Tennis Video Enrichment with Content Layer Separation and Real-Time Rendering in Sprite Plane

Jui-Hsin Lai [1] and Shao-Yi Chien [2]

Media IC and System Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
BL-421, 1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
[1] larry@media.ee.ntu.edu.tw, [2] sychien@cc.ee.ntu.edu.tw

***Abstract*—Sport video enrichment can provide viewers more interaction and user experiences. In this paper, with tennis sport video as an example, two techniques are proposed for video enrichment: content layer separation and real-time rendering. The video content is decomposed into different layers, like field, players and ball, and the enriched video is rendered by re-integrated these layers information. They are both executed in sprite plane to avoid complex 3D model construction and rendering. Experiments shows that it can generate nature and seamless edited video by viewers' requests, and the real-time processing speed of 30 720x480 frames per second can be achieved on a 3GHz CPU.**

I. Introduction

The amount of sport video contents has rapidly increased in this decade, and the research works on sport video applications have dramatic growth. For example, to enhance the user experiences in watching sport games, Inamoto and Saito proposed the free viewpoint synthesis to let viewers decide their own viewing angles[1]; for advertisement and content insertion, Li et al. provide a method to detect proper timing and proper location automatically[2], and Yu et al. use 3D projection to model play field for stadium replacement[3]; furthermore, Wikstrand and Eriksson use animations to present a football game which are enlightening and enjoyable for viewers[4]. However, it seems that there needs a uniform processing scheme to provide viewers all above effects and even creates more functions to enrich the viewing experiences. Besides, rather than viewing the enriched sport video in graphics animations[1][4], nature enriched sport video is more desirable, and the real-time rendering ability is also important to enable viewers to interact with the sport video contents.

In this paper, a video processing scheme for tennis video enrichment is proposed, which consists of two main techniques: video content layer separation and real-time content rendering. To avoid complex 3D model construction and rendering, both techniques are executed in a sprite plane in bird's eye view, where a fiducial coordinate system is adopted. Fig. 1 shows the concepts of the proposed video processing scheme. For video content layer separation, with the help from the sprite plane, video content is decomposed into different layers like field, score box, players and ball, as shown in Fig. 1(a). According to viewers' requests, the enrichment video is real-time rendered by re-integrating these layered information with the following

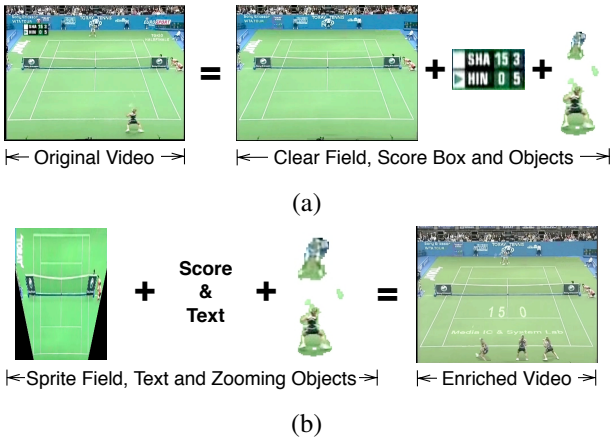


Fig. 1. (a) The concept of video content separation. (b) The concept of video content rendering.

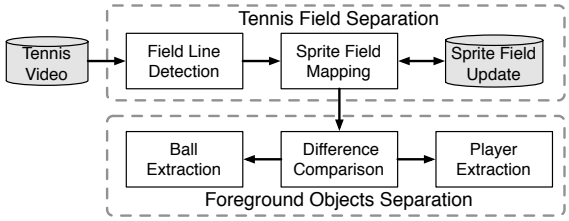


Fig. 2. The processing scheme of video content separation.

procedures: sprite field processing, foreground objects painting, and view rendering, as shown in Fig. 1(b). It should be noticed that the rendering video content will be presented in plentiful formats according to viewers' different requests and the provided materials. Different from the previous works, the enrichment video is nature and more vivid, providing more enlightening viewing effects, and giving viewers the ability to watch a customized sport game videos.

II. Video Content Separation

The processing flow of the proposed content separation is shown as Fig. 2, which composes two parts: tennis field separation and foreground objects separation.

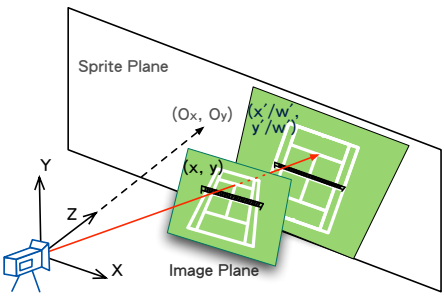


Fig. 3. The projection from the image plane to the sprite plane.

A. *Tennis Field Separation*

Sprite field plane is the tennis field in a fiducial coordinate system in bird's eye view. To warp the video frames from image planes to the sprite plane, we use the well-known perspective motion model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{w} \end{bmatrix} = \mathbf{H} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ m_7 & m_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (1)$$

where $\mathbf{H}$ is the transformation matrix from video frame coordinate (x, y) to sprite field coordinate $(\dot{x}/\dot{w}, \dot{y}/\dot{w})$. The parameters of the transformation matrix $\mathbf{H}$ are also rewritten as an eight-tuple vector $\mathbf{M}$,

$$\mathbf{M} = \begin{bmatrix} m_1 & m_2 & \dots & m_8 \end{bmatrix}^T. \quad (2)$$

Fig. 3 illustrates the operation of image projection from video frame to the sprite plane. To compute the transformation matrix $\mathbf{H}$, the intersections of field lines would be the feature points for image projection. So, it needs to detect the field lines and calculates the line intersections in each video frame. First, we apply edge filter to extract the binary map $E(x_i, y_i)$ from video frame $I(x_i, y_i)$,

$$E(x_i, y_i) = Edge(I(x_i, y_i)), \forall i \in N. \quad (3)$$

Then, the Hough transform is used to detect straight lines in $E(x_i, y_i)$. The equation of Hough transform is shown below,

$$\underline{b} = -x_i \underline{a} + y_i, \forall E(x_i, y_i) = true. \quad (4)$$

Because field lines in tennis are only distributing over vertical and horizontal directions, we can limit the parameter spaces of $\underline{ab}$ to increase the detecting result and decrease the computation. The line functions are detected by the peak distributions of $E(x_i, y_i)$ on $\underline{ab}$ parameter spaces. After that, the intersection coordinates (x_k, y_k) are calculated from these line functions.

The intersection coordinates $\mathbf{P}$ of a video frame and the corresponding coordinates $\mathbf{Q}$ in sprite plane are employed to calculate the initial translation camera motions $\mathbf{M_0}$ by the regression equations.

$$\mathbf{M_0} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{Q}. \quad (5)$$

It should be noticed that if the number of detected intersections is less than four, or the coordinates of detected intersections form the homogeneous solution. The M^{t-1}, the transformation matrix of previous video frame at time $t-1$, would be the initial M_0.

After that, the transformation matrix is further refined by the Levenberg-Marquardt iterative minimization algorithm[5] in gradient descent to minimize the cost function E,

$$E = \sum_{i \in N} |e(i)|^2 = \sum_{i \in N} |I(x_i, y_i) - I'(x_i', y_i')|^2, \quad (6)$$

where $I(x_i, y_i)$ is the luminance value of pixel (x_i, y_i) in the video frame, $I'(x_i', y_i')$ is the luminance value of the corresponding position (x_i', y_i') in the sprite plane, and N is a set of effective pixels. The iterative procedure of gradient decent is shown as follows.

$$\mathbf{M_d} = \mathbf{M_{d-1}} + \mathbf{A}^{-1}\mathbf{B}, \quad (7)$$

$$A_{k,j} = \sum_{i \in N} \frac{\partial e(i)}{\partial m_k} \frac{\partial e(i)}{\partial m_j}, B_k = \sum_{i \in N} -e(i) \frac{\partial e(i)}{\partial m_k},$$

where $\mathbf{M_d}$ is the transformation matrix at the d-th iteration, $\mathbf{A}$ is an 8×8 matrix, and B is an eight-tuple vector. The iterative process is repeated until the improvement of each parameter is smaller than a threshold, or the number of iteration is larger than the maximum number of iteration.

After the transformation matrix is derived, the main efforts of sprite plane generation are transforming the video frame to the fiducial coordinate and removing the foreground objects. In general situation, foreground objects do not occupy the field region for a long time, and we can assume that the maximum histogram bin of the pixel value distribution in temporal domain should be the background, which can be described with the following equations.

$$h_{x_i,y_i}(k) = \#\{I_t(x_i, y_i) | I_t(x_i, y_i) = k, \forall t \in [t_1, t_2]\},$$
$$S(x_i, y_i) = \arg\max_k h_{x_i,y_i}(k), \quad (8)$$

where $h_{x_i,y_i}(k)$ is the histogram of pixel values for a period time $[t_1, t_2]$ at the position (x_i, y_i), and $S(x_i, y_i)$ is the pixel value of sprite field. During the game proceeding, some new background objects may appear in the tennis field, and the sprite field can also be updated by equation (8).

B. *Foreground Objects Separation*

After tennis field separation, the clear field is generated by warping sprite field with the transformation matrix $\mathbf{H^{-1}}$, as shown in Fig. 1(a). The foreground regions D of a video frame are then derived by frame difference between the video frame $I(x_i, y_i)$ and clear field $I_c(x_i, y_i)$,

$$D = \{I(x_i, y_i) | |I(x_i, y_i) - I_c(x_i, y_i)| > Th, \forall i \in N\}, \quad (9)$$

where Th is a threshold for change detection. The foreground regions D include moving audiences, players, and ball. To find out player objects, a player sieve is proposed to eliminate objects of audiences and ball, which contains a size sieve and a position sieve. In the size sieve, the object size of players can be estimated from the transformation matrix $\mathbf{H^{-1}}$, and

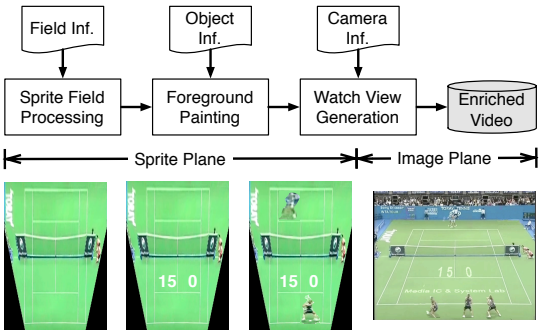


Fig. 4. The rendering flow of the enrichment video.

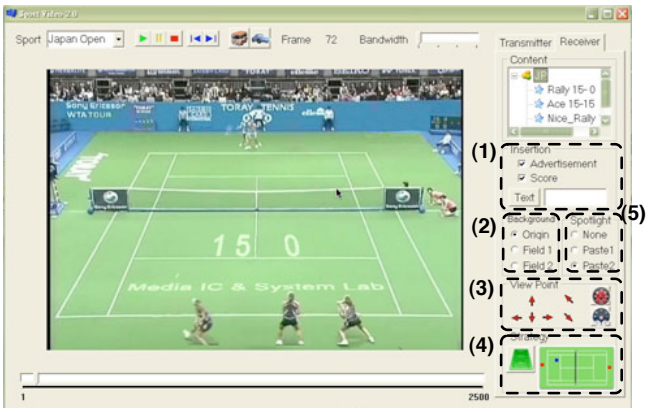


Fig. 5. User interface of enrichment functions: (1)Insertion, (2)Background, (3)ViewPoint, (4)Strategy, and (5)Spotlight.

the objects with too small or large size are eliminated. In the position sieve, the positions of player objects must locate on the field, which can be used to eliminate most audience objects.

Contrary to players, the size of ball is much smaller, and the ball sometime disappears due to high speed movement or video coding. Therefore, it cannot be correctly extracted with only change detection approach. In the proposed algorithm, the ball is extracted with a ball sieve and Kalman filter. In the foreground regions D, the objects of ball candidates are detected with a ball sieve, which contains a size sieve, a position sieve, and a color sieve. In the size sieve, the ball size is estimated also by the transformation matrix $\mathbf{H^{-1}}$ and is often very small; in the position sieve, the ball positions must appear inside the field, which can be used to exclude most error candidates; in color sieve, the color of ball objects must fall into the ball color range. Next, from these ball candidates got from the ball sieve, the proposed method based on Kalman filter is applied to predict and complete the ball trajectory. The predict model is showed as followings.

$$\mathbf{X_t} = \mathbf{F_t}\mathbf{X_{t-1}} + \mathbf{B_t} + \mathbf{W_t}, \tag{10}$$

$$\mathbf{F_t} = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{B_t} = \begin{bmatrix} P_i \\ V_i \\ A_i \end{bmatrix}, \mathbf{W_t} = \begin{bmatrix} \omega_p \\ \omega_v \\ \omega_a \end{bmatrix},$$

where $\mathbf{X_t}$ is the state of ball position, velocity and acceleration at time t, $\mathbf{F_t}$ is the state transition model which is applied to the previous state $\mathbf{X_{t-1}}$, $\mathbf{B_t}$ is the control-input model which models the ball hitting by players, and $\mathbf{W_t}$ is the process noise which is assumed to be drawn from a zero mean normal distribution. Finally, the ball trajectory is extracted by recursively updating the model and predicting.

III. Enrichment Video Rendering

With the generated layered video contents, the enriched video is generated in real-time by the rendering flow shown in Fig. 4. It consists of three main steps: sprite field processing, foreground painting, and view projection, where the first two steps are executed in the sprite plane, and the third step transform the enriched video from sprite plane to image plane. In sprite field processing, the tennis field can be modified. For example, we can paint the scores on the sprite field that gives viewers a fresh experience different from normal score boxes. The viewers can even also paint comments or any texts on the field for video annotation. Besides, the sprite field could be changed from tennis field to any viewer providing image, where it seems like that the players are playing the game on another place. Moreover, the advertisements can be seamlessly inserted on the field.

After field processing, the foreground objects are also painted on the sprite field. The size of the foreground players are set by considering the relative size to the field on the sprite plane. Since the background maybe replaced in the sprite field processing, in order to achieve seamless editing, the object luminance and color are adjusted to fit the background color by Poisson image editing [6]. Note that, the above operations are all implemented in sprite plane, which makes the operations simple for real-time rendering.

Finally, the enriched video frame is transformed to the view selected by viewers with the transformation matrix $\dot{\mathbf{H}}$ and the following equation:

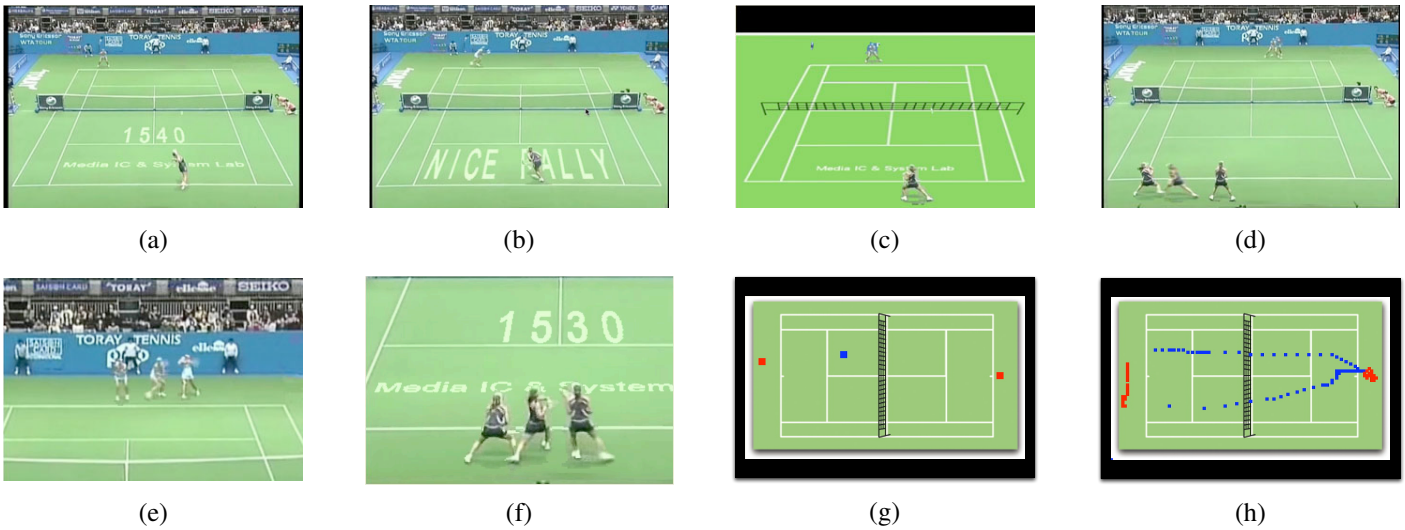
$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{w} \end{bmatrix} = \dot{\mathbf{H}} \begin{bmatrix} \dot{x}/\dot{w} \\ \dot{y}/\dot{w} \\ 1 \end{bmatrix}, \tag{11}$$

where $(\ddot{x}/\ddot{w}, \ddot{y}/\ddot{w})$ is the coordinate of the output frame, and $(\dot{x}/\dot{w}, \dot{y}/\dot{w})$ is the coordinate of sprite plane. With modifying $\dot{\mathbf{H}}$, viewers can generate the enriched video from different viewing angles. The viewing angle of the rendering video frame is equal to that of the original video when $\dot{\mathbf{H}} = \mathbf{H^{-1}}$.

IV. Experimental Results

To present the enriched tennis video, we design a user interface as shown in Fig. 5, which contains five functions. Viewers are able to input requests, and the enrichment video will be rendered in real-time. The video demonstration is also presented in our previous work [7]. The followings are the function descriptions.

Insertion enables the content provider to render advertisement



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. Enrichment functions: (a)Insertion with score count and advertisement. (b)Insertion with viewer's comments. (c)Background change from tennis field to a virtual game field. (d)Spotlight with three contiguous motions. (e)ViewPoint with local focus and Spotlight. (f)ViewPoint with player tracking and Spotlight. (g)Strategy map showing the positions of players and ball. (h)Strategy map showing the trajectories of players and ball.

or score count on the field and enables the viewers to paint text commends for video annotation without disturbing the game proceeding. The visual effect of score and advertisement insertion is shown as Fig. 6(a). Viewers also have the ability to insert commends for video annotation as shown in Fig. 6(b). **Background** enables viewers to change the background from tennis field to the user provided image as shown in Fig. 6(c). The visual effect is like that players are playing the game in the virtual world, which gives viewers a fresh viewing experience. **Spotlight** enables viewers to watch games with multiple players in contiguous motion and enriches viewing experiences, as shown in Fig. 6(d). With this function, viewers can analyze the postures of players with display at very low frame rate. **ViewPoint** enables viewers to decide their own view angle of watching a game. Fig. 6(e) shows an example that the camera is focusing on one player with **Spotlight** function. Besides, **ViewPoint** is able to automatically control the virtual camera motions to track the players, which is very convenient for viewers to focus on the favorite player, as shown in Fig. 6(f). **Strategy** synchronously displays players and ball positions on the map as shown in Fig. 6(g), where the positions of balls and players are recorded. Viewers can discuss the winning strategies of players or easily analyze trajectories of ball and players by this function shown in Fig. 6(h).

Comparison to previous works [1] [2] [3] [4], the proposed work not only achieves these visual effects, corresponding to **ViewPoint**, **Insertion**, **Background** and **Strategy** respectively, but also provides **Spotlight** to enrich the video content. The experimental results demonstrates that the proposed method can generate nature edited video in real-time. As for the computation time of video rendering, the speed of 30 720x480 video frames per second can be achieved, which meets the real-time requirement, on a PC with Pentium IV 3 GHz CPU.

V. Conclusions

In this paper, we propose a scheme for tennis video enrichment with two techniques: video content separation and real-time content rendering. For video content separation, the video frame is transformed to sprite plane and decomposed into different layers. According to viewers' requests, the enrichment video can be real-time rendered by re-integrating these layers information. The experimental results show that the designed user interface gives viewers more enjoyable viewing experience and provides more enlightening visual effects with the real-time processing capability of 30 720x480 frames per second on a 3GHz CPU.

References

[1] 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 ACE '04*, September 2004, pp. 42–50.

[2] Y. Li, K. W. Wan, X. Yan, and C. Xu, "Real time advertisement insertion in baseball video based on advertisement effect," in *Proceedings of the 13th annual ACM International Conference on Multimedia MULTIMEDIA '05*, November 2005, pp. 343–346.

[3] X. Yu, X. Yan, T. T. P. Chi, and L. F. Cheong, "Inserting 3d projected virtual content into broadcast tennis video," in *Proceedings of the 14th annual ACM International Conference on Multimedia MULTIMEDIA '06*, no. 619-622, 2006.

[4] G. Wikstrand and S. Eriksson, "Football animations for mobile phones," in *Proceedings of the Second Nordic Conference on Human-Computer Interaction NordiCHI '02*, October 2002, pp. 255–258.

[5] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, 1963.

[6] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Transactions on Graphics (TOG) , ACM SIGGRAPH 2003 Papers SIGGRAPH '03*, vol. 22, pp. 313–318, July 2003.

[7] J.-H. Lai and S.-Y. Chien, "Video program: Tennis video 2.0: a new framework of sport video applications," in *Proceedings of the 15th International Conference on Multimedia MULTIMEDIA '07*, September 2007, pp. 1087–1088.