International Journal of Innovative Computing, Information and Control
Volume **8**, Number **10(A)**, October **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6733–6745

AN ADAPTIVE SEARCH WINDOW ALGORITHM FOR PLAYER TRACKING IN BROADCAST TENNIS VIDEOS

Kuan-Ting Lai[1], Chaur-Heh Hsieh[2], Jenn Dong Sun[3] and Yao-Chuan Jiang[4]

[1]Graduate Institute of Electrical Engineering
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan
kuantinglai@gmail.com

[2]Department of Computer and Communication Engineering
Ming Chuan University
No. 5, De-Ming Road, Gwei-Shan, Taoyuan 333, Taiwan
harrishsieh@gmail.com

[3]Department of Computer Science
Chinese Culture University
No. 55, Hwa-Kang Road, Yang-Ming-Shan, Taipei 11114, Taiwan
jdsun@faculty.pccu.edu.tw

[4]Department of Information Engineering
I-Shou University
No. 1, Sec. 1, Syuecheng Road, Dashu District, Kaohsiung 84001, Taiwan

Received January 2011; revised September 2011

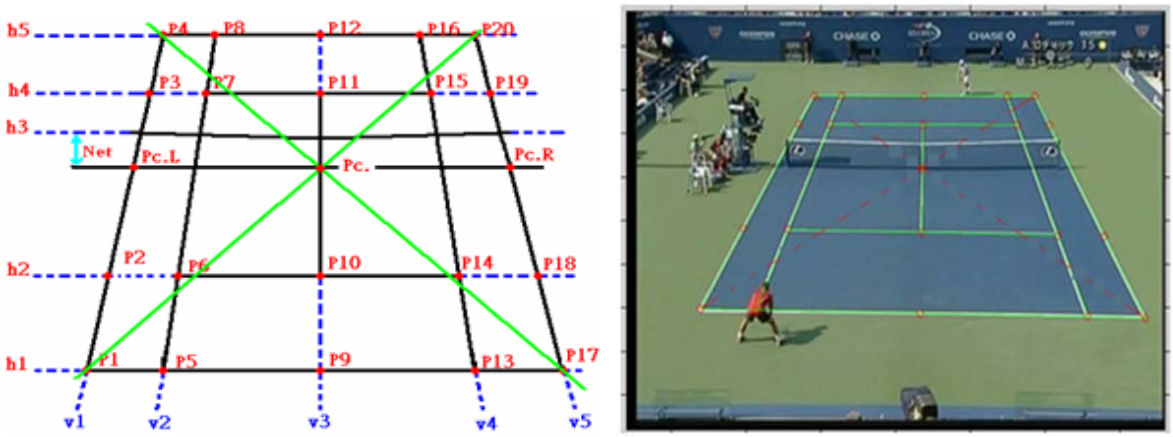
Abstract. *Player tracking is an essential technique in content analysis of tennis videos. A traditional tracking method consists of defining search window, extracting player's features, and utilizing the features to track in video. Previous methods usually have fixed search windows. Due to the angles of video camera, the court in a tennis video is a trapezoid rather than a rectangle. As a result, the size of the player figure continuously changes while moving from position to the other. Even in the same position, the player body is deformable and very versatile. The fixed search window method not only affects the accuracy of player feature extraction, but also incurs unnecessary calculation. In this paper, we propose an adaptive search window algorithm, which automatically adjusts search window in real-time by referring to player's position and speed. Experimental results demonstrate that the proposed method is effective and efficient for the tracking of players.*
Keywords: Sport video analysis, Event detection, Player tracking

1. **Introduction.** Detecting events in broadcast sport videos is one of most active research topics in video semantics analysis. The reason is that a sport game contains well-defined and limited events, and thus analyzing contents of sports is more feasible than other videos, e.g., movies and TV programs. Sports video analysis has wide applications such as video retrieval system, automatic summarization system, to name a few.

The most challenge part of processing video semantics lies in bridging semantic gap. The semantic gap represents the difference between extracted information and user expectation, and exists in almost every multimedia system. Every sport game has specific rules and courts, which is also known as specific features or domain knowledge. In addition, a sport game video contains lots of repeating patterns. By using those repeating patterns

and domain knowledge, it becomes practical to bridge semantic gap. As a result, extraction of the repeating patterns is very important for semantic analysis. Segmentation of objects from video is the first step to retrieve repeating patterns. After segmenting target objects, the temporal and spatial relationships between objects need to be further derived by tracking the extracted objects. Those relationships are employed for recognizing high level semantics, e.g., serving and volleying in tennis, or shooting and passing in football. Moreover, researchers can further analyze a player's style or a team strategies. Because of the reasons mentioned above, tracking objects in sport video has become a popular and valuable research topic.

Many segmentation or tracking algorithms for general videos have been presented in the literature [1]. In recent years, many researchers have devoted to study sport videos, and numerous algorithms were proposed. With regard to tennis sport, C.-M. Kuo et al. [2] presented a playfield segmentation method for baseball videos based on a GMM. The adaptive GMM model is constructed by a novel training pixel-selection scheme, which automatically selects the appropriate pixel(s) from input video for parameter estimation in the expectation-maximization (EM) process. G. Sudhir et al. [3] proposed a method that uses court color to extract player segments from a video, then searches player location by movement estimation, and finally utilizes player trajectories to recognize events like serving, volleying and stroking. C. Calvo et al. [4] developed an automatic annotation system of tennis video. The system also utilizes court color to eliminate non-related regions and keep the lower part of the court. Hough transform is applied to detect court lines, and edge detection is used to find strong edges. Those edges that have different directions from court lines are edges of players, which can be used to track players as well as detect events. N. Rea et al. [5] proposed to use Radon transform to detect court lines, particle filter to track players, and Hidden Markov Model (HMM) to establish the connection between player trajectories and game events. J. Han et al. [6,7] took a similar approach as the methods in [2], but changed the coordinate system from image domain to real-world domain. Object tracking is conducted in real-world domain, and therefore increases the tracking accuracy. In terms of soccer games, the system developed by A. Ekin and A. M. Tekalp et al. [8] first identifies scenes of long-range, medium-range, and close-up shot, and utilizes additional information of penalty box and production features to recognize events. The proposed system is able to extract soccer events as scoring, penalty, and judging events. O. Utsumi et al. [9] designed an object detection system based on color. The system utilizes green color to detect court, and finds players by sport shirt color and edge information. The algorithm proposed by L. Xie and S.-F. Chang et al. [10] uses low-level features including color and motion intensity, and HMM (Hidden Markov Model) to separate play period from break period in a sport game video. C. J. Needham et al. [11] applied silhouette matching to detect players, and employed particle filter to track multiple players simultaneously. With regard to tennis video analysis, tracking of a player is the fundamental step of the whole process. To track a player, we need to define a search window initially, and then extract player features from the window. In previous methods, the search windows are usually set to be fixed during the tracking process. Due to the angle of video camera, in image domain, the shape of tennis court is trapezoid rather than rectangular in real world domain. As a result, the player figure size constantly changes from one position to another. In addition, the player body is deformable during the fierce competition. So even in the same position of a court, the player figure size is still changing. Using a fixed search window not only decreases tracking performance of players, but also increases non-necessary calculations. In this paper, we propose an adaptive search window algorithm, which can automatically adjust window size according to player's location and maximal moving speed. Experimental



(a) Court lines model

(b) A frame image overlapped with calculated court lines

FIGURE 1. Court line model and calculated court lines

results demonstrate the proposed method is effective and efficient for the tracking of players.

2. Adaptive Search Window.

2.1. **Initial search window.** The process of player tracking consists of two phases: detection and tracking. Detection phase locates player position at the first frame of an input video. Tracking phase extracts player figure for the subsequent frames using information of previous frames. To detect the player in the first frame, we should define an initial search region. To define an initial search range of players, we need to retrieve reliable and essential court information. The important court information contains: court line intersections and projection of a net on the court. Figure 1(a) is the court lines model including all lines and intersections. The model defines 5 horizontal lines (h1-h5), 5 vertical lines (v1-v5), middle net line, and 21 intersections (P1-P20 and Pc.). By making use of the court line detection technique developed by the authors [12], we can calculate all the positions and intersections of court lines in a frame. Figure 1(b) shows the calculation results (solid lines) overlapped with original image.

With the court lines model, we can define the initial search windows for tennis players. The search windows for upper and lower fields are listed as follows:

- Initial Search Window for Upper Court
  - Horizontal range (Width)

$$\begin{aligned} \text{Left} &: X_{P4} - \frac{1}{2} * (X_{P4} - X_{P1}) \text{ or } (Zero) \\ \text{Right} &: X_{P20} + \frac{1}{2} * (X_{P17} - X_{P20}) \text{ or } (ImageWidth) \end{aligned}$$

  - Vertical range (Height)

$$\begin{aligned} \text{Top} &: \max(Y_{P4}, Y_{P20}) - \frac{2}{3} * \max(Y_{P4}, Y_{P20}) \text{ or } (Zero) \\ \text{Bottom} &: Y_{Pc} \end{aligned}$$

- Initial Search Window for Lower Court

– Horizontal range (Width)

$$\text{Left} \;:\; X_{P1} - \frac{2}{3} * X_{P1} \text{ or } (Zero)$$

$$\text{Right} \;:\; X_{P17} + \frac{2}{3} * (ImageWidth - X_{P17}) \text{ or } (ImageWidth)$$

– Vertical range (Height)

$$\text{Top} \;:\; Y_{Pc}$$

$$\text{Bottom} \;:\; \max(Y_{P1}, Y_{P17}) - \frac{2}{3} * (ImageHeight - \max(Y_{P1}, Y_{P17})) \text{ or } (ImageHeight)$$

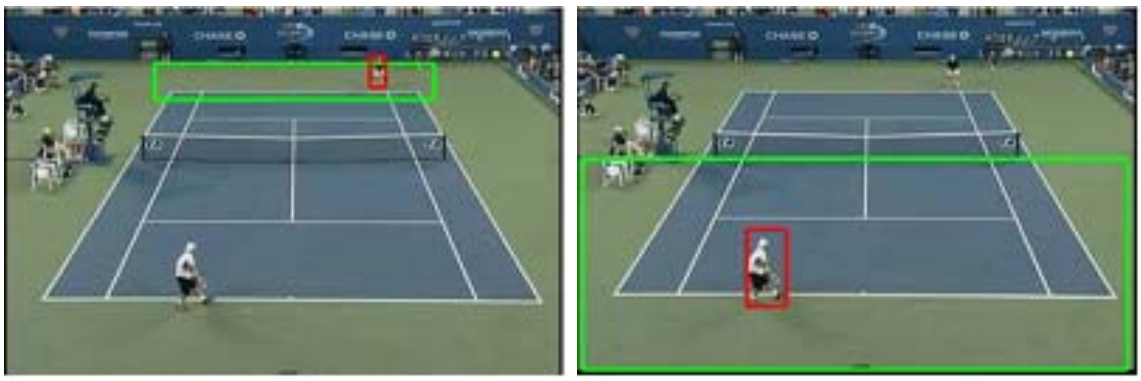
The initial search windows of upper and lower courts are illustrated in Figure 2.

FIGURE 2. Initial search windows for upper and lower courts

After defining the search windows, we can start to extract players. There are many detection methods, and we employ the algorithm introduced by authors in [13]. In [13], a non-dominant color extraction and edge detection are combined to extract the player figure. The non-dominant extraction scheme removes non-player pixels using adaptive thresholds in HSV color space. The adaptive thresholds can accommodate the changes of colors with different courts and different lighting. During the fierce competition of a game, players perform various actions, such as swing or serve, which may cause false detection between player body and background. In order to enhance detection reliability, edge detection is utilized to compensate non-dominant color detection. The well-designed combination method is capable of producing a correct and complete player body, as shown in Figure 3. Using a minimal bounding box to enclose the player body generates a player window, which is a rectangular shape, as the small box shown in Figure 4.

2.2. **Player tracking.** After the player detection, we use a player window with size of δW$*\delta$H to enclose the detected player. The next step is to estimate the player's position for the subsequent frames, which is the so-called tracking phase. We define a search window with size W $\times$ H centered at the bounding box, as shown in Figure 5. Within

FIGURE 3. Extraction of player figure



(a) Detect upper player (b) Detect lower player

FIGURE 4. Initial search area (big box) and the result of player detection (small box) (a) for upper player, and (b) for lower player

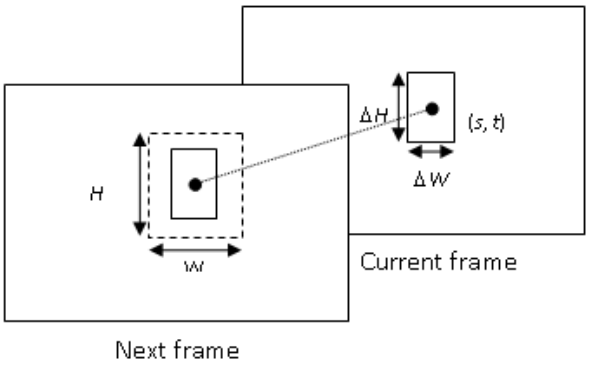


FIGURE 5. Search window centered at the player window

the search window, a full search scheme slides the player window to find a new position which generates maximal object area for each frame, as

$$Player's\ position = \arg\max_{s,t}(area_{bit-1}(B_{s,t}))$$

where $area_{bit-1}()$ denotes the total number of $bit-1$ pixels with value 1 (player pixels) in the binary image B obtained in the detection phase.

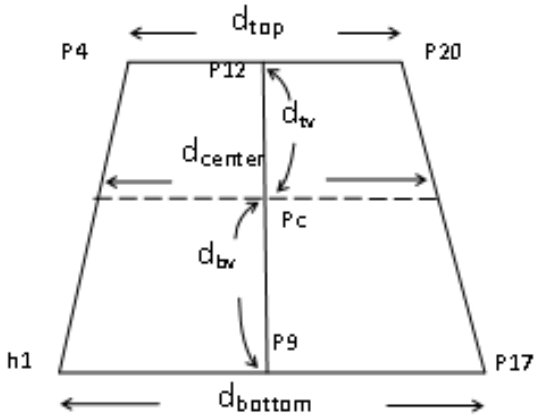


FIGURE 6. Pixel distances of the court

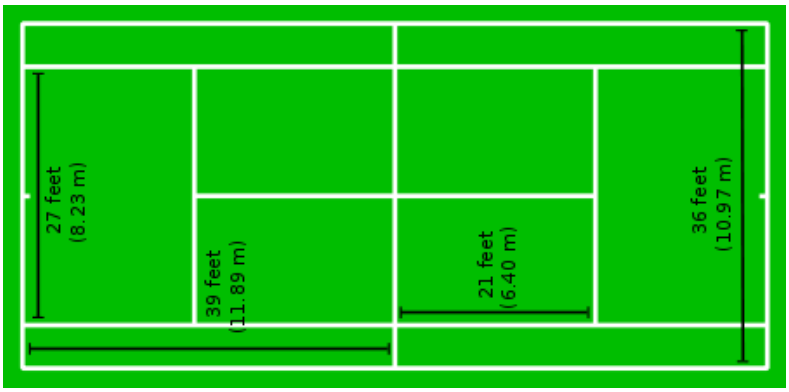


FIGURE 7. Real tennis court model

2.3. **Adaptive search window algorithm.** The most popular video recording format is 30 frames per second. The distance that a player can move within 1/30 second is very limited. Hence, the next location of the player can be estimated by his previous location. The movement of a player can be classified into vertical, horizontal, and diagonal movement. The adaptive search window can be defined as the maximal possible range that a player can reach with 1/30 second. Under the same moving distance, the vertical and horizontal movements will generate the maximal moving range. Therefore, the diagonal movement can be ignored. The maximal moving range is defined as pixels, so any distance mentioned in the following is defined by pixel distance.

Pixel distance definitions: Referring to the court lines model in Figure 1(a), we define pixel distance of each court line as below: Lower court horizontal distance $\overline{P_1P_{17}}$ is d_{bottom} pixel; lower net horizontal distance $\overline{P_{cL}P_{cR}}$ is d_{center} pixel; upper court horizontal distance $\overline{P_4P_{20}}$ is d_{top} pixel; lower court vertical distance $\overline{P_9P_c}$ is d_{bv} pixel; upper court horizontal distance $\overline{P_cP_{12}}$ is d_{tv} pixel. The definitions are illustrated in Figure 6.

Real distance definitions: Referring to the real court model in Figure 7 [14], we can get the real dimension of a tennis court. The width of the court is 36 feet (10.97 m), and height of half court is 39 feet (11.89 m). The authors in [15] mentioned that the maximal moving speed of a player is around 2-7 m/s. In other words, the maximal moving distance in one frame is 2/30 -7/30 m (in 1/30 second). This is real moving speed. By combining the information of pixel distance, real distance and real moving speed, we can calculate the maximal moving distance of a player as the maximal number of moving pixels for

each frame below:

$$\begin{aligned} upper\ court\ horizontal\ movement &= (7/30)/(10.97/d_{top})\ pixels \\ lower\ net\ horizontal\ movement &= (7/30)/(10.97/d_{center})\ pixels \\ lower\ court\ horizontal\ movement &= (7/30)/(10.97/d_{bottom})\ pixels \\ upper\ court\ vertical\ movement &= (7/30)/(11.89/d_{tv})\ pixels \\ lower\ court\ vertical\ movement &= (7/30)/(11.89/d_{bv})\ pixels \end{aligned} \tag{1}$$

The values of the five parameters d_{bottom}, d_{top}, d_{center}, d_{tv} and d_{bv} are all different. The maximal moving distance is smaller as the player location is closer to upper court. Therefore, we should generate different sizes of search windows according to different player locations. Parameters d_{bottom}, d_{top} and d_{center} affect the width of the search window, while d_{tv} and d_{bv} affect the height. The adaptive search window is obtained by using those parameters to re-calculate search window in every frame. As shown in Figure 1(a), h1 and h5 are parallel (or close to parallel), and hence h1 and h5 have the same slope m with different interception b ($y = mx + b$). Using the characteristics of linear equation we can define the variation range of b value, as shown in Equation (2). The variation of distance can further be calculated by Equation (3). Finally, we can calculate any location of the player by solving parameter d and then obtain the search region. The equations are defined in (4) and (5).

The variation ranges of upper court and lower court are (R_{top} and R_{bottom}):

$$\begin{aligned} R_{top} &= |b_{center} - b_{top}| \\ R_{bottom} &= |b_{bottom} - b_{center}| \end{aligned} \tag{2}$$

where b_{bottom}, b_{top} and b_{center} represent the b values of $\overline{P_1P_{17}}$, $\overline{P_4P_{20}}$ and $\overline{P_{cL}P_{cR}}$, respectively.

Horizontal movement distance variation (ρ_{top} and ρ_{bottom})

$$\begin{aligned} \rho_{top} &= \frac{d_{center} - d_{top}}{R_{top}} \\ \rho_{bottom} &= \frac{d_{bottom} - d_{center}}{R_{bottom}} \end{aligned} \tag{3}$$

During the tracking process, the previous search window of player has already been known. Assume that the location in upper court is (x_{tp}, y_{tp}), and in lower court is (x_{bp}, y_{bp}). Drawing a line using player's coordinate which is parallel to h1 and h5, we can calculate the b value as:

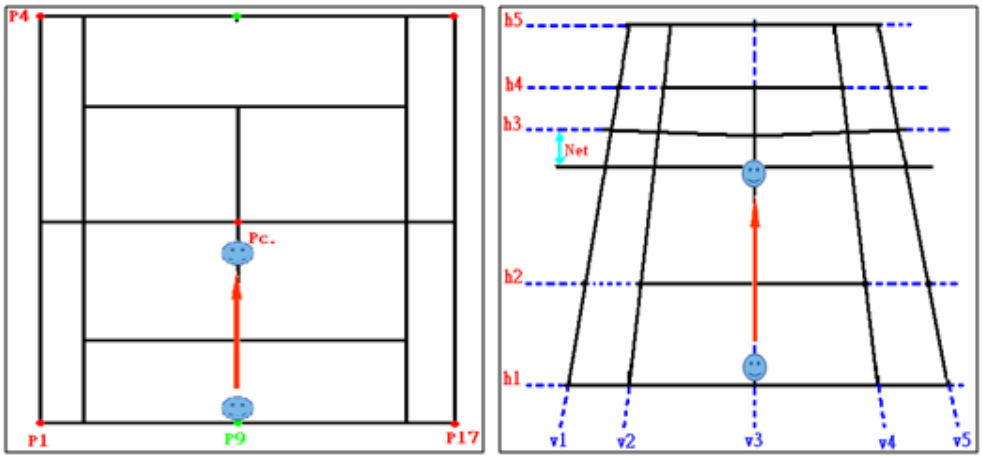
$$\begin{aligned} b_{tp} &= y_{tp} - m_{h\_avg} * x_{tp} \\ b_{bp} &= y_{bp} - m_{h\_avg} * x_{bp} \end{aligned} \tag{4}$$

where m_{h_avg} represents the average slope of h1and h5.

By using Equations (2) and (3), we can calculate the maximal horizontal movements of players in upper and lower courts (d_{tp} and d_{bp}):

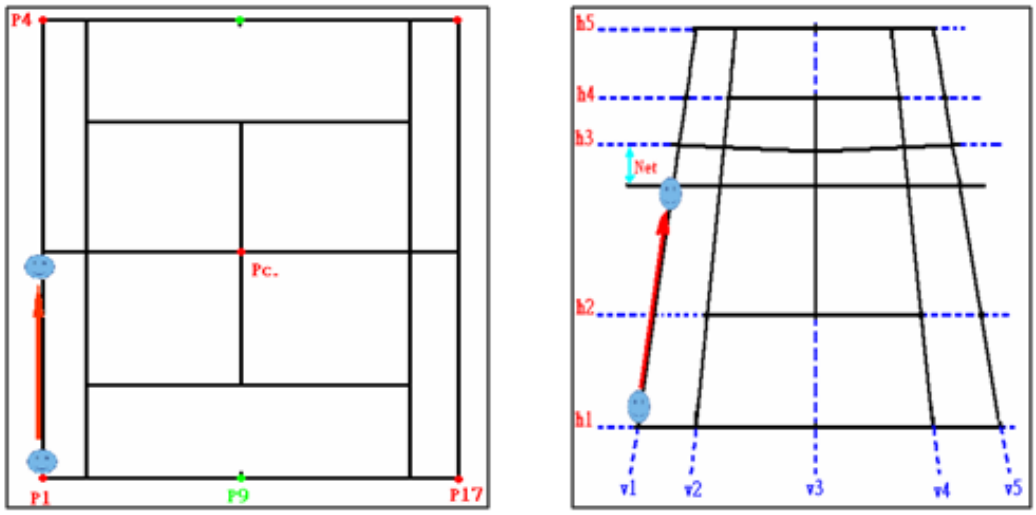
$$\begin{aligned} d_{tp} &= d_{top} + |b_{tp} - b_{top}| * \rho_{top} \\ d_{bp} &= d_{bottom} + |b_{bp} - b_{bottom}| * \rho_{bottom} \end{aligned} \tag{5}$$

Finally, we can get the maximal horizontal moving range of a player: maximal horizontal moving distance of upper court is $(7/30)/(11.89/d_{tp})$ pixel; maximal horizontal moving distance of lower court is $(7/30)/(11.89/d_{bp})$ pixel. After obtaining horizontal range, we need to calculate the limitation of vertical movement. The estimation of height parameters



(a) Trajectory in real domain (b) Trajectory in image domain

FIGURE 8. Player moving vertically near the serving line



(a) Trajectory in real domain (b) Trajectory in image domain

FIGURE 9. Player moving vertically in the left side of the court

consider the largest possible vertical distance a player can make, which happens when the player is moving along serving line: $\overline{P_cP_{12}}$ or $\overline{P_9P_c}$.

Because we only consider vertical and horizontal movements, the height of the window is only affected by vertical movement. Therefore, we classify vertical movements into two scenarios (real domain and image domain), and explain the reason that taking half court distance on serving line as parameter: (1) When moving vertically in real world, if the player is moving along the serving line, the vertical pixel distance is equal to the vertical distance pixel along the court side, as shown in Figure 8 and Figure 9. (2) When moving vertically in image domain, moving along the serving line has the same distance as in real world. However, due to different camera angle, the moving distance in image domain is larger than in real world. In other words, the vertical moving range is largest while moving along the serving line of the court. By considering both scenario (1) and (2), we can define the best vertical moving range as: maximal vertical moving range of upper court is $(7/30)/(11.89/d_{tv})$ pixels; maximal vertical moving range of lower court is $(7/30)/(11.89/d_{bv})$ pixels.

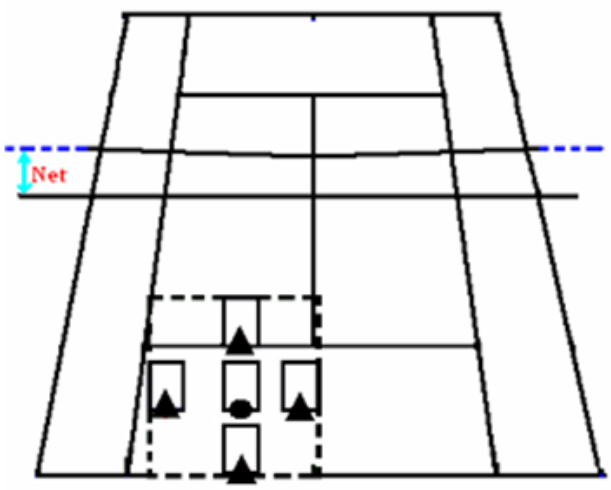
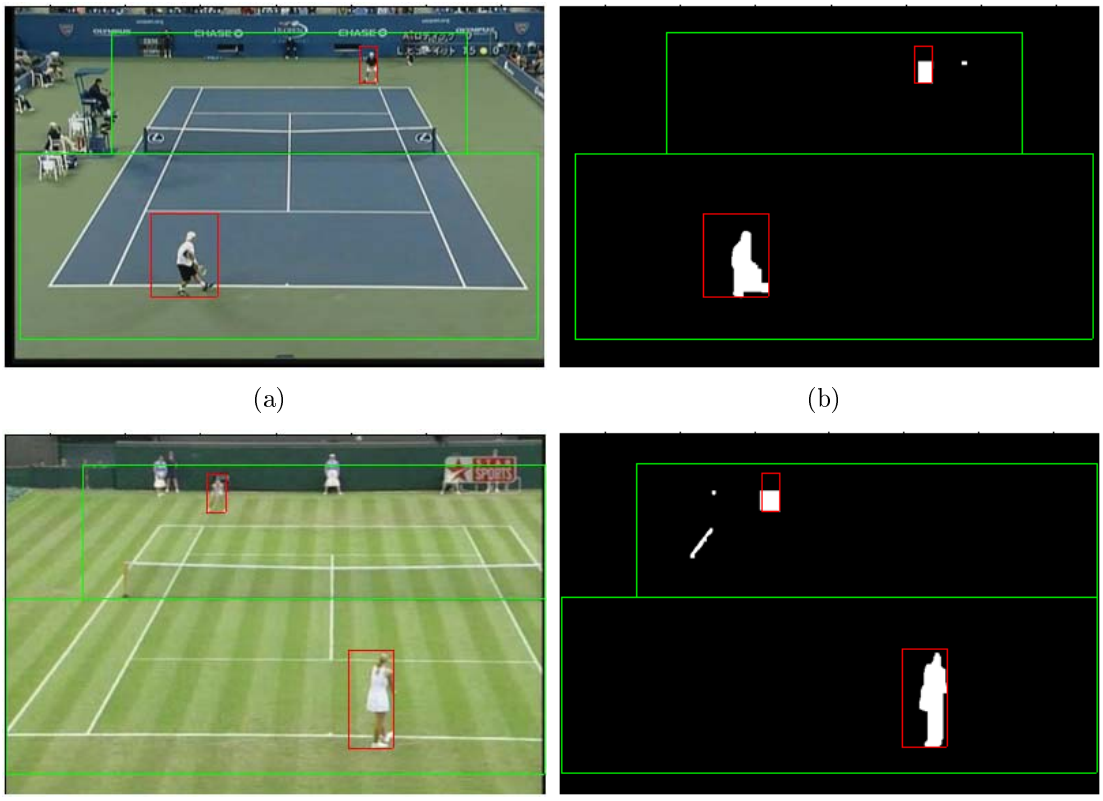


FIGURE 10. Illustration of the generation of adaptive search window

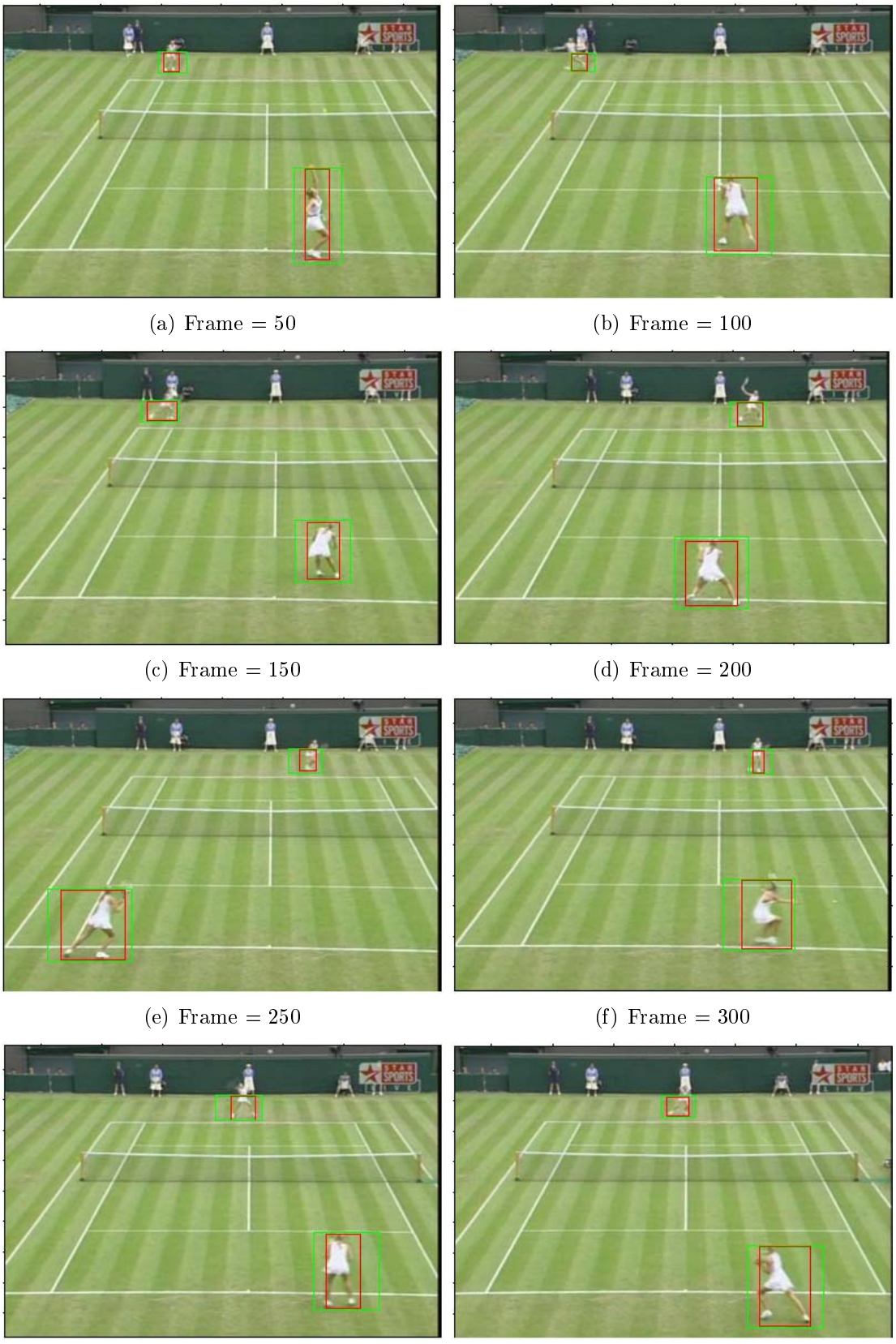


(a) (b)

(c) (d)

FIGURE 11. Initial search window and player window

As mentioned above, we can calculate maximal horizontal and vertical moving ranges of a player. As shown in Figure 10, the middle black dot is the player position, which corresponds to the feet position of the player, and the four triangle marks denote maximal moving range of the player. Each rectangular mark represents a player window. Thus, a search window is constructed from the four player windows estimated (region in dashed-lines).



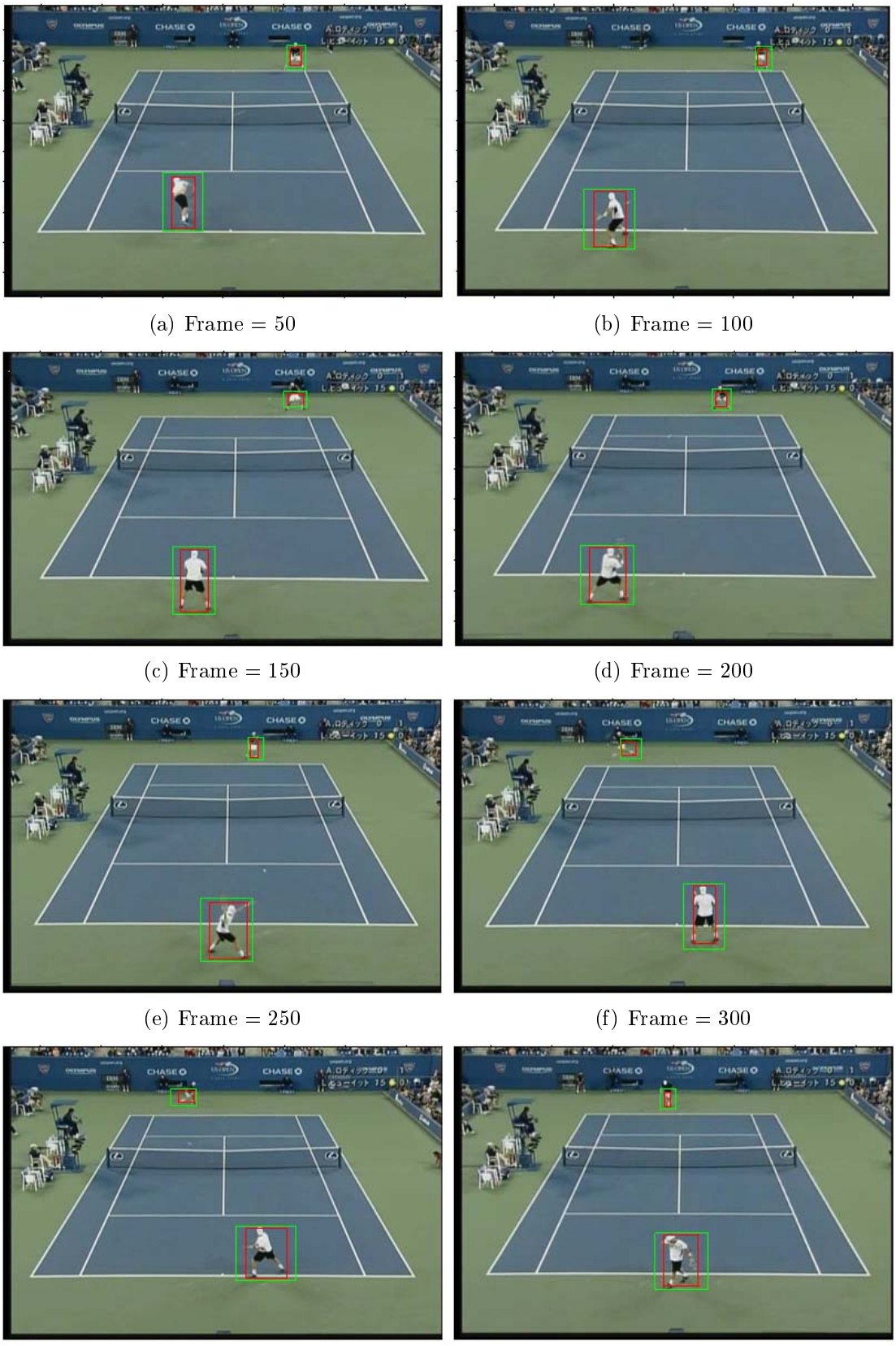
(a) Frame = 50 (b) Frame = 100

(c) Frame = 150 (d) Frame = 200

(e) Frame = 250 (f) Frame = 300

(g) Frame = 350 (h) Frame = 400

FIGURE 12. Adaptive search window (big box) and updated player window (small box)



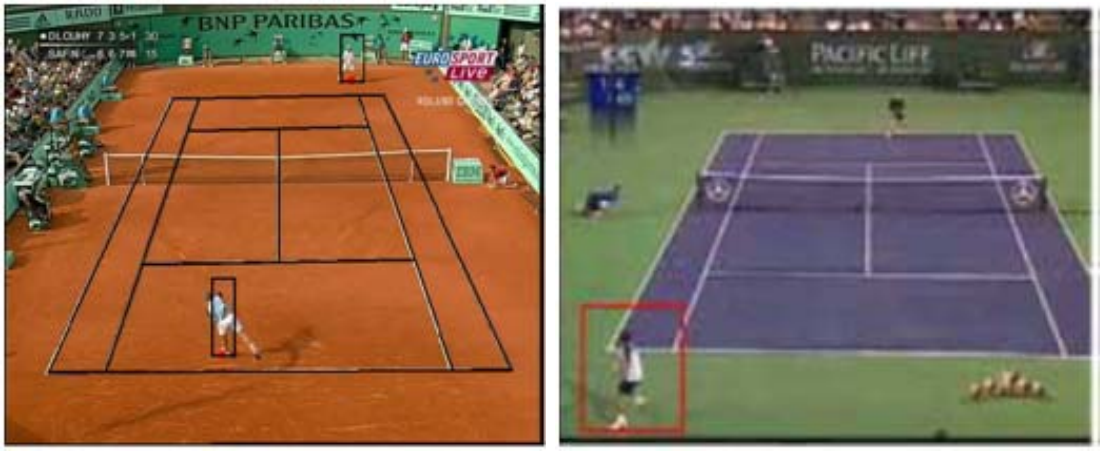
(a) Frame = 50 (b) Frame = 100

(c) Frame = 150 (d) Frame = 200

(e) Frame = 250 (f) Frame = 300

(g) Frame = 350 (h) Frame = 400

FIGURE 13. Adaptive search window (big box) and updated player window (small box)



(a) (b)

FIGURE 14. Other fixed search windows in [7]: (a) window is too small, and (b) window is too large

3. **Experimental Results.** The broadcast videos of 2006 US Open and 2006 Wimbledon are chosen as test dataset. The video format is encoded by MPEG-2, 480*720, 30 FPS. The test program is written by MATLAB. As shown in Figure 11, the initial search window (big box) and player window (small box) are both fixed size. Figures 12(a) and 12(b) demonstrate the adaptive search window (big box) and updated player window (small box) under tracking. The updated player window is the minimum region that contains whole player body. To find the updated player window, we can use court lines information to remove noises with fixed locations, e.g., court lines or logos, and utilize closing operation in morphology to extract complete player body. The extracted results are shown in right part of Figure 11.

The adaptive search windows are illustrated in Figure 12 and Figure 13. Obviously, the search window constantly changes with the player body. The more the player stretches, the larger the search window. Moreover, the size of the search window also varies with player positions. The search window of the player in upper court is smaller than the one in lower court. The results indicate that the size of the proposed search window changes over time according to the player size. Therefore, during tracking, our method not only reduces the interference of noises, but also increases the computational efficiency. Figure 14 shows the fixed search windows of the conventional algorithms [7]. It can be seen that the fixed size windows are either too small or too large for specific frames. However, our algorithm exempts from these problems.

4. **Conclusions.** In this paper, we propose a new adaptive search window algorithm for tracking players in broadcast tennis videos to overcome the drawbacks of the conventional fixed search window schemes. The maximal possibility range of player motion is computed by using his/her previous location and maximal speed. The experimental results show that the proposed method successfully detects players in a large amount of test videos. The search window size automatically adapts to the position and posture of the player. Although in a few frames, the player limbs are not completely contained in the search window. The errors are caused by that the player detection method used in this paper, which could miss some parts of player body under certain circumstances. The proposed method is very robust and effective when the player body is correctly segmented.

Acknowledgment. This work was supported in part by National Science Counsel Granted NSC 99-2221-E-130-011-MY3.

REFERENCES

[1] D. Zhong and S.-F. Chang, Long-term moving object segmentation and tracking using spatiotemporal consistency, *IEEE International Conference on Image Processing*, vol.2, pp.57-60, 2001.

[2] C.-M. Kuo, M.-H. Hung and C.-H. Hsieh, Baseball playfield segmentation using adaptive Gaussian mixture models, *The 3rd International Conference on Innovative Computing, Information and Control*, pp.360-360, 2008.

[3] G. Sudhir, C. Lee and K. Jain, Automatic classification of tennis video for high-level content-based retrieval, *Proc. of IEEE International Workshop on Content Based Access of Image and Video Databases*, pp.81-90, 1998.

[4] C. Calvo, A. Micarelli and E. Sangineto, Automatic annotation of tennis video sequences, *Proc. of the 24th DAGM Symposium on Pattern Recognition*, London, UK, pp.540-547, 2002.

[5] N. Rea, R. Dahyot and A. Kokaram, Classification and representation of semantic content in broadcast tennis videos, *Proc. of IEEE Int. Conf. Image. Processing*, Genoa, Italy, pp.1204-1207, 2005.

[6] J. Han, D. Farin, P. H. N. de With and W. Lao, An automatic analyzer for sports video databases using visual cues and real-world modeling, *Proc. of the IEEE International Conf. on Consumer Electronics*, 2006.

[7] J. Han, D. Farin, P. H. N. de With and W. Lao, Multi-level analysis of sports video sequences, *SPIE Conference on Multimedia Content Analysis, Management, and Retrieval*, San Jose, USA, vol.1, 2006.

[8] A. Ekin, A. M. Tekalp and R. Mehrotra, Automatic soccer video analysis and summarization, *IEEE Trans. on Image Processing*, vol.12, no.7, pp.796-807, 2003.

[9] O. Utsumi, K. Miura, I. Ide, S. Sakai and H. Tanaka, An object detection method for describing soccer games from video, *Proc. of IEEE ICME*, Lausanne, Switzerland, 2002.

[10] L. Xie, P. Xu, S.-F. Chang, A. Divakaran and H. Sun, Structure analysis of soccer video with domain knowledge and hidden Markov models, *Pattern Recognition Letters*, pp.767-775, 2004.

[11] C. J. Needham and R. D. Boyle, Tracking multiple sports players through occlusion, congestion and scale, *The 12th British Machine Vision Conference*, Manchester, UK, pp.93-102, 2001.

[12] Y. C. Jiang, C. H. Hsieh, C. M. Kuo and M. H. Hung, Court line detection and reconstruction for broadcast tennis videos, *The 21th IPPR Conference on Computer Vision, Graphics and Image Processing*, 2008.

[13] Y. C. Jiang, K. T. Lai, C. H. Hsieh and M. F. Lai, Player detection and tracking for broadcast tennis video, *Pacific-Rim Symposium on Image and Video Technology 2009*, Tokyo, Japan, 2009.

[14] http://upload.wikimedia.org/wikipedia/commons/thumb/f/f7/Tennis_court_imperial.svg/220px-Tennis_court_imperial.svg.png.

[15] J. Han and P. H. N. de With, 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.

[16] N. Rea, R. Dahyot and A. Kokaram, Classification and representation of semantic content in broadcast tennis videos, *IEEE International Conference on Image Processing*, vol.3, pp.1204-1207, 2005.

[17] Z. Zivkovic, M. Petkovic and R. van Mierlo, Two video analysis applications using foreground background segmentation, *Proc. of the VIE-2003 Conference on Visual Information Engineering*, Surrey, Guildford, pp.310-313, 2003.

[18] H. Miyamori and S.-I. Iisaku, Video annotation for content-based retrieval using human behavior analysis and domain knowledge, *Proc. of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp.320-325, 2000.

[19] *http://www.kshs.kh.edu.tw/student/page/tennis/rule.htm.*