A TRAJECTORY-BASED BALL DETECTION AND TRACKING ALGORITHM IN BROADCAST TENNIS VIDEO

Xinguo Yu[a], Chern-Horng Sim[b], Jenny Ran Wang[c], and Loong Fah Cheong[b]

[a] Institute for Infocomm Research,
21 Heng Mui Keng Terrace,
Singapore 119613
xinguo@i2r.a-star.edu.sg

[b] Electrical and Computer Eng,
National University of Singapore,
Singapore 117543
{chern-horng.sim, eleclf}@nus.edu.sg

[c] School of Computer Science & Eng,
The University of New South Wales,
Sydney 2052, Australia
jennyw@cse.unsw.edu.au

ABSTRACT

Ball locations over frames facilitate tennis video analysis to a great extent. But so far no algorithm is able to obtain satisfactory result in locating the ball in broadcast tennis video (BTV). This paper presents a trajectory-based algorithm to detect and track the ball in BTV. Unlike the object-based algorithm, it does not decide whether an object is the ball. Instead it decides whether a candidate trajectory is a ball trajectory. This algorithm is able to obtain ball locations for most frames in a BTV, making use of four cues, namely, (1) an anti-model method to produce ball candidates from each frame, (2) a trajectory-based scheme to generate, identify, and extend the ball trajectories from a set of candidates, (3) a method to infer the ball locations according to players' locations and the points of hitting, (4) a method to estimate missing ball locations from known ball locations. The experimental results show that our algorithm obtains the ball locations for above 96% frames in a sufficient accuracy for summarization.

Keywords: Broadcast tennis video, Trajectory-based algorithm, Candidate feature image, Trajectory processing, Hitting point.

1. INTRODUCTION

Time is precious to everything, even in the sports area, for we are living in a fast pace era. Viewers want to watch sports video summaries fast; coaches want to find the tactical portion of sports video quickly. These needs stimulate the interest to automatically analyze the sports video. Since tennis is one of the most popular sports, tennis analysis receives much attention. This interest is further motivated by the possible applications over a wide range of topics, such as tactic analysis, indexing, retrieval, and summarization [2-5].

Since the "ball" is the most important object in tennis, detecting and tracking the ball become crucial in tennis video analysis. Unfortunately, so far, no algorithm is able to obtain satisfactory results in locating the ball in broadcast tennis video (BTV) due to the following challenges:

- The ball has serious deformation due to the reflection.
- The ball is very small, especially at the far side of the camera. For some frames, balls are so small that human eyes are unable to see.
- The ball is often occluded by the players and the net. It may mix with audience and complicated background.
- The appearance of the ball varies irregularly over frames. Its size, shape, and velocity change irregularly over frames.

The above-listed challenges make it difficult to build a ball representation to identify the ball in a frame. In other words, like the ball detection and tracking problem in broadcast soccer video (BSV), this problem is object-indistinguishable [8].

This paper proposes a trajectory-based algorithm for ball detection and tracking in BTV. The proposed algorithm can be viewed as another instance of the trajectory-based approach [6-8]. At the same time, the proposed algorithm has its own new elements. Firstly, it uses locations of the players at hitting points (a point when the racket hits the ball) to infer the locations of the ball. Secondly, it uses these hitting points to infer the turning points of the ball route. Last, it infers which player hits the ball from the data of player locations, hitting points, and ball candidate locations.

In this paper, we assume that the video has already been cut into segments showing the tennis court. The proposed algorithm is for locating the ball in each frame of the segment. Our algorithm has five components as depicted in Figure 1: player detection and tracking, candidate detection, hitting point detection, candidate trajectory generation, and trajectory processing. In the *Player Detection and Tracking* component, we locate the locations of far-player and near-player in the game for each frame using mainly the color of players, the far-player (near-player) being the one who is far from (near) the camera. In the *Candidate Detection* component, we produce the ball candidates for each frame by removing the identified non-ball objects. Then we classify the candidates into two categories. In the *Hitting Point Detection* component, we use the algorithm presented in [5] to detect the hitting points in each segment from audio. Then we differentiate hittings into far-player hittings or near-player hittings. In the *Candidate Trajectory Generation* component, we first create the various candidate feature images (CFI). Then we use a Kalman filter-based procedure to produce the candidate trajectories from each CFI. In the *Trajectory Processing* component, we take three steps to obtain the ball trajectories. Firstly, we evaluate each candidate trajectory. Then we identify the ball trajectories through a selection procedure. Finally, we extend the ball trajectories according to the hitting points and player locations.

Related work of locating the ball in sports videos exists. Some papers proposed object-based algorithms to identify the ball [1]. However, these algorithms can only obtain satisfactory results for selected segments of the video. Yu et al [6-9] developed trajectory-based algorithms for locating the ball in BSV. Under encouraging of the good performance of the algorithms in [6-8], this paper designs a trajectory-based ball detection and tracking algorithm for BTV. This algorithm has overcome the unique challenges of the current problem, which are listed in the above.

The rest of the paper is organized as follows: Section 2 presents the proposed algorithm. Section 3 gives the various experimental results. We conclude our paper in Section 4.

2. BALL DETECTION AND TRACKING ALGORITHM

This section explains the components of the proposed algorithm depicted in Figure 1 in turn. The procedures that have similar ones in [6-9] will be given in summaries due to the limited space, whereas the novel procedures will be presented in more details.

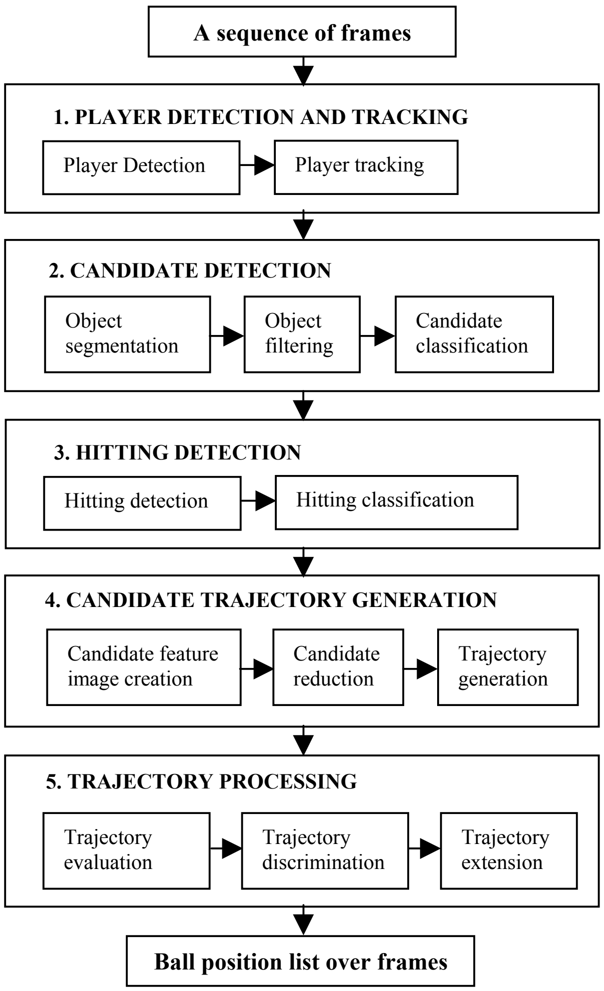

Figure 1. Block diagram of the proposed algorithm.

2.1. Player Detection and Tracking

For each frame of a video sequence, we first find the boundary of the court and remove all long lines and the net. We then use the pixel growing procedure to segment the objects in the frame. Knowing that every frame has two-side players (far-player and near-player), we, therefore, conclude that two/four relatively large objects should be the players. In addition, the criteria below are also considered: (a) the near-player should be at the lower part of the frame and the far-player should be at the upper part of the frame; (b) there should be one or two players on one side of the net; (c) two sides of the net should have the same number of players. Without losing the generality, following discussion will consider single match only, where there are two players in total.

2.2. Ball Candidate Detection

Since many objects look like the ball in a frame, we cannot identify the ball within the frame. We use sieves to remove as many non-ball objects as possible in each frame. Then the remaining objects are considered as the ball candidates of the frame. Candidate trajectories are generated from the candidate feature images, which present all the ball candidates of a given sequence together. A candidate trajectory is a trajectory in which each element is a ball candidate, which also is a ball-trajectory candidate. A procedure is used to identify the ball trajectories from the candidate trajectories based on the confidence index of each candidate trajectory and the knowledge that there is at most one ball in each frame. Below shall present the above steps in turn.

Candidate generation: To produce the ball candidates for each frame, the below sieves are built to remove the non-ball objects.

Ball Size Sieve Θ_1 : We filter out the objects out of the ball size range, which are estimated from the detected net.

Line Sieve Θ_2 **:** We filter out all long lines, including straight lines and curves as the ball cannot be deformed into a long line.

Ball Color Sieve Θ_3 : We filter out objects without ball color pixels.

Shape Sieve Θ_4 : The ball image can have a shape quite different from a circle, but in most frames its width-to-height and height-to-width ratios are less than 2.5 according to the results of our statistical analysis.

Ball Location Sieve Θ_5 : At the instance of hitting, we remove the objects that are far from the hitting player. We will explain how to find the player hitting the ball in Section 2.3.

Each sieve Θ_i is a Boolean function on domain $\mathrm{O}(F) = \{o : o \text{ is an object in the field of frame } F\}$.

$$\Theta_i(o) = \begin{cases} 0 & \text{if sieve } \Theta_i \text{ removes it,} \\ 1 & \text{otherwise.} \end{cases} \quad (1)$$

After sieving, the remaining objects of $\mathrm{O}(F)$ form the ball candidate set $C(F)$ of frame F.

$$\mathrm{C}(F) = \{o\!: o \in \mathrm{O}(F), \quad \Theta_i(o) = 1 \text{ for } i = 1 \text{ to } 5.\}. \quad (2)$$

Candidate classification: Following references [6, 8], we classify the candidates into two categories according to the probability that the candidate is the ball. Let $f_1, f_2, \cdots, f_k$ be all features (such as color, circularity, isolation, etc) used to evaluate the candidates. Let $o \in C(F)$ be a ball candidate of frame F and $P_i(o)$ $(i = 1, 2, \cdots, k)$ be the probability that o is a ball according to the feature f_i . The choice of features is such that they are independent with relatively small error. This assumption is attractive because the probability that o is a ball has a simple formula by the Bayesian rule.

$$P(o) = \prod_{i=1}^{k} P_i(o). \quad (3)$$

According to the probability $P(o)$, the candidates in $C(F)$ can be divided into two categories. Category 1 and 2 contain the objects with high and low probability respectively. In the context of BTV, the features used are the time unit between the considered frame and the frames where hittings occur, the surrounding of the candidate, color, and size.

2.3. Hitting Detection

Hitting detection: The sound emitted by the racket hitting the ball is distinct. In this paper, we use the algorithm by Xu et al [5] to obtain frames where hittings occur.

Hitting classification: The detected hittings are divided into two categories: near-player hittings and far-player hittings, which are hit by the near-player and by the far-player, respectively. This classification is achieved based on two cues: (a) in frames around instances of near-player hittings, we can obtain good and clear ball candidates; (b) the ball must be hit alternatively by the two players in a tennis game.

Ball location inference: The above hitting classification can tell us which player hits the ball at each hitting point. Thus, we can estimate the ball location according to the player location as the ball is in the vicinity of the player when he/she is hitting the ball. The detected hittings will also serve as the start and end points of the ball trajectory. In addition, the hitting can help in removing non-ball objects as the ball must be near the hitting player.

2.4. Candidate Trajectory Generation

Candidate feature image creation: A candidate feature image (CFI) is an image that draws a combination of the candidate numeral features over the frames in a given sequence. The candidate Y-image is created in such a way that the width of the created image is the frame number of the sequence and the height of the created image is the height of the frame. Each candidate draws a pixel in Y-image with the column being the frame serial number in the sequence and the row being the row in the video frame. The candidate XY-image, a 3D image, is created in such a way that the length of the image is the frame number of the sequence, and the width and height of the image are the width and height of the frame in the tennis video. Each candidate draws a pixel (t, x, y) in the candidate XY-image with t being the frame serial number in the sequence and x (y) is the number of the column (row) of the candidate centered in the video frame (see papers [6-8] for more CFIs). Figures 2, 4, and 5 show a sample candidate Y-image in various processing stages, which corresponds to the sequence from frame 36970 to frame 37177.

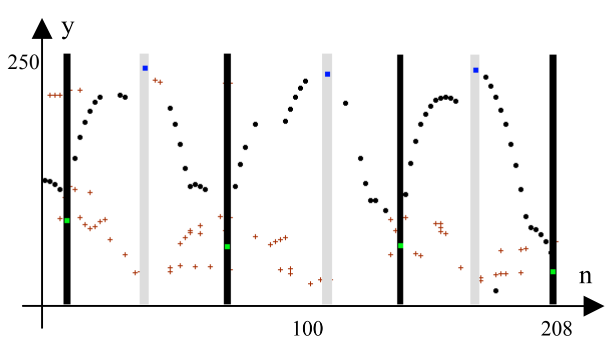

Figure 2. The obtained ball candidates. The black dots and red crosses stand for categories 1 and 2, respectively.

In Figures 2, 4, and 5, black vertical lines and grey vertical lines denote frames where the near-player hittings and the far-player hittings occur respectively. In this paper, we still call the CFI with these lines as CFI.

Trajectory generation: Following references [6, 8], we use a Kalman filter based procedure to generate the candidate trajectories in the various CFIs. However, a unique idea below is used in this paper. Since we can obtain good and clear candidates around the hitting points by the near-player, we start to find candidate trajectories from the location of the near-player when he is hitting the ball, which are illustrated by the green rectangles on the black lines (blue rectangles on the grey lines illustrate the locations of the far-player).

2.5. Ball Trajectory Processing

Trajectory confidence index: Let $\Gamma = \{T: \textit{candidate trajectory}\}$ be the trajectory set of a given video segment in the candidate feature image. Let $\lambda_1, \lambda_2, \cdots, \lambda_m$ be all properties of trajectory T. These properties characterize the statistical properties of objects in T (such as percentage of candidates in category 1, the length of the trajectory, etc). A function $\Omega_i(\lambda_i)$ computes the confidence index that T is a ball trajectory with respect to λ_i. The confidence index $\Omega(T)$ that T is a ball trajectory is defined below:

$$\Omega(T) = \sum_{i=1}^{m} \Omega_i(\lambda_i) \qquad (4)$$

Trajectory discrimination: In broadcast tennis video, there is at most one ball in the video. Thus, once we identify a ball trajectory, we know that all other candidate trajectories that overlap with it are non-ball trajectories, i.e. we can discard them. With this domain knowledge, we propose the following procedure to select the ball trajectories.

Let Γ be the set of all candidate trajectories.
SET *the ball trajectory set B to be empty.*
  WHILE *(Γ is not empty)* ***DO***
    Move the trajectory T with the highest index into B.
    Discard the trajectories that overlap with T in Γ.

Figure 3. Ball trajectory selection procedure.

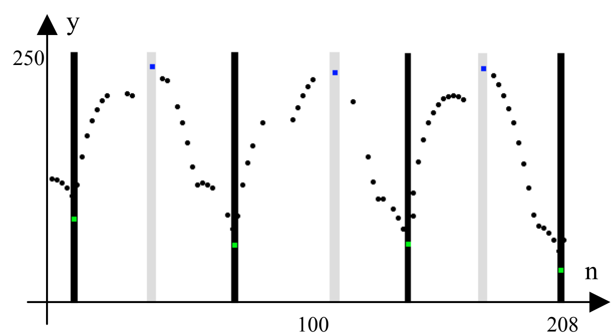

Figure 4. The obtained trajectories after discrimination.

Trajectory extension: Following [7-8], we use a Kalman filter-based model procedure to track the ball. However, not many balls can be identified through tracking because they fail to be segmented.

Ball location computation: When the ball is near the far-player, it may be occluded by the player, too small to be seen, or mixed with audience. Thus, it is useless to track the ball. Hence, we compute the ball location by extending the trajectory to the far-player's hitting location. This location is inferred from the obtained player's location and the obtained hittings as the ball is near the player who hits it. The ball location extended in the above-described way is not exact, but it can greatly facilitate the content analysis. For example, this location is enough to identify the winning-pattern of the game.

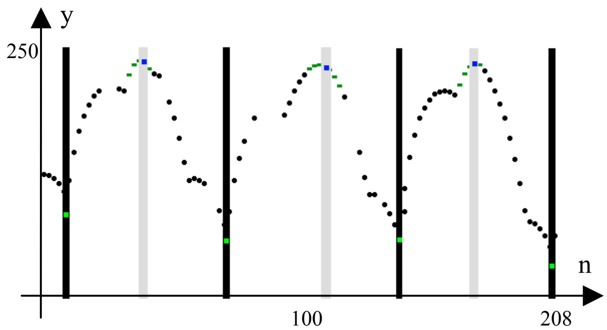

Figure 5. The obtained final ball trajectories.

3. EXPERIMENTAL RESULTS

The proposed algorithm has been tested on 7 segments (total of 120 seconds), which are from mpeg1 video recorded from TV signal. The content of video is the Men's Final of FRENCH OPEN 2003, which is the game between Juan Carlos Ferrero from Spain and Martin Verkerk from Holland, held on June 8, 2003. Note that, each frame must have two players. Hence, their ground truths are the number of frames in the considered segment. The experimental results of player and hitting detection are shown in Table 1, in which column "player" records the number of frames each player is detected within a segment; column "hitting" shows the ratio of hitting frames to the total number of successfully detected hitting frames associated with that player within the segment.

Table 1. Results of Player Detection and Tracking

segments	frame number	near-player		far-player	
		player	hitting	player	hitting
05355-06132	778	767	11/11	732	10/10
08905-09334	430	430	6/6	358	4/5
14526-14866	341	341	3/3	341	2/2
19025-19274	250	232	3/3	250	2/2
24492-24981	490	449	6/6	476	6/7
27520-27891	372	372	2/2	372	2/2
36960-37310	351	349	4/4	351	3/3

The experimental results of identifying the ball locations are shown in Table 2, in which column "balls/frames" gives the ratio of total frames to frames containing the ball; column "d+t" gives the numbers of detected balls and tracked balls separately; column "(d+t)%" gives the percentage of "d+t" to the number of frames containing the ball; column "final" shows the number of the obtained balls by detection, tracking, and computation; column "final%" shows the percentage of "final" to the number of frames containing the ball. You may notice that the percentage of the ball detection and tracking are not very high. This is because the ball cannot be seen when it is close to the far-player. The promising indication is that the proposed algorithm computes the ball locations from the obtained player's hitting locations. And the computed ball locations are accurate enough for further event detection and win-pattern identification, which are presented in other papers.

Table 2. Results of Ball Detection and Tracking

segments	balls/ frames	detected+tracked		computed	
		d + t	(d+t)%	final	final%
05355-06132	738/778	336+121	62.0%	738	100%
08905-09334	376/430	265+17	75.0%	361	96.0%
14526-14866	294/341	250+11	88.4%	294	100%
19025-19274	171/250	100+38	80.7%	171	100%
24492-24981	441/490	300+20	72.6%	426	96.6%
27520-27891	275/372	146+174	80.0%	275	100%
36960-37310	349/351	269 +10	79.9%	349	100%

4. CONCLUSION AND FUTURE WORK

We have presented a trajectory-based ball detection and tracking algorithm for broadcast tennis video. This algorithm can be viewed as an instantiation of the trajectory-based approach proposed by Yu et al in [6-8]. It possesses the merits of the trajectory-based approach, as well as making additional contributions. Firstly, it makes use of the location relation between players and ball to improve the ball candidate quality as the hitting player must be near to the ball in tennis. Secondly, it uses the hittings to decide the start and end points of the ball trajectories. In conclusion, this algorithm makes use of not only the merits of the trajectory-based algorithms, but also the domain knowledge of tennis video. The presented algorithm has the ability to obtain the ball locations that are accurate enough for further event detection and win-pattern identification.

In the future, we will detect the location of hitting-player's racket to improve the estimation of ball locations. We will further detect the tactics, exciting highlights and winning-patterns in broadcast tennis video based on the obtained ball trajectories.

5. REFERENCES

[1] T. D'Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences, In Proc. of ICPR 2002, Canada, Aug 11-15, 2002.

[2] H. Miyamori. Automatic annotation of tennis action for content-based retrieval by integrated audio and visual information, CIVR 2003, LNCS 2728, pp331-341

[3] G. Sudhir, J. C. M. Lee, and A. K. Jain. Automatic classification of tennis video for high-level content-based retrieval, Int. Workshop on Content-Based Access of Image and Video Databases, Jan 3, 1998, Bombay, India, pp81-90

[4] J. R. Wang and N. Parameswaran. Survey of sports video analysis: research issues and applications, Conference in Research and Practice in Information Technology (VIP2003), Australia, Vol. 36.

[5] M. Xu, N. C. Maddage, C. Xu, M. Kankanhalli, and Q. Tian. Creating audio keywords for event detection in soccer video, In Proc. of ICME 2003, Vol II, pp281-284.

[6] X. Yu, Q. Tian, and K. W. Wan. A novel ball detection framework for real soccer video, In Proc. of ICME 2003, Vol II, pp265-268.

[7] X. Yu, C. Xu, Q. Tian, and H. W. Leong. A ball tracking framework for broadcast soccer video, In Proc. of ICME 2003, Vol II, pp273-276.

[8] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang and K. W. Wan. Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video, In Proc. of ACM MM 2003, Nov 2-8, 2003, Berkley, CA, USA, pp11-20.

[9] X. Yu, C. Xu, Q. Tian, X. Yan, K. W. Wan and Z. Jiang. Estimation of the ball size in broadcast soccer video, IEEE Pacific-Rim Conference on Multimedia, Singapore, Dec 15-18, 2003.