



Tracking Table Tennis Balls in Real Match Scenes for Umpiring Applications

K. C. P. Wong^{1*} and L. S. Dooley¹

¹*Next Generation Multimedia Technology Group, Department of Communication and Systems,
The Open University, Milton Keynes, MK7 6AA, United Kingdom.*

Research Article

Received 22nd July 2011
Accepted 1st September 2011
Online Ready 15th September 2011

Abstract

Judging the legitimacy of table tennis services presents many challenges where technology can be judiciously applied to enhance decision-making. This paper presents a purpose-built system to automatically detect and track the ball during table-tennis services to enable precise judgment over their legitimacy in real-time. The system comprises a suite of algorithms which adaptively exploit spatial and temporal information from real match video sequences, which are generally characterised by high object motion, allied with object blurring and occlusion. Experimental results on a diverse set of table-tennis test sequences corroborate the system performance in facilitating consistently accurate and efficient decision-making over the validity of a service.

Keywords: Video processing; image segmentation; object detection; object tracking; table tennis; umpiring;

1 Introduction

Table tennis match umpiring is a very demanding task, with the service stage being particularly difficult to judge because so many observations need to be made within a very short period of time. According to Delano (n.d.), up to 31 separate observations are required to be checked within a one second time window, with the umpire needing to decide upon the legitimacy of a service shortly after it is conducted. To compound this problem, some observations relating to the height and deviation of the ball rise are very hard for a human being to judge. Section 2.06.02 of the laws of International Table Tennis Federation (2008) 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."

It is an especially difficult undertaking for a human to correctly and repeatedly assess the height and deviation of the ball rise by mere visual inspection. A more pragmatic approach is to employ

* Corresponding author; Email: k.c.p.wong@open.ac.uk;

computerised tools capable of making accurate and fast measurements of the ball rise, to aid the umpire in making correct decisions. An intuitive, non-disruptive way of evaluating the ball rise is to capture every service using a video camera and to then detect and track in real time, the *object of interest* (OOI) i.e., the ball, on a frame-wise basis. However, for a standard computer system to accurately segment and track in real time an OOI in match situations is extremely challenging for a myriad of reasons, including:

- **High object motion:** The ball travels fast. If the shutter speed of the camera is not sufficiently high, the object can become blurred, colour faded and distorted in shape.
- **Multiple moving objects:** Apart from the ball, the players, table-tennis bats and the crowd all exhibit different motion, so ball detection based purely on motion is ineffectual other than for the simplest scenes.
- **Uneven lighting:** Light sources are usually located in the ceiling, which tends to make the upper portion of ball appear brighter than the lower part.
- **Occlusion:** The ball can be blocked by the player, the bat, clothing, or for other reasons, such as its disappearance from view when it is thrown too high during a service.
- **Merging:** When the contrast between the ball and background is low, the ball may become indistinguishable from the background.
- **Object Confusion:** Background and foreground objects which have a similar colour, size and shape may be confused with the ball, such as for example, with the circular characters on the poster and the white ball in the SIF (Source Input Format) resolution *Table Tennis* sequence in Figure 1.
- **Small size:** The size of the ball is often only a few percent of the size of the frame, which renders conventional histogram-based detection methods (An et al., (2010)) unsuitable.
- **Time constraint:** As this is a real time application, the latency incurred for detecting and tracking the ball must be minimised which precludes computationally intensive algorithms from being adopted.

While considerable literature exists in relation to general object detection and tracking, the corresponding literature relating to table tennis applications is limited. Desai et al. (2005) for instance, proposed a motion-based multiple filter banks method for object tracking which was effective in tracking table tennis balls, though the test sequences used were relatively simple in nature comprising a plain background and with the ball being the only moving object. Furthermore, no experimental results from an actual table tennis match scene were presented. In contrast, Chen and Zhang, (2006) used Kalman filtering and incremental Bayesian algorithms to track the ball in real match scenes, and while the ball was successfully tracked, the objective was the automatic extraction of game highlights rather than fast precise object detection and tracking and no corresponding accuracy evaluation was provided. Maggio and Cavallaro (2009) presented an alternative object tracking algorithm based upon a combination of Mean Shift search and CONDENSATION in a particle filtering framework and a multiple semi-overlapping colour histograms based target representation. Results revealed this technique was able to detect the ball in certain frames of the SIF resolution *Table Tennis* sequence used by the image processing community (Maggio and Cavallaro, 2009; Comaniciu et al., 2003), though accuracy was not sufficiently adequate for umpiring applications.

While all three approaches, to some extent, successfully tracked the ball under prescribed conditions, reliable accurate and efficient object detection and tracking allied with low computational overheads to enable real time processing, remained a key objective. This provided

the motivation to investigate the development of a new system for fast and precise object detection to assist umpires in making correct rulings on the ball rise during the short service phase of a table tennis game. An innovative framework is presented to achieve this aim which exploits key spatial and temporal features of the OOI, with experimental results confirming the efficacy of the system in different table-tennis match sequences.

The rest of the paper is organized as follows: Section 2 discusses the underlying principles of the ball detection and tracking algorithms used in the new system. An experimental results analysis is then presented in Section 3, with some conclusions being provided in Section 4.

2 Ball Detection Method

To address the ball detection challenges described in Section 1, the proposed system employs various object detection and analysis algorithms, which are selectively applied from an initial object segmentation of a sequence to identify so-called candidate balls. These algorithms exploit both spatial and temporal information including object size, shape, colour, motion and predicted trajectory to assist in the decision-making process.

2.1 Object Segmentation

The first stage in the detection process is to segment the candidate balls from a frame in the sequence. A candidate ball is defined as an object that exhibits similar properties to the OOI, which is the table tennis ball. Motion and colour-based segmentation (Dooley and Karmakar, 2003) have been considered, though both have fundamental drawbacks. For example, as there are multiple moving objects in a typical match scene, the motion of the ball can often be difficult to accurately establish due to overlapping and motion-based segmentation is also generally computationally intensive. In contrast, for colour-based segmentation, clustering and thresholding techniques were analysed (Image Processing Toolbox DEMOS, n.d.). The accuracy of the former heavily relies on obtaining a good estimate of the initial number of colour clusters. For match scenes, securing an accurate and reliable estimate of the number of clusters is an intractable problem as it can often vary between frames. This provided the impetus to investigate for this particular application, the use of a threshold-based segmentation approach, especially as it is straightforward to implement and computationally efficient. Firstly, a binary differential image is created where all pixels with a colour difference with the ball less than a defined threshold are turned white, with those whose colour difference is greater than the threshold are turned black. Neighbouring white pixels are then merged together to form objects (Shankar, 2008)). The segmentation accuracy however, is entirely dependent on the initially selected threshold value, which is very sensitive to noise which makes automatically choosing the most appropriate threshold difficult. To overcome this sensitivity problem, a two-pass thresholding (TPT) method has been proposed by Wong (2009).

2.2 The Two-Pass Thresholding Method

This technique significantly improved the precision of the threshold tolerances so enabling accurate OOI segmentation in high-resolution still images (Wong, 2009), while TPT has subsequently been extended by Wong and Dooley, (2010) to enable OOI to be successfully segmented in lower resolution sequences. TPT processes each frame in two passes using two different thresholds. A coarse threshold is employed during the first pass so all pixels with a

dissimilar colour to the ball are filtered. The remaining pixels form candidate ball objects in an analogous manner to the standard thresholding method. Due to the coarseness of the threshold used, the shape of a candidate ball can be distorted, especially around its base due to light shading. The purpose of the first pass however, is to solely approximate candidate ball locations. Any distorted candidate balls will be restored in the second pass, where a more relaxed threshold is applied, but only to those regions where in the original frame, candidate balls and their neighbourhoods have been identified.

The relaxed precision for both thresholds is an attractive feature of the TPT technique. A poorly chosen threshold in the first pass for instance, only impacts on the object size while object locations remain approximately constant. In the second pass, the regions of interest (ROI) are restricted to either one or more small areas where there is normally only one candidate ball with a high contrast background. A less precise threshold therefore does not compromise the overall segmentation performance.

2.3 Automatic Tuning Of The Two-Pass Thresholds

While the TPT method tolerates lower precision thresholds, it is still desirable for these to be automatically determined. This can be achieved using a known reference location for the ball, which is provided for example, by the user during the calibration stage. Wong and Dooley (2010) developed an iterative algorithm for tuning the thresholds, with the basic principles being summarised below, 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 selected to provide a larger ROI in order to tune the *Pass 1* threshold. Apply TPT algorithm to segment the OOI.
- If multiple *candidate balls* (objects) are produced in *Pass 1*, increase the level of the threshold by $u\%$ for the next iteration and continue *Pass 1* threshold tuning.
- If no *candidate ball* remains after *Pass 1*, reduce the threshold by $u\%$ and continue *Pass 1* threshold tuning.
- If only one object remains after *Pass 1*, this is the desired threshold value and *Pass 2* tuning commences.
- 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 found (A_o) closest to the given ball location.
- If $(A_b - A_o)/A_b$ lies within $\pm v\%$ then set 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.

2.4 Adaptive Control Of The Region Of Interest

As the OOI is very small compared with the size of typical a frame ($\approx 0.06\%$), searching the entire frame for the OOI is computationally very expensive. A more efficient strategy is to define a ROI where the probability of finding the OOI is high. Furthermore, if the locations of the OOI in current and previous frames are known, its approximate location in the next frame can be predicted using extrapolation, and this predicted location then set as the centre of the ROI. As for the size of ROI, it is desirable to be made adaptive based on the search successes. For example, when the OOI is not found in the current ROI, the size of the ROI needs to be enlarged. Likewise, when the OOI is located within the current ROI, the size of the ROI can be minimised. Wong and Dooley (2010) proposed an adaptive algorithm to dynamically adjust the dimensions of the ROI and this has been embedded in the new system. For completeness, it is reproduced below, where both j and k are positive constants which are empirically determined during initialisation:

- In frame #1, 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 OOI
- 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).

2.5 Object Evaluation

Following segmentation, each identified *candidate ball* is analysed using a two stage evaluation. The first stage checks: i) if it has a *rounded upper contour (RUC)*, ii) its location is consistent with the predicted location (T), and iii) if it exhibits motion at both its centre (M_c) and the predicted location (M_p). The respective parameters RUC , T , M_c and M_p are formally defined in Table 1. A *candidate ball* must satisfy at least two of the above four conditions in order to fulfil the first test, whose purpose is to both eliminate *candidate balls* which are unlikely to be the OOI, while concomitantly including *candidate balls* with imperfections. For instance if the *candidate ball* shape is distorted i.e., no RUC due to insufficient lighting, but the object is in the vicinity of the predicted location and also exhibits motion, this *candidate ball* passes the first test.

The second stage of the evaluation uses a set of spatial geometric measurements, which are: area (A), maximum width (W), maximum height (H), perimeter (P), roundness (R) and the error function E_{RUC} . These parameters are defined in Table 2. An error function E can now be defined as:

$$E = \frac{E_{RUC} + w_A \frac{|A_c - A_b|}{A_b} + w_W \frac{|W_c - W_b|}{W_b} + w_H \frac{|H_c - H_b|}{H_b} + w_P \frac{|P_c - P_b|}{P_b} + w_R \frac{|R_c - R_b|}{R_b}}{(n_p + 1)n_c} \quad (1)$$

where w are the weightings applied to the spatial parameters, n_p is the number of spatial parameters, n_c is the number of conditions a *candidate ball* satisfied in the Stage-1 test, and A_c , W_c , H_c , P_c , R_c and A_b , W_b , H_b , P_b , R_b are the spatial parameters of the *candidate ball* and actual ball (OOI) respectively.

Table 1. Summary of the spatial and temporal parameters used in the candidate ball stage-1 test

Rounded Upper Contour (RUC)	$RUC = 1 \text{ if } E_{RUC} < t_{RUC}$ $= 0 \text{ if } E_{RUC} \geq t_{RUC}$ <p>where t_{RUC} is a preset threshold and E_{RUC} is an objective (error) function defined as:</p> $E_{RUC} = \frac{\sum_{i=1}^n \left \frac{d_i - r}{r} \right }{N} \quad \text{and}$ $d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}$ <p>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.</p>
Position (T)	$T = 1 \text{ if } L_{diff} < t_T$ $= 0 \text{ if } L_{diff} \geq t_T$ <p>where L_{diff} is the Euclidean distance between the OOI actual and predicted locations and t_T is a threshold. The predicted location is the linear extrapolation of OOI locations in previous frames.</p>
Motion (M_c)	$M_c = 1 \text{ if } OC_{diff} \geq t_M$ $= 0 \text{ if } OC_{diff} < t_M$ <p>where OC_{diff} is the Euclidean distance between pixels at the centre of the candidate ball and the OOI in previous frames and t_M is a preset threshold.</p>
Motion (M_p)	$M_p = 1 \text{ if } OP_{diff} \geq t_M$ $= 0 \text{ if } OP_{diff} < t_M$ <p>where OP_{diff} is the Euclidean distance between pixels at the centre of the predicted ball location and the OOI in previous frames.</p>

Table 2. Summary of spatial parameters for the Stage-2 test

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)	Length of the object boundary
Roundness (R)	Given by: $R = \frac{4\pi A}{P^2}$ <p>where $0 < R \leq 1$ and $R=1$ is a circle.</p>

The candidate ball with the smallest error E in (1) is then classified as the OOI. The rationale for employing this 2-stage test is to directly address the following three object detection challenges:

- i. There are other objects in the background scene with similar geometric features and dimensions as the ball. These may be misclassified as the OOI if only the Stage 2 test is applied. The checking mechanism for the predicted location and motion in Stage 1 prevents such misclassification.
- ii. The shape of the *candidate ball* is distorted due to insufficient lighting, low frame rates or merging with other objects. The *candidate ball* may not then be detected by the Stage 2 test alone.
- iii. The *candidate ball* is either partially or fully occluded. The *candidate ball* may not be correctly identified. Checking the predicted location and motion in Stage 1 assists in detecting the OOI.

2.6 Block Matching Detection

While the OOI detection method described performs well under many varied lighting conditions, in certain scenarios, particularly when processing low contrast sequences, the thresholding-based methods can fail. As a consequence, a *block matching* (BM) detection method has been included in the system, though this significantly increases the computational costs incurred. A *sliding window* (SW) is used to recursively process the ROI on a pixel-by-pixel basis, and compares the pixel-wise difference between the current and reference blocks so forming a BM error E_{BM} defined as:

$$E_{BM} = \frac{\sum_{i=1}^{W_{ROI}} \sum_{j=1}^{H_{ROI}} \sqrt{(pcr_{ij} - prr_{ij})^2 + (pcg_{ij} - prg_{ij})^2 + (pcb_{ij} - prb_{ij})^2}}{W_{ROI} H_{ROI}}$$

respectively, and i and j are indices.

The reference block is generated by extracting a block of pixels from the largest inner square enclosed by the circumference of the OOI, which in 2D will be a circle. The size of the SW is exactly the same as the reference block, so the OOI is considered to be the block with the lowest E_{BM} , provided it is less than an acceptable error level. The centre of the OOI is the centre of the block.

BM techniques do have some drawbacks. They are computationally expensive because of the inherent search overheads and are also unable to detect OOI which are partially occluded. As a consequence, BM is only applied in the system under specific conditions, namely when an OOI is not identified using the TPT method and the ROI is minimised by the ROI adaptive control algorithm described in Section 2.3.

3 Results Analysis

To evaluate the performance of the new table tennis ball detection system, a set of five test sequences including a mixture of resolutions, and varying object motions were chosen to evaluate all the assorted detection challenges identified in Section 1. *Sequence 1* was extracted from the widely adopted SIF *Table Tennis* sequence. The service segment comprises 79 frames and contains a number of detection challenges including camera (global) and object (local) motion, a blurred OOI due to the lower spatial and temporal resolutions and some object occlusion. *Sequences 2 to 5* were captured from actual match environments, though only the service elements were analysed. The full characteristics and detection challenges of each sequence are summarised in Table 3.

Table 3. Full characteristics and detection challenges of the five test sequences

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
	Table Tennis	ITTF	Timo Boll	Local league	High Throw
No of frames	79	45	30	67	39
Frame (pixels)	352 x 240	352 x 240	640 x 480	720 x 576	512 x 384
Capture rate	30 fps	30 fps	30 fps	100 fps	30 fps
Ball size area (pixels)	50-177	30-64	254-572	24-93	25-50
Ball Colour	White	Orange	White	White	Orange
Frame size	Small	Small	Large	Large	Medium
Variation in Ball size	Large	Medium	Medium	Large	Medium
Object motion	High	High	High	Low	High
Global (camera) motion	High	None	Low	None	None
Blurring of OOI	41 frames	7 frames	28 frames	None	17 frames
Merging	No	No	No	2 frames	9 frames
Occlusion	12 frames	No	No	1 frame	17 frames
Lighting	Good	Acceptable	Good	Dark	Dark

To ensure an equitable comparison, key system constants were empirically determined during initialisation and maintained throughout all the experiments, with the only user inputs required being the diameter, colour and location of the ball in frame #1 of a sequence. Numerical results showing the detection rates and computational times are presented in Table 4, while the qualitative results for samples frames from each test sequence are displayed in Figures 1 to 5 respectively.

Table 4. Numerical test 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 with the maximum number of TPT tuning iterations being set to 20

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
No. of correct detections	64	42	28	66	14
No. of incorrect detections	1	0	0	0	0
No. of undetected (frames with OOI occlusion excluded)	14 (2)	3	2	1 (0)	25 (8)
Detection rate (frames with OOI occlusion excluded)	81% (96%)	93%	93%	99% (100%)	36% (64%)
Time for ball rise (secs)	0.56 (7 frames)	0.55 (5 frames)	0.82 (8 frames)	1.08 (9 frames)	3.52 (12 frames)

Additional test results including actual video outputs, the five test sequences and further discussion can be found at http://xgmt.open.ac.uk/table_tennis

The service illustrated in *Sequence 1* is in fact illegal, since the ball is not stationary on the player's palm before the service starts, and it is fully occluded by the player's hand for 12 frames. The reason for analysing this particular sequence is because it is widely used in object detection research and comparative results are available. The sequence has a complex textured background

comprising objects of similar shape and size to the OOI and exhibits global motion (camera zooming) together a number of other detection challenges. The system successfully detected the OOI in 64 of the 79 frames and incorrectly detected an object as the OOI in only a single frame. Interestingly, neither the camera zooming nor the confusing OOI-like objects on the wall poster caused false OOI detections, while such features provided degraded detection performance in the previous system (Wong and Dooley, 2010). Despite *Sequence 1* not being designed for table tennis umpiring applications, the system still gave an overall detection rate of 81%, and if the frames where the OOI was occluded by the player during the service are excluded from the analysis, the detection rate was 96%. Some example frames are shown in Figure 1. The ROI is denoted by the yellow square; the red circles and crosses are the detected OOI contour and its centre respectively; and the green and blue circles are the locations of OOI derived using the mean shift and CONDENSATION algorithms (Maggio and Cavallaro, 2009) respectively. The detection method proposed by this paper outperforms both the mean shift and CONDENSATION algorithms, with only the computationally expensive mean shift search in a Particle Filtering framework combined with a multiple semi-overlapping colour histograms based target representation (Maggio and Cavallaro, 2009), producing comparable results, which cannot be seen in Figure 1 because they are overlapped by the red circles. Frame #78 shows where the OOI can be very difficult to detect because of fast camera zooming. This was not tested by Maggio, E., Cavallaro, A. (2009), yet the proposed method correctly detected the ball as evidenced in Figure 1(e). All the locations of the detected balls and the trajectory of the complete service are shown in Figure 1(f). The trajectory was formed by fitting a curve over the detected ball locations using cubic-Spline interpolation and was used to recover undetected OOI locations. This trajectory was also used to estimate the height and deviation of the ball rise.

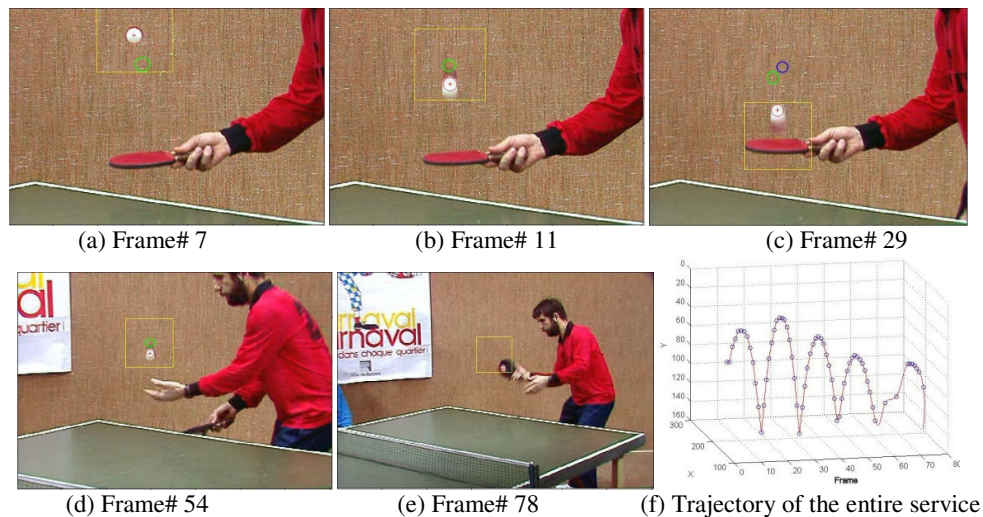


Figure 1: Sample processed frames for *Sequence 1* – (a), (b) and (c) show the OOI bouncing on the bat; (d) the OOI at its zenith during the service, (e) the penultimate frame of the service, and (f) the full service trajectory

Sequence 2 was specifically designed for training purposes, so the camera is positioned to provide an umpire's view of a service. Although the OOI, which is orange in colour, appears small and sometimes blurred due to the low frame rate used, the system still achieved a 93% detection rate

with no incorrect detections. It is important to highlight that in a number of frames light variations on the red-coloured table tennis bat produced a strong correlation between the colour intensities of the ball and bat (the merging problem discussed in Section 1). This was the reason for the unsuccessful OOI detection in just three frames, though when the ball strikes the bat in the final frame #45 of the service and becomes blurred, the system still correctly detected the OOI. Example frames from *Sequence 2*, covering various stages of the service are shown in Figure 2.

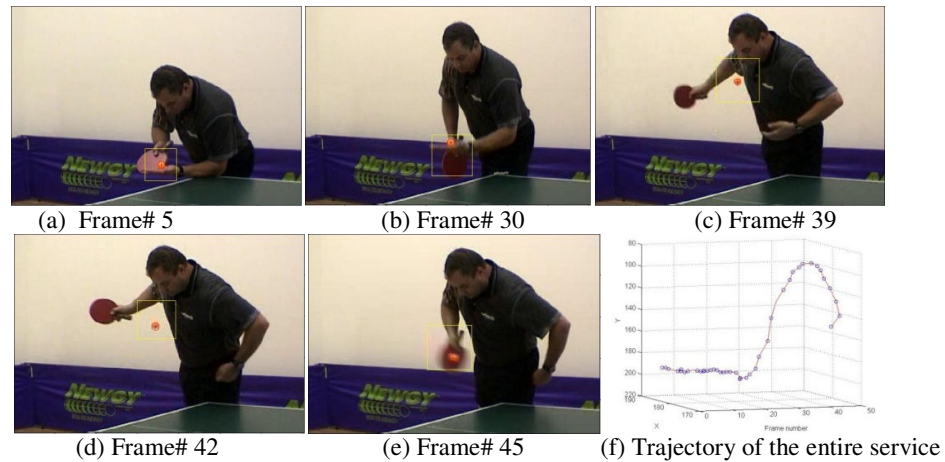


Figure 2: Sample processed frames for *Sequence 2* – (a)-(e) respectively show the OOI on the palm, rising, at the highest point, falling and being hit; and (f) the full service trajectory

Sequence 3 (ttCountenance, 2009) shows a service conducted by a table tennis player at an international match.

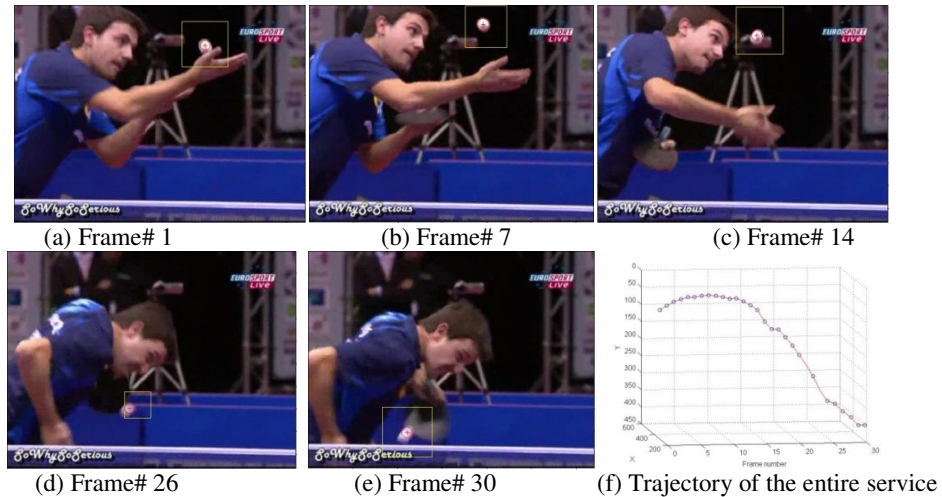


Figure 3: Sample processed frames for *Sequence 3* – (a)-(e) respectively show the OOI on the palm, rising, at the highest point, falling and being hit; and (f) the full service trajectory.

The sequence features a close-up of the player, so the ball appears to be large and in fast motion. As the capture rate is low, the OOI predominantly appears blurred in this sequence. There is also global motion as the viewing angle changes and camera shake occurs. Despite these detection challenges, the system achieved a detection rate of 93% with no incorrect detections. Figure 3 (a)–(e) illustrates some example frames from *Sequence 3*, covering respectively where the ball is on the palm of the hand, rising, at its zenith, falling and being struck.

Sequence 4 also shows an umpire’s angle and view. This sequence was captured at a substantially higher frame rate and spatial resolution so the ball appears to be travelling slower and is much clearer. However, the wide angle makes the OOI appear to be very small compared to the frame size and high capturing rate makes the video appear darker due to the much shorter exposure time used. When the OOI approaches the net, it becomes partially and then totally occluded, yet the system successfully detected the OOI in all frames, except for one frame when the ball was completely invisible. Example processed frames for *Sequence 4*, covering where the ball is on the palm, at the peak, being hit, and examples of partial and full ball occlusion are provided in Figure 4.

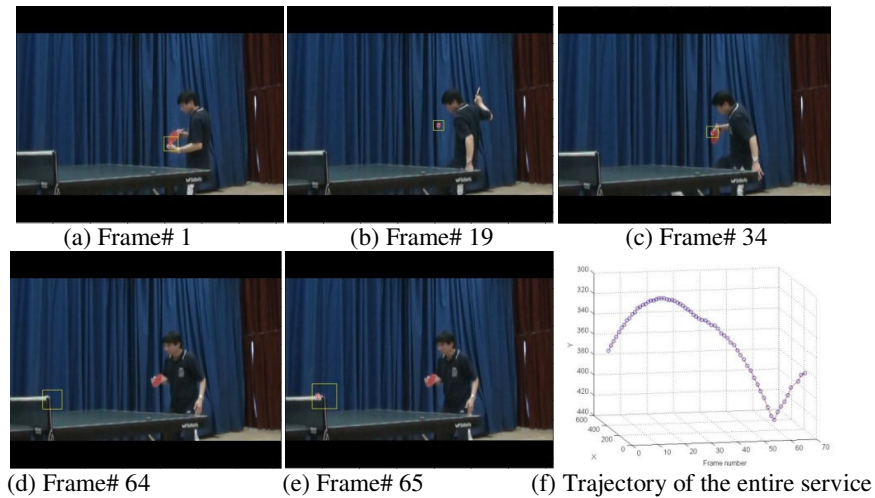


Figure 4: Sample processed frames for *Sequence 4* – (a)-(e) respectively show the OOI on the palm, at the highest point, being hit, completely occluded and half occluded; and (f) the full service trajectory

Sequence 5 represents a very challenging scenario, where the ball has actually been thrown too high so it disappears for a significant number of frames (17 out of 39 frames) in the service sequence. Additionally, the orange coloured ball temporarily merges with either the palm or face of the player in 9 of the frames. The height of the ball rise of this service far exceeds the rules and in this regard would be observed by the umpire. However, the purpose for analysing this particular sequence was to ascertain the processing limitations of the system in such a challenging detection scenario. The sequence was captured using a high shutter speed so the OOI appears clear though the side effect is the video becomes darker. The OOI nevertheless still appears blurred in 17 frames as it falls at high speed and is struck. If the 17 frames where the OOI was not inside the frame (and hence cannot be detected) are excluded from the analysis, an overall detection rate of

64% was attained. Figure 5 illustrates some sample processed frames for *Sequence 5*, covering where the ball is on the palm, rising, disappeared from the view, falling and being hit. Despite the OOI not being detected for 25 frames, the missing ball locations were still reasonably estimated by the trajectory plot (using Cubic-Spline interpolation) displayed in Figure 5(f).

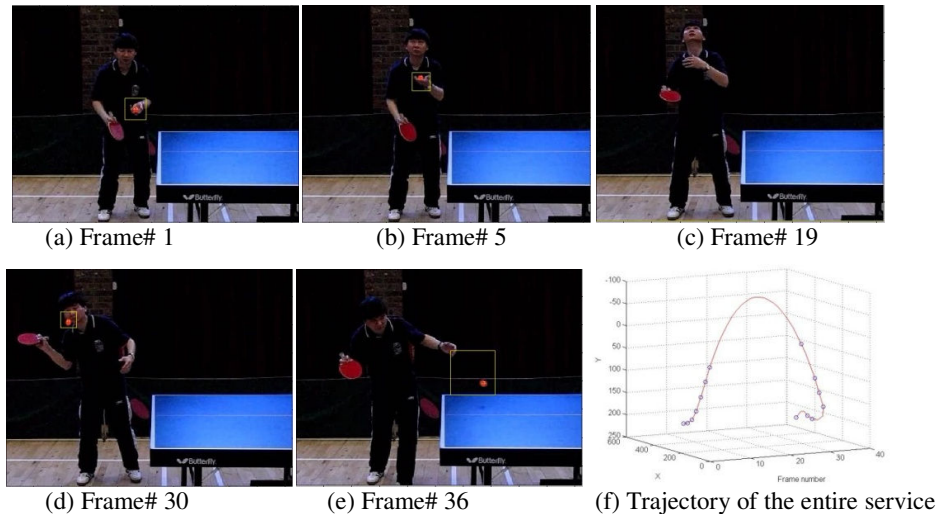


Figure 5. Sample processed frames for *Sequence 5* – (a)-(e) respectively show the OOI on the palm, rising, disappeared from the view, falling and being stuck; and (f) the full service trajectory

Finally, from a computational efficiency perspective, whenever the system was able to track the ball, the adaptive ROI technique constrained the OOI search area to a square whose lengths were three times ($j=3$) the ball diameter. When tracking was lost however, the sides of the ROI were 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, to measure the ball rise height of a service *Sequences 1 to 4* incurred less than one second to determine the ball rise time, and as highlighted in Section 1, this latency is more than adequate to assist the umpire judge the legality of a particular service. In contrast *Sequence 5* incurred a processing overhead of $3.52secs$ due to the high search cost for frames where the OOI was not visible. While the computational time for this sequence is considerably longer, pragmatically it would be acceptable as the high throw service takes longer and the umpire would not require technological aids to judge the legality of the ball rise as it clearly exceeded the height requirement.

These results clearly corroborate the system performance in accurately and efficiently detecting a table tennis ball from complex real match sequences involving a variety of camera angles, capture rates and match conditions, where both object blurring and occlusion are key challenges to be effectively resolved.

8 Conclusion

Many sports are increasingly exploiting 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.

It is important to stress that detecting objects in a highly complex environment and in real time, a purpose built detection system, as demonstrated here, often outperforms more generic systems which detect universal objects. Furthermore, the algorithms developed are modular and flexible so they can be easily adapted to detect other objects of interest such as other ball types, tyres on vehicles and bearing or rollers on machines.

References

- An, Y., Riaz, M, Park, J. (2010). CBIR based on adaptive segmentation of HSV Colour space. 12th International Conference on Computer Modelling and Simulation, 24-26 March 2010. Cambridge, UK.
- Chen, W., Zhang, Y.J. (2006). Tracking Ball and Players with Applications to Highlight Ranking of Broadcasting Table Tennis Video. IMACS Multi-conference on Computational Engineering in Systems Applications. Volume 2, 4-6 Oct. 2006 Page(s), 1896 – 1903, China.
- Comaniciu, D., Ramesh, V., Meer, P. Kernel-based object tracking, IEEE Trans. Pattern Analysis Machine Intell. 25(5), 564–577.
- Delano, L.F. (n.d.). Level 1 Seminar for Umpires. ITTF Umpires and Referees Committee. http://www.ittf.com/urc/courses/Umpire_Level_1_v1_files/frame.htm. Accessed 24/06/2009.
- Desai, U.B., Merchant, S.N., Mukesh, Z., Ajishna, G., Purohit M., Phanish H.S. (2005). Small object detection and tracking: Algorithm, analysis and application. Lecture Notes in Computer Science. Springer. Berlin 2005. ISSN 0302-9743.
- Dooley, L.S., Karmakar, G.C. (2003). A Fuzzy Rule-based Colour Image Segmentation Algorithm. *IEEE International Conference on Image Processing (ICIP'03)*, pp I-997–980. Barcelona. Sept 2003.
- Image Processing Toolbox DEMOS (n.d): Colour-Based Segmentation Using the L*a*b* Color Space. Image Processing Toolbox 6.4. The MathWorks Inc, http://www.mathworks.com/products/image/demos.html?file=/products/demos/shipping/images/ip_xfabric.html. Accessed on 24/06/2009.

International Table Tennis Federation (2008). International Table Tennis Handbook 2008/2009. http://www.ittf.com/ittf_handbook/ittf_hb.html. Accessed on 24/06/2009.

ITTF Umpires and Referees Committee (n.d.), Umpires training videos. http://www.ittf.com/_front_page/ittf4.asp?category=urc. Accessed on 10/01/2007.

Maggio, E., Cavallaro, A. (2009). Accurate appearance-based Bayesian tracking for maneuvering targets. *Computer Vision and Image Understanding*, 113, 544-555.

Shankar, J. (2008). Color-Based Segmentation with Live Image Acquisition. Matlab Image Processing Toolbox webcast, The MathWorks Inc. http://www.mathworks.com/products/demos/image/color_segmentation. Accessed on 24/06/2009.

ttCountenance (2009). Europe Top 12: Timo Boll-Vladimir Samsonov. [video online] Available at: <<http://www.youtube.com/watch?v=C9D88AcmLjI>> Accessed 21 December 2009.

Wong, K.C.P. (2009). Identifying table tennis balls from real match scenes using image processing and artificial intelligence techniques. *International Journal of Simulation Systems, Science & Technology*, 10(7), 6-14.

Wong, K.C.P., Dooley, L.S. (2010). High-motion table tennis ball tracking for umpiring applications. In: IEEE 10th International Conference on Signal Processing, 24-28 Oct 2010. Beijing, China.

© 2011 Wong & Dooley; This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.