A Real Time Player Tracking System for Broadcast Tennis Video

Bao Dang[1], An Tran[1], Tien Dinh[1], and Thang Dinh[2]

[1] Faculty of Information Technology, University of Science, VNU-HCMC, 227 Nguyen Van Cu, Ho Chi Minh City, Vietnam
{dangvinhbao,tranminhan.my}@gmail.com, dbtien@fit.hcmus.edu.vn
[2] Computer Science Department, University of Southern California, USA
thangdin@usc.edu

Abstract. This paper proposes a novel framework for tennis player detection and tracking. The algorithm is built on (1) a powerful court-line pixel detection method utilizing intensity and texture pattern, (2) a fast RANSAC-based line parameter estimation which also determines line extents, and (3) a player segmentation and tracking algorithm exploiting knowledge of tennis court model. The content of the video is then explored at a highly semantic level. The framework was tested extensively on numerous challenging video sequences with various court environments and lighting conditions. The results show the robustness and the promising direction of our algorithm.

Keywords: Object detection, object tracking, sports analysis, feature extraction, camera calibration.

1 Introduction

Automatic analysis of sport videos is an interesting application of content analysis, since it allows people to query information of their specific interest easily. Since each group of users has their own preferences and requirements, applications in sport video analysis are very broad [14]. To analyze a tennis video at a higher semantic level such as activity recognition or content classification, a robust detection and tracking algorithm is required. This paper presents a novel content analysis framework based on an automatic players and court detection and tracking method in tennis domain.

Due to the highly dynamic nature of sport videos, existing state of the art methods for single object tracking [3, 4, 7] cannot be applied effectively. Camera motion, player motion, player-articulated poses and scene changes are just some of the difficulties. Therefore, a successful sport analysis system should exploit game specific information rather than apply a generic solution. In court games such as tennis and soccer, typical and useful information is the uniform court color and white court-lines. Moreover, players should only be in the playing field or close to the surrounding boundary. In [9, 10], Hough transform is adopted to detect court-lines, which are subsequently used for calibration. However, the approach is slow and not robust because of the computation complexity and inaccuracy of Hough transformation. In

N.T. Nguyen, M.T. Le, and J. Świątek (Eds.): ACIIDS 2010, Part II, LNAI 5991, pp. 105–113, 2010.

[13], a court-line detection algorithm is described using straight-line detection method to build up a court model. However, it does not provide good results when the court is partially occluded.

In this paper, we propose a real-time and robust framework for developing a fully automatic tennis player tracking system. The flowchart of our system is presented in Fig. 1. First, a court detector is applied to determine precise position of the court in the current frame. This is done by a simple white pixel detection procedure and a RANSAC-based line detection algorithm which also helps to determine the extent of the lines. Some heuristic pruning techniques are also applied to reduce the number of possible candidates for the actual court position. Second, the positions of players are detected in the image by background subtraction. In this step, court model information is used to exclude distracters. It enhances the robustness of our method on broadcast tennis videos impressively. Finally, the positions of players with respect to the court are obtained by combining the position information of the court, the players, and the court model.

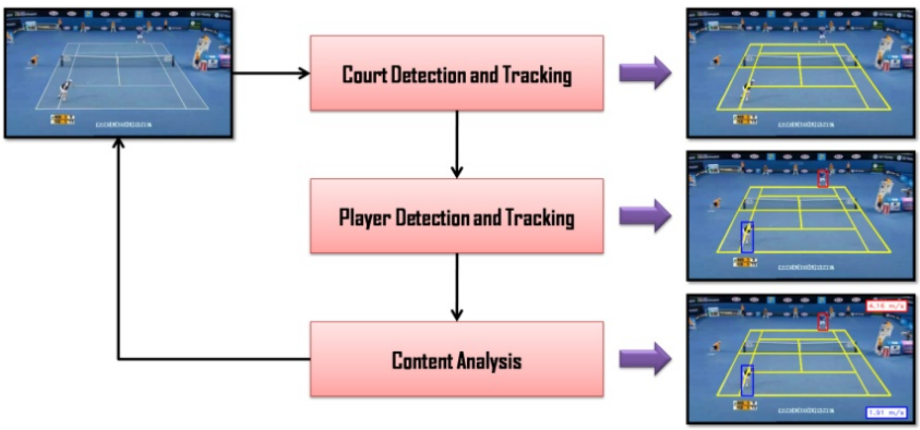


Fig. 1. The overview of proposed system

The rest of this paper is organized as follows. The details of the court detector are presented in Section 2. Player detection and tracking algorithm is described in Section 3. Experimental results on various video sequences and some applications are shown in Section 4, followed by conclusions and future work in Section 5.

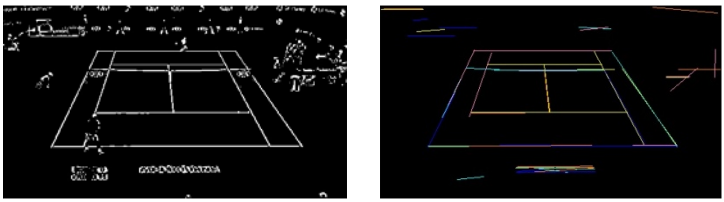
2 Court Detection and Tracking

2.1 White Pixel Extraction

Based on the observation that the color of court-lines is always white, [10] suggested a luminance threshold method to extract white pixels with an additional constraint to exclude large white areas from the detection result. In order to eliminate large areas of white pixels, which usually do not belong to court-lines, darker pixels are checked if they can be found at a horizontal or vertical distance τ from the candidate pixel. In which, τ is the approximate line width. This means white court-line pixels must be enclosed either horizontally or vertically by darker pixels.

2.2 Line Parameter Estimation

From the set of obtained white pixels, court-line candidates will be detected by RANSAC algorithm [6]. The most dominant lines are determined and removed from the dataset in an iterative process, until we get enough line candidates. Dominant lines are defined by the start and the end positions of line segments. Those segments are further linked together to make longer segments if possible and classified as horizontal or vertical lines. These steps will be explained clearly in the following sections.



(a) (b)

Fig. 2. (a) White-pixels image and (b) Detected lines by RANSAC

2.2.1 RANSAC Based Dominant Line Detection

RANSAC is a randomized algorithm that hypothesizes a set of model parameters and evaluates the quality of the parameters. After several hypotheses have been evaluated, the best one is chosen. Specifically, a line is hypothesized by randomly selecting two court-line pixels. Then, the line parameter g is computed. An evaluation score is defined as

$$s(g) = \sum_{(x,y)\in P} \max(\tau - d(g,x,y), 0)\,, \tag{1}$$

where $d(g,x,y)$ is the distance of the pixel (x,y) to line g, P is the set of court-line pixels, and τ is the line width from Section 2.1 which has default value of 4. This score effectively indicates the support of a hypothesized line as the number of white pixels close to the line and also determine two end points of the line. The process is repeated until about 25 hypothesized lines are generated randomly [6]. Finally, the hypothesized line with the highest score is selected, as illustrated in Fig. 2.

2.2.2 Line Segment Refinement and Classification

The RANSAC based line detector produces some neighboring short-line segments that actually belong to one court-line. This misalignment happens because of poor quality of the video or the court itself. To address this issue, we merge two neighboring segments l_1 and l_2 if $(\widehat{l_1, l_2}) < 0.75°$ and $d(l_1, l_2) < 5$. Obtained lines are also classified in advance as horizontal lines or vertical lines based on two end point coordinates. A candidate line having two end points (x_1, y_1) and (x_2, y_2) is considered as a horizontal line if $|y_1 - y_2| < 20$. Furthermore, horizontal lines are sorted from the top to the bottom; whereas, vertical lines are sorted from the left to the right. Both steps aim to reduce the number of possible hypotheses in calibration parameters searching space and hence improve the performance of the algorithm. All above parameters are set empirically according to the resolution of the input video.

2.3 Model Fitting

The model fitting step determines the correspondences between the detected lines and the lines in the standard court model. Once these correspondences are known, we can construct the homography between real-world coordinates and image coordinates. According to [12], an image coordinate (x, y) is related to a court model coordinate (α, β) by a homography matrix H

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{22} & h_{23} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix} = H \times \begin{pmatrix} \alpha \\ \beta \\ 1 \end{pmatrix}. \tag{2}$$

With some manipulations, we can rewrite the relation as

$$\begin{aligned} h_{11}\alpha + h_{12}\beta + h_{13} - h_{31}\alpha x - h_{32}\beta x - h_{33}\mathrm{x} = 0\,, \\ h_{21}\alpha + h_{22}\beta + h_{23} - h_{31}\alpha y - h_{32}\beta y - h_{33}\mathrm{y} = 0\,. \end{aligned} \tag{3}$$

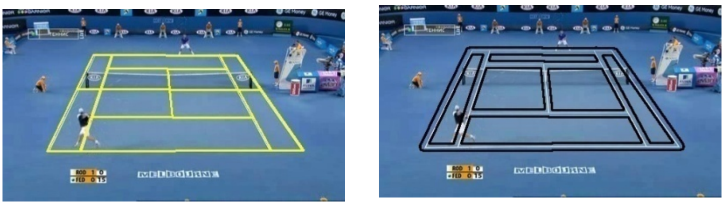
These two linear equations have nine unknown parameters $h_{11} \dots h_{33}$ of the linear mapping matrix H. If dividing each term in the equations by the element h_{33}, we are left with only eight unknown parameters. Matching two candidate lines with two model lines, we acquire a complete system of linear equations, which can be solved by the Gaussian elimination method.

In order to find the best homography that fits the dataset, [10] proposed a matching score approach. After all transformation matrices have been evaluated, the matrix with the highest score is selected as the ultimate solution. In this paper, the Extreme Fast Line Algorithm [1] is used for the scoring step. Although the computation for each calibration setting can be done fast, the evaluation of the model support is computationally intensive. Therefore, some simple and quick tests to reject physically impossible calibration parameters are applied.

In addition, another evaluation step is proposed to confirm the calibration settings. Let (μ_R, μ_G, μ_B) and $(\delta_R, \delta_G, \delta_B)$ be defined as the means and the standard deviations of red, green and blue channel of non-white pixel set inside the court, respectively. Due to the uniform of the court color, calibration parameters are invalid if $\max(\delta_R, \delta_G, \delta_B) > \theta$, where θ is a threshold. In other words, the detection result is unreasonable when the color inside the court varies too much.

2.4 Model Tracking

The previous detection algorithm only has to be applied once in the bootstrapping process at the first frame of every new playing shot. Given that the change in camera speed is small, the camera parameters of subsequent frames are predictable. Similar to [9], the detected court-lines in the previous frame are extended to obtain a search region for the current frame, shown in Fig. 3. The same detection technique is executed, but now within the local search area, to track the model. The obvious benefits are that the court-lines are estimated more accurately and that the calibration setting can be determined rapidly by matching each court-line with the closest predicted model line. Our court detection and tracking method has been experimented in different types of court under various lighting conditions. In case of court occlusion or camera in motion, only two precise court-lines are required to reconstruct the court successfully.



(a) (b)

Fig. 3. (a) Previous court detection result and (b) Local search area for the current frame

3 Player Detection and Tracking

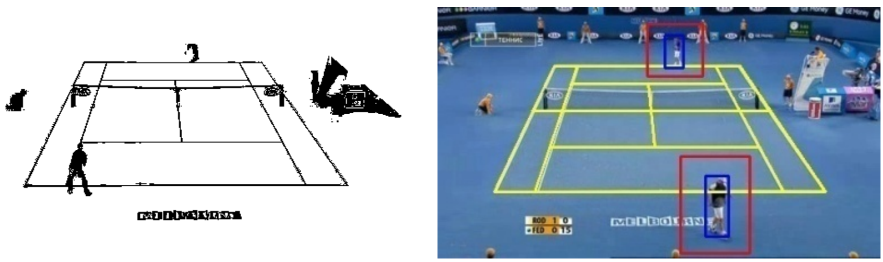
This section explains the process to obtain the real-world positions of the players, starting with the pixel-level detection, up to the final trajectory computation at the object-level. In general, the moving area of a player is limited to the playing field inside the court and partially the surrounding area. Moreover, the color of the playing field and the surrounding area is almost uniform. These features allow us to separately construct background models for the field inside the court and the surrounding area, instead of creating a complete background for the whole image [9]. This approach has two advantages. First, the background image cannot be influenced by any camera motions. Second, only color and spatial information are considered when constructing the background models, which avoids complex motion estimation.

The RGB color space is used for modeling the background. The stored background model for the playing field inside the court is $[\mu_R, \mu_G, \mu_B, \delta_R, \delta_G, \delta_B]$, defined in Section 2.3. A pixel inside the court is marked as foreground if

$$|r - \mu_R| > \alpha\, \delta_R \ \vee \ |g - \mu_G| > \alpha\, \delta_G \ \vee \ |b - \mu_B| > \alpha\, \delta_B\,, \tag{4}$$

where α is a constant, depending on the color variance of the court, and r, g, b denote the red, green, blue values of the current pixel, respectively. The same process is applied to all pixels in the area surrounding the court. This step produces a foreground image as shown in Fig. 4a. Here, the background model is assumed to be persistent during a playing shot; therefore, only the first frame background is modeled.

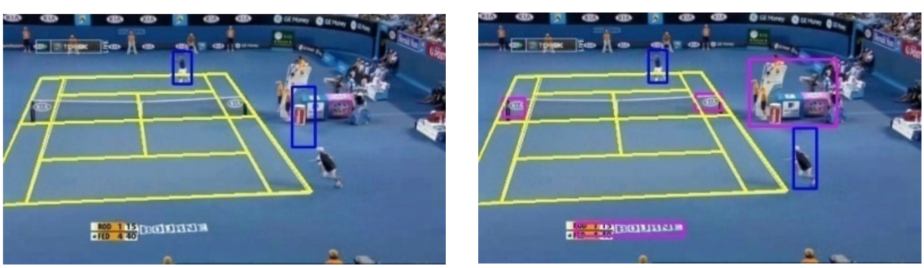
Exploiting the knowledge of the game court, two searching windows in the foreground image, above and below the tennis net line are used. By assuming that the player sizes in the video do not change much, the player boundary for each searching window is defined as a fixed-size rectangle which encloses the maximum foreground pixels (Fig. 4b). The initial windows are fixed for the first frame of every playing shot, whereas the windows in subsequent frames are adjusted, surrounding the previous locations of the players [11]. The reference point of the player's position in the image plane is defined as the lowest vertical point on the boundary rectangle, which is transformed using Eq. (2) to obtain the real-world location of the player.



(a) (b)

Fig. 4. (a) An example of foreground image and (b) Adaptive searching windows

Unfortunately, the player is not the only foreground object in the video frame. As shown in Fig. 5a, other objects, such as referees and ball boys, can affect the tracking result. This issue has not been mentioned in any recent works. Thus, we propose a heuristic method to solve the problem. Obviously, the positions of distracters are usually fixed on the court during a playing shot. We take advantage of this fact to detect and track those objects and enhance the player tracking process, even when the camera is moving. After the players are detected in the first frame, all remaining medium-sized foreground blobs are considered as the distracters. Their image coordinates are then converted to the real-world model by using the calibration parameters. In successive frames, these coordinates are transformed back to the image domain by the new camera setting to locate all of those distracters in order to remove them from the foreground image. The proposed method can significantly eliminate the false detection (Fig. 5b).



(a) (b)

Fig. 5. (a) False player detection problem and (b) Our proposed method

4 Experimental Results and Applications

In this section, experimental results of our system on four video sequences recorded from Grand Slam tournaments [2], which are the most important tennis events of the year, are provided. All sequences have the resolution of 480 by 270 pixels, the frame rate of 25 fps, and the length of 9000-11000 frames. The system is implemented in C++ and runs at 75 fps on a Intel Core 2 Duo 2.0 GHz processor. This efficiency performance is due to the combination of a specialized court-line pixel detector, a RANSAC-based line detector and exploiting the knowledge of the game court. Hence there is room for further processing steps can be applied to extract high level semantic in content analysis applications which required real-time performance.

Table 1. Court detection results

	Playing frames	Detected	Correct	Miss	False
Australian Open	10135	10123	10118 (99%)	17	5
French Open	5026	4985	4970 (98%)	56	15
Wimbledon	7667	7413	7396 (96%)	361	107
US Open	8327	8282	8279 (99%)	48	3

Table 2. Player detection results

	Playing frames	Player 1	Player 2
Australian Open	10135	9127 (90%)	9336 (92%)
French Open	5026	4786 (95%)	4415 (88%)
Wimbledon	7667	5478 (71%)	5622 (73%)
US Open	8327	7863 (94%)	7521 (90%)

The system achieves an impressive 96-99% court detection rate (Table 1). Our algorithm is very robust to occlusion, partial court views, poor lighting, and camera motion as shown in Fig. 7 since it requires only two precise court-lines to track the court model. This result outperforms the algorithms in [5, 13], which only handle full court view frames. The player detection accuracy is about 87% (Table 2), where the criterion is that at least 70% of the body of the player is included in the detection window. Fig. 6 illustrates some visual results, where the top and bottom players are bounded with red boxes and blue boxes, respectively.

According to [8], the tracking results are analyzed to extract higher semantic information, such as:

- **Instant speed of the player:** The speed of each player is definitely an important factor to reveal the current status of a player (running or still). It also indicates the intensity of the match. The speed is estimated based on the court model size and player positions during the shot, as shown in Fig. 6.
- **Speed change of the player:** Acceleration and deceleration of a player occurs during changes in action behavior.

- **Relative position of the player on the court model:** This position is the main source for the recognition of those events that are characterized by a particular arrangement of players on the playing field. **Fig. 7** shows our event detection results of three typical events, which are service, baseline rally and net approach.
- **Temporal relations among each event:** In some sports games like tennis and baseball, there are strong temporal correlations among key events. For example, in a tennis video, service is always at the beginning of a playing event, while the baseline rally may interlace with net approaches.

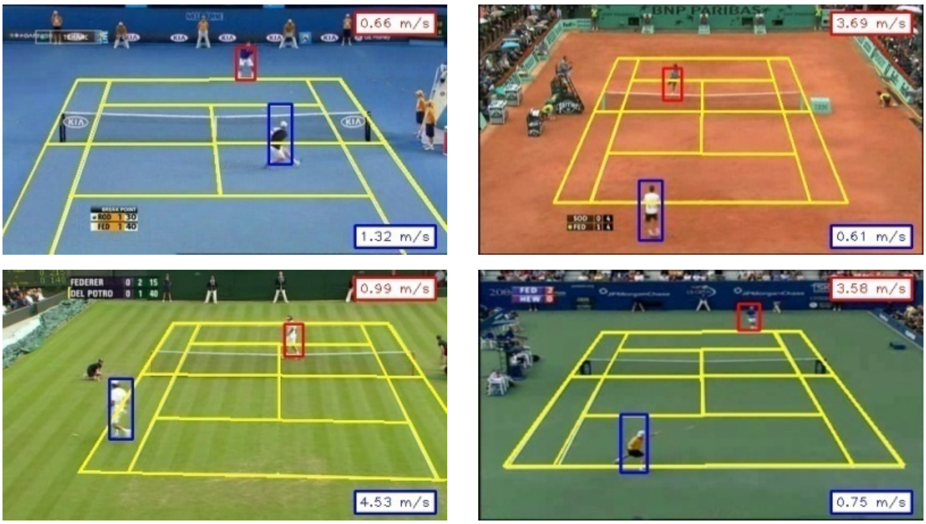


Fig. 6. Visual experimental results (the court is indicated by yellow lines, the red and blue rectangles represent the players) in different court types, lighting conditions, and camera views

(a) (b) (c)

Fig. 7. Some examples of event detection: (a) Service, (b) Baseline Rally and (c) Net Approach

5 Conclusions and Future Work

We have presented a novel framework for tennis sports tracking and content analysis. The algorithm includes the playing field detection using court-line detection and tracking, player segmentation. The system provides a high-level semantic content analysis in tennis video with a limited human-interaction at the beginning. Possible future directions of the system would be upgrading of homography matrix into full camera matrix, including the curved line segments into the court model, or employing the popular HMM model to describing the dynamic of the video events.

Acknowledgements

We would like to acknowledge the support of AI Lab of the University of Science, VNU-HCMC for providing facilities for this research.

This work is a part of the KC.01/06-10 project supported by the Ministry of Science and Technology, 2009-2010.

References

1. Extremely Fast Line Algorithm, `http://www.edepot.com/linee.html`
2. Roger Federer Highlights Video, `http://www.rogerfedererpoints.com`
3. Arulampalam, S., Maskell, S., Gordon, N., Clapp, T.: A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. In: IEEE Transactions on Signal Processing, pp. 174--188 (2002)
4. Avidan, S.: Ensemble Tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, USA (2005)
5. Calvo, C., Micarelli, A., Sangineto, E.: Automatic Annotation of Tennis Video Sequences. In: Proceedings of the 24th DAGM Symposium on Pattern Recognition, pp 540--547. London, UK (2002)
6. Clarke, J., Carlsson, S., Zisserman, A.: Detecting and tracking linear features efficiently. In: Proc 7th British Machine Vision Conference, pp. 415--424. Edinburgh (1996)
7. Collins, R., Liu, Y., Leordeanu, M.: Online Selection of Discriminative Tracking Features. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1631--1643 (2005)
8. Farin, D., Han, J., de With, P.H.N.: Multi-level analysis of sports video sequences. In: SPIE Conference on Multimedia Content Analysis, Management, and Retrieval, pp. 1--12. San Jose, USA (2006)
9. Farin, D., Han, J., de With, P.H.N., Lao, W.: Automatic Tracking Method for Sports Video Analysis. In: 26th Symposium on Information Theory, pp. 309--316. Brussels, Belgium (2005)
10. Farin, D., Krabbe, S., de With, P.H.N., Effelsberg, W.: Robust Camera Calibration for Sport Videos Using Court Models. In: SPIE Storage and Retrieval Methods and Applications for Multimedia, pp. 80--91 (2004)
11. Jiang, Y. C., Lai, K.T., Hsieh, C.H., Lai, M.F.: Player Detection and Tracking in Broadcast Tennis Video. In: Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology, pp. 759--770. Tokyo, Japan (2009)
12. Segarra, D. L.: Tennis Stroke Detection in Video Sequences. In: Master Degree Project. Stockholm, Sweden (2005)
13. Sudhir, G., Lee, C., Jain, K.: Automatic Classification of Tennis Video for High-level Content-based Retrieval. In: Proceedings of International Workshop on Content-Based Access of Image and Video Databases, pp. 81--90. Washington DC, USA (1998)
14. Wang, J. R., Parameswaran, N.: Survey of sports video analysis research issues and applications. In: Workshop on Visual Information Processing, pp. 87--90, Sydney, Australia (2004)