos of real match is one of the distinguishing features of TRP.

3.1.2 Clip Selection for Hitting

Unlike moving clip selection, the major factor considered in clip selection for hitting is the similarity of texture and shape between the final selected moving clip and the hitting clip. As the transition example in Figure 5 illustrates, the primary challenge is how to choose the hitting clip Hit to connect to the current frame CF_t in a seamless cascade. A reasonable assumption for a suitable connection is that the initial frames in a successive clip should be visually similar to the current clip. This requires computing the similarity between frames of the current clip and the initial frames of clips in the Hit category. HD_i, the distance between the initial frame of clip i and the current frame, is defined on the basis of textural and shape features. It is written as

$$HD_i = D_{tex,i} + c_h D_{shape,i}, \tag{5}$$

$$D_{tex,i} = \sum_x \sum_y |Hit_{i,1}(x,y) - CF_t(x,y)|^2, \tag{6}$$

where $D_{tex,i}$ is the textural similarity between the current frame $CF_t(x,y)$ and the initial frame of the successive clip $Hit_{i,1}(x,y)$. Note that the positions of the player masks in these clips are normalized (Section 2.1), and no alignment process is required in measuring textural similarity. $D_{shape,i}$ stands for the shape similarity derived with the Hausdorff-distance [7]. c_h is a weighting coefficient to balance the effectiveness of shape similarity. The clip Hit_i with the minimum HD_i is chosen as the successor clip.

For hitting properties, the hitting strengths are based on statistics from the player's performance in real videos. As an example in Figure 5, the blue and red charts show statistics for forehand and backhand strengths, respectively, in each direction (extracted in Section 2.2). To simulate the game, a Gaussian variable is added to the direction and strength of each hit. Note that different players in different videos have different statistics—a property which makes the proposed interactive game system more realistic. The rendering of hitting based on player characters is another distinguishing feature of TRP.

3.2 Smoothing Transitions

Occasionally, the next selected clip may not be sufficiently similar to the current clip, in which case the rendering result will appear awkward when the two are directly connected. For the example in Figure 5, the rendering result is not smooth if we cascade CF_t and $Hit_{1,1}$. In our observations, the dissimilarity between two clips comes from shape, color, and motion. To smooth the transition, we propose to insert transition frames between two cascading clips. The transition frames are calculated from the current clip and the next selected clip by considering the smoothness of shape, color, and motion. The number of transition frames can be dynamically determined by the value of the similarity measure in (5).

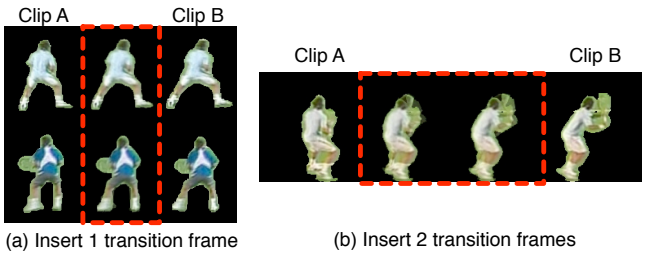

Figure 6: Shape smoothing. (a) Insert one transition frame. (b) Insert two transition frames.

For shape smoothing, we attempt to interpolate the transition postures between two cascading clips. A well-known approach for shape transition is the image morphing method proposed by Seiz and Dyer [24]. With transition points labeled by users, image morphing can generate smooth transitions between two different images. Furthermore, a hierarchical and tile-based image warping scheme proposed by Gao and Sederberg [4] can improve the results. However, neither method can be directly applied to our application because it is impossible to manually label the transition points between any two clips in a massive database. Therefore, another challenge is how to automatically label the transition points. With the help of the feature detection method proposed in [18] and the feature descriptor in [16], feature points on the images can be automatically detected and matched. Suppose we wish to find the transition frames between two images: $I_1(i,j)$ and $I_2(i,j)$. $P_1(k)$ and $P_2(k)$ are the positions of feature points on $I_1(i,j)$ and $I_2(i,j)$, respectively. We propose to modify the cost function for view morphing by adding the distance between feature points as shown in the following equation.

$$W = \sum_i \sum_j |I_1(i,j) - \tilde{I}_2(i,j)|^2 + \lambda \sum_{k=1}^{n} |P_1(k) - \tilde{P}_2(k)|^2, \tag{7}$$

where $\tilde{I}_2(i,j)$ is the warping result of $I_2(i,j)$, $\tilde{P}_2(k)$ are the positions of the feature points in $\tilde{I}_2(i,j)$, n is the number of matched feature points, and λ is the weighting coefficient. The morphing process employs hierarchical and tile-based warping with the cost function (7). The process iteratively finds the minimum value of W and stops when W converges. As an example of shape smoothing in Figure 6(a), we insert one transition frame between clips A and B in which the player in the transition frame has an intermediate posture. Figure 6(b) is another example of shape smoothing with two transition frames. It can be observed that transition frames can effectively smooth the clip connection.

For color smoothing, the clips in the database are segmented from different time periods in a video. Therefore, each clip may have a different background because of changes in the weather. Occasionally, changes in background lighting conditions lead to luminance variation in the clips, making the transition unpleasing. To solve this problem, all clips in the database are normalized to the color of the court with Poisson Editing [19].

For motion smoothing, clips may have different movement speeds and directions. A discontinuity in the motion will

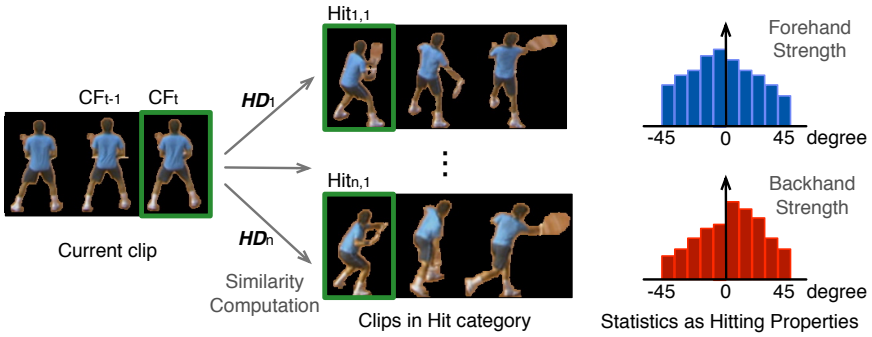

Figure 5: A suitable clip is chosen when the hitting clip has higher similarity to the current clip. The hitting properties of a player depend on statistics from real videos. The blue and red charts show hitting statistics for forehand and backhand strengths, respectively, in each direction.

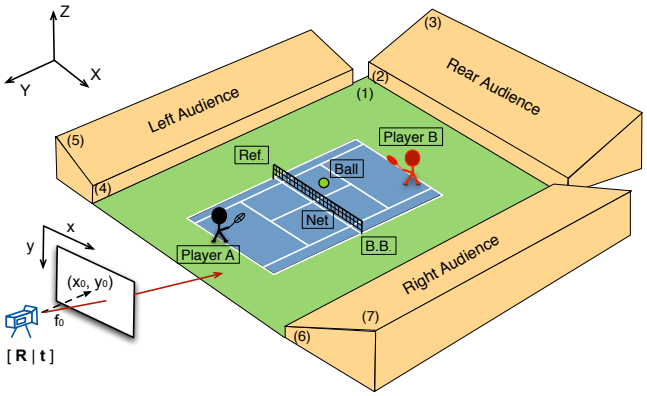

Figure 7: The structure of game system.

then make the rendering result discrete. To reduce discontinuities in motion, we provide an intermediate motion state to the transition frames. The intermediate motion state can be a linear interpolation of speed and direction between the two clips.

3.3 Game System

The entire game system includes rendering of not only the player rendering but also the background. The background is also an important component of a game system because a better rendering of it will increase the game's realism and user enjoyment. Inspired by the method proposed by Horry et al. [6] in "Tour Into the Picture" and the improved methods in [10], in TRP, 3D scenes are rendered from a 2D image once the user manually labels the 3D structure of the image. As the illustration in Figure 7 shows, the 3D structure of a tennis court can be roughly modeled by seven boards: (1) the floor, (2) the base of the rear audience, (3) the top of the rear audience, (4) the base of the left audience, (5) the top of the left audience, (6) the base of the right audience, and (7) the top of the right audience. The 2D scene rendered from the 3D structure is controlled by intrinsic and extrinsic parameters of the camera as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_0 & 0 & x_0 \\ 0 & f_0 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R} \mid \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \qquad (8)$$

Table 2: The computations per frame.

Items	Computation Units
Clip Selection	45
Smooth Transition	160 ∼ 270
Background Rendering	100
Foreground Rendering	20 ∼ 90

where f_0 is focal length and $[x_0, y_0]$ are the offset coordinates of the intrinsic parameters. The rotation matrix $\mathbf{R}$ and translation matrix $\mathbf{t}$ are extrinsic parameters. By modifying these camera parameters, we can render virtual 2D scenes from the 3D structure in any viewing angle. Incorporating foreground rendering into the 3D structure, the foreground objects in Figure 7 are rendered in the following order: Player B, net, referee, ball boy, ball[4], and Player A. In order to achieve more vivid player effects, we also model the light source and draw player shadows by warping player shapes according to the position of light sources. To model the motion blur effect of a fast-moving ball, alpha blending and multiple-ball rendering are used. More vivid rendering results can be generated by attending to these details.

4. EXPERIMENTS AND DISCUSSION

4.1 Rendering Results

We designed a graphic user interface to show rendering results at a resolution of 720 × 480 (Figure 1(a)), XY positions of players and ball on fiducial coordinates (Figure 1(b)), YZ positions of players and ball on fiducial coordinates (Figure 1(c)), and motion paths of players A and B (Figure 1(d)). In particular, Figures 1(b) and (c) clearly illustrate that the game system is built upon a 3D model. Figure 1(d) shows the potential motion paths of players, all of which are based on a database extracted from real videos of matches to record movement properties. Table 1 shows the information of tennis players in the game and also includes the match videos where the players extracted from. Because many *Motion* and *Hit* clips in the player database are similar, we only select some clips with various moving

[4]If the depth of the ball is deeper than that of the net, the rendering priority of the ball is higher than that of the net.

Table 1: Information of player database and the match videos where the players extracted from.

Tennis Video	Player Name	# of Clips in Motion	# of Clips in Hit
2009 French Open Semi Final	Roger Federer	27	12
2009 Wimbledon Open Semi Final	Serena Williams	22	12
2007 Australia Open Final	Roger Federer	29	26
2009 French Open Semi Final	Juan Martin del Potro	28	12
2009 Wimbledon Open Semi Final	Elena Dementieva	22	12

paths and hitting postures in the real-time game rendering. As shown in Table 1, 5 real tennis players are available in the game, and the number of clips in *Motion* and *Hit* states are also listed.

To control the player in the game, we analyzed the XYZ-axis acceleration signals of Wiimote through the Blue-tooth protocol for user's gesture recognition. The hitting strength of a player in the game is proportional to the statistics in the match video and user's force feedback. We suppose the hitting strength from statistics is $\mathbf{F_s}$, the user's force feedback is $\mathbf{F_u}$, and the hitting strength of the player in the game is $\mathbf{F_s} \cdot \alpha\mathbf{F_u}$, where α is the parameter for the normalization.

4.1.1 Viewing Effect

Further rendering results are shown in Figure 8. Figures 8(a) and (b) show players competing on a court at the French Open. To give more vivid visual effects, shadows of foreground objects are added to the court surface. Furthermore, the score is seamlessly painted on the court with alpha blending. To increase excitement, the player's hitting energy is shown accumulating during the game with the bars in the upper-right and lower-left corners. The player produces a powerful stroke accompanied by a fire ball when the hitting energy is full, as shown in Figure 8(b). Figure 8(d) shows players competing on a court at Wimbledon. Figure 8(e) shows a player and an animated character on a court at the Australia Open, and Figure 8(f) shows two animated characters on a court at the US Open. These images demonstrate that the rendering effects are quite realistic and resemble real videos.

The position of the camera is far from the tennis court in Figures 8(b), (d), (e), and (f), whereas it is behind player A in Figures 8(c), (g), (h), and (i). With changes in the viewing angle, the visual effects of TRP are more vivid and offer more novel experiences to users. Furthermore, the proposed methods of database selection can determine suitable clips and connect them to form various player movements and postures. The smoothing transitions effectively reduce the awkward effects caused by directly connecting two clips. A demo video showing the player and background renderings is available from the link of system overview in Section 1.

We find that the some rendering clips were still not smooth enough. Because the morphing process needs to iteratively refine the transformation, the much dissimilar clips will need more computation time to get better results. However, the requirement of real-time performance is always the critical factor in the game rendering; thus some clip connections cannot be refined to be perfect during the game. The number of discontinuous motion will decrease if the computer as the game server has higher computation capability.

4.1.2 Computational Analysis

TRP is an interactive tennis game and requires real-time rendering performance during user interaction. As mentioned in Section 3, the computations for rendering include clip selection, smoothing transitions, background rendering, and foreground rendering. Note that, the computation time of player rendering in proportion to the number of player clips, and the numbers of clips are listed in Table 1.

We set the computation of background rendering as 100 computation units (CUs) per frame and normalized the CUs for the other steps as shown in Table 2. Foreground rendering requires 20 to 90 CUs per frame depending on the position of camera. For example, the computational load is heavy when the camera position is close to the player as in Figure 8(c), because a larger foreground area must be rendered. Note that the computations for background rendering do not decrease when foreground computations increase, because the former is independent of the latter. Clip selection would process and depend on the current pose of the user. Due to the partial selection of player database, clip selection only costs 45 CUs, and it would be linearly increased when clip number increases. Smoothing transitions require extensive computations to detect feature points, execute the morphing process, and perform Poisson Editing. In the experiments, smoothing transition requires 160 to 270 CUs per frame depending on the size of foreground players, which costs the most computation in the game rendering. We made a multi-thread program and employed a PC with Intel i7 2.6GHz CPU to achieve a rendering performance of 720×480 and 30 fps, providing users with a more comfortable gaming experience.

4.2 Game Prediction with Player Statistics

Player rendering with hitting statistics and movement properties extracted from real videos is a key feature of TRP. A player's performance in TRP may reflect that player's performance in a real video. Therefore, we designed an experiment to observe whether the performances of players in TRP correlate with those in a real video. The experimental results are shown in Table 3. Two real videos were used: the men's semi-final of the French Open and the women's semi-final of Wimbledon, both in 2009. From the match records, the percentage of games won by Roger Federer in the former is 51%. The percentage of games won by Serena Williams in the latter is 54%.

To simulate a match with TRP, both players were controlled by the computer. A Gaussian variable was added in the direction and strength of hits to model the player in the real video. The simulations were run for 5 and 3 sets for the French Open and Wimbledon, respectively. The per-

Figure 8: Rendering results. Rendering results. (a)-(c) Two players competing on a French Open court. (d),(g) Two players competing on a Wimbledon court. (e),(h) A player and an animated character competing on an Australia Open court. (f),(i) Two animated characters competing on a US Open court.

centage of games won by Roger Federer was 55%, and that by Serena Williams was 61%. Therefore, the performance of a player in TRP can reflect that player's performance in real videos, although the former slightly overestimates the latter. We realize that match results are difficult to predict because player performance depends not only on hitting and moving but also on emotions, the weather, and chance. Nevertheless, we might still test how well the simulated results of TRP hold in general. For example, it would be interesting to use TRP to predict the results of Federer's performance in the 2009 French Open final and in the 2007 French Open. This could potentially show whether Federer's technique has advanced or regressed.

4.3 Subjective Evaluations

For the user study, we designed subjective evaluations for twenty undergraduates who played TRP for the first time, and the game environment was captured in our demo video. Of the twenty evaluators, eleven had a habit of watching videos of tennis matches whereas the rest did not. Sixteen evaluators had a habit of playing games on PS3 or Wii, whereas the others did not. Five questions were designed to evaluate the experience of playing TRP, and four were designed to compare the experiences of playing TRP, Wii Sports, and Top Spin 3 on PS3.

Before the subjective evaluations, evaluators were required to watch videos of tennis matches. Subsequently, they were required to play TRP and score their satisfaction on a five-point scale, i.e., 1, very unsatisfied; 2, somewhat unsatisfied; 3, no difference; 4, somewhat satisfied; and 5, very satisfied. The five questions were as follows:

Q.1 Did you have interactions with the video content from playing TRP?

Q.2 Did you have an immersive experience with the game of tennis from playing TRP?

Q.3 Was it entertaining and interesting to play TRP?

Q.4 Do you think that TRP is an innovative multimedia application?

Q.5 Are you more willing to play TRP after watching videos of tennis matches?

The average scores and standard deviations of the evaluations are listed in Figure 9. The results show that evaluators identify with increased interaction, immersive experience, and enjoyment from playing TRP. Furthermore, they highly

Table 3: The percentage of games won in real videos and in results simulated by Tennis Real Play.

Game Video	Name of Player A	Name of Player B	Game Points A-B	Game(%)	Simulation(%)
2009 French Open S.-F.	Roger Federer	Juan Martin del Potro	3-6, 7-6, 2-6, 6-1, 6-4	51 : 49	55 : 45
2009 Wimbledon Open S.-F.	Serena Williams	Elena Dementieva	6-7, 7-5, 8-6	54 : 46	61 : 39

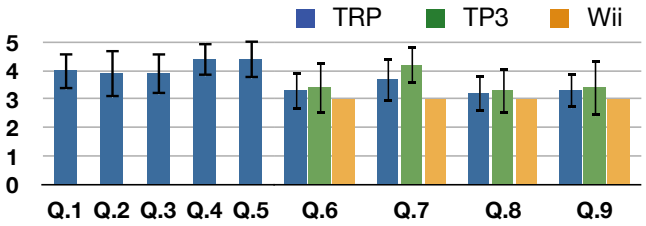

Figure 9: Results of subjective evaluation. The bars' heights are the average scores, and the black lines show the standard deviations.

agree that TRP is an innovative multimedia application and are more willing to play it after watching videos of tennis matches.

In the next phase, evaluators were required to play the tennis games in Wii Sports(Wii), Top Spin 3(TP3) on PS3, and TRP. They were told to use Wii as the standard of comparison and give a score of 1 to 5 for their experience with TP3 and TRP, i.e., 1, much worse; 2, somewhat worse; 3, no difference; 4, somewhat better; 5, much better. The four questions were as follows:

Q.6 Comparing the entertainment levels of each game, what do you think of the performance of TP3 and TRP?

Q.7 Comparing the realism of the visual effects, what do you think of the performance of TP3 and TRP?

Q.8 Comparing the interactiveness of each game, what do you think of the performance of TP3 and TRP?

Q.9 Comparing your preferences for each game, what do you think of the performance of TP3 and TRP?

The average scores and standard deviations of the evaluations are listed in Figure 9. The primary advantages of Wii are the innovations in user-interactive dialogue (e.g., Wiimote). TRP also employs Wiimote for interactive dialogue. From the results in Figure 9, the performances of TRP in regard to visual effects, interactiveness, and preference are all higher than for Wii. Some subjects noted that TRP has vivid rendering effects and realistic player properties which provided them with a more interesting and enhanced experience. Compared to TRP, the primary advantages of TP3 are its vivid rendering effects of the court and the players. The performances of TRP are slightly lower than those of TP3 in terms of entertainment, visual effects, and preference. However, we feel that the performances of TRP are still outstanding because TP3 requires dozens of individuals to build the game model and draw textures. In contrast, all of the materials in TRP are simply extracted from real videos. Furthermore, this feature may also lead to a new framework in game production; the latest game of TRP will be available after a real tennis match is played.

5. CONCLUSION AND EXTENSION

Inspired by video analysis/annotation, video-based rendering, and interactive sports games, an interactive tennis game—TRP—constructed using models extracted from videos of real tennis matches is proposed. As techniques for player model creation, we propose a database normalization process and a 4-state-transition behavioral model of tennis players. For player rendering, we propose clip selection, smoothing transitions, and a framework combining a 3D model with video-based rendering. Experiments show that vivid rendering results can be generated with low computational requirements. Moreover, the player model can adequately record the ability and condition of a player, which can then be used to roughly predict the results of real tennis matches. User studies reveal that subjects like the increased interaction, immersive experience, and enjoyment from playing TRP. They also show that evaluators rate the visual effects, interactiveness, and preference for TRP higher than those for Wii Sports but slightly lower than those for Top Spin 3. However, unlike building complex scene models or drawing player textures in Top Spin 3, all of the materials in TRP are extracted from videos of real matches. This property can also provide a new framework for game production; the latest game of TRP will be available after a tennis match is played.

Basically, some steps of the application can be improved by utilizing a powerful computation server, which can increase the rendering smoothness discussed in Section 4.1.1, or the existing techniques. For example, a few recent methods for shape [8] [17] and color interpolation [15] which might be useful to render more vivid viewing effects. However, the real-time constraint is one of reasons why we do not prefer these techniques. Furthermore, there are some papers addressing on pose estimation of tennis player [28] and retrieval of hitting statistics [27], but we only put emphasis on player rendering and system integration.

Limitations of the current system include the restrictions in the viewing angles and resolutions of the rendered game frame. For example, the system cannot render arbitrary views of the player, and the rendered game frame is blurry if the resolution of the tennis video is insufficiently high. Nevertheless, the limitation in viewing angles can be overcome if multiple videos from different cameras are made available. By constructing a database of multiple court models and players, the system can render game frames from any viewing angle. To overcome the limitation of low resolution, super-resolution techniques may be employed to preserve more details from the real video.

For future studies, our top priority is to extend the application of the proposed methods to other sports videos. The proposed methods in player model creation and player rendering will be modified. For example, the techniques of database normalization, clip selection, and smoothing transitions can be applied to videos of football games. Specifi-

cally, the proposed four-state-transition model of tennis players can be replaced by a transition model for football players (i.e., shot-pass-stop-motion). In this way, the framework in TRP can be extended to other sports videos to create games such as Football Real Play and Baseball Real Play.

6. REFERENCES

[1] G. Colqui, M. Tomita, T. Hattori, and Y. Chigusa. New video synthesis based on flocking behavior simulation. In *International Symposium on Communications, Control and Signal Processing*, pages 936–941, 2008.

[2] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 726–733, 2003.

[3] M. Flagg, A. Nakazaway, Q. Zhangz, S. B. Kang, Y. K. Ryu, I. Essa, and J. M. Rehg. Human video textures. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, pages 199–206, 2009.

[4] P. Gao and T. W. Sederberg. A work minimization approach to image morphing. *The Visual Computer*, pages 390–400, 1998.

[5] J. Han, D. Farin, and P. H. N. de With. Broadcast court-net sports video analysis using fast 3-d camera modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1628–1638, 2008.

[6] Y. Horry, K.-I. Anjyo, and K. Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 225–232, 1997.

[7] D. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the hausdorff-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.

[8] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. In *Proceedings of ACM SIGGRAPH 05*, pages 1134–1141, 2005.

[9] N. Inamoto and H. Saito. Free viewpoint video synthesis and presentation of sporting events for mixed reality entertainment. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 42–50, 2004.

[10] H. W. Kang, S. H. Pyo, K.-I. Anjyo, and S. Y. Shin. Tour into the picture using a vanishing line and its extension to panoramic images. *Computer Graphics Forum*, 20(3):132–141, 2001.

[11] Y. Kitamura, R. Rong, Y. Hirano, K. Asai, and F. Kishino. Video agent: interactive autonomous agents generated from real-world creatures. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 30–38, 2008.

[12] J.-H. Lai, C.-L. Chen, C.-C. Kao, and S.-Y. Chien. Tennis video 2.0: A new presentation of sports videos with content separation and rendering. *Journal of Visual Communication and Image Representation*, 22(3):271–283, 2011.

[13] J.-H. Lai and S.-Y. Chien. Tennis video enrichment with content layer separation and real-time rendering in sprite plane. In *Proceedings of IEEE International Workshop on Multimedia Signal Processing, MMSP 2008*, pages 672–675, 2008.

[14] J.-H. Lai, C.-C. Kao, and S.-Y. Chien. Super-resolution sprite with foreground removal. In *IEEE International Conference on Multimedia and Expo*, pages 1306–1309, 2009.

[15] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, 23(3):689–694, 2004.

[16] D. Lowe. Distinctive image features from scale-invariant keypoint. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[17] D. Mahajan, F.-C. Huang, W. Matusik, R. Ramamoorthi, and P. Belhumeur. Moving gradients: A path-based method for plausible image interpolation. *ACM Transactions on Graphics*, 28(3):42:1–42:11, 2009.

[18] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, pages 63–86, 2004.

[19] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. In *Proceedings of ACM SIGGRAPH 03*, pages 313–318, 2003.

[20] P. M. Phillips and G. Watson. Generalising video textures. In *Proceedings of the Theory and Practice of Computer Graphics*, pages 726–733, 2003.

[21] M.-C. Roh, B. Christmas, J. Kittler, and S.-W. Lee. Gesture spotting for low-resolution sports video annotation. *Pattern Recognition*, 41(3):1124–1137.

[22] A. Schodl and I. A. Essa. Controlled animation of video sprites. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 121–127, 2002.

[23] A. Schodl, R. Szeliski, D. H. Salesin, and I. Essa. Video textures. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 489–498, 2000.

[24] S. M. Seitz and C. R. Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive*, pages 21–30, 1996.

[25] J. Wang, C. Xu, E. Chng, K. Wah, and Q. Tian. Automatic replay generation for soccer video broadcasting. In *Proceedings of the 12th Annual ACM International Conference on Multimedia MULTIMEDIA '04*, pages 32–39, 2004.

[26] J. R. Wang and N. Parameswaran. Analyzing tennis tactics from broadcasting tennis video clips. In *Proceedings of the 11th International Multimedia Modelling Conference*, pages 102–106, 2005.

[27] X. Yu, C. H. Sim, J. R. Wang, and L. F. Cheong. A trajectory-based ball detection and tracking algorithm in broadcast tennis video. In *Proceedings of IEEE International Conference on Image Processing*, pages 1049–1052, 2004.

[28] G. Zhu, C. Xu, Q. Huang, W. Gao, and L. Xing. Player action recognition in broadcast tennis video with applications to semantic analysis of sports game. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 431–440, 2006.