Tracking a table tennis ball for umpiring purposes

Conference or Workshop Item

Tracking a table tennis ball for umpiring purposes

Hnin Myint, Patrick Wong, Laurence Dooley
The Open University, United Kingdom
Email: {hnin.myint, patrick.wong,
laurence.dooley}@open.ac.uk

Adrian Hopgood
Sheffield Hallam University
United Kingdom
Email: a.hopgood@shu.ac.uk

Abstract

This study investigates tracking a table-tennis ball rapidly from video captured using low-cost equipment for umpiring purposes. A number of highly efficient algorithms have been developed for this purpose. The proposed system was tested using sequences capture from real match scenes. The preliminary results of experiments show that accurate and rapid tracking can be achieved even under challenging conditions, including occlusion and colour merging. This work can contribute to the development of an automatic umpiring system and also has the potential to provide amateur users with open access to a detection tool for fast-moving, small, round objects.

1. Introduction

In table tennis, umpiring a match is a challenging task, even for trained personnel. Disagreement between players and umpires does occur. The service part of a rally is particularly difficult to umpire because up to 31 observations have to be undertaken within approximately a second [1], and the rules defining a good service is very complicated, e.g. one of the rules says the ball has to rise near vertically for 16 cm after leaving the server's fully opened palm. With the advances of computer vision technologies, it is possible that a specially designed computer system could perform better than human umpires. This forms the motivation for this research project. The success of such a system highly depends on its ability to accurately track the location of the ball. This paper therefore focuses the discussion on the ball detection and tracking part of the system.

Tracking a table tennis ball in a real match scene is difficult as the ball is small and often travels at high speed against a complex background. This can cause the image of the ball be merged with the background or be occluded by the players or their bats. Furthermore, the light condition of the scene and quality of the camera(s) used can cause additional difficulties as the image of the ball can become blurry and shape distorted.

In the literatures [2]–[10], ball detection and tracking generally consists of three fundamental processes, namely segmentation, detection and tracking. [2]–[6] employ colour thresholding (CT) based segmentation methods to extract candidate balls from each video frame. As the ball is of a uniformly colour (either white or orange), pixels that has values within a threshold range are considered to be from the ball. Although the method is simple, the images of the extracted objects are often distorted due to uneven lighting or colour merging with the background. Furthermore, the result of CT is very sensitive to the threshold setting and it is also difficult to set the threshold appropriately. To this end, [6] proposed a two-pass colour thresholding method, which uses a harsh threshold in the first pass to identify the locations of the candidate ball, and a lenient threshold on the second pass applying only to the regions identified in the first pass to recover the missing pixels filtered in the first pass. On the other hand, [7] proposed a dilate based method known as growth of sampled points to recover the distorted ball shape. Although both methods achieved some improvements, the performance of CT is generally poor when the ball is against a complex background.

An alternative segmentation method is background subtraction (BS) which works by subtracting the current frame from the background. BS is used by [7], [8] for extracting candidate balls. While it is simple and effective, it will work only if the surrounding area of the ball remains unchanged and the ball is in motion across consecutive frames. In real table-tennis match sequences, not only the ball but also players, table tennis bats and the crowd exhibit different motions. Therefore, ball detection purely based on motion is ineffectual.

Furthermore, a major challenge for segmentation is occlusion, which happens quite often in table tennis match sequences when the image of the ball is blocked by the players or their bats. To address this problem, multiple cameras approach have been incorporated for ball detection [2]–[5], [7]–[10]. With multiple cameras, the chance of capturing the ball is higher. An additional benefit is that the depth information can be derived by two overlapping views. The depth information can be used to predict three dimensional (3D) trajectory [3]–[5].

The segmentation process often results in a set of candidate balls, rather than just one ball. This is because of the existence of background objects of similar colour to the ball, or objects surrounding the ball exhibit motions. A ball detection process is therefore required to identify the ball. In the literature, feature based detection is commonly used [2]–[10]. The features are usually colour, size, shape and location of the ball. The features of candidate balls are compared with the expected values of the actual ball. The one with minimal error is deemed the detected ball. While Hough transform and Canny edge detection are commonly used for evaluating the shape, a trajectory model is used for estimating the ball location.

As for ball tracking, various trajectory models have been used. While [2], [9] and [10] employed Kalman filter, [4] and [7] used the aerodynamic model, [5] applied a bouncing model which is based on a physical motion model but will reinitialise the velocity calculations after a bounce is detected, [6] applies cubic extrapolation and [7] and [8] uses least square based 3D flight path model. Although [2]–[10] achieved some degrees of success in detecting and tracking a ball in their test environments, [2]–[5], [7], [8] mainly tested their algorithms in scenes of relatively simple background. While the proprietary Hawk-eye system [9], [10] is applied in real matches, it has to employ multiple broadcast-grade high speed cameras and fitted them at high locations to provide aerial views of the ball against a simple background of the court. The proposed automatic umpiring system is aimed to be portable and accessible to amateur users. The hawk-eye system is fixed and expensive and is therefore not suitable. In contrast, this paper reports the investigation and development of a low cost but high efficient ball tracking system, which is suitable for the proposed automatic umpiring system.

The remainder of the paper is organized as follows: Section 2 presents the proposed detection and tracking strategy while Section 3 briefly discusses the experimental set-up and the chosen test sequences. Experiment results and performance comparison are given to verify the effectiveness of the proposed algorithms in Section 4. Finally, a conclusion is provided in Section 5.

2. Proposed methodology

To tackle the challenges of tracking a table tennis ball for umpiring purpose, this paper proposes an efficient and effective detection and tracking strategy that can track a table tennis ball from stereo videos taken by two single view low-cost cameras by form of stereo vision. Similar to approaches discussed in the literature [2]–[10], the proposed system also follows the segmentation-detection-tracking approach, but with emphasis on segmentation and detection improvements. Furthermore to address the occlusion problem, a stereo vision system with an inter-view correction mechanism is also proposed.

The proposed system comprises of four main modules: combined adaptive colour thresholding and motion detection (ACTMD) module for ball segmentation, second order motion model (SOMM) module for trajectory prediction, feature based ball detection (FBD) module and inter-view self-correction (IVSC) module. A block diagram illustrating the connections of the modules is shown in Figure 1.

To make the system efficient, it must process only the essential region of a frame which contains the ball, known as region of interest (ROI). In the proposed system, ROI is defined as a small squared region with the width of the side set to several times (e.g. 2-3) the length of the diameter of the ball. This ensures the region is small yet big enough to fully contain the ball. The location of the centre of ROI is dynamically adapted to where the predicted centre of the ball is at each frame. The predicated ball location (PBL) is determined by the SOMM module, which is based on the second order motion model and can be calculated using Equation (1):

$$P_n = \begin{cases} C_1 & if \;\; n = 1 \; or \; 2 \\ B_{n-1} + v_n \Delta t & if \;\; n = 3 \\ B_{n-1} + v_n \Delta t + \frac{1}{2} \, a_n \, \Delta t^2 & if \;\; n > 3 \end{cases} \qquad (1)$$

where P_n is the predicted ball location of n^{th} frame and C_1 is the centre of the ball in the first frame and is given by the user. Although C_1 can be detected, the system is more robust if the initial location of the ball is obtained accurately from the user. B_{n-1} is the detected ball location in the previous frame, v_n is the velocity at frame n, Δt is the time difference between the two frames in which the ball is successfully detected and a_n is the acceleration at frame n. The velocity v_n and acceleration a_n are calculated by dividing Δt from the change in position and velocity respectively between the two immediately previous frames in which the ball was detected successfully. The location of the ball is tracked by using the predicted location for the current frame together with the saved locations of the previous frames. While this model is generally reliable, the predicted location can be erratic if the detected location in the previous frame(s) is wrong. To prevent this error from propagating, the predicted location will be corrected if its value is significantly different from the centre of the object that is most likely the ball (OMLB) by ACTMD and FBD.

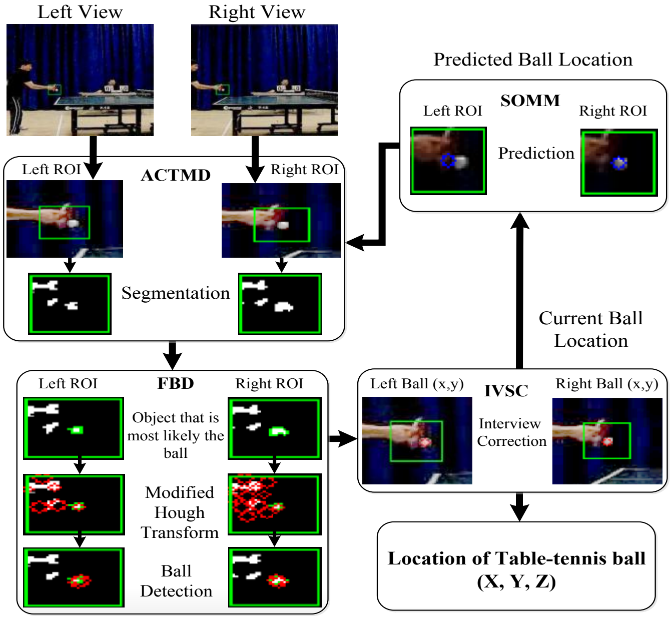

Figure 1. Block diagram of the proposed system

As for segmentation, the two commonly used methods, CT and BS, both have their own merits and drawbacks. Furthermore, the segmentation results often sensitive to the threshold value used. It is therefore desired to have a combined CT and BS method with its thresholds dynamically adapted. ACTMD is such a method and it filters the ROI by automatically selective CT and BS. In a typical match scene, the background is complex and often has multiple moving objects (e.g. players, bats, umpire and audience). The fluctuated intensity of the ball can be apparent if the video is captured at a rate much higher than the frequency of the electricity supply of the lights (e.g. 50Hz). ACTMD addresses this segmentation challenges by firstly checking for motion by using BS. If a segmented object of which the contour encloses the centre of the predicted location and with the smallest distance between itself and the predicted location, that object will be deemed the OMLB. Otherwise, CT will be attempted. If OMLB is found, it will proceed to the detection stage. If not, no ball can be detected in this view. However, the ball location may be recovered from the inter-view correction process, which is to be discussed later.

Once a ball is detected, a statistically based algorithm will tune the threshold of CT. The algorithm extracts the pixels of an area at the detected location of the ball in the previous frame. This area is slightly smaller than the ball (e.g. 80% radius) in order to prevent pixels of the background being incorrectly included if the detected centre of the ball is slightly inaccurate. Subsequently, the mean and standard deviation of pixel values of the area are calculated and the acceptable range of pixel values (R_i) is defined using Equation (2), where μ_i and σ_i are the mean and standard deviation colour values of the area, m_i is a user defined multiple, e is a real number between 1 and 2 that allows the user to include an error margin, and i is the index of the colour channels. The HSV (hue, saturation, value) colour scheme is chosen as the separation of a colour into hue, saturation and intensity values assists the thresholding of pixels which has the same colour but different intensity.

$$e(\mu_i - m_i\sigma_i) < R_i < e(\mu_i + m_i\sigma_i) \qquad (2)$$

As for BS, the system can establish the background by averaging a set of previous frames. However, due to the 50Hz variation of the intensity of the lights, a good background cannot be made this way. As a result, the previous frame only is used as the background. The drawback of this approach is that when the ball is not traveling very fast, the images of the ball in the background and current frame are partially overlapped. The subtraction of these two images often results in two crescent shaped objects situated at equal distance from the PBL. This makes determining the OMLB difficult. To remove the crescent corresponding to the ball image in the background, the direction of ball travel (calculated by SOMM) is used to guide the removal (e.g. if the ball is travelling from left to right, the crescent on the left will be removed).

After the objects of interest are segmented, the FBD module is used to decide which one is OMLB on each view by finding which object is nearest to the predicted ball location. As the contours of these objects are not necessarily circular, the centre of the object is therefore calculated by averaging positions of all the pixels enclosed by its contour. The object that has the shortest Euclidean distance between the centres of the object and the predicted ball is deemed OMLB. If only one object is segmented and its distance is significantly different from the predicted location, this may indicate the PBL is wrong. In such a case, the segmented object will be OMLB and the PBL will be reset to the centre of OMLB.

Subsequently, FBD will determine the centre and the radius of the ball by finding a circle that best fit the contour of OMLB and nearest to the predicted location. To obtain a good fit, a large number of candidate circles need to be trial, but this has a cost implication. Therefore, the proposed strategy is to first generate large number of candidate circles, but eliminate those that are of wrong sizes or orientations using a prior knowledge about the radius of the ball and the fact that the correct circles should contain the centre of OMLB. To this end, a modified Hough transform algorithm is employed to find and eliminate candidate circles. The circle with the smallest Euclidean distance from the PBL and radius error is chosen to be the detected ball. If the Euclidean distance and/or radius error are smaller than predefined thresholds, one point will be awarded to its detection confidence value for each satisfied criterion. This confident value will be used to guide the inter-view correction. As the ball detection is conducted independently on each view, when the ball in one view is undetected (e.g. due to occlusion) or wrongly detected, its location will be estimated using the location of the detected ball from the other view and the known disparity value from a previous frame where balls from both views are successfully detected. If the confidence value of one view is lower than the other, its ball location will be corrected using the information from the other view. This method can significantly improve the detection accuracy.

3. Experimental setup and tested sequences

The system was tested against a set of three sequences obtained from the Open University table tennis database (OUTTDB) [11] and an evaluation made of its ability to handle ambiguous conditions such as a sudden change of trajectory, occlusion, uneven illumination, scale variation, deformation, motion blur, noise and disappearance from the point of view. The first sequence is the same single-view video previously used by [6]. This particular sequence was selected to enable a performance comparison, discussed in Section 4. Both the second and third sequences are stereo videos captured during real matches. While the second sequence only views the left half of a table, the third sequence show the full table. As a result, the size of the ball in the third sequence appears much smaller. An example frame of each sequence is shown in Figure 3, on which the red and blue circles and green square indicate the result of the detected and predicted positions of the ball and ROI respectively. A summary of the key features of the tested sequences is shown in Table 1. All experiments were conducted on an Intel® Core™ 2 Duo CPU T8100 @ 2.10 GHz computer with the detecting and tracking algorithms implemented in C++ employing OpenCV computer vision library.

Table 1. Summary of the features of tested sequences

Features of Tested sequence	**Comparison**		
	Sequence 1	**Sequence 2**	**Sequence 3**
No of frames	46	200	400
Size of frame (pixels)	352×240	512×384	512×384
Capture rate	30 fps	300 fps	300 fps
Initial ball radius	4.5 pixels	3 pixels	1.95 pixels
Ball colour	Orange	White	White
Key detection challenges	Low frame rate, object blurring, merging, occlusion	colour merging, ball reflection, multiple moving objects and occlusion	Complex background, colour merging, ball reflection, multiple moving objects, occlusion, very small ball's size

4. Performance comparison

The detection performance was evaluated using a ground truth provided by [11]. The detection rate, root mean squared (RMS) error between the detected ball locations and the ground truth and the average process time per frame can be found in Table 2. If the Euclidean distance between the detected location and the ground truth is less than the diameter of the ball, the detection is deemed correct. The proposed system can accurately detect the balls with a high success rate, despite occlusion, colour merging, reflection, blurring and shape distortion occurs in the test sequences. Even with the very challenging Sequence 3, which has complex background and the ball being very small, the detection rate is still over 91%. The only occasions where the detection failed were when the ball was occluded or severely merged with the background. A trajectory comparison of the ground truth and the detected ball for each of these sequences is shown in Figure 4.

Table 2. Summary of the detection results

Quantitative Result	**Sequence 1**		**Sequence 2**		**Sequence 3**	
	Left View	**Right View**	**Left View**	**Right View**	**Left View**	**Right View**
Total frames	46	46	200	200	400	400
Detection rate	100%		95.5%		91%	
RMS error	1.38 pixels		1.6 pixels		1.9 pixels	
Process time	18 ms		47 ms		59 ms	

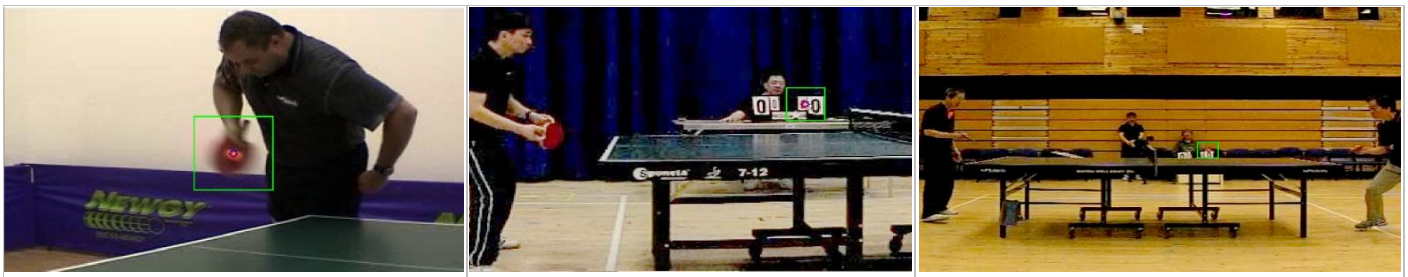

Figure 3a. Tested sequence 1 Frame #45

Figure 3b. Tested sequence 2 Frame #73

Figure 3c. Tested sequence 3 Frame #163

Figure 3. Tested sequences

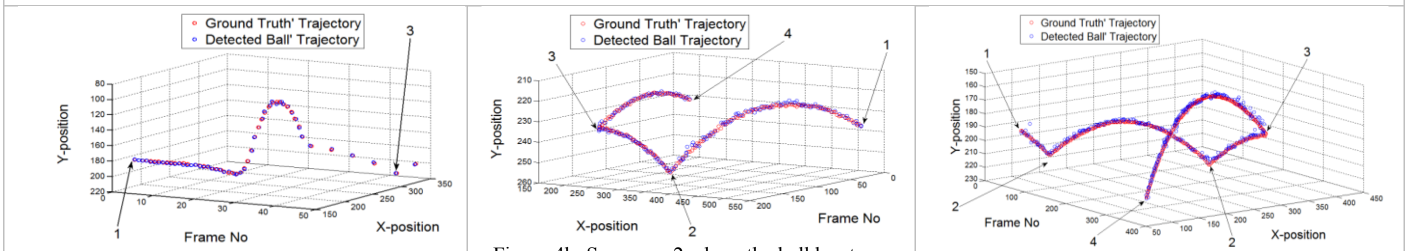

Figure 4a. Sequence 1 where the ball is located at the player's palm and served

Figure 4b. Sequence 2 where the ball has travelled from right to left

Figure 4c. Sequence 3 where the ball has travelled from left to right

Figure 4. Trajectory Comparison between the Ground Truth and the Detected Balls
Label 1- Ball at first frame (or) incoming ball, 2 - The ball has hit the table and bounced up, 3 - The ball has hit the bat and bounced back, 4- Outgoing ball

It is clear from the comparison that the trajectories are highly aligned, indicating successful detection throughout the whole sequence. The performance comparison between the proposed system and [6] is given in Table 3. As can be seen, the proposed system outperformed it on both detection rate and speed.

Table 3. Performance comparison

	Proposed			[6]		
Sequence	**1**	**2**	**3**	**1**	**2**	**3**
Detection rate	100%	96%	91%	98%	-	-
Process time (ms)	18	47	59	100	-	-

5. Conclusions

The proposed system, which was designed with highly efficient and effective algorithms, has demonstrated its ability to track a table-tennis ball accurately and rapidly. Despite the videos were produced by entry level cameras resulting in poor quality, the proposed algorithms can overcome these detection challenges. This shows the proposed system can be contributed to an automatic umpiring system. Furthermore, the techniques are applicable to the more general problem of tracking fast-moving small round objects and their low cost opens up the prospect of improved access by amateur users.

References

[1] Delano and L.F. (n.d.), “Level 1 Seminar for Umpires. ITTF Umpires and Referees Committee.” [Online]. Available: http://www.ittf.com/urc/courses/Umpire_Level_1_v1_files/frame.htm. [Accessed: 15-Dec-2014].

[2] M. Arenas, J. Ruiz-del-Solar, S. Norambuena, and S. Cubillos, “A Robot Referee for Robot Soccer,” in *RoboCup 2008: Robot Soccer World Cup XII*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Springer Berlin Heidelberg, 2009, pp. 426–438.

[3] X. Chen, Q. Huang, W. Zhang, Z. Yu, R. Li, and P. Lv, “Ping-pong trajectory perception and prediction by a PC based High speed four-camera vision system,” in *2011 9th World Congress on Intelligent Control and Automation (WCICA)*, 2011, pp. 1087 –1092.

[4] X. Chen, Y. Tian, Q. Huang, W. Zhang, and Z. Yu, “Dynamic model based ball trajectory prediction for a robot ping-pong player,” in *2010 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2010, pp. 603–608.

[5] H. Bao, X. Chen, Z. Wang, M. Pan, and F. Meng, “Bouncing model for the table tennis trajectory prediction and the strategy of hitting the ball,” in *2012 International Conference on Mechatronics and Automation (ICMA)*, 2012, pp. 2002 –2006.

[6] K. C. P. Wong and L. S. Dooley, “High-motion table tennis ball tracking for umpiring applications,” in *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, 2010, pp. 2460–2463.

[7] Z. Zhang, D. Xu, and M. Tan, “Visual Measurement and Prediction of Ball Trajectory for Table Tennis Robot,” *IEEE Trans. Instrum. Meas.*, vol. 59, no. 12, pp. 3195 –3205, Dec. 2010.

[8] J. Liu, Z. Fang, K. Zhang, and M. Tan, “Improved high-speed vision system for table tennis robot,” in *2014 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2014, pp. 652–657.

[9] B. Bal and G. Dureja, “Hawk Eye: A Logical Innovative Technology Use in Sports for Effective Decision Making,” *Sport Sci. Rev.*, vol. XXI, no. 1, pp. 107–119, Apr. 2012.

[10] N. Owens, C. Harris, and C. Stennett, “Hawk-eye tennis system,” in *International Conference on Visual Information Engineering, 2003. VIE 2003*, 2003, pp. 182 – 185.

[11] “The Open University table tennis database.” [Online]. Available: http://users.mct.open.ac.uk/xgmt/outtdb/. [Accessed: 13-Jun-2014].