# Player Detection, Tracking and Segmentation in Broadcast Tennis Video

**Chaur-Heh Hsieh*, Chin-Pan Huang, Yao-Chuan Jiang**

*Department of Computer and Communication Engineering, Ming Chuan University*

## ABSTRACT

Automatic sports video analysis has attracted many research interests recently. Player figure extraction from videos is a fundamental but also most challenging task. This paper presents a new method for player detection, tracking and segmentation in broadcast tennis video. Utilizing player color and edge information to extract a player figure, several novel schemes are developed. Initially, court lines are detected with adaptive thresholding. Then an adaptive search window which generates varying player bounding box is designed to make the tracking more efficient and reliable. The detected box is further processed by using the combination of non-dominant color extraction and edge detection filters to obtain the raw player figure. Moreover, a modern shadow removal scheme is developed to refine the figure. The experimental results indicate that the proposed algorithm achieves excellent tracking performance and yields a complete player figure under different environmental conditions.

**Keywords:** sport video analysis, player detection and segmentation, automatic annotation

# 廣播網球視訊中球員的偵測、追蹤和分割研究

***謝朝和、黃金本、姜耀川**

銘傳大學 資訊傳播工程學系

## 摘　　要

近年來，全自動的運動視訊分析已經引起了廣泛的研究興趣。從視訊中擷取球員外形是一個基本而具挑戰性的任務。本文提出一新的偵測、追蹤和分割廣播網球視訊中球員的方法，即利用球員的顏色和邊緣信息，開發新的擷取球員外形的方法。首先，採用調適的門檻值以偵測球場線；其次設計調適追蹤窗以適應隨時間變動之球員的身形，來提升追蹤的效率及可靠性；最後，藉著應用結合非主要色彩擷取技術和邊緣偵測濾波器，進一步處理該偵測到的窗框，以獲得初始的球員外形；此外，更採用新的陰影去除方法來精緻化偵測到的球員外形。實驗結果顯示此演算法可以達到非常卓越的追蹤性能，且在各種不同環境條件下獲得完整的球員外形。

**關鍵詞：**運動視頻分析，球員的偵測與分割，自動標註

文稿收件日期 102.1.31; 文稿修正後接受日期 102.11.4; *通訊作者
Manuscript received January 31, 2013; revised November 4, 2013; * Corresponding author

# Ⅰ. INTRODUCTION

Automatic sports video analysis has received extensive attention recently, because sport video appeals to large audiences. The analysis of sports video generates various valuable applications such as highlighting, summarization, indexing/retrieval, athlete's training and entertainment. In the past few years, significant content analysis has been performed on various kinds of sports such as soccer, tennis, baseball, American football, and hockey. [1-25].

The major challenge of sport video analysis is how to bridge the gap between low-level features and high-level semantics. The existing content analysis schemes for sport videos could be roughly classified into four different levels of semantics: video structure analysis, shot (scene) classification, highlight detection, and event detection/recognition [25]. More recently, action recognition and/or behavior analysis of players has received much attention since it is very helpful for higher-level understanding of the sport games. The extraction of a complete player figure plays an important role for action recognition and behavior analysis. Therefore, accurate segmentation of a player body is significant. However, the segmentation performance is highly related to the detection and tracking of players. Therefore, the three tasks are often considered together. In this work, we focus on the detection and segmentation of players for broadcast tennis videos. The proposed method can be applied to other racket sport games like badminton and volleyball.

Player detection, tracking and segmentation for broadcast videos are much more difficult than typical videos due to the following reasons [24]:

(a) Cameras are not stationary; they are zoomed and rotated and often follow the players.

(b) The background changes frequently and the player moves randomly during the play.

(c) A player may be segmented into multiple regions because of the differences in the color of shorts, jerseys, and socks used.

(d) Court colors and textures change with different stadiums such as US Open, Wimbledon Open and French Open.

(e) Shadows cast by the players or other objects in the scene.

Many researches which are highly related to our work have been published in past years [26-34]. Early works explore temporal information of frame difference and then perform morphology operations [28, 34, 35] to extract player figure. Those methods are simple and fast, but easily affected by spectator movement or camera view change. Another approach is background subtraction, which constructs a background model to separate players [26, 27, 30, 31, 32, 34]. Major background models include empty court image, mean of continuous frames, and mean of dominant color. The empty court image is hard to obtain because there usually have players in the court such that the background subtraction method is unrealistic. Using continuous frames to set up the statistic background model shows great performance at fixed camera view, but not suitable for circumstance of frequently changed perspectives. The dominant color method has merits of computation simplicity and robustness under different perspectives. However, dominant color selection and range determination are still open problems requiring more efforts. Furthermore, in player detection, the existing algorithms often employ a fixed search window which does not fit a player figure well. Although it makes no difference in tracking players, high level operations, like action recognition, still demand a best fit window and a complete body. In this paper, we propose a novel method for player detection, tracking and segmentation. The contribution of the work includes:

(a) An effective approach for the court line detection is presented which is mainly based on a novel adaptive thresholding scheme. The adaptive thresholding is robust to the change of stadiums and lighting conditions, as well as noise corruption.

(b) An adaptive search window with varying player bounding box is designed to make the extraction of player body more complete and the tracking more efficient and reliable.

(c) A player segmentation method which combines non-dominant color extraction and edge detection is proposed to effectively separate players from background.

(d) A novel cast shadow removal scheme is presented to refine the player figure extraction and player search window.

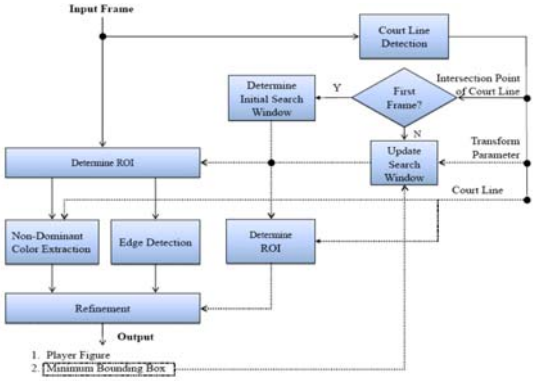
The paper is organized as follows. We first give a brief overview of the proposed system in Section 2. The major units of this system are described in detail in the following sections. Section 3 presents the court line detection algorithm which contains adaptive thresholding and noise removal schemes. The design of adaptive search window for player detection and tracking is presented in Section 4. Player segmentation method is described in Section 5. Section 6 demonstrates the experimental results of different tennis games. Finally, the conclusion is given in Section 7. Some of the results in this paper were presented before in [36].

## Ⅱ. THE PROPOSED SYSTEM

The proposed system for player detection, tracking and segmentation is illustrated in Fig. 1. The objective of the system is to obtain a complete player figure from a tennis video. To achieve high performance, we employ the court model in real-world space. The relationship between image space and real-world space is derived with the help of court lines. Thus, we need to detect court lines as preliminary processing for every input frame. The next step is to determine an appropriate search window for locating a player figure. The search window determines the region of interest (ROI) to be processed. The initial search window for the first frame is fixed, whereas the search windows in the subsequent frames are varying (adaptive). The data in ROI are fed into player segmentation unit which combines non-dominant color extraction with edge detection to extract player information. The segmented result is further refined to achieve a complete player figure. The major units of the system are listed below and their details are described in the following sections.

(a) Court line detection: detect all court lines and calculate their intersection points.
(b) Design of adaptive search window: generate adaptive varying ROI for detection from frame to frame which employs the player speed in real-world space.
(c) Player segmentation: extract complete player figure.
- Extract player figure by combining non-dominant color extraction and edge detection.
- Player figure refinement.
- Calculation of player moving trajectory.

Fig. 1. Flow chart of the proposed system for player detection, tracking and segmentation.




## Ⅲ. COURT LINE DETECTION

Court lines are usually white. Thus to detect white pixels is quite straightforward and efficient for detection of court lines. However, the pixel intensity often changes due to different lighting, courts and climate conditions. Thus, fixed color threshold of white cannot segment lines well enough. In this paper, we propose an adaptive thresholding to increase the robustness of line detection using white pixel.

The court line detection is performed for the first frame of a wide-angle shot which covers a whole court scene. The RGB color space is first transformed into HSV (Hue, Saturation, Value) space. The detection of court line is then performed in V channel. Through binarization and noise removing, Radon transformation (RT) [37] projects the court line image into peaks in Radon space, equations of court lines are then detected from the Radon image. The block diagram of court-line detection process is shown in Fig. 2, and the details will be described in the following.

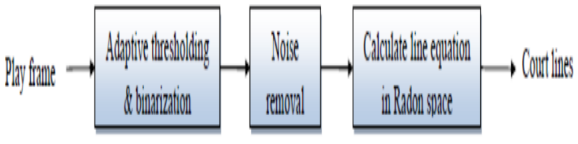



Fig. 2. Flow chart of court line detection.

### 3.1 Adaptive Thresholding and Binarization

Our experiences indicate that for a whole court view, the non-court line pixels occupy more than 80% of the total number of pixels in a frame. Thus, the court lines usually occupy a small number of white pixels that have higher intensity value. Therefore, these pixels of white court lines usually fall on the bins which have low counts and locate on the most right of the histogram of V channel. Fig. 3(a) and 3(b) show a court view and its histogram. The court line pixels probably locate in the bins with a red box. To detect the court line pixels, we need to find a threshold at the left boundary of the red box. The pixels below the threshold are non-court-line pixels, and those above the threshold are court-line ones. As mentioned before, the threshold should adapt to the changes of environments. We propose an adaptive thresholding scheme to detect court line pixels.

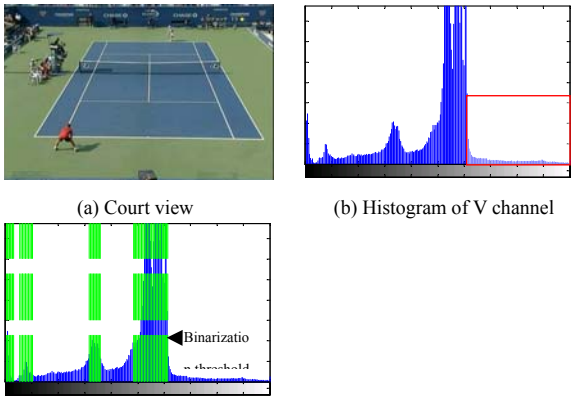
Recall that the court lines often occupy a small number of pixels and have a high intensity (V) value. Therefore, we define the weights of all bins of V histogram as

$$w_i = \frac{h_i}{(i+1)}, \text{ for } i = 0,1,...,255. \qquad (1)$$

where $h_i$ means the pixel count of the $i$th bin. The higher $w_i$ of a pixel, the higher probability of non-court line pixel it belongs to. We sort the histogram bins according to the descending order of weights. The candidates of non-court line pixels can be selected from the histogram bins with larger weight values. So, we select the candidates by accumulating the pixel counts of the sorted bins until the number of pixels accumulated reaches 80% of the total number of pixels. From the candidates selected, we assign the maximal bin as a threshold, as illustrated in Fig. 3(c). The threshold is then employed to determine each pixel as court-line pixel (greater than threshold) or non-court-line pixel (less than threshold). The result is shown Fig. 4(a).

### 3.2 Noise Removing

After the binarization with the adaptive threshold, we obtain the binary image as shown in Fig. 4(a).




(a) Court view (b) Histogram of V channel

(c) Candidate bins in green dash

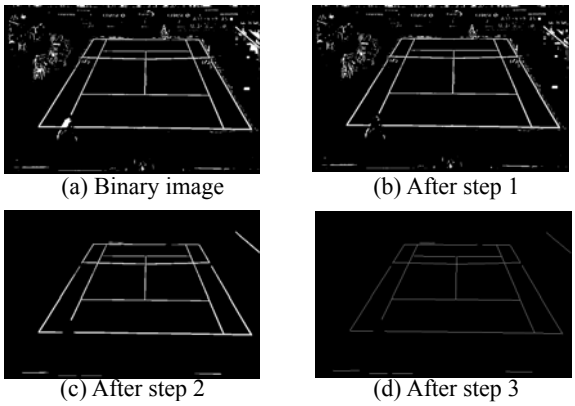
Fig. 3. Illustration of search adaptive threshold value.

It contains white pixels of other objects such as a player wearing in white, a white logo, and so on. These non-line pixels are regarded as noises. To improve the accuracy of line detection, we intend to remove the noises using morphological operations, which contain three steps as follows.

1) Blob removing: A referring structuring element (SE) is a rectangle, e.g. a 7 5 matrix of bit-1. We perform opening (erosion and then dilation) with the SE to the binary image, which removes the blobs larger than the structuring element. The result is shown in Fig. 4(b).
2) Linear object refinement: After blob removing, linear objects are refined by opening operations using flat linear SEs. SEs of l-length and -angle are prepared, e.g. l=20 and = 0°, 1°,⋯, 179°. Then we refine linear objects by the repeated union of the opening results using these SEs and the result is shown in Fig. 4(c).
3) Thinning: At last, the linear objects are thinned into lines of 1-pixel wide as shown in Fig. 4(d).

### 3.3 Peak Search in Radon Space

After noise removal, the binary image is transformed into Radon space through Radon transform [37], and the projection into radius ($r$) and angle ($\theta$) is illustrated in Fig. 5. We found that some light spots or strong peaks appear in the four ranges of 20°~40°, 80°~100°, 140°~160° and 175°~180°. These strong peaks are projected from those thinned lines.



(a) Binary image (b) After step 1

(c) After step 2 (d) After step 3

Fig. 4. Illustration of noise removal.

We can search strong peaks within each range. In Fig. 5, the four search ranges are in red. Blue circles in 80°~100° are the projection of horizontal lines, and green circles for vertical lines.

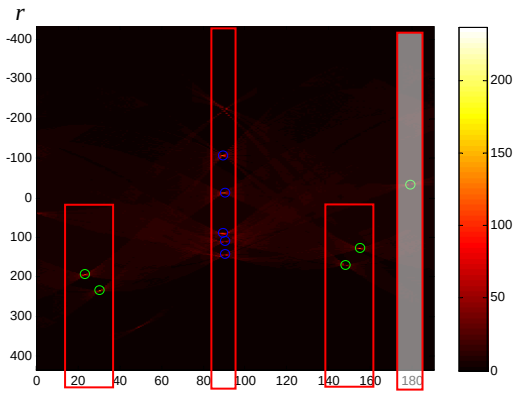



Fig. 5. Radon transformation of thinning lines.

Once the locations of strong peaks are obtained, we can calculate line equations according to pairs of the peaks using Eq. (2). With the line equations and the court model, we can redraw court lines on the image of the court view, as shown in Fig. 6.

$$x\cos\theta + y\sin\theta = r \qquad (2)$$

Fig. 6. Court line detection result (highlighted in green).

# Ⅳ.DESIGN OF ADAPTIVE SEARCH WINDOW

At the first frame, the player position is unknown. By referring to the court model shown in Fig.7, we define initial search areas around the court as follows, where denotes the coordinate of a point. The search areas contain upper court and lower court, as illustrated in Fig.8. The definition of the initial search window reduces the search range which not only avoids the noise interference but also speeds up the detection process of players.

**Upper Court:**

Left: $\left[ X_{P4} - \frac{1}{2} \times ( X_{P4} - X_{P1} ) \right]$ or 0

Right:

$$\left[ X_{P20} + \frac{1}{2} \times \left( X_{P17} - X_{P20} \right) \right] \text{ or } (width\ of\ image)$$

Top: $\left[ \max(Y_{P4}, Y_{P20}) - \frac{2}{3} \times \max(Y_{P4}, Y_{P20}) \right]$ or 0

Bottom: $Y_{Pc}$

**Lower Court**:

Left: $\left[ X_{P1} - \frac{2}{3} X_{P1} \right]$ or 0

Right:

$$\left[ X_{P17} - \frac{2}{3}(width\ of\ image - X_{P17}) \right] \text{or} \left(width\ of\ image\right)$$

Top: $Y_{Pc}$

Bottom:

$$\left\{ \max(Y_{P1}, Y_{P17}) - \frac{2}{3}\left[ height\ of\ image - \max(Y_{P1}, Y_{P17}) \right] \right\}$$

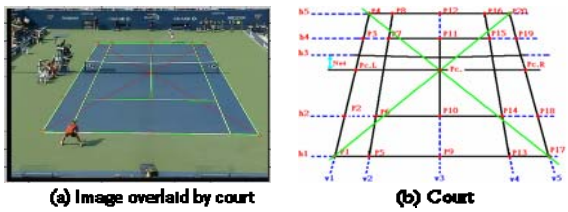


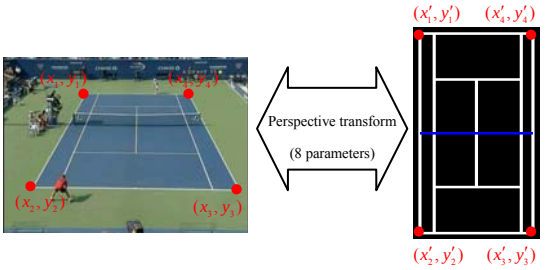

Fig. 7 Court model in image space.

Fig.8. Initial search windows for the first frame.

The initial search window is used to locate the player for the first frame of a video. Once the players are detected, their locations in the subsequent frames can be tracked with a much smaller search window (referred to as tracking window hereafter). In the literature [29, 32, 34], the tracking window size is often fixed for all frames of the video. In addition, the rectangular box which encloses the player (called player window) is fixed as well. Since a player is not a rigid object, and the player posture is varying from frame to frame, the fixed size of windows is inappropriate. In this work, we propose an adaptive search window to efficiently track the deformable player figure. It not only keeps the complete player figure but also reduces the search time and avoids noise interference.

According to [30], the speed of a player is around 2~7 meters per second. As a result, the maximal movement distance of a frame time is 7m/fps, where fps is the frame rate. The distance is employed for the calculation of search window of each frame. Since the player speed is only true in real-world space, we apply the two-dimensional (2-D) perspective transform to relate the coordinate in image space to that in real-world space. The eight transform parameters are estimated from the coordinates of the four corners of the court in image space, and the corresponding coordinates of those in real-world space, as illustrated in Fig. 9 and Eq. (3).

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & x_1x'_1 & y_1x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & x_1y'_1 & y_1y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & x_2x'_2 & y_2x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & x_2y'_2 & y_2y'_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & x_3y'_3 & y_3x'_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & x_3y'_3 & y_3y'_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & x_4x'_4 & y_4x'_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & x_4y'_4 & y_4y'_4 \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4 \end{bmatrix} \quad (3)$$

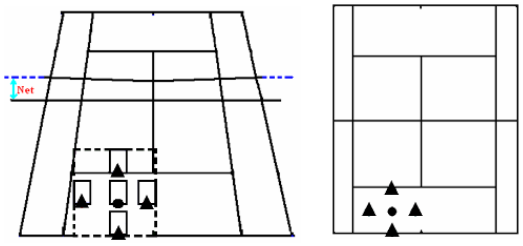



(a) Court in image (b) Court in real-world

Fig. 9 Illustration of the court: (a) in image space and, (b) in real-world space.

The generation of the adaptive search window is illustrated with the lower court in Fig.10 and the procedures are described below:

1. In image space, detect a player and calculate its representative position. Here, we use the center of the bottom line of the player window as the representative position, as the dot shown in Fig. 10(a).
2. Map the representative position back into real court model by perspective transform, as the dot shown in Fig. 10(b).
3. Calculate the maximal possible displaced locations in 4 directions (left, right, up, down) in real-world space, as the triangles shown in Fig. 10(b).
4. Map the four locations in real court model into image space using 2-D perspective transform, as shown the triangles in Fig. 10(a). The resulting locations indicate the possible displaced positions of a player in image space.
5. Each possible displaced position in image space corresponds to a minimal rectangular bounding box. Using the minimal bounding box, we obtain a new search window, which is highlighted by the rectangle of dotted lines, as shown in Fig. 10(a).




(a) Possibleplayer locations in image space (b) Possible player locations in real-world space

Fig. 10. Illustration of adaptive search window.

# V.PLAYER SEGMENTATION

## 5.1 Non-dominant Color Extraction and Edge Detection

After deciding the region of interest of player detection, we can start to extract the player figure. To handle camera viewpoint change, non-dominant color detection is employed in our system. In addition, colors of different parts of the court are affected by light, shadow or camera viewpoint. To get more accurate value, we take advantage of court knowledge and use average color of the field where the player belongs to. According to the model of Fig. 7, we distinguish the court into four areas: inner field of upper court, outer field of upper court, inner field of lower court, and outer field of lower court. The court is split horizontally by net line, while inner and outer fields are defined by court lines. Fig.11 demonstrates the inner and outer fields of lower court.

Upper court inner field:

$$\left[ z \in \left( h5_{\text{down}} \cap v1_{\text{right}} \cap v5_{\text{left}} \right) \middle| y \text{ of } z > Pc \right]$$

Upper court outer field:

$$\left[ z \in h4_{\text{up}} \mid z \notin \left( h4_{\text{up}} \cap h5_{\text{down}} \cap v1_{\text{right}} \cap v5_{\text{left}} \right) \right]$$

Lower court inner field:

$$\left[ z \in \left( h1_{\text{up}} \cap v1_{\text{left}} \cap v5_{\text{right}} \right) \middle| y \text{ of } z < Pc \right]$$

Lower court outer field:

$$\left[ z \in h2_{\text{down}} \mid z \notin \left( h2_{\text{down}} \cap h1_{\text{up}} \cap v1_{\text{right}} \cap v5_{\text{left}} \right) \right]$$

(a) HSV channel image of lower court

(b) Dominant colors of inner field (purple) and outer field (blue)

Fig. 11. Dominant color detection of lower court.

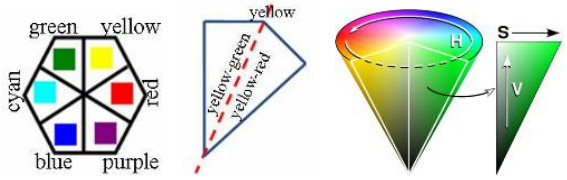
There are various color spaces in the literature. In this paper, we adopt HSV color space since it is very similar to human vision and works well for natural illumination [38]. We select hue and value channels to detect non-dominant color pixels. First we calculate the mean $\mu$ and variance $\sigma^2$ of each channel in the selected region of a court, then use Eq. (4) to determine the non-dominant color pixels.

$$NDC(x,y) = \begin{cases} 1, & if \ |P_H - \mu_H| > \alpha\sigma_H^2 \text{ or } |P_V - \mu_V| > \alpha\sigma_V^2 \\ 0, & otherwise. \end{cases}$$

$$\text{where } \alpha = \frac{0.5 * \beta + \sigma_H^2}{\sigma_H^2}. \tag{4}$$

$P_H$ and $P_V$ denote a pixel value of H and V channels, respectively.

The parameter α is an adjustable parameter which is varied with the court conditions such as different courts or different lighting conditions of the same court. To determine the value of $\beta$ in the above equation, we quantize H into 6 dominant colors, as shown in Fig. 12(a), so the quantization step size is 1/6 (H is normalized into 0 to 1). We define $\beta$ as the maximal variation of a dominant color like "yellow" shown in the Fig.12 (b), which is one half of the quantization step size, that is 0.5*1/6=1/12. The experimental results indicate that the novel non-dominant color extraction (NDC) approach is robust against the varying court colors, which is demonstrated by the examples shown in Fig. 13.

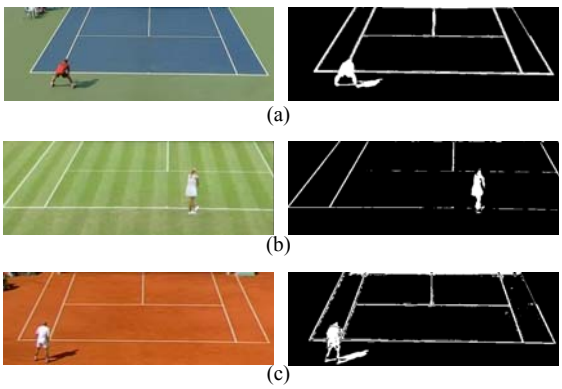



(a) Quantization of H channel

(b) Yellow dominant color and its variation

(c) HSVcolor space [17]

Fig. 12. Illustration of the quantization of color space.



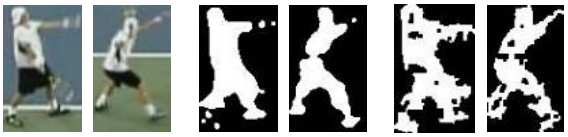
(a)

(b)

(c)

Fig.13. Non-dominant color extraction: (a) US Open, (b) Wimbledon Open, (c) French Open.

Although the color is a robust feature for court analysis, it still faces several difficulties. For example, some players wear white color dress which interferes with the court line detection. In addition, during the fierce competition of a game, players perform various actions, such as swing or serve, which may cause false detection between player figure and background. In order to enhance detection reliability, we add edge detection and utilize the result to compensate non-dominant color extraction. Two examples are shown in Fig. 14, where images in second column are non-dominant color extraction results, and in third column are the results by performing edge detection and then closing operation. As we can see, some parts of player figure are lost in non-dominant color extraction but preserved in edge map, and vice versa. A well-designed combination method is capable of producing a correct and complete player figure.



(a) Original image (b) Extracted with NDC (c) Extracted with edge detection

Fig.14. Results of non-dominant color extraction and edge detection.

Fig.15 illustrates the edge generation procedures. First, the input search image is processed by Sobel filter to produce the horizontal and vertical edge images. Second, each obtained image is smoothed with averaging filter to reduce noises and then binarized by comparing with an adaptive threshold of $2 \times Max\left(\sigma^2_{inside}, \sigma^2_{outside}\right)$, where $\sigma^2_{inside}$ and $\sigma^2_{outside}$ denote the variances of colors of inside field and outside field, respectively. Note that the adaptive threshold is robust to the variation of different courts. Eventually, the horizontal edge map and vertical edge map are generated.

## 5.2 Refinement of Player Figure

The final step, refinement, is to remove undesired information and refine player figure. Fig.16 shows the flowchart of the refinement algorithm. The major steps include:

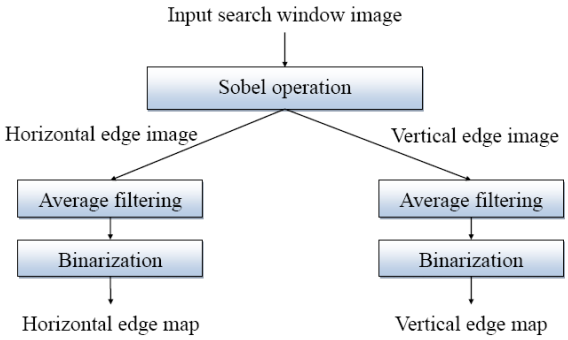



Fig.15. Edge map generation procedures.

1. Remove court lines.
2. Combine horizontal and vertical edge maps.
3. Combine non-dominant color and new edge map.
4. Remove cast shadow.

Three images including NDC map, horizontal and vertical edge maps are fed into the refinement part. First, court lines are removed from the three images. Second, we combine horizontal and vertical edge maps by performing OR operation and use connected component labeling to remove noises. Third, the binary image of non-dominant color and new edge map are merged by OR operation, which generates the binary image as shown in Fig. 17.

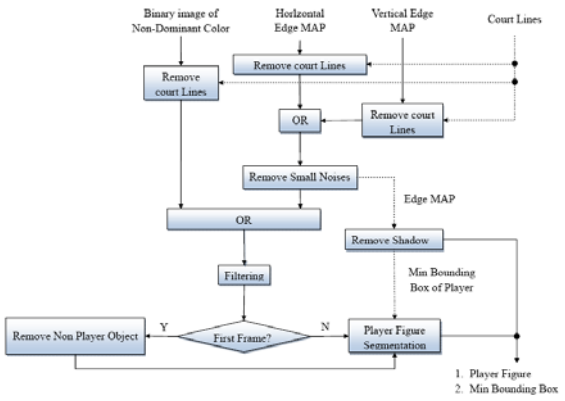



Fig.16. Refinement processes of player segmentation.

Fig. 17. The binary image by merging NDC map and edge map.

The merged result may contain shadows, thus we propose a new shadow removal technique. The shadows can be roughly classified into self shadows and cast shadows [39]. Fig. 18 shows the two types of shadows. The self shadows are highlighted in green and cast shadows are in red. The major cast shadow is in the right-hand side of the player, which affects the player window and player figure extraction significantly.

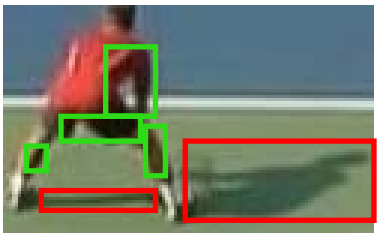


Fig. 18. Self-shadow (highlighted in green) and cast shadow (highlighted in red).

Since the pixels of the self-shadow are the neighbors of the player figure and their colors are very close to the player pixels around them, removing self shadow is error prone and frequently eliminates parts of the player figure as well. Since our goal is to maintain the integrity of the player figure, we concentrate on dealing with cast shadows. However, if we use the visual features of shadows directly, both cast-shadow and self-shadow will be removed. To attack the problem, we develop a novel method as follows. It first detects the range of the cast shadow of the right side of the player as shown in Fig.18. Then, it removes the cast shadow under the player body.

The color of shadow is gray or black, which has high saturation (S), and low value (V) in HSV color space. In addition, the hue (H) value is greater than that of the court color. We first apply the following shadow detection formulas to the edge pixels of the player window, which correspond to the white pixels of the edge map in Fig. 19(c). More precisely, the edge pixels of the player window is

$$P_e(x,y) = P(x,y) \cdot E(x,y) \qquad (5)$$

where $P(x,y)$ is the original player-window-image (Fig. 19(a)) and $E(x,y)$ is the corresponding edge map (Fig.19(c)). Applying shadow detection formula, Eq. (6), to the $P_e(x,y)$ image, we obtain the shadow edge map as shown in Fig. 19(d). The shadow edge map is further filtered by removing the upper two-thirds part of the player window and get Fig. 19(e). By subtracting the filtered result (Fig. 19(e)) from edge map (Fig. 19(c)), the cast shadows in right side and under the player body are completely eliminated, as shown in Fig. 19(f). From this result, we can calculate the range in the horizontal direction of the right-side cast shadow. Based on the range, we crop the NDC map (Fig. 19(b)) and the filtered shadow edge map, and the results are shown in Fig. 19(g) and Fig. 19(h), respectively. By subtracting Fig. 19(g) from Fig. 19(h) and then removing small isolated noises, we get player figure without shadow as demonstrated in Fig. 19(i).

$$-\alpha\sigma_H^2 \le p_H - \mu_H < \frac{1}{6} \text{ and } p_S - \mu_S \ge -\alpha\sigma_S^2 \text{ and } p_V - \mu_V \le \alpha\sigma_V^2 \qquad (6)$$

where $p_H$ and $p_S$ denote the pixel values of H and S channels of $P_e(x,y)$ image in Eq. (5), respectively. The $\mu$ and $\sigma^2$ denote the mean and variance of each signal, separately, and the parameter α is referred to Eq. (4).

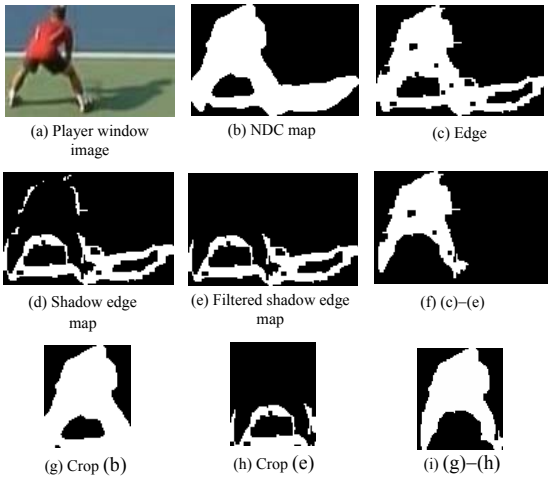



Fig. 19. Illustration of shadow removal.

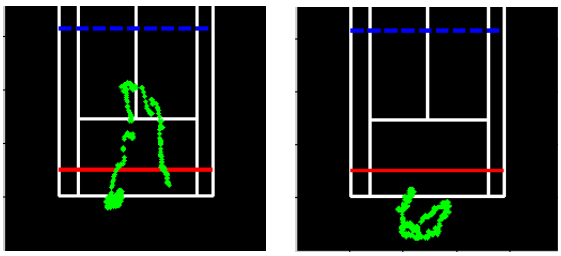
## 5.3 Player Trajectory Calculation

The process in Fig.16 generates player figure and the location of the player window. As stated before, we use the center of the bottom

line of the player window as the location of a player. Using the player location of the first frame and the extracted player figure, we can track the player and extract its figure for the subsequent frames. And then we can obtain the player moving trajectories in a video as demonstrated in the experimental section. It is noted that our work first segments the player figure and then track its position for all frames in a video. The tracking procedures are stated below.

(a) For the next frame, design a search window centered at the player window according to the procedure illustrated in Fig.10. It is noted that the search window is a binary image which contains 1 (player pixel) or 0 (non-player pixel).

(b) Using the player window to slide in the search window pixel by pixel from left to right and then top to bottom. For each search location, we count the number of 1 within the player window, which represents the player area. After all locations are searched, record the location which gives the maximal player area. The procedures are repeated until all the frames of the video are done. Consequently, all locations recorded yield moving trajectories of a player as demonstrated in Fig. 20.



(a) Net approach event (b) Ground stroke event

Fig. 20. Moving trajectories of a player in real- world space.

# VI. EXPERIMENTAL RESULTS

In this section, we provide experiment results of the major units of the proposed system including court line detection, adaptive search window, tracking performance and player segmentation, respectively. The experimental data are selected from 10 videos of US Open, Wimbledon Open and French Open. The proposed algorithms are proved being robust and effective in different courts and under varying lighting conditions.

## 6.1 Court Line Detection

Using the intensity histogram of a video frame, we design a rule to determine an adaptive threshold for court line detection. The adaptive threshold is used to segment white pixels, which is robust against the color variation of courts due to the changing of different courts and different climates, as demonstrated in Fig. 21.

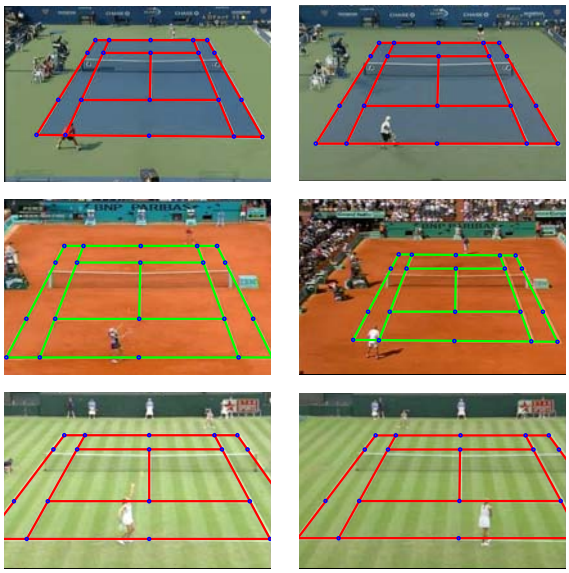


Fig. 21. Results of court line detection.

## 6.2 Adaptive Search Window

Most tracking methods in the literature such as [29, 32, 34] employ fixed size of search windows during the tracking period. The search windows are either too small or too large. Smaller search windows lead to lose parts of player figure, and incur false judgment of player actions. Lager search windows contain too much noise and redundant information, which make tracking inefficient.

Fig. 22 shows the adaptive search window (marked in black) and player window (marked in red) during the tracking period. It can be seen that both windows are changing frame by frame, which is adaptive according to the deformable player. The adaptive window method is more suitable for high level automatic annotation

system.

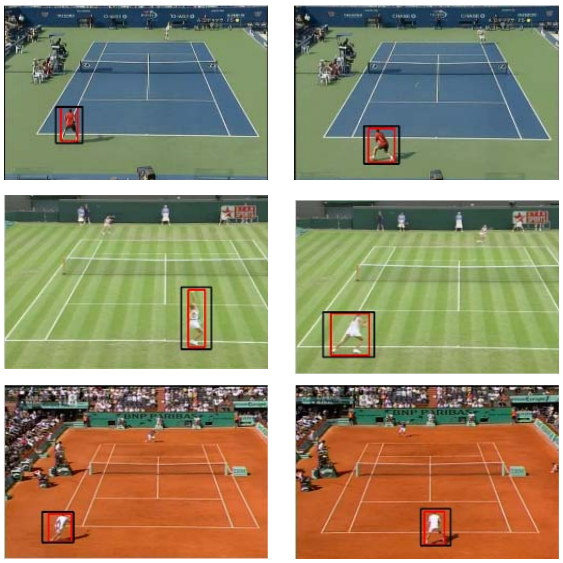


Fig. 22. Experimental results of proposed adaptive search window (search window highlighted in black; player window highlighted in red).

## 6.3 Tracking Performance

In this subsection, we evaluate the tracking performance of the proposed method, and compare it with the method in [40]. In [40], the authors presented a robust tracking algorithm in 3-D domain, and they also proposed an adaptive Double Exponential Smoothing (DES) filter to further smooth the tracking trajectory.

Fig. 23 shows an example of the trajectory generated by the proposed method. The tracking results of the method in [40] without and with DES are demonstrated in Fig. 24(a) and Fig. 24 (b), respectively. It is seen that the significant tracking errors occur in the approach without DES tracking filter. Meanwhile, even with DES, some significant errors still exist such as the area marked with the red rectangle. On the contrast, our approach that does not employ any tracking filter gives better tracking performance. Comparison of the proposed method and the method in [5] with DES for various tennis videos are summarized in Table 1. It indicates that our method achieves better performance than that of the method in [5].

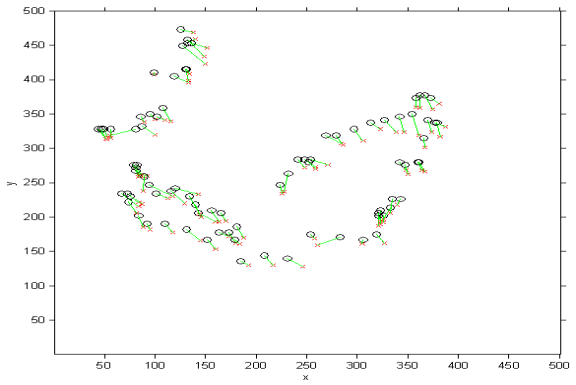
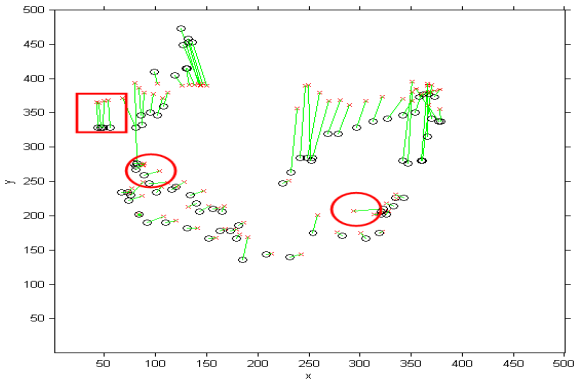
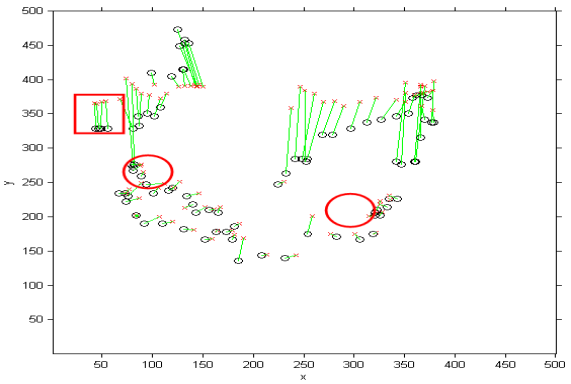


Fig. 23. Tracking performance of the proposed method. The marks o and x denote the ground truth and tracking result, respectively, and the green line segment denotes the tracking error vector.



(a) without DES



(b) with DES

Fig. 24. Tracking performance of the method in [5]. The marks o and x denote the ground truth and tracking result, respectively, and the green line segment denotes the tracking error vector.

Table 1 Comparison of the tracking performance of the proposed method and the method in [5] with DES tracking filter

| | Proposed | [5] with DES |
|---|---|---|
| US Open (with shadow) | 0.3204(m) | 0.3225(m) |
| US Open (without shadow) | 0.1489(m) | 0.2993m) |
| Wimbledon Open | 0.2089(m) | 0.4395(m) |
| French Open | 0.1603(m) | 0.1505(m) |

### 6.4 Player Segmentation

The factors affecting accuracy of player segmentation are player window and segmentation algorithm. The proposed adaptive window and player figure extraction algorithm are robust and effective for different courts and lighting conditions, so the method achieves excellent segmentation results as shown in Fig. 25. The performance of the player figure segmentation is essential to the higher level analysis such as player action recognition and behavior analysis.

Fig.25. Experimentalresults of player segmentation.

### 6.5 Computation Load Analysis

In this subsection, we evaluate the computation load of the proposed method. Our algorithm consists of two major parts: courtline detection and player tracking (segmentation). The average computation times of these two parts for 10 videos with different lengths are shown in Table 2. The proposed algorithm is implemented with Matlab codes. Programs are run on a personal computer with Intel Core i7 CPU and Windows 7 operating system (64-bit version). The average time for a frame is about one second. This can be further reduced with code optimization. This result indicates that the proposed algorithm is highly potential for real time applications.

Table 2 Computational load analysis

| Video | Frame Number | Line Detection (sec) | Player Tracking (sec) | Total Time (sec) |
|---|---|---|---|---|
| 1 | 703 | 0.588 | 0.519 | 1.107 |
| 2 | 5062 | 0.577 | 0.702 | 1.279 |
| 3 | 817 | 0.438 | 0.506 | 0.944 |
| 4 | 420 | 0.607 | 0.51 | 1.117 |
| 5 | 1436 | 0.517 | 0.922 | 1.439 |
| 6 | 760 | 0.466 | 0.513 | 0.979 |
| 7 | 564 | 0.638 | 0.521 | 1.159 |
| 8 | 624 | 0.609 | 0.52 | 1.129 |
| 9 | 4118 | 0.624 | 0.889 | 1.513 |
| 10 | 7610 | 0.559 | 0.514 | 1.073 |
| Average Time (sec) | | | | 1.174 |

## VII. CONCLUSIONS

In this paper, a new algorithm for player detection, tracking and segmentation in broadcast tennis videos has been presented. The method has been able to extract a complete player figure under the different courts and lighting conditions. Several novel schemes have been developed to overcome problems of deformable player figure, varying lighting conditions, camera viewpoint change, and different tennis courts including court line detection with adaptive thresholding, adaptive search window utilizing relation between image space and real-world space, fusion of non-dominant color extraction and edge detection filter. In addition, a novel shadow removal method has been proposed to refine the player figure. Regarding the adaptive search window, we have employed court knowledge and using perspective transform to calculate the search window; for non-dominant color extraction, hue and value have been used as parameters and the region of interest has been deliberately selected; for edge detection, a Sobel filter has been applied for retrieving horizontal and vertical edge maps, which are associated with non-dominant color extraction result to refine the player figure. Around 50 video segments from 12 tennis games have been used to test the algorithm. Experimental results have

demonstrated that the proposed algorithm achieves highly robust court line detection, possesses adaptive search window according to the deformable player, reaches excellent tracking performance, and yields a complete player figure under different environmental conditions. Those prominent features are very useful for the higher-level processing like player action recognition and behavior analysis, which is currently being investigated.

## ACKNOWLEDGEMENTS

This work was supported in part by National Science Counsel, Taiwan under the grant NSC 99-2221-E-130-011-MY3 and NSC 99-2632-E-130-001-MY3.

## REFERENCES

[1] Zhou, W., Dao, S., and Kuo, C. C. J., "On-line knowledge- and rule-based video classification system for video indexing and dissemination," Information Systems, Vol. 27, No. 8, pp. 559-586, 2002.

[2] Ekin, A., Tekalp, A. M., and Mehrotra, R., "Automatic soccer video analysis and summarization," IEEE Trans. on Image Processing, Vol. 12, No. 7, pp. 796-806, 2003.

[3] Zhang, D., and Chang, S.-F., "Real-time view recognition and event detection for sports video," Journal of Visual Communication and Image Representation, Vol. 15, No. 3, pp. 330-347, 2004.

[4] Li, B., Errico, J. H., Paoand H., and Sezan, I., "Bridging the semantic gap in sports video retrieval and summarization," Journal of Visual Communication and Image Representation, Vol. 15, No. 3, pp. 393-424, Sep. 2004.

[5] Xie, L., Xu, P., Chang, S.-F., Divakaran, A., and Sun, H., "Structure analysis of soccer video with domain knowledge and hidden Markov models," Pattern Recognition Letters, Vol. 25, No. 7, pp. 767-775, 2004.

[6] Leonardi, R., Migliorati, P., and Prandini, M., "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains," IEEE Trans. on Circuits Syst. Video Techn., Vol. 14, No.. 5, pp. 634-643, 2004.

[7] Gong, Y., Han, M., Hua, W., and Xu, W., "Maximum entropy model-based baseball highlight detection and classification," Computer Vision and Image Understanding, Vol. 96, No. 2, pp. 181-199, 2004.

[8] Shih, H.-C., and Huang, C. L., "MSN: statistical understanding of broadcasted baseball video using multi-level semantic network," IEEE Trans. on Broadcasting, Vol. 51, No. 4, pp. 449-459, 2005.

[9] Duan, L.Y., Xu, M., Tian, Q., Xu, C., Jin, J. S., "A unified framework for semantic shot classification in sport video," IEEE Trans. on Multimedia, Vol. 7, No. 6, pp. 1066-1083,. 2005.

[10] Sadlier, D., and O'Connor, N., "Event detection in field sports video using audio-visual features and a support vector machine," IEEE Trans. on Circuits Syst. Video Techn., Vol. 15, No. 10, pp. 1125-1233, 2005.

[11] Huang, C. L., Shih, H.-C., and Chao, C.-Y., "Semantic analysis of soccer video using dynamic Bayesian network," IEEE Trans. on Multimedia, Vol. 8, No. 4, pp. 749-760, 2006.

[12] Zhang, D., and Chang, S.-F., "Event detection in baseball video using superimposed caption recognition," in Proceedings of the tenth ACM international conference on Multimedia, France, pp. 315-318, 2002.

[13] Liua, J., Tong, X., Lic, W., Wang, T., Zhang, Y., and Wang, H., "Automatic player detection, labeling and tracking in broadcast soccer video," Pattern Recognition Letters, Vol. 30, No. 2, pp.103-113, 2009.

[14] Jian, J.-L., Hung, M.-H., Hsieh, C.-H., Chang, Y., "Real-Time scene classification for baseball videos," in Proc. CVGIP, Taiwan, pp. 115-122, 2005.

[15] Su, Y.-M., and Hsieh, C.-H., "A novel caption extraction scheme for various sports captions," ICPR 2006, Hong Kong, pp. 1054-1057, 2006.

[16] Su, Y. M., and Hsieh, C.-H., "A novel model-based segmentation approach to extract caption contents on sports videos," ICME 2006, Toronto, pp. 1829-1832, 2006.

[17] Ekin, A., and Tekalp, A. M., "Shot type classification by dominant color for sports video segmentation and summarization," ICASSP 2003, Hong Kong, Vol. 3, pp. 173-176, 2003.
[18] Pei, S.-C., and Chen, F., "Semantic scenes detection and classification in sports videos," in Proc. CVGIP 2003, Taiwan, pp. 210-217, 2003.
[19] Chu, W.-T., and Wu, J.-L., "Development of realistic applications based on explicit event detection in broadcasting baseball videos," in Proceedings of International Multimedia Modelling Conference, pp. 12-19, 2006.
[20] Assfalg, J., and Bertini, M., "Semantic annotation of soccer videos: automatic highlights identification," Computer Vision and Image Understanding, Vol. 92, No. 2-3, pp. 285-305, 2003.
[21] Huang, Y., Llach, J., and Bhagavathy, S., "Players and ball detection in soccer videos based on color segmentation and shape analysis," Proc.MCAM'07, Lecture Notes in Computer Science, Springer-Verlag, Vol. 4577,pp. 416-425, 2007.
[22] Babaguchi, N., Kawai, Y., Ogura T., and Kitahashi, T., "Personalized abstraction of broadcasted American football video by highlight selection," IEEE Transactions on Multimedia, Vol. 6, No. 4, pp. 575-586, 2004.
[23] Li, F., Woodham, R. J., "Analysis of player actions in selected hockey game situations," in Proc. of 2nd Canadian Conference on Computer and Robot Vision(CRV 2005), pp. 152-159, 2005.
[24] Pallavi, V., Mukherjee, J., Majumdar, A. K., and Sural, S., "Graph-Based multiplayer detection and tracking in broadcast soccer videos," IEEE Transactions on Multimedia, Vol.10, No. 5, pp. 794-805, 2008.
[25] Hung, M.H., and Hsieh, C. H., "Event detection of broadcast baseball videos," IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 12, pp. 1713-1726, 2008.
[26] Zhong, D., Chang, S.-F., "Long-term moving object segmentation and tracking using spatiotemporal consistency," IEEE International Conference on Image Processing, Vol. 2, pp. 57-60, 2001.
[27] Zhong, D., Chang, S.-F., "Real-time view recognition and event detection for sports video," Journal of Visual Communication and Image Representation, Vo1. 5, pp. 330-347, 2004.
[28] Miyamori, H., and Iisaku, S.I., "Video annotation for content-based retrieval using human behavior analysis and domain knowledge," Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 320-325, 2000.
[29] Han, J., Farin, D., and de With, P. H. N., "Multi-level analysis of Sports video sequences," Proc. SPIE of Multimedia Content Analysis, Management, and Retrieval, Vol. 6073, 2006.
[30] Han, J., and de With, P. H. N., "A unified and efficient framework for court-net sports videos analysis using 3-D camera modeling," SPIE Electronic Imaging, Vol. 1, pp. 6506-6515, 2007.
[31] Bertini, M., Cucchiara, R., Del Bimbo, A., Prati, A., "Semantic adaptation of sports video with user-centered performance analysis," IEEE Transactions on Multimedia, Vol. 8, No. 3, pp. 433-443, 2006.
[32] Rea, N., Dahyot, R., and Kokaram, A., "Classification and representation of semantic content in broadcast tennis videos," IEEE International Conference on Image Processing, Vol. 3, pp. 1204-1207, 2005.
[33] Zivkovic, Z., Petkovic, M., Mierlo, R. J., Keulen, M., Heijden, F., Jonker, W., and Rijnierse, E., "Two video analysis applications using foreground/background segmentation," Proceedings of the 2003 Conference on Visual Information Engineering, pp. 310-313, 2003.
[34] Zhu, G., Huang, Q., Xu, C., Xing, L., Gao W., and Yao, H., "Human behavior analysis for highlight ranking in broadcast racket sports video," IEEE Transactions on Multimedia, Vol. 9, No. 6, pp. 1167-1182, 2007.
[35] Sudhir, G., Lee, J. C. M., and Jain, A. K., "Automatic classification of tennis video for high-level content-based retrieval," in Proc. Int. Workshop on Content-Based

Access of Image and Video Databases, Bombay, pp. 81-90, 1998.

[36] Jiang, Y. C., Lai, K. T., Hsieh, C. H., Lai, M. F., "Player detection and tracking in broadcast tennis video", The 3rd Pacific Rim Symposium on Advances in Image and Video Technology (PSIVT'09), Lecture Notes in Computer Science, Springer-Verlag, Vol. 5414, pp. 759-770, 2009.

[37] Deans, S. R., The radon transform and some of its applications, New York, John Wiley & Sons, 1983.

[38] http://en.wikipedia.org/wiki/Color_model.

[39] Prati, A., Mikic, I., Trivedi, M., and Cucchiara, R., "Detecting moving shadows: formulation, algorithms and evaluation" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 7, pp. 918-923, 2003.

[40] Han, J., Farin, D., and de With, P. H. N., "Multi-level analysis of sports video sequences," in SPIE Conference on Multimedia Content Analysis, Management, and Retrieval, Vol. 1, pp. 145-153, 2006.