

# High-Motion Table Tennis Ball Tracking for Umpiring Applications

K. C. P. Wong and L. S. Dooley

Next Generation Multimedia Technology Group, Department of Communication and Systems  
The Open University, Milton Keynes, MK7 6AA, United Kingdom  
{k.c.p.wong; l.s.dooley}@open.ac.uk

**Abstract**— Table-tennis umpiring presents many challenges where technology can be judiciously applied to enhance decision-making, especially in the service facet of the game. This paper presents a system to automatically detect and track the ball during table-tennis services to enable precise judgment over their legitimacy. The system comprises a suite of algorithms that adaptively exploit spatial and temporal information from real match videos, which are generally characterized by high object motion, allied with object blurring and occlusion. Experimental results on various table-tennis test videos corroborate the system performance in facilitating accurate and efficient decision-making over the validity of a service.

**Keywords**- *image processing; image segmentation; object detection; object tracking; computer applications*

## I. INTRODUCTION

The umpiring of table-tennis services is a challenging task with up to 31 separate observations needing to be made within a 1 second interval to decide the legitimacy or otherwise of a service [1]. Two of the most difficult judgments concern determining the height and deviation of the table tennis ball rise. Section 2.06.02 of the laws of the game [2] state:

*"The server shall project the ball near vertically upwards, without imparting spin, so that it rises at least 16cm after leaving the palm of the free hand and then falls without touching anything before being struck."*

For a human to correctly and repeatedly gauge the height and deviation of a table tennis ball rise by mere visual inspection is extremely difficult, so adopting computerized tools to provide fast measurement information for umpires, will help considerably in aiding their judgments. An intuitive, non-disruptive method of evaluating the ball rise is to analyze the video sequence of a service and accurately detect and track the ball. There are two challenges in ball detection in match scenarios: i) its relative size to other objects in the frame, i.e., players, table, and background and ii) the generally high object motion content, with the ball often being occluded or merging with foreground or background objects which have similar pixel intensity and shape. Furthermore, the ball can become blurred, off-color and its shape distorted in low frame rate sequences. All these factors influence the detection accuracy and can lead to incorrect decisions. Desai et al [3] proposed a method that successfully tracked table tennis balls using motion-based multiple filter banks, though the test sequences used were not from an actual match and the ball was the only moving object. In contrast, Chen et al [4] applied Kalman

filtering and an incremental Bayesian algorithm to track the ball in real match scenes, though the objective was to automate extraction of game highlights rather than fast precise object detection and tracking. These two requirements were the impetus for the research as they are of paramount importance in helping umpires make correct rulings on the ball rise during the service phase of a game. This paper presents an accurate and efficient system for detecting and tracking a ball by exploiting key spatial and temporal properties, with experimental results confirming the efficacy of the developed algorithms for different table-tennis sequences.

The remainder of the paper is organized as follows: Section 2 discusses the basic principles of the ball detection method, while the proposed techniques for improved detection accuracy and speed are presented in Section 3. Experimental results analysis is provided in Section 4, with some conclusions given in Section 5.

## II. BALL DETECTION METHOD

Table tennis sequences are usually characterized by multiple object motion so ball detection by exploiting temporal correlations alone is ineffectual. The method adopted in this paper uses frame-based object segmentation together with spatial and temporal clues such as size, shape and motion trajectory. Two color-based object segmentation techniques, thresholding [5] and clustering were considered for this particular application, with the former being the more appropriate due to its computational efficiency and there being only a single *object-of-interest* (OOI) – the table-tennis ball. Clustering-based segmentation [6] in contrast relies on securing an accurate estimate of the initial number of color clusters before iterative clustering commences, which cannot be guaranteed as the cluster number can change from frame to frame.

### A. Two-Pass Thresholding

The threshold-based color segmentation method in [7] was initially applied to create a binary image identifying pixels with a color similar to the reference OOI. These pixels were subsequently merged with neighboring pixels to form objects. The drawback however was that detection was solely dependent on the threshold, which is very sensitive to noise. To reduce the sensitivity, a *two-pass thresholding* (TPT) technique [8] which has been successfully applied to detect table tennis balls in high-resolution still images, has been extended to video sequences with varying resolutions. TPT uses different

thresholds in each pass with the first applying a coarse threshold, to find all pixels with a similar color to the OOI, which are merged together with neighboring pixels to form *candidate balls*. The aim is to approximate the location of each *candidate ball*, as defined by the centre of these objects. This approach can however remove portions of the ball, especially in and around its base due to variations in color and light shading. To recover the missing portions, the second pass uses a more relaxed threshold, which is only applied to those regions in the original frame in the neighborhood of a *candidate ball*. The benefit of TPT is it loosens threshold selection so the value in each pass can be less precisely set. The first pass estimates the location of a *candidate ball*, with the choice of different thresholds only impacting upon the object size whilst the location remains approximately constant. In the second pass, the *region of interest* (ROI) is restricted to either one or a small number of areas so any error in the threshold is insignificant [8].

### B. Object Evaluation

To assess which *candidate ball* is the OOI, a set of geometric measures based upon the respective spatial and temporal parameters of each object is employed. These are summarized in Table 1.

TABLE I. SUMMARY OF SPATIAL AND TEMPORAL OBJECT PARAMETERS

Parameters	Description
Area ( $A$ )	Object size (number of pixels)
Maximum width ( $W$ )	Horizontal distance between the left and rightmost pixels of the object.
Maximum height ( $H$ )	Vertical distance between the top and bottom most pixels of the object
Perimeter ( $P$ )	The length of object boundary
Roundness ( $R$ )	This is given by: $R = \frac{4\pi A}{P^2}$ where $0 < R \leq 1$ and $R=1$ is a circle.
Rounded Upper Contour ( $RUC$ )	$RUC = 1$ if $E_{RUC} < t_{RUC}$ $= 0$ if $E_{RUC} \geq t_{RUC}$ where $t_{RUC}$ is a preset threshold and $E_{RUC}$ is an objective (error) function defined as: $E_{RUC} = \frac{\sum_{i=1}^n  d_i - r }{N}$ and $d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$ where $(x_i, y_i)$ is a pixel in the upper contour, $i$ the pixel index, $N$ the number of pixels, $r$ the radius and $(x_c, y_c)$ the centre obtained by solving the equation of a circle for $(x_i, y_i)$ . $d_i$ is the distance between pixel $i$ and the centre.
Motion ( $M$ )	$M = 1$ if $O_{diff} \geq t_M$ $= 0$ if $O_{diff} < t_M$ where $O_{diff}$ is Euclidean distance between pixels at the centre of the OOI in the current and previous frames and $t_M$ is a preset threshold.
Trajectory ( $T$ )	$T = 1$ if $L_{diff} < t_T$ $= 0$ if $L_{diff} \geq t_T$ where $L_{diff}$ is Euclidean distance between the OOI's actual and predicted locations and $t_T$ is a threshold. The predicted locations are the linear extrapolation of OOI locations in previous frames.

During a table tennis service, the ball must be visible to both the opponent and umpire at all times [2]. While the ball is

normally a round object, when the ball is on the palm of the hand it becomes merged and can be occluded by the palm. Despite this the top portion of the ball will still exhibit a rounded upper contour (RUC) which is the initial clue exploited by the detection algorithm to establish whether the segmented candidate ball is the OOI. This could equally however be a background object having similar color and shape features to the OOI, so to improve detection accuracy further clues such as whether the object follows a predicted trajectory or exhibits motion are employed. Both these in certain circumstances can be unreliable as if for instance, the object was wrongly detected in a previous frame. To reduce this effect, each candidate ball is checked as to whether it: - i) has an RUC; ii) is in the predicted trajectory and iii) has motion. Only candidate balls satisfying at least two of these three criteria will be further considered. The advantage of this strategy is it resolves a number of detection ambiguities such as when:

- There are ball-shaped objects in the background scene which have a RUC, but neither motion nor trajectory. These objects will be rejected.
- The shape of the ball is distorted due to insufficient lighting, low frame rates or merging with other objects which do not have a high RUC, but the OOI still has trajectory and motion. This category of object will be accepted.
- The ball does not follow the predicted trajectory due to being incorrectly detected in previous frame but it has a RUC and motion. These objects will be accepted.

The five spatial geometric parameters summarized in Table 1 for each *candidate ball* are then measured i.e.,  $A$ ,  $W$ ,  $H$ ,  $P$  and  $R$ , with an objective (error) function  $E$  formulated:

$$E = \frac{\sum_{i=1}^n w_i \frac{|p_i - B_i|}{B_i}}{n} \quad (1)$$

where  $w_i$  is a weight applied to each geometric measure,  $n$  is the number of spatial parameters, and  $p_i$  and  $B_i$  are the respective candidate ball and actual ball parameters, so  $p_1 = A$ ,  $p_2 = W$ ,  $p_3 = H$ ,  $p_4 = P$  and  $p_5 = R$ . The candidate ball with the minimum  $E$  in (1) is then classed as the OOI.

### III. ENHANCED DETECTION ACCURACY AND EFFICIENCY

#### A. Adaptive control of the region of interest

In a typical frame, the OOI occupies a very small area ( $\approx 0.06\%$ ) and searching the entire frame for this object is computationally expensive. If the location of the OOI can be estimated, a ROI can then be established which limits the search area for the OOI and is centered about the predicted object location. In this application, once a ball is detected in a frame, its location in the next frame is predicted and an adaptive technique applied to modify the search area accordingly. This algorithm may be summarized as follows where  $j$  and  $k$  are both constants which are empirically set during initialization:

- In the first frame, set the size of ROI equal to the size of the frame.
- If the *OOI* is found, reduce ROI for next frame to a small square of which the length of the side is  $j$  times the diameter of the ball.
- If no *OOI* is found then scale the length of the ROI in the next frame by  $k$
- If the width (height) of the ROI is greater than the frame width (height), reduce to the frame width (height).

#### B. Automatic Tuning of Two-Pass Thresholding

While TPT makes threshold setting for object segmentation more straightforward, it is still desirable if both values could be automatically determined. To achieve this purpose, once an initial estimate of the *OOI* location is provided by the user, the following iteratively procedure tunes the TPT technique, where  $m$ ,  $g$ ,  $u$  and  $v$  are empirically defined constants:

- In the first iteration, set the ROI for the current frame to  $m$  times the ball diameter, where  $m$  is set to provide a larger ROI in order to tune the *Pass 1* threshold. Then search for the *OOI* using the TPT algorithm.
- If multiple *candidate balls* (objects) are produced in *Pass 1*, increase the level of the threshold by  $u\%$  for the next iteration and retune *Pass 1* threshold.
- If no *candidate ball* remains after *Pass 1*, reduce the threshold by  $u\%$  and retune *Pass 1* threshold.
- If only one object remains after *Pass 1*, this is the desired threshold value and *Pass 2* can commence.
- If the maximum number of *Pass 1* iterations is reached and no suitable threshold is found, then select the threshold that produced the minimum number of *candidate balls* and start tuning the *Pass 2* threshold.
- Calculate area difference between the ball ( $A_b$ ) and the object ( $A_o$ ) closest to the given ball location.
- If  $(A_b - A_o)/A_b$  lies within  $\pm v\%$  then set the current threshold as the *Pass 2* threshold and terminate tuning.
- Otherwise, if  $A_o > A_b$  then increase the *Pass 2* threshold by  $g\%$  in the next iteration, while if  $A_o < A_b$ , then reduce the *Pass 2* threshold by  $g\%$  and continue *Pass 2* threshold tuning.
- If the maximum number of *Pass 2* tuning iterations is reached with no suitable threshold being found, select the threshold with lowest  $(A_b - A_o)$  value.

#### IV. DISCUSSION OF RESULTS

To experimentally evaluate the performance of the system, three different test video sequences were used, with their respective temporal and spatial resolutions given in Table 2. Each sequence had multiple moving objects and distinctive characteristics in terms of the type of motion and object occlusion. *Sequence 1* was the popular SIF *Table Tennis*

sequence extensively used in video processing research, while *Sequences 2* and *3* were both shot in real match environments. *Sequence 3* was specifically produced with the goal of determining the best frame rate, spatial resolution, and camera angle/location for this application, while all sequences were edited so only the service element was analyzed, with the corresponding frame lengths given in Table 2. The key system constants were all empirically determined for initialization and maintained throughout the experiments. All experiments were conducted on an Intel Core2 Duo 2.2G PC with the detecting and tracking algorithms being implemented in Matlab R2009a. The system requires the user to only input the initial ball diameter, color and location for each sequence, with the corresponding test results being presented in Table 3. *Sequence 1* interestingly illustrates an illegal service since the ball is fully occluded by the player's hand for 8 frames. It has a complex background comprising objects of similar shape and size to the *OOI* and the video has global motion (camera zooming). The system successfully detected the *OOI* for the first 56 frames before incorrectly detecting objects in the background scene (see Figure 1(b) to see ROI as part of the wall poster). Camera zooming also created similar objects to the *OOI* close to the trajectory of the ball, together with making the actual size of the object smaller, hence the large variable ball size in Table 3. In spite of *Sequence 1* not being designed for umpiring applications, the system still achieved a 72% successful detection rate, while when frames where the *OOI* was occluded by the player are excluded, a detection rate of 80% was attained.

TABLE II. SPATIAL AND TEMPORAL RESOLUTIONS FOR EACH TEST SEQUENCE.

	<i>Sequence 1</i>	<i>Sequence 2</i>	<i>Sequence 3</i>
	<i>Table Tennis</i>	<i>ITTF [9]</i>	<i>Local league</i>
<b>No of Frames</b>	78	46	31
<b>Frame (pixels)</b>	352 x 240	352 x 240	720 x 576
<b>Capture rate</b>	30 fps	30 fps	100 fps
<b>Key Video Features</b>	Global & object motion. Similarity in fore & background objects (ball). Object blurring	Low frame rate. Object motion only, Small ball size. Blurred object (ball)	Large frame size. High frame rate Object motion only. Small ball size.

TABLE III. EXPERIMENTAL RESULTS WITH THE SYSTEM PARAMETERS BEING:  $J=3$ ,  $M=2J$ ,  $K=1.3$ ,  $U=V=10$ ,  $G=(100|A_b - A_o|)/A_b$  AND THE MAXIMUM NUMBER OF TPT TUNING ITERATIONS = 20

	<i>Sequence 1</i>	<i>Sequence 2</i>	<i>Sequence 3</i>
<b>Ball size (area)</b>	50—177 pixels	30—64 pixels	24—93 pixels
<b>Detection Rate</b>	56/78 (72%)	45/46 (98%)	31/31 (100%)
<b>Time for ball rise (secs)</b>	0.56 (7 frames)	0.55 (5 frames)	1.08 (9 frames)
Additional test results including actual video outputs, test sequences and further discussion can be found at <a href="http://xgmt.open.ac.uk/table_tennis">http://xgmt.open.ac.uk/table_tennis</a>			

*Sequence 2* in contrast was specifically designed for training purposes so the camera is positioned to provide an ideal umpire's view of a service. Although the frame rate is low, which causes some object blurring and the ball, which is orange color, appears small, the system achieved a 98% detection rate. It is important to highlight in a number of frames light variations on the red-colored table tennis bat

meant a strong correlation between the color intensities of the two moving objects existed, yet in only one frame did the system fail to correctly detect the ball, and that was when the OOI hit the bat in the final frame of the service when the ball was very blurred. If only frames from the ball starting to rise to its zenith are considered, which are essential in evaluating the height of ball rise, the detection rate was 100%. *Sequence 3* also displays an umpire's angle and view. This sequence was captured at a substantially higher frame rate and spatial resolution and although the ball appears to be very small compared to the frame size, the system successfully detected and tracked the OOI in all frames. From a computational efficiency perspective, whenever the system was able to track the ball, the adaptive ROI technique restricted the OOI search area to only three times ( $j=3$ ) the ball diameter. However, when tracking of the ball was lost, the ROI was scaled by 30% (i.e.  $k=1.3$ ) and the time taken to relocate the OOI became commensurately longer. Depending on the number of objects within the ROI, the average time required for processing a single frame in all experiments was approximately *100ms*. In terms of OOI detection and tracking for the entire table-tennis service, *Sequence 2* incurred approximately *550ms* to determine the ball rise time, but as highlighted in Section 1, this latency is more than adequate to assist the umpire in judging the legality of a particular service. These results clearly demonstrate the capability of the system to accurately and efficiently detect a table tennis ball from complex real match sequences for a variety of camera angles, capture rates and match conditions where both object blurring and occlusion are major challenges to be effectively resolved.

## V. CONCLUSIONS

Many sports are increasingly turning to technology for verification purposes in key umpiring decisions. Table-tennis has a myriad of diverse rules governing the legality of a service and this paper has presented an accurate and efficient system for detecting and tracking table-tennis balls during the complex

high motion service stage of a game. The system segments potential objects into *candidate balls* prior to adaptively exploiting both spatial and temporal information from real match videos to detect and track the actual ball. Experimental results on different test sequences confirm the system's consistent performance in enabling fast and precise decision-making over the validity of a table-tennis service.

## REFERENCES

- [1] Delano Lai Fatt, "Level 1 Seminar for Umpires", [http://www.ittf.com/urc/courses/Umpire\\_Level\\_1\\_v1\\_files/frame.htm](http://www.ittf.com/urc/courses/Umpire_Level_1_v1_files/frame.htm), ITTF Umpires and Referees Committee, accessed on 24/06/2009.
- [2] "International Table Tennis Handbook 2008/2009", [http://www.ittf.com/itf\\_handbook/itf\\_hb.html](http://www.ittf.com/itf_handbook/itf_hb.html), accessed on 24/06/2009.
- [3] Desai U. B., Merchant S. N., Zaveri Mukesh, Ajishna G., Purohit M., Phanish H. S., "Small object detection and tracking: Algorithm, analysis and application", Lecture Notes in Computer Science, Springer, Berlin 2005, ISSN 0302-9743.
- [4] Wei Chen, Yu-Jin Zhang, "Tracking Ball and Players with Applications to Highlight Ranking of Broadcasting Table Tennis Video", IMACS Multiconference on Computational Engineering in Systems Applications, Volume 2, 4-6 Oct. 2006 Page(s):1896 – 1903, China.
- [5] Xiayi H. and Boulgouris N.V., "Robust Object Segmentation using Adaptive Thresholding", *IEEE International Conference on Image Processing (ICIP'07)*, pp 1-45–48, San Antonio, Sept 2007.
- [6] Dooley L.S. and Karmakar G.C., "A Fuzzy Rule-based Colour Image Segmentation Algorithm", *IEEE International Conference on Image Processing (ICIP'03)*, pp 1-997–980, Barcelona, Sept 2003.
- [7] "Image Processing Toolbox DEMOS: Identifying Round Objects", Image Processing Toolbox 6.4, <http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ipexroundness.html>, The MathWorks Inc, accessed on 24/06/2009.
- [8] Wong, Patrick K. C., "Developing an Intelligent Table Tennis Umpiring System: Identifying the ball from the scene", *Second Asia International Conference on Modelling & Simulation*, 13-15 May 2008, Kuala Lumpur, Malaysia.
- [9] ITTF Umpires and Referees Committee, Umpires training videos. [http://www.ittf.com/\\_front\\_page/itf4.asp?category=urc](http://www.ittf.com/_front_page/itf4.asp?category=urc) Accessed on 10/01/2007.

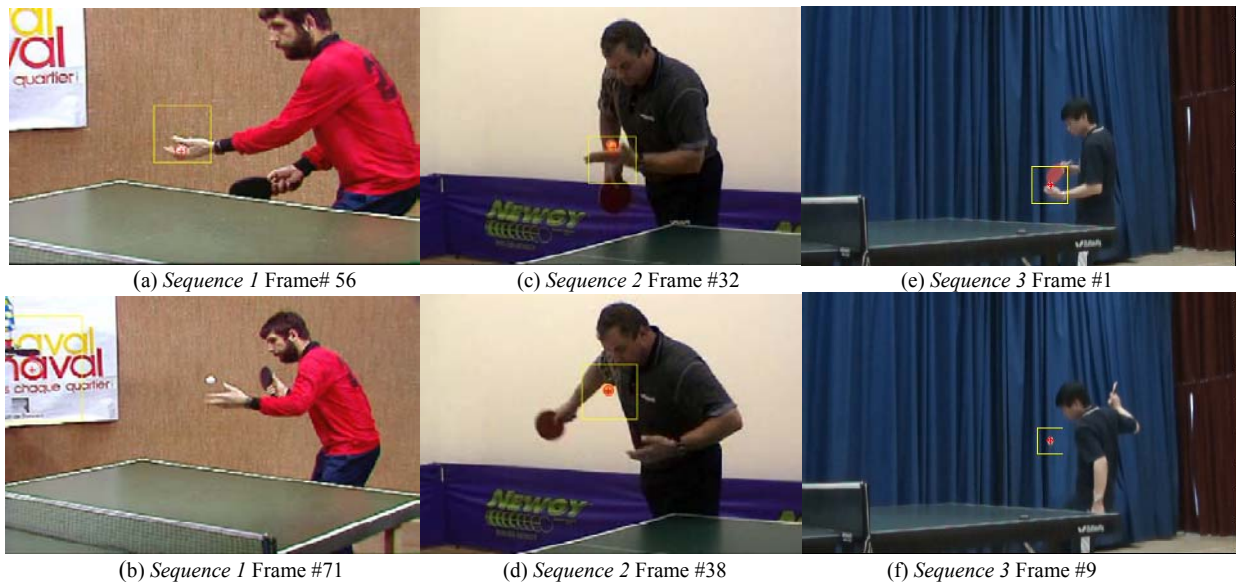


Figure 1. Example processed frames for the 3 test sequences. The ROI is represented by the yellow square; the red circles and crosses are the detected OOI contour and its centre respectively. (a), (c) and (e) show the OOI on the palm of the hand and (b), (d) and (f) at the highest point in the service.