J. Vis. Commun. Image R. 22 (2011) 271–283

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Tennis Video 2.0: A new presentation of sports videos with content separation and rendering

Jui-Hsin Lai, Chieh-Li Chen, Chieh-Chi Kao, Shao-Yi Chien *

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

ARTICLE INFO

Article history:
Received 4 August 2010
Accepted 3 January 2011
Available online 6 January 2011

Keywords:
Interactive video
Video analysis
Video enrichment
Video rendering
Event search
Scalable video
Video annotation
Object segmentation
Sprite
Matting

ABSTRACT

This paper proposes a new method for presenting sports videos. Tennis videos are used as an example for the implementation of a viewing program called as Tennis Video 2.0. For the methods in video analysis, background generation by considering the pixels in temporal and spatial distribution is proposed; foreground segmentation combining automatic trimap generation and matting model is proposed. To provide more functions in watching videos, the rendering flow of video contents and the semantic Scalability are proposed. With the new analysis and rendering tools, the presentation of sports videos has three properties—Structure, Interactivity, and Scalability. The experiments show that several broadcasting game videos are employed to evaluate the robustness and performance of the proposed system. For user study, 20 evaluators highly identify that Tennis Video 2.0 is a new presentation of sports videos and give people better viewing experience.

1. Introduction

Watching sports videos is a popular entertainment. Many people watch sports on television or computer and later discuss the highlights with friends. However, people just only watch sports in the manner in which they are broadcast and cannot watch games in a manner preferred by them. For example, sport videos are often interrupted by the advertisements, which is undesirable as they reduce the excitement involved in watching a game. Progress in broadcasting technology has made it possible for people to interact with TV contents and be able to choose their preferred sports channels, submit requests to broadcast, and even watch customized sports videos. Therefore, there is much research that can be conducted and various applications that can be developed for interactive broadcasting. Examples of two research topics regard, how to generate interactive video and how many presentation functions can be provided to enrich game videos.

In recent years, there has been a dramatic growth in research on sport video applications. Kokaram et al. provided a complete review on content indexes and retrieval methods of sports videos [1]; Inamoto and Saito proposed the synthesis of free viewpoint that would let viewers decide their own viewing angles in order to enhance their experience of watching sport games [2]; Li et al. provided real-time advertisement insertion on game court in order to avoid game interruption [3]; Tang et al. presented a content-adaptive system for streaming the goal events of soccer games over a network with low bandwidth limitations [4]; and Wikstrand and Eriksson employed animations to represent a football game on mobile phones [5]. However, when considering the entire genre of sports video applications, some suggestions still need to be considered:

- The video contents of broadcasts should be fully annotated. In other words, each ball trajectory, player gesture, and highlight should be recorded on the video. People should be able to browse the video contents and quickly get game information on demand.
- More applications should be provided that enrich the viewing experience. Broadcasts should provide functions that will allow viewers to choose the manner in which they wish to watch game videos—replaying their favorite highlights and analyzing players' postures in detail. Moreover, all applications must enable real-time interaction with viewers, and their requests must be responded to instantly.
- Game videos need to provide different video bitstream for different viewing devices like mobile phones, computers, or

* Corresponding author.
E-mail addresses: juihsin.lai@gmail.com (J.-H. Lai), chiehli.chen@gmail.com (C.-L. Chen), chiehchi.kao@gmail.com (C.-C. Kao), sychien@cc.ee.ntu.edu.tw (S.-Y. Chien).

1047-3203/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jvcir.2011.01.001

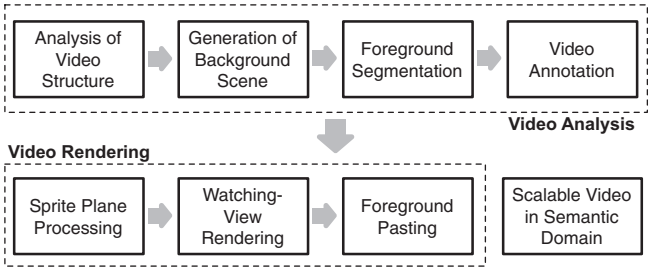

Fig. 1. Block diagram of the processing flow in Tennis Video 2.0. We propose methods of the background generation, foreground segmentation, steps of video rendering, and scalable video in semantic domain.

televisions. Considering the limitation of transmission bandwidth, the video contents should have scalable properties.

We develop new methods in the analysis and rendering of video contents to achieve above requirements. Fig. 1 is the block diagram of the processing flow in Tennis Video 2.0, which includes two parts: video analysis and video rendering. First, a video is partitioned into several shorter clips by analyzing the video structure. Next, the background scenes are generated from each clip and used to segment foreground objects, and the information of foreground objects is employed for video annotation. After that, three steps of video rendering are proposed to reintegrate the video content and implement the scalable video in semantic domain. In this paper, the contributions of the proposed methods are listed below.

- For the generation of background scene, we propose a method to generate the background scenes from the video frames by calculating the pixel value with maximum probability in temporal and spatial distribution. Note that the method can preserve the details of background scene and remove foreground objects without human-assistance.
- In the foreground segmentation, the method to generate the trimap[1] and a segmentation method with the matting model is proposed that can precisely segment the foreground objects from the complex background. Unlike the binary mask of segmentation results in the previous works, the segmentation results are presented with alpha values that not only accurately present the distribution of foreground object but also give the better viewing effect in the video rendering.
- We propose a brand-new processing flow to render videos, which introduces the sports videos with a new presentation and provides viewers functions with more interaction and customized experience. In addition, the capacity of real-time processing of video rendering lets viewers to watch the customized video immediately.
- Scalable video is a topic to reduce video bitstream size under different bandwidth constrains. Contrary to decreasing the video quality to reduce video bitstream, the smaller bitstream size in the proposed methods is achieved by abandoning the video contents with less semantic importance. The experimental results show that the bitstream size of a video is effectively decreased and the visual quality is still maintained in the proposed method.

With the development of new content analysis and rendering methods, we propose an integral framework to provide new sports videos that have more functions. In this study, we use tennis as an example for the presentation of the new sports video called Tennis Video 2.0. Contrary to the traditional game videos, Tennis Video 2.0 provides three properties: Structure, Interactivity, and Scalability.

- **Structure**. People can browse the game videos and watch the highlights in time-based or event-based sorting. Moreover, the graphic user interface (GUI) of Tennis Video 2.0 also provides a search function for game strategy. The strategy search is different from general event search because it can search for events on a higher semantic level.
- **Interactivity**. Instead of interrupting the game to play the advertisement, the game continues while the advertisements are rendered on the tennis court. The spotlight is a function that enriches the viewing experience with multiple hitting postures of the players. The tracker view animates the camera so that it focuses on the player or the ball, which provides an alternative to the traditional experience of a steady camera. Note that all of the above functions provide real-time responses so people are able to personalize their game watching experience.
- **Scalability**. The GUI supports different bitstream size for different transmission bandwidths. Instead of lowering the bitstream size by reducing the video quality, the proposed scalable video has four different bitstream sizes based on information reduction on semantic level. By methods of video background reuse, video clip abandonment, video content abandonment, and video in animation, the requirements of lower bandwidth are achieved.

The remainder of this paper is structured as follows: Analysis of video contents is described in Section 2, and the rendering of video contents is described in Section 3. Section 4 shows the experimental results and has discussions. Finally, the paper is concluded in Section 5.

2. Analysis of video contents

In tennis game videos, the video contents repeat the following iteration: service, game play, and game stop. One service-play-stop video clip can be divided into a Video Unit as shown in Fig. 2. Each Video Unit usually represents an event in a tennis video, that begins with a serve and ends before the next serve. With such a regular structure, the process of temporal structure analysis can be done by finding all play shots in the game video [6]. Afterward the whole video can be decomposed into Video Units. To further analyze the information in the play shot of each Video Unit, we propose methods to separate the background court and foreground objects. More details will be described in the following sections.

2.1. Generation of background scenes

The generation of the background scenes is an important step in Tennis Video 2.0 because background scenes can be used for the foreground segmentation and provide the functions of background editing in the video rendering. To build the background scenes, the sprite plane is employed. A sprite is an image constructed from a video sequence, and it is also used as a highly efficient video coding method [7,8]. For our purpose, the sprite plane is the tennis court composed of the play shot in a Video Unit. First, the video frames

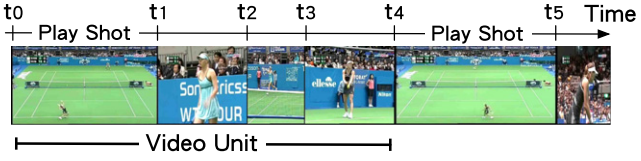

Fig. 2. Video Unit of a tennis video. It represents one service-play-stop video clip in a tennis game.

[1] Trimap is a map indicating the background, foreground, and unknown region on an image.

are projected to the sprite plane with the perspective motion model in (1). The perspective motion model can model the possible camera motions in the tennis video including panning, tilting, and zooming

$$\begin{bmatrix} x_s \\ y_s \\ w_s \end{bmatrix} = \begin{bmatrix} m_{v1} & m_{v2} & m_{v3} \\ m_{v4} & m_{v5} & m_{v6} \\ m_{v7} & m_{v8} & 1 \end{bmatrix} \begin{bmatrix} x_v \\ y_v \\ 1 \end{bmatrix}, \tag{1}$$

where m_{v1}–m_{v8} are the transformation parameters from frame coordinates (x_v,y_v) to the sprite plane coordinates $(x_s/w_s,y_s/w_s)$, and feature detection and feature matching are the methods to project the frame coordinates to court coordinates. The first step in Fig. 3 illustrates the projection from the video frame to the sprite plane, and we employ the matching of feature points to calculate the projection parameters. The feature detection can be implemented by the Harris Corner Detector [9], and the descriptor of each feature point is presented by scale-invariant feature transform (SIFT) [10]. After that, the parameters of projective transform can be calculated by solving the equation of matching feature pairs on the spite plane and video frames [11]. Sometimes, there are potion-misalignments of the features that would decrease the accuracy of frame projection. To extract more precise projection, the transformation parameters calculated from the feature matching are further refined by minimizing the cost function E,

$$E = \sum_{i\in N} |e(i)|^2 = \sum_{i\in N} |I(x_v,y_v) - I'(x'_v,y'_v)|^2, \tag{2}$$

where $I(x_v,y_v)$ is the luminance value of pixel (x_v,y_v) in the video frame, $I'(x'_v,y'_v)$ is the luminance value of the corresponding position (x'_v,y'_v) in the sprite plane, and N is the set of pixels on the overlap region. The minimization process is implemented by use of Levenberg–Marquardt iterative minimization algorithm [12] as shown below

$$\mathbf{M}_d = \mathbf{M}_{d-1} + \mathbf{A}^{-1}\mathbf{B}, \tag{3}$$

$$A_{kj} = \sum_{i\in N} \frac{\partial e(i)}{\partial m_{vk}} \frac{\partial e(i)}{\partial m_{vj}}, \quad B_k = \sum_{i\in N} -e(i) \frac{\partial e(i)}{\partial m_{vk}}, \tag{4}$$

where $\mathbf{M}_d$ is the transformation matrix at the dth iteration, $\mathbf{A}$ is an 8×8 matrix, $\mathbf{B}$ is an eight-tuple vector, $\partial e(i)$ is the differential of pixel difference in (2), and m_{vk} and m_{vj} are the transformation parameters from m_{v1} to m_{v8}. The iterative process is repeated until the improvement of each parameter converges, or the number of iteration is larger than the maximum number of iteration.

There are several previous studies proposed methods to generate sprite images. For the early developments of sprite, Smolic et al. proposed the technique for long-term global motion estimation and applied the sprite on video coding [13], and Lu et al. proposed an efficient static sprite-generation and the complete compression scheme for background video coding [14]. However, they needed manually label the foreground objects in a sequence before the process of sprite generation because the motion vectors of foreground objects are usually different from the background scenes. Nevertheless, it would be impractical to manually label the foreground regions in a game video, and we propose the method to generate sprite images with foreground removal. The main idea to remove foreground objects is based on the observation that the moving objects do not occupy a fixed position on the game court for a long time. Thus, we can assume that the maximum temporal distribution on each pixel location should be the background

$$E^{(1)}_{x_i,y_i} = \arg\max_k h_{x_i,y_i}(k), \tag{5}$$

$$h_{x_i,y_i}(k) = \sum_{t=t_1}^{t_2} \delta(I_t(x_i,y_i) - k), \quad \forall k \in \mathbf{C}, \tag{6}$$

where $E^{(1)}_{x_i,y_i}$ is the pixel-value with maximum appearance probability at the coordinates (x_i,y_i); the index 1 implies that $E^{(1)}_{x_i,y_i}$ is the initial pixel-value of the generated sprite, $h_{x_i,y_i}(k)$ is the appearance number of pixel-value k under a period time $[t_1,t_2]$ at the coordinates (x_i,y_i) on the sprite, $\delta(\cdot)$ is the impulse function, and $\mathbf{C}$ is the RGB color space.

The assumption that a pixel-value with the number of maximum appearance on temporal distribution is a background pixel is true when foreground objects exhibit rapid movement. Nevertheless, the foreground objects and background scenes may have equivalent appearance probabilities if the foreground objects occupy a fixed region for a period of time. Under such conditions, the foreground pixels would be chosen as the background pixels. To solve this problem, information on the temporal distribution of pixels is not enough. Pixel correlation of spatial co-appearance also needs to be considered for determining the background pixels. The pixel correlation of the spatial co-appearance is described in Fig. 4. Each chosen pixel on the sprite has an individual co-appearance probability corresponding to the surrounding pixels. The current pixel-value, $E^{(n)}_{x_i,y_i}$ is updated by the co-appearance probability of surrounding pixels in the correlation region $\mathbf{R}$. The probability of co-appearance can be mathematically expressed as in (7)

$$s^{(n)}_{x_i,y_i}(k,j) = \frac{\sum_{t=t_1}^{t_2} \delta(I_t(x_i,y_i) - k)\delta(I_t(x_j,y_j) - E^{(n)}_{x_j,y_j})}{\sum_{t=t_1}^{t_2} \delta(I_t(x_j,y_j) - E^{(n)}_{x_j,y_j})}, \tag{7}$$

where $s^{(n)}_{x_i,y_i}(k,j)$ is the co-appearance probability of pixel-value k at coordinates (x_i,y_i) under the pixel-value $E^{(n)}_{x_j,y_j}$ at coordinates (x_j,y_j) in a period of time $[t_1,t_2]$, and the index n is the iteration number of pixel updates. Note that each pixel in the region $\mathbf{R}$ has an effect on the update of the current pixel. Then, the current pixel-value is updated by the value k with maximum summation probabilities. The updated pixel-value $E^{(n+1)}_{x_i,y_i}$ is written in (8)

$$E^{(n+1)}_{x_i,y_i} = \arg\max_k \sum_{\forall j\in \mathbf{R}} s^{(n)}_{x_i,y_i}(k,j). \tag{8}$$

It should be noticed that each pixel-value on the sprite is iteratively updated by (7) and (8). The update process is repeated, until all the

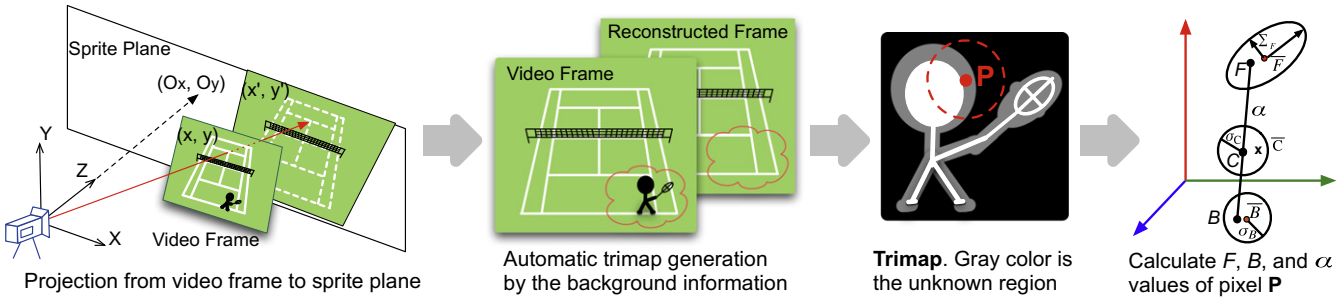

Fig. 3. Processing flow of background projection and foreground segmentation.

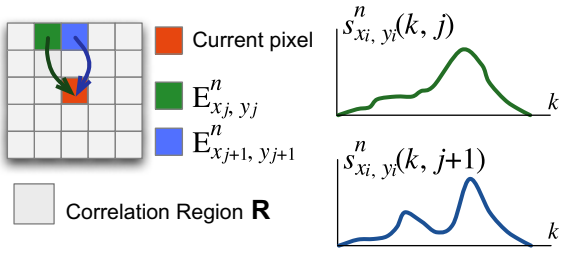

Fig. 4. The current pixel is updated by the pixel-value k with maximum co-appearance probabilities.

pixel-values on the sprite are converged or the number of iteration is larger than the threshold. Finally, the pixel-values on the sprite are used as the background scene.

2.2. Segmentation of foreground objects

Some studies describing the segmentation by the background information can be found in the previous literature. Han et al. trained the dominant background color based on Gaussian mixture models and extracted foreground objects by removing the pixel belong to background color [15]. This method could effectively remove the foreground objects in the background with homogeneous color but failed in the non-homogeneous background. For methods with background modeling, Chien et al. proposed an efficient method to build background scene while foreground objects were segmented by frame difference [16]. However, this method only worked under the steady camera. For tennis videos, the possible camera motions are the panning, titling, and zooming, which make the segmentation process become more difficult.

In this section, a segmentation method combining background information and matting model is proposed. First, the video frame without foreground objects can be reconstructed from the sprite plane by the transformation parameters in (1), which is also called reconstructed frame in Fig. 3. Next, the foreground objects can be segmented by the frame difference [17]. However, the segmentation results were sometimes unacceptable under the situation that the foreground color is similar to the background color. It is because how to set a proper threshold to detect the region belong to the foreground or background is difficult. To improve the segmentation results, some methods called soft segmentation were proposed like Bayesian Matting [18]. Bayesian Matting models both the foreground and background color distributions with spatially-varying sets of Gaussians, and assumes a fractional blending of the foreground and background colors to produce the final output. The equation of color blending is shown in the following equation

$$C = \alpha F + (1 - \alpha)B, \tag{9}$$

where C, F, and B are pixel's composite, foreground, and background color, respectively, and α is the pixel's opacity component used to linearly blend between the foreground and background. The segmentation results of Bayesian Matting are better than the results of frame difference; however, the trimap needs to be manually labeled before the matting procedure. As shown in Fig. 3, the trimap is a map indicating the foreground (white), background (black), and unknown (gray) regions on the image. The generation of a trimap is time consuming and requires human-assistance, and therefore previous researchers have developed some methods to facilitate trimap generation. For example, Chuang et al. used optical flow and background estimation for trimap prediction to reduce the efforts of human-assistance [19]. Unfortunately, fully-automatic generation of a trimap was not achievable in these studies.

For the automatic generation of a trimap, the pixel difference between the reconstructed frame and the video frame is calculated to decide the foreground, background, and unknown regions. Unlike setting a threshold T to label a pixel belonging to the foreground or background, we set an obviously low threshold T_l and an obviously high threshold T_h to decide the background region and the foreground region, respectively. The pixel difference between T_l and T_h is labeled as the unknown region as the gray region in Fig. 3. After that, the trimap is automatically generated without human-assistance. After trimap generation, a pixel's composition C can be modeled as a Gaussian probability distribution center as the predicted color with standard deviation σ_C. The spatial coherence of the image is estimated to model foreground distribution. Log likelihood for the foreground can be modeled as an oriented elliptical Gaussian distribution center as $\overline{F}$ with a weighted covariance Σ_F. Although the background color B is already known in our application, we still model a Gaussian probability distribution center as $\overline{B}$ with the standard deviation σ_B to model the camera noise. The equation of matting procedure is derived as the following equation:

$$\begin{bmatrix} \Sigma_F^{-1} + I\alpha^2/\sigma_C^2 & I\alpha(1-\alpha)/\sigma_C^2 \\ I\alpha(1-\alpha)/\sigma_C^2 & I(1/\sigma_B^2 + (1-\alpha)^2/\alpha^2) \end{bmatrix} \begin{bmatrix} F \\ B \end{bmatrix} = \begin{bmatrix} \Sigma_F^{-1}\overline{F} + C\alpha/\sigma_C^2 \\ \overline{B}/\sigma_B^2 + C(1-\alpha)/\sigma_C^2 \end{bmatrix}, \tag{10}$$

where I is a 3×3 identity matrix. The details of equation solving are explained in [18]. The last step in Fig. 3 illustrates the model of matting equation in (10), and we can see that a pixel's composition C is modeled as a Gaussian probability distribution center as the color $\overline{C}$ on the frame in *RGB* color space with standard deviation σ_C. $\overline{B}$ is the pixel value on corresponding position in the sprite plane, and the background pixel B is modeled as a Gaussian probability distribution with the standard deviation σ_B. The pixel value of foreground F can be modeled as an oriented elliptical Gaussian distribution center as $\overline{F}$ with a weighted covariance Σ_F in *RGB* color space. Note that the variation of Gaussian distribution of foreground pixel is modeled as the weighted covariance but not a unit distribution σ_F because $\overline{F}$ is an average pixel-value summarized by the surrounding pixels In other words, the foreground pixel is modeled with the distribution of surrounding pixels that is different from the model of camera noise of the observed pixel C and the known pixel B. The weighted covariance Σ_F is the pixel distribution calculated from the surrounding pixels.

The background pixel B, foreground pixel F, and α can be extracted by solving (10). Unlike the results of binary segmentation, the segmentation mask of matting results is gray-level distribution. The brighter regions mean the larger blending ratio of foreground object, and the darker regions mean the less blending ratio of foreground object.

2.3. Event detection

Facing the large quantity of sport videos, viewers can easily browse the game video and find out hot events and highlights with video analysis tools. Several previous works on sport video annotation can be found in literatures. For instance, Tien et al. provided a favorable way, called as Sports Wizard, for the user to browse sports videos based on semantic concepts or game structure [20]. Zhang and Chang detected baseball event using superimposed caption recognition [21]. Hung and Hsieh combined captions and visual features for semantic event detection of baseball video [22]. It can be found that lots information can be extracted from the videos and used for video annotation. Especially, the score box provides plenty of event information in a sports video, and it can be applied on a tennis video as well. The score box usually has the same style, character type, and exists in the corner of screen in a broadcasting video. Therefore, we can set score box location and save all number/character images in advance as a prior knowledge. To recognize the numbers in score box, template

matching is employed to find the best matched pre-stored number images. With this method, the score numbers and server in tennis games are extracted.

Next, the semantic information can be retrieved from the foreground segmentation. In many sports videos, the moment, when the player hits the ball, is a key event and contains semantic information for video annotation, like shots in soccer [23], strikes in baseball [24], shoot in basketball [25], and rallies in tennis. The player's positions at the hit moment are the important information for video annotation. Several previous works proposed methods to detect the hit moment. For example, Cai et al. detected highlight sound in an audio stream and employed Hidden Markov modes (HMMs) to model those sound effects for event detection [26]. Chen et al. presented a physics-based scheme which utilized the motion characteristics to extract ball trajectory from lots of moving objects [27]. Tien et al. presented an approach that employed visual and aural cues to perform event detection in tennis videos [28]. A Kalman-based prediction model was proposed to model the ball trajectories in a tennis video, and a moment was recognized as the hit time when the ball was close to the player [17]. In this paper, we adapt Cai's method to detect the hit sound in an audio stream but not to detect ball trajectory in a video. The main reason is that the ball in tennis videos is extremely small, and the ball often disappears from the video frame because of the high speed movement, quantization of video coding, or being occluded by players. After hit time detection, the player's position at the hit moment is represented as the label in Fig. 5, and these positions in each Video Unit are recorded as hit-position patterns like (**c d e d**) or (**c c a**). Note that the rally count of each Video Unit is easily presented as the length of a position pattern. Combing the score box information and hit position patterns, each Video Unit is annotated by the events shown in Table 1. The annotation result includes seven events: ace, fault or net, double fault, hit-before-net, break point, rally, and null. For example, an ace event is judged as the server's score, and only one player hits the ball.

Table 1
Information for video annotation.

Event	Score info.	Pattern of hit position
Ace	Server gets score	Length equal to 1
Fault or net	Score fixed	Length equal to 1
Double fault	Receiver gets score	Length equal to 1
Hit-before-net	Do not care	a or b exists
Break point	Final score and server loses	Do not care
Rally	Score changed	Length longer than 1
Null	Score fixed	Length equal to 0

3. Rendering of video contents

A number of studies have been published on sports video analysis [29,30], highlight detection [31–33], and content enrichment of sports videos [34,35]. These previous works have provided people with a convenient way to watch sports highlights and an interesting way to enjoy games. To further improve the viewing experience, game videos should provide interactive functions or customized video contents to viewers. However, only a few pieces of literature on video interaction can be found. In this section, we propose video rendering to achieve above requirements.

3.1. Flow of video rendering

Instead of directly playing the game video, the proposed video rendering reintegrates the extracted information in Section 2 and generates video contents according to the user's requests. The concept of video rendering brings the viewers a fresh viewing experience and provides more enjoyment from watching games. The proposed video rendering can provide real-time responses to immediately satisfy a user's requests. Fig. 6 is the proposed rendering flow, which contains three main steps: sprite-plane processing, watching-view generation, and foreground pasting.

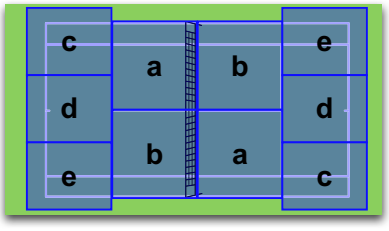

Fig. 5. Labels of player's position at the hit time.

3.1.1. Sprite-plane processing

The first step of video rendering is to modify the sprite plane in order to achieve some visual effects. Several possible functions are described.

- **Score** is a function that displays the game scores on the court, which gives viewers a fresh alternative to score boxes. To achieve this function, the score is painted on the target region of the sprite plane as seen in the first step in Fig. 6. In order to make the painted score seamless with the court, the luminance and color of the painted score are adjusted to fit the background color through Poisson image editing [36].
- **Comment** is a function that lets viewers insert comments or annotate texts on the play court. Sometimes, these insertions could even be a watermark to protect the video contents from being illegally copied. To achieve this function, the text comments are painted on sprite plane.
- **Advertisement** is a function that lets the video provider display advertisements in video or text formats during the game. Note that the insertion of advertisements will not interrupt the game proceedings, which should not detract from the excitement of watching game videos. To achieve this function, the advertisements are displayed on a specific region of the sprite plane. By adjusting the alpha value of luminance and color, the advertisements can be seamlessly integrated into the video.

3.1.2. Watching-view rendering

After sprite-plane processing, the watching view is rendered from the edited sprite plane with the transformation matrix in the following equation

$$\begin{bmatrix} x_w \\ y_w \\ w_w \end{bmatrix} = \begin{bmatrix} m_{s1} & m_{s2} & m_{s3} \\ m_{s4} & m_{s5} & m_{s6} \\ m_{s7} & m_{s8} & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ w_s \end{bmatrix}, \tag{11}$$

where $(x_s/w_s, y_s/w_s)$ are the coordinates in the sprite plane, $(x_w/w_w, y_w/w_w)$ are the coordinates in the watching view, and m_{s1}–m_{s8} are the transformation parameters. The watching view is equal to the original video frame when (11) is equal to the inverse of (1). In addition, the rendering view can be different from the original video by modifying the transformation parameters. For example, the watching view is shifting to the left half of the court by decreasing the m_{s3}, and it is focused on a specific player by increasing the m_{s1} and m_{s5}. Two functions of game watching are created by modifying the transformation parameters.

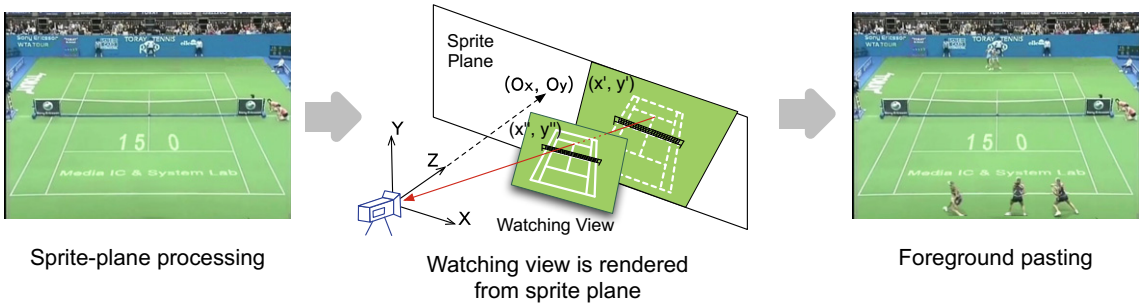

Fig. 6. Proposed rendering flow, which consists of three steps: sprite-plane processing, watching-view rendering, and foreground pasting.

- **Player Tracker** is a function that controls the camera to track a specific player using zooming in. This viewing effect is like putting a magnifying glass focusing on the player. A game video that uses the zoom-in effect can provide a better watch experience for devices with smaller screens. To achieve this function, the zooming factors, m_{s1} and m_{s5}, are increased in order to create a camera zoom-in effect. The translation parameters, m_{s3} and m_{s6}, add translation velocity V_h and V_v in the horizontal and vertical direction respectively to make a player appear inside the window.
- **Ball Tracker** is a function that controls the movement of the camera by following the ball's trajectory. Contrary to focus on the player's position in Player Tracker, the transformation parameters are calculated using information of hit moment described in Section 2.2. In order to provide more comfortable rendering results, the camera movement is modeled as the quadratic movement between two adjacent players' positions at the hit moments.

3.1.3. Foreground pasting

Before pasting foreground objects on the rendered frame, the objects' positions and sizes should be calculated and adjusted because they are all referenced to the video frames. First, the coordinates $(x_v^{(f)}, y_v^{(f)})$ of player feet in the video frame can represent the player's positions on the court. By using (1) and (11), the feet coordinates of the player are transformed from video frame to watching view, and $(x_w^{(f)}/w_w^{(f)}, y_w^{(f)}/w_w^{(f)})$ are the corresponding coordinates. For the adjustment of an object's size, it should be noticed that the object stands on the court and cannot be directly pasted on the watching view by the transformation parameters in (1) and (11). Nevertheless, the object's size should be scaled according to the standing position on the court. The scale ratio of the object from the video frame to watching view can be retrieved by the differentiation results. The foreground objects should be zoomed by the factor calculated from the following equation

$$\text{Scaleratio} = \frac{\partial \frac{x_w^{(f)}}{w_w^{(f)}}}{\partial x_v} \frac{\partial \frac{y_w^{(f)}}{w_w^{(f)}}}{\partial y_v}. \tag{12}$$

In foreground pasting, two functions are created by modifying the pasting number.

- **Continuous** is a function that enables viewers to watch the game videos with multiple players in contiguous motion. Viewers can see the complete highlights of player gestures and also have more interesting viewing experiences. To achieve this function, the players in current time and previous times are pasted. In the experiments, three players are pasted at the time t, $t-1$, and $t-2$ s.
- **Hit Time** is a function that displays multiple players at the current time and previous hit moments. The player's position and gesture at hit moment represent important semantic information in a tennis game. Viewers can learn the game strategy by the positions and gestures of the players at the hit moments.

After the three steps shown in Fig. 6, the enriched video is rendered by enabling the above functions. By using such viewing functions, viewers can have a more interesting and enjoyable viewing experience.

3.2. Video rendering with Scalability

The proposed rendering flow not only provides interesting viewing experience but also gives the scalable property on video bitstream size. Scalability is the property that models video transmission under different bandwidth limitations. Scalable video coding (SVC), the scalable extension of advance video coding (AVC) [37], is a current standardization project for video compression under different transmission bandwidths. It provides variable video sizes by reducing the video resolution (spatial domain), decreasing video frame number (temporal domain), and increasing quantization parameters (PSNR domain). However, the viewing quality for lower video sizes is often seriously decreased and is not acceptable.

Contrary to scaling the bitstream sizes in spatial, temporal, and PSNR domains, the proposed Scalability provides different bitstream sizes in the semantic domain. As mentioned in previous sections, the game video has been decomposed into play shot and non-play shots in the temporal domain, and the video content of play shots are further separated into different layers. By reintegrating this extracted information, the video content can be rendered in different formats for different bitstream sizes. In other words, smaller size is achieved by reducing unnecessary video contents in semantic consideration, but the visual quality is still maintained. The proposed Scalability in semantic domain is classified into four levels as shown in Fig. 7.

3.2.1. Video background reuse

It can be observed that the view of the tennis court rapidly appears in a game video, which covers a large percentage of the area of play shots. If the background court can be reused in the video broadcast, it will obviously decrease the bitstream size. Thus, the bitstream reduction in Level 1 is achieved by reusing the background scenes. Notice that the background court and foreground objects of the play shots are individually transmitted, but the background court is only transmitted once and reused in the subsequent video. Although the background court is abridged in the subsequent transmission, the rendered video in Level 1 is still identical to the original game video. Thus, the bitstream reduction in Level 1 is achieved by reducing the redundant transmission of the background court. It should be noted that non-play shots are transmitted in the single layer without further video processing due to the complexity of the scenes.

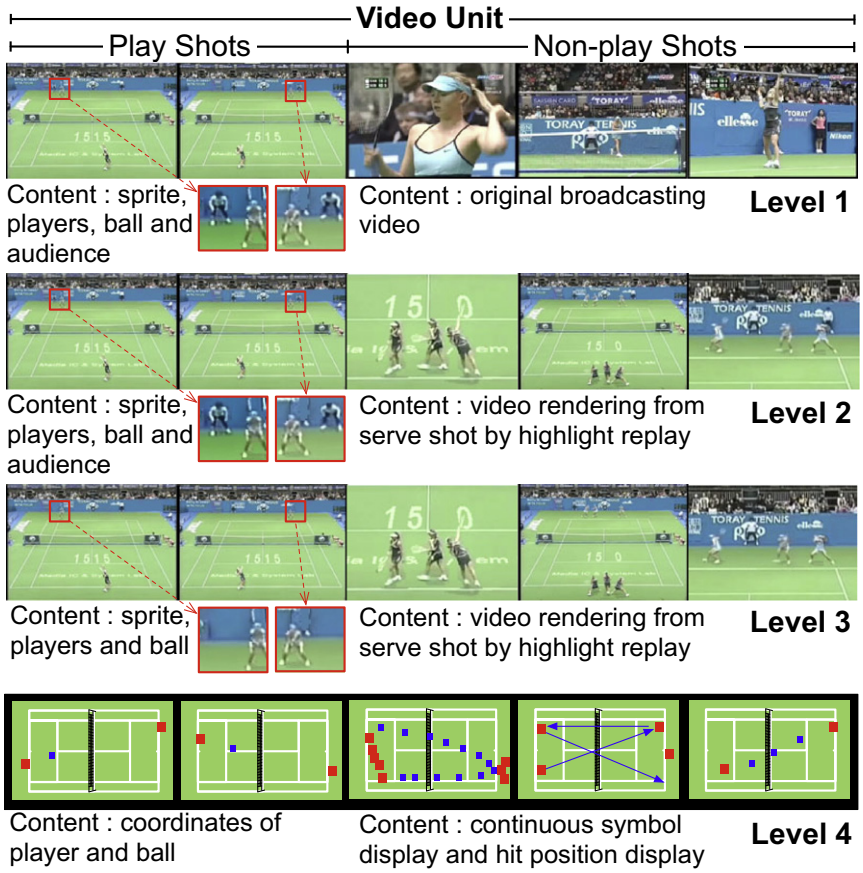

Fig. 7. The illustration of a scalable video. Scalability provides four levels of video content in the semantic domain.

3.2.2. Video clip abandonment

To decrease the bitstream size in Level 2, the approach is to abandon the video clips with less semantic importance. With regard to semantic importance, play shots have more important game information than non-play shots which are usually event replays or the close view of players. Furthermore, the average bitstream sizes for the non-play shots are greater than play shots due to the rapid scene changes. According to the above considerations, the non-play shots are abandoned and the total bitstream size is extremely reduced. To fill the absence from non-play shots, highlights in the play shots are played. The highlights can be a video clip from the play shots with longer player running distances because the situation is exciting when the opposite makes the player run a long distance to return the ball.

3.2.3. Video content abandonment

To further decrease the bitstream size in Level 3, the approach is to abandon certain video contents in the play shots. While watching game videos, people pay most attention to the players and less attention to non-essential objects like the referee, ball boys, and people in the stadium. However, these non-attractive objects take up lots of transmission bandwidth. Therefore non-attractive objects are abandoned to reduce the bitstream size, and only the ball and players are transmitted. To fill in the empty time cause by the absence of the non-play shots, highlight replays, similar to the process described in Level 2, are rendered. In addition, there is an interesting effect in Level 3 that all the objects, besides the ball and players, are static in the video.

3.2.4. Video in animation

To extremely decrease the bitstream size in Level 4, animation is employed to represent the game video. By extending the reduction described in Level 3, only the positions of the ball and players are transmitted. Level 4 is proposed for extremely low transmission bandwidths. Although there are no player postures or other detailed game information, the state of the game is still roughly represented by these coordinates. During the time of non-play shots, statistic data are shown such as player trajectories and hit positions. With the trajectories of the ball and players, the techniques of computer graphics can provide users new experiences of watching game videos [38].

4. Experiments and discussion

Several broadcasting game videos with resolution 720×480 are employed to evaluate the robustness and performance of the proposed system. Notice that all test videos are applying the same threshold in video analysis and rendering. The video demonstrations of Tennis Video 2.0 are available on the website.[2]

4.1. Results of video content analysis

For the settings of background generation in Section 2.1, the parameter **R** is the correlation region to update the current pixel. We have the experiments with different settings of correlation region, e.g., 1×1, 3×3, 5×5, and 7×7 pixels. The results show that the generated sprite plane has less temporal defect with setting a larger parameter **R** but the computation increases exponentially. In addition, we find that sprite defects would be almost removed with setting area larger than 3×3, so we set a range of 5×5 pixels in the experiments.

[2] http://media.ee.ntu.edu.tw/larry/tennis2/.

For the settings of trimap generation in Section 2.2, a pixel is labeled as the background when the difference of pixel-value is less than T_l. A higher setting value of T_l would enlarge the background region in a trimap, and a lower setting value would enlarge the unknown region. The experiments with different setting values of T_l, from 50, 100, 200, 300, 400 to 500, show that there are some over-segmentation in the results with a setting value less than 50 and some defects in the segmented foreground with a setting value larger than 300. Therefore, the value of T_l is set between 50 and 300 to get the better performance. For the setting of T_h, a pixel is labeled as the foreground when the difference of pixel-value is larger than it. A lower setting value of T_h would reduce the unknown region in the trimap, but a higher setting value may reduce the foreground region. We have set various values of T_h, from 500, 600, 700, 800, 900 to 1000, as the experiments and found that segmentation results are almost independent from the T_h if we set a value larger than 500. One of the reasons is that most foreground objects have different color distribution from that of background scenes. As for the situation that the foreground color similar to the background color, the pixel would be labeled as the unknown region in a trimap, and the proposed matting model in (10) would calculate the distribution of foreground and background color. That means, T_h and T_l are not crucial parameters, and the system can work well with fixed values. Note that all segmented results in the figures and demo videos in this paper are with the same settings, $T_h = 900$ and $T_l = 100$.

Fig. 8(a) and (b) is the input video frames with the moving foreground players. Fig. 8(d) and (e) is the initial background scenes composed of pixel-values with maximum probability in temporal distribution in (5). It can see that the moving foreground can be completely removed. Unlike the moving players, the background audiences occupy the fixed region and have gesture change all the time. The pixel-values with temporal peak distribution cannot correctly present the background scene like Fig. 8(c). These inconsistent pixels belong to different objects and make the background scene look like blended with noise. After updating process by the spatial correlation in (8), the inconsistent pixels are iteratively deleted and the background quality is improved in Fig. 8(f). We can see that the update procedure is effective to remove the defect from temporal filter.

Fig. 9 shows the comparisons of foreground segmentation between the proposed methods and previous works [15,17]. Fig. 9(a), (f), (k) and (p) is the input video frames with foreground players, Fig. 9(b), (g), (l) and (q) is the segmentation results of [15], and Fig. 9(c), (h), (m) and (r) is the results of [17]. We find that the methods in [15] can precisely segment the players under the homogeneous background but failed in the non-homogenous background like Fig. 9(p). The methods in [17] can accurately segment the players under the complex background scenes, but the results need to be improved under the foreground color is similar to the background color. With the automatically generated trimap, the proposed segmentation method with matting model calculates the α values in the unknown region. For the results of trimap generation in Section 2.2, Fig. 9(d), (i), (n) and (s) are the automatically generated trimap by the proposed method, and we can see that the regions where the foreground color similar to the background color are labeled as the unknown region (gray color). Fig. 9(e), (i), (o) and (t) are the segmentation results of the proposed method, which show that the proposed methods can correctly segment the players under complex background and the regions where foreground color similar to the background color. In our opinion, the player's shadow is like the blending results of background scene blended and a black color, and it is difficult to segment if the blending ratio of the black color is weak. Therefore, the shadow would only be complete removed or preserved in the previous method like [15,17]. Different from binary segmentation results, the segmentation mask of matting results is gray-level distribution. The brighter regions mean the larger blending ratio of foreground object, and the darker regions mean the less blending ratio of foreground object. Especially, the player's shadow in Fig. 9(f) lightly projects on the court. The shadow would be removed in the binary segmentation methods but preserved in our method. From Fig. 9(j), we can see that the gray-level mask can accurately present the property of shadow. In addition, the shadow can be removed from the segmentation results in Fig. 9(j) with setting a threshold, e.g., 128. However, we do not quantize the segmented results as the binary masks because the gray-level masks not only show the blending property of foreground objects but also bring more seamless pasting in the video rendering. As shown in Fig. 11(b), (e), (g), (i), and (j), the gray-level masks of foreground objects make the video rendering with more natural visual effects.

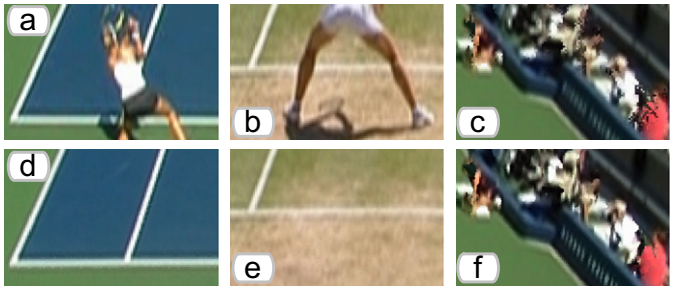

Fig. 8. Experimental results of background generation. (a, b) Input video frames with foreground players. (d, e) Background scenes with the pixels in the temporal peak distribution. (c, f) Background scenes before and after the process of spatial correlation.

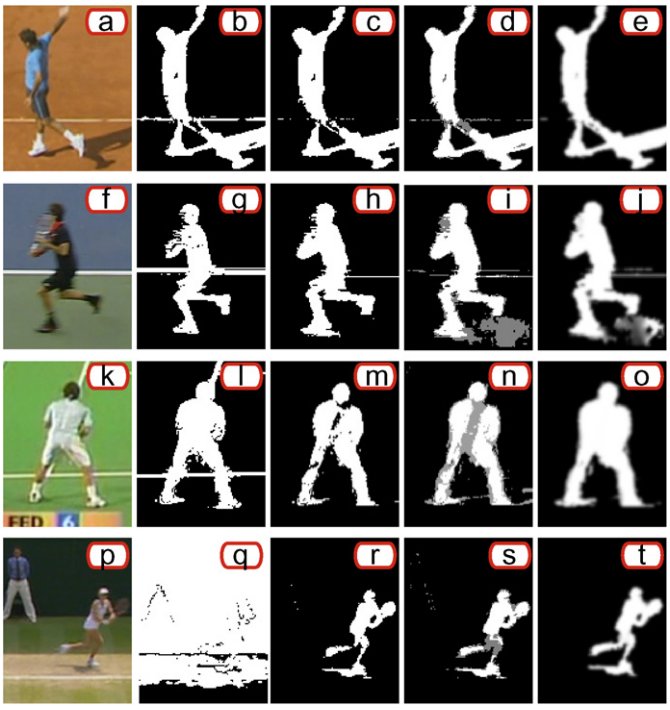

Fig. 9. Experimental results of foreground segmentation. (a, f, k, p) Input video frames. (b, g, l, q) Segmentation results of [15]. (c, h, m, r) Segmentation results of [17]. (d, i, n, s) Proposed methods for trimap generation. (e, i, o, t) Segmentation results of the proposed methods.

4.2. Results of Structure

Structure is the property that shows the video events and provide a method of quickly browsing video content. Viewers can

browse a game's proceedings and watch the highlights using time-based or event-based sorting. Moreover, the GUI of Tennis Video 2.0 in Fig. 10 provides a search function for play strategy. This search function is different from general event searches, and gives viewers the ability to search for a specific game strategy.

For event annotation, score information and hit position patterns in Table 1 are important clues and provide high confidence information. 554 Video Units are used as the experiments. The results of event annotation are shown in Table 2, where both the average recall and precision rates are 93.1%. With regard to the lower rates in the ace event, the lost detection of the hit moment is the main factor that also induces the lower precision rate in a null event. It is because the hitting sound is sometimes covered by the audiences' cheer or a broadcaster's voice, which makes the recognition more difficult. To improve annotation results, the analysis of player gestures and ball trajectories may be the helpful clues in the detection of hit moment.

With regard to the results of strategy search, five search patterns randomly selected were employed for the evaluation. The notations of search patterns are referenced to Fig. 5. For example, search pattern (**e e c a**) is the hit pattern in Fig. 10. In Table 3, the average precision rate is 89.3%, and the average recall rate is 80.6%. It can be observed that the recall rate is not high enough due to the lost detection of hit moments. By adding player gesture analysis and detection of ball trajectories [24,27], some missing or error detection of hit moments would be reduced, and the search results would also be better.

4.3. Results of Interactivity

Fig. 11 shows the experimental results of Interactivity. Fig. 11(a) is the result of **Score**. The original score box on the lower left screen is removed and replaced by a score painted on the court. Using luminance and color adjustments, the score text is now more seamless with the background court. Fig. 11(b) is the result of **Comment**. Text comments from user input can be painted anywhere on the sprite plane, which brings an amazing visual effect. Fig. 11(c) and (d) is the results of **Advertisement** in text and video formats, respectively. We can observe that foreground objects are vividly standing on the advertisement, and these insertions will not interrupt game proceedings. With color blending and luminance adjustment, the insertion has a vivid and seamless visual effects. Fig. 11(e) and (f) is the results of **Hit Time**, and the players at the previous hit time indexes are pasted. The player positions and gestures at hit times can provide significant semantic information

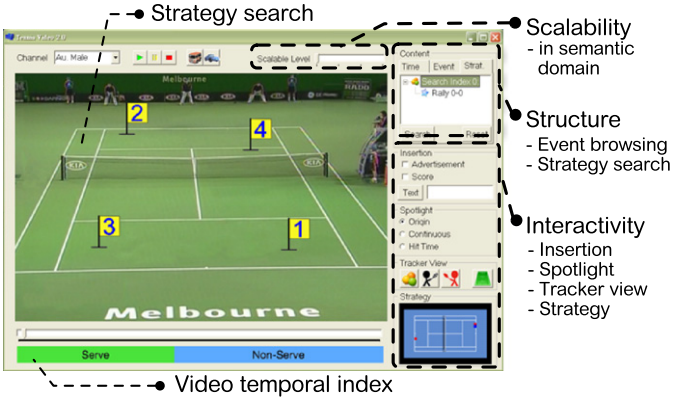

Fig. 10. The graphical user interface of Tennis Video 2.0, which has three properties: Structure, Interactivity, and Scalability to enrich the watching experience. Users can click the bottoms to enjoy the proposed functions and use double-click on the court to search the play strategy.

Table 2
Results of event annotation.

Event	Quantity	Precision (%)	Recall (%)
Ace	21	71.4	75.0
Fault or net	49	81.6	100
Double fault	9	100	100
Hit before net	97	85.6	90.2
Break point	31	100	100
Rally	335	96.7	95.9
Null	12	100	50.0
Total	554	93.1	93.1

Table 3
Results of pattern search.

Search pattern	Quantity	Precision (%)	Recall (%)
e e c	26	92.0	88.5
e e c a	14	90.9	71.4
c c e	30	92.9	86.7
c c e a b	11	80.0	72.7
e e d d e	12	80.0	66.7
Total	93	89.3	80.6

to viewers. Viewers can watch the players in a difficult situation if there is a long distance between adjacent hit time positions. Fig. 11(g) is the result of **Player Tracker** and **Continuous**, and the rendering view is focused on the player with pasting multiple players. Moreover, the zooming effects are suitable to watch the game videos on the smaller display screens. With such a zoom effect, it can provide more comfortable viewing quality. Fig. 11(h) shows the result of **Ball Tracker** and **Continuous**. The rendering view is focused on the ball trajectory, which brings more exciting game experience and is also suitable for the watching on smaller display screen. All the functions proposed in Section 3.1 can be combined on the rendering flow, and Fig. 11(i) and (j) is the results of function combination. This gives viewers a totally fresh experience when watching a game. Fig. 11(k) and (l) is the results of **Strategy**, the window in the right-bottom of GUI in Fig. 10. The coordinates of the players and the ball are illustrated on the court map, and the statistic information of each Video Unit is also presented.

In addition to photo results, more demo videos of Interactivity are available on the website.[3] It should be noted that all of the above experiments represent real-time responses. The computation time of video rendering reaches a speed of 30 720 × 480 video frames per second, which meets the real-time requirement, on a PC with Pentium IV 3 GHz CPU.

4.4. Results of Scalability

The proposed scalable video in semantic domain is presented on four levels in Fig. 7. As mentioned in Section 3.2, lower transmission bit rates are achieved by reducing video contents, but the video quality is still maintained. The corresponding bit rate of video transmission in each level is shown in Fig. 12. Contrary to the method of compressing the whole video frame as seen in general video encoders, the video frames of play shots are decomposed into the background court and foreground objects. Then each component is individually encoded and transmitted.

For the compression in Level 1, the background court is encoded using Lossless JPEG [39], and the average compression rate is 6.453 times. In other words, a background court with a 1080 × 720 resolution in RGB format has a 362 KB file size. Notice that the sprite

[3] http://media.ee.ntu.edu.tw/larry/tennis2/.

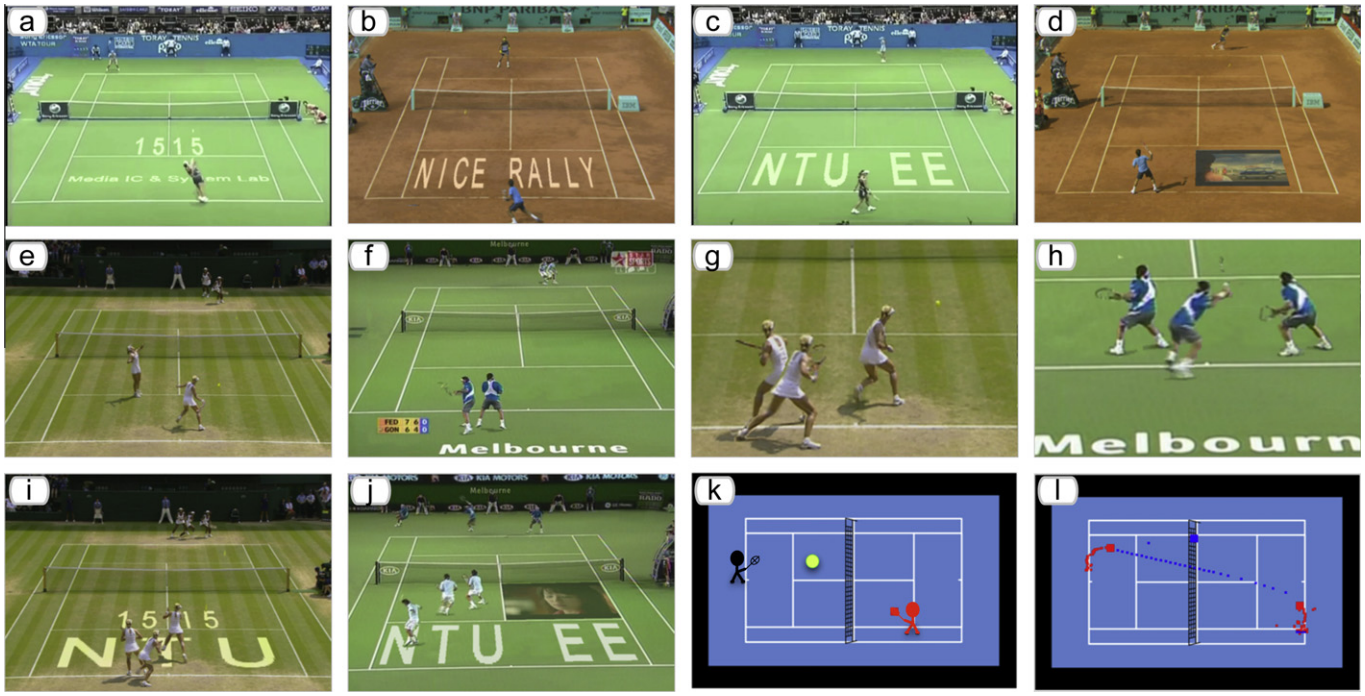

Fig. 11. Experimental results of Interactivity: (a–d) Insertion of Score and Advertisement. (e, f) Hit Time. (g) Ball Tracker with Continuous. (h) Player Tracker with Continuous. (i, j) Function combinations. (k, l) Strategy display.

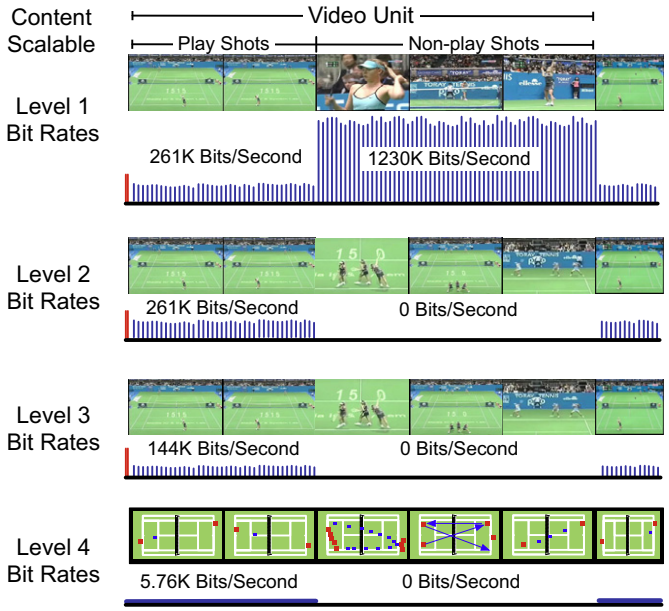

Fig. 12. The bit rates of video transmission on each level. The red bin at the beginning is the transmission bit rate of the background court, and the blue bins are the average bit rates for video contents. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

image is reused in the video, so the it only requires to be transmitted once in the beginning as the red[4] bin in Fig. 12. For the compression of the foreground objects, all of the foreground objects are encoded by the main profile of the H.264 video encoder (JM15) [37], and the average bitrates are 261 K bits per second (bps). The non-play shots are also encoded by the main profile of the H.264 video encoder (JM15), and the average bitrates are 1230 Kbps. We can see that the background reuse in the play shots allows for lower bit rates in comparison with the bitrates of non-play shots.

For the compression bitrates in Level 2, the background court and foreground objects of the play shots are individually encoded by the same method as Level 1. Due to the abandonment of non-play shots, highlight replays are rendered from the play shots, which do not require additional transmission data. The bit rates in the time interval of the non-play shots are zero, but it requires the additional cost of a buffer to store the contents of the play shots. We can see that the abandonment of non-play shots has a dramatic effect on bit rate reduction in comparison with Level 1.

For the compression bitrates in Level 3, the background court and foreground objects are individually encoded similar to in Level 2, the difference being that the foreground objects only include the ball and players. The average bit rate for the foreground objects is only 144 Kbps, which is about half size required by Level 2. Because of the abandonment of non-play shots, the bit rates of the highlight replays are zero, the same as Level 2. We have observed that the viewing quality in Level 3 is almost identical to Level 2, even though the non-essential foreground objects were abandoned.

For the compression bitrates in Level 4, only the position coordinates of the ball and the players are transmitted. Instead of video frames, the positions of the ball and players are displayed on the court map. The data size of these coordinates is only 5.76 Kbps. The statistic data during the time interval of non-play shots is retrieved from the coordinates of play shots. The total bit rate in Level 4 is extremely lower than other levels.

The comparisons of bitstream size and visual quality between the Scalability and SVC are shown in Fig. 13. For SVC in spatial domain, the lower bitstream size is achieved by reducing the video resolution, but the visual quality is declined from the smaller display regions. For SVC in temporal domain, the lower bitstream size is achieved by reducing the frame rate, however the visual quality

[4] For interpretation of color in Figs. 2–13, the reader is referred to the web version of this article.

Fig. 13. The comparisons of bitstream size and visual quality of SVC in spatial, temporal, PSNR domain, and the proposed Scalability in semantic domain.

is decreased from the discontinuous camera motion and player postures. For SVC in PSNR domain, the lower bitstream size is achieved by increasing the quantization parameters (QP), nevertheless the visual quality is declined from the blurred video. The lower bitstream size of the proposed Scalability is achieved by abandoning the contents in the priority of semantic importance, and the visual quality can still be maintained. Notice that the ball boy is disappeared in the second level of the proposed Scalability, but people may not find it out in watching the game videos. We see that the visual quality of the proposed Scalability is better than SVC. For example, third level in spatial domain, third level in temporal domain, second level in PSNR domain, and second level in the proposed Scalability roughly have the same bitrate, and the visual effect of the proposed Scalability is more acceptable. The demo videos are available on the website,[5] which show the detailed visual effects of the comparisons. Furthermore, we also employ the user study to evaluate the performance between SVC and Scalability in Section 4.5.

4.5. Subjective evaluation

For the evaluation by user study, we design a subjective evaluation for 20 evaluators, who are the graduate students. Among the evaluators, 10 have the habits in watching tennis videos, and the others do not have the habits in watching tennis videos. Eleven questions are designed to evaluate the properties of Tennis Video 2.0.

For subjective evaluation of Structure, the evaluator is required to browse game videos by Windows Media Player. After that, the evaluator is required to browse game videos by the functions of Structure and give the score (1–9) of satisfaction, ex: 1: Very unsatisfied, 3: Unsatisfied, 5: No difference, 7: Satisfied, 9: Very satisfied.

Questions about Structure:

Q1.1 Do you feel more convenient with the functions of Structure in watching game videos?
Q1.2 Do you think the functions of Structure help you to experience this game?
Q1.3 Are you willing to watch game videos with the functions of Structure?

For subjective evaluation of Interactivity, the evaluator is required to watch game videos by Windows Media Player. After that, the evaluator is required to watch game videos by the functions of Interactivity and required to give the score (1–9) of satisfaction, ex: 1: Very unsatisfied, 3: Unsatisfied, 5: No difference, 7: Satisfied, 9: Very satisfied.

Questions about Interactivity:

Q2.1 Do you have more fun with the functions of Interactivity in watching game videos?
Q2.2 Do you think the functions of Interactivity help you to have more interaction in watching game videos?

[5] http://media.ee.ntu.edu.tw/larry/tennis2/.

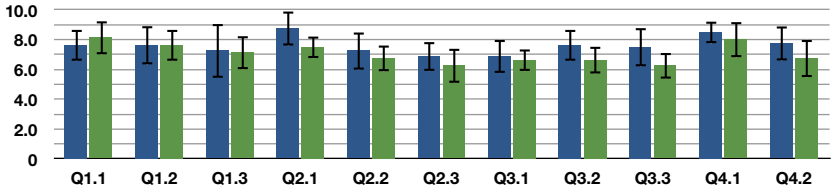

Fig. 14. Results of subjective evaluation. Blue bar is the average score of evaluators with habits in watching tennis videos, and green bar is the average score of evaluators without habits in watching tennis videos. The black lines show the standard deviations. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Q2.3 Are you willing to watch game videos with the functions of Interactivity?

For subjective evaluation of Scalability, the evaluator is required to watch the videos under bitstream reduction by SVC and the functions of Scalability. Three questions are designed to evaluate the visual quality, and the evaluators are required to give the score (1–9) of satisfaction, ex: 1: Very unsatisfied, 3: Unsatisfied, 5: No difference, 7: Satisfied, 9: Very satisfied.

Questions about Scalability:

Q3.1 Do you think the visual quality is more acceptable and game information is clearer under bitstream reduction?
Q3.2 Do you think the functions of Scalability are more practical solutions in the video broadcasting?
Q3.3 Are you willing to watch game videos with the functions of Scalability?

For subjective evaluation of whole system of Tennis Video 2.0, the evaluator is required to give the score (1–9) of Tennis Video 2.0. Questions about Tennis Video 2.0:

Q4.1 Do you think Tennis Video 2.0 is a new presentation of sports videos and gives people better watching experience?
Q4.2 Are you willing to watch game videos with Tennis Video 2.0?

Fig. 14 shows the average scores and standard deviations of the evaluation. For the results of Structure, the results show that evaluators identify the functions of Structure which bring more convenience and help them to experience the game in watching the videos. For the results of Interactivity, the evaluators are all identify the contribution from getting more fun and having more interaction with the game in watching the videos. For the results of Scalability, the evaluators identify the visual quality of semantic Scalability is more acceptable and the game information is clearer than SVC. Furthermore, they think the proposed semantic Scalability are more practical solutions than SVC in video broadcasting. In the evaluation of whole system, the evaluators highly identify that Tennis Video 2.0 is a new presentation of sports videos and give people better viewing experience. In addition, they are willing to watch the game videos using the functions of Tennis Video 2.0. Finally, there is another interesting phenomenon that the scores from evaluators with habits are higher than the scores from evaluators without habits. It seems that people, who often watch tennis games, much identify the contributions of Tennis Video 2.0 and prefer to have these functions.

5. Conclusions and future works

In this paper, a background generation with considering the pixels in temporal and spatial distributions is proposed. Next, a segmentation method combining automatic trimap generation and matting model is proposed to separate the video contents into layers. To provide more functions in watching videos, the rendering flow of video contents and the semantic Scalability are proposed. With new video analysis and rendering tools, a new presentation method for sports videos is proposed with properties of Structure, Interactivity, and Scalability. Structure is the property that ensures video content is well annotated and provides people a convenient way to watch what they want. Interactivity is the property that gives people the ability to customize the video content and watch the enriched game videos. Scalability is the property that enables video content to have different transmission bit rates under different bandwidth limitations.

For the extension to broadcasting systems, Tennis Video 2.0 can be implemented in two ways: game videos processed in broadcasting servers and game videos processed in user clients. For game videos processed in broadcasting servers, the computation of video analysis is done in broadcasting servers, and users only receive the extracted information. In other words, the broadcaster can employ many computation units for real-time video analysis, and game videos with extracted information are received in user clients, which have real-time rendering ability. For game videos processed in user clients, the personal computers, home servers, or set-up-boxes are used for video analysis. Due to the lower computation capability than the devices of the broadcasters, it may spend several hours for one game video. Thus, users can enjoy the game videos only after the off-line computation is finished.

Finally, the proposed methods in the background scene generation and foreground objects segmentation not only effective to tennis videos but also can be applied to other sports videos. The proposed the methods in video rendering and the concepts of Scalability in semantic domain can also be extended to other sports videos. We think the properties of Tennis Video 2.0 can be promoted to more sports videos, for example: Football Video 2.0 and Baseball Video 2.0.

References

[1] A. Kokaram, N. Rea, R. Dahyot, M. Tekalp, P. Bouthemy, P. Gros, I. Sezan, Browsing sports video: trends in sports-related indexing and retrieval work, IEEE Signal Processing Magazine 23 (2) (2006) 47–58.
[2] N. Inamoto, 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, 2004, pp. 42–50.
[3] Y. Li, K.W. Wan, X. Yan, C. Xu, Real time advertisement insertion in baseball video based on advertisement effect, in: Proceedings of the 13th annual ACM International Conference on Multimedia, 2005, pp. 343–346.
[4] Q. Tang, I. Koprinska, J.S. Jin, Content-adaptive transmission of reconstructed soccer goal events over low bandwidth networks, in: Proceedings of the 13th Annual ACM International Conference on Multimedia, 2005.
[5] G. Wikstrand, S. Eriksson, Football animations for mobile phones, in: Proceedings of the Second Nordic Conference on Human–Computer Interaction NordiCHI'02, 2002, pp. 255–258.
[6] J.-H. Lai, S.-Y. Chien, Baseball and tennis video annotation with temporal structure decomposition, in: Proceedings of IEEE International Workshop on Multimedia Signal Processing, MMSP 2008, 2008, pp. 676–679.
[7] C.-Y. Chen, S.-Y. Chien, Y.-H. Chen, Y.-W. Huang, L.-G. Chen, Unsupervised object-based sprite coding system for tennis sport, in: Proceedings of the International Conference on Multimedia and Expo 2003, vol. 1, 2003, pp. I–337–340.

[8] D. Farin, P.H. de With, Enabling arbitrary rotational camera-motion using multi-sprites with minimum coding-cost, IEEE Transactions on Circuit and Systems for Video Technology (2006) 492–506.
[9] K. Mikolajczyk, C. Schmid, Scale and affine invariant interest point detectors, International Journal of Computer Vision 60 (1) (2004) 63–86.
[10] D. Lowe, Distinctive image features from scale-invariant keypoint, International Journal of Computer Vision 60 (2) (2004) 91–110.
[11] M. Brown, D.G. Lowe, Recognising panoramas, in: Proceedings of Ninth IEEE International Conference on Computer Vision, 2003.
[12] S. Baker, I. Matthews, Lucas-kanade 20 years on: a unifying framework, International Journal of Computer Vision (2004) 221–255.
[13] A. Smolic, T. Sikora, J.-R. Ohm, Long-term global motion estimation and its application for sprite coding, content description, and segmentation, IEEE Transactions on Circuit and Systems for Video Technology 9 (8) (1999) 1227–1242.
[14] Y. Lu, W. Gao, F. We, Efficient background video coding with static sprite generation and arbitrary-shape spatial prediction techniques, IEEE Transactions on Circuit and Systems for Video Technology 13 (5) (2003) 394–405.
[15] J. Han, D. Farin, 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 (2008) 1628–1638.
[16] S.-Y. Chien, S.-Y. Ma, L.-G. Chen, Efficient moving object segmentation algorithm using background registration technique, IEEE Transactions on Circuit and Systems for Video Technology 12 (7) (2002) 577–586.
[17] J.-H. Lai, 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, 2008, pp. 672–675.
[18] Y.-Y. Chuang, B. Curless, D. Salesin, R. Szeliski, A bayesian approach to digital matting, in: Proceedings of IEEE Society Computer on Computer Vision and Pattern Recognition, 2001, pp. 264–271.
[19] Y.-Y. Chuang, A. Agarwala, B. Curless, D.H. Salesin, R. Szeliski, Video matting of complex scenes, in: ACM SIGGRAPH 2002, 2002.
[20] M.-C. Tien, Y.-T. Lin, J.-L. Wu, Sports wizard: sports video browsing based on semantic concepts and game structure, in: Proceedings of the Seventeen ACM International Conference on Multimedia, 2009, pp. 1133–1134.
[21] D. Zhang, S.-F. Chang, Event detection in baseball video using superimposed caption recognition, in: Proceedings of the International Conference on Multimedia MULTIMEDIA'02, 2002, pp. 315–318.
[22] M.-H. Hung, C.-H. Hsieh, Event detection of broadcast baseball videos, IEEE Transactions on Circuits and Systems for Video Technology (2008) 1713–1726.
[23] C.-L. Huang, H.-C. Shih, C.-Y. Chao, Semantic analysis of soccer video using dynamic bayesian network, IEEE Transactions on Multimedia 8 (2006) 749–760.
[24] H.-T. Chen, H.-S. Chen, M.-H. Hsiao, W.-J. Tsai, S.-Y. Lee, A trajectory-based ball tracking framework with visual enrichment for broadcast baseball videos, Journal of Information Science and Engineering 24 (2008) 143–157.
[25] H.-T. Chen, M.-C. Tien, Y.-W. Chen, W.-J. Tsai, S.-Y. Lee, Physics-based ball tracking and 3d trajectory reconstruction with applications to shooting location estimation in basketball video, Journal of Visual Communication and Image Representation 20 (2009) 204–216.
[26] R. Cai, L. Lu, H.-J. Zhang, L.-H. Cai, Highlight sound effects detection in audio stream, in: IEEE International Conference on Multimedia and Expo, 2003, pp. 37–40.
[27] H.-T. Chen, H.-S. Chen, S.-Y. Lee, Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2007, pp. 1097–1100.
[28] M.-C. Tien, Y.-T. Wang, C.-W. Chou, K.-Y. Hsieh, W.-T. Chu, J.-L. Wu, Event detection in tennis matches based on video data mining, in: Proceedings of the International Conference on Multimedia and Expo, 2008, pp. 1477–1480.
[29] X. Yu, C. Xu, H.W. Leong, Q. Tian, Q. Tang, K.W. Wan, Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, in: Proceedings of the Eleventh ACM International Conference on Multimedia, 2003, pp. 11–20.
[30] J. Wang, C. Xu, E. Chng, K. Wah, Q. Tian, Automatic replay generation for soccer video broadcasting, in: Proceedings of the 12th Annual ACM International Conference on Multimedia MULTIMEDIA'04, 2004, pp. 32–39.
[31] N. Rea, R. Dahyot, A. Kokaram, Classification and representation of semantic content in broadcast tennis videos, in: Proceedings of IEEE International Conference on Image Processing, vol. 3, 2005, pp. 1204–1207.
[32] I. Kolonias, J. Kittler, W.J. Christmas, F. Yan, Improving the accuracy of automatic tennis video annotation by high level grammar, in: Proceedings of IEEE International Conference on Image Analysis and Processing Workshop, ICIAPW 2007, 2007, pp. 154–159.
[33] G. Zhu, Q. Huang, C. Xu, L. Xing, W. Gao, H. Yao, Human behavior analysis for highlight ranking in broadcast racket sports video, IEEE Transactions on Multimedia 9 (6) (2007) 1167–1182.
[34] X. Yu, X. Yan, T.T.P. Chi, 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, 2006, pp. 619–622.
[35] C.-H. Chang, K.-Y. Hsieh, M.-C. Chiang, J.-L. Wu, Virtual spotlighted advertising for tennis videos, Journal of Visual Communication and Image Representation 21 (2010) 595–612.
[36] P. Pérez, M. Gangnet, A. Blake, Poisson image editing, ACM SIGGRAPH 22 (2003) 313–318.
[37] ITU, Recommendation H.264: Advanced video coding for generic audiovisual services, International Telecommunication Union, 2003.
[38] D. Liang, Q. Huang, Y. Liu, G. Zhu, W. Gao, Video2cartoon: Generating 3d cartoon from video-cartoon: generating 3d cartoon from broadcast soccer video, IEEE Transactions on Circuits Electronics 53 (3) (2007) 1138–1146.
[39] ITU, Recommendation JPEG Standard, International Telecommunication Union, 1993.