Reconstruction of 3D Trajectories for Performance Analysis in Table Tennis

Sho Tamaki, Hideo Saito

Graduate School of Science and Technology, Keio University

3-14-1, Hiyoshi, Kohoku-ku, Yokohama, Kanagawa, Japan

`{tamaki, saito}@hvrl.ics.keio.ac.jp`

Abstract

A method of reconstructing the 3D trajectories of a table tennis ball is introduced, which was developed to solve the problem with conventional analysis in table tennis. There are several methods of reconstructing 2D ball trajectories or 3D trajectories of balls heavier than those in table tennis. However, these methods cannot be adopted to reconstruct the 3D trajectories of table tennis balls, because there are problems that are attributed to the dimensions of the trajectories and weight of the balls. The method proposed in this paper could reconstruct the 3D trajectories of a table tennis ball. The key feature of the method is that it approximates that a ball is traveling on tilted planes. This approximation makes reconstruction robust against failure to measure 3D ball positions. A system using two RGB cameras was developed based on the new method. The system experimentally demonstrated that it could provide accurate information for match analysis. A system using an RGB-D camera was then developed to optimize usability for practitioners. We experimentally demonstrated that the system could provide accurate information for service analysis.

1. Introduction

There has recently been growing interest in tactical and technical analysis though videos and statistics, *i.e.* so-called performance analysis, in sports sciences[4, 8, 9] where coaches can use the results of performance analysis to optimize decision making. There has been some research that has aimed at applying methods of performance analysis to table tennis[5, 7].

However, performance analysis has rarely been seen in practical scenario in table tennis because this analysis in table tennis requires enormous amounts of time or huge budgets due to the heavy workload imposed by data collection. The inefficiency of methods of data collection makes it difficult to carry out analysis, and it is not currently available to many practitioners.

We developed computer systems to solve this problem.

One of the systems was developed without any computer vision technologies. and it made data collection more efficient by supporting input operations by just using a graphical user interface (GUI). This system certainly advanced solutions to the problem. For example, the system enabled performance analysis at the London Olympics in 2012. While the system achieved advances, it still had a major problem in that they required a lot of time to collect shot data, such as the positions where the ball was hit and where it bounced. We have called analysis with software "conventional analysis" in this paper and the main problem with this originated from its dependence on the operation of manual input. Automated data collection is therefore urgently required. This problem could be solved if we could obtain the 3D trajectories of table tennis balls.

Here, we propose a vision-based method of reconstructing the 3D trajectories of table tennis balls. There are several methods of reconstructing 2D ball trajectories or 3D trajectories of balls that are heavier than table tennis balls. However, they cannot be adopted to reconstruct the 3D trajectories of table tennis balls because there are problems that are attributed to the dimensions of the trajectories and weights of a balls. The proposed approach in this paper can be used to reconstruct the 3D trajectories of table tennis balls. The key feature of the proposed method is approximation where the ball is traveling on tilted planes. This approximation makes the reconstruction robust against failures to measure 3D ball positions in various frames. In addition, the proposed approach can successfully be used to reconstruct trajectories with ball candidates including false positives. The method of dealing with false positives is based on the method proposed by Fei *et al*. [1]. We developed a system using two RGB cameras based on the new method. The system obtained shot data for match analysis and was expected to solve the problem with conventional analysis. We then developed a system using an RGB-D camera to optimize usability for practitioners. Although the system using the RGB-D camera could not reconstruct the trajectories of fast moving or rotating balls due to functional issues with the RGB-D camera, the system was useful for analyz-

ing services. Moreover, we expected that the system would be able to reconstruct the positions of any balls in the future when the frame rates and the measurement range of depths of RGB-D cameras were enhanced. The proposed method experimentally demonstrated that it made performance analysis of table tennis more efficient and more effective through these developments in the systems.

This paper consists of six sections. Section 2 briefly describes conventional analysis. Section 3 first considers technical issues in reconstructing the 3D trajectories of table tennis balls with usable photographic devices for performance analysis. Next, we describe our method that overcame several technical issues. Section 4 introduces the system that use two RGB cameras, which we experimentally evaluated. Section 5 introduces the system using an RGB-D camera, which we also experimentally evaluated. Finally, we conclude the paper in section 6.

2. Conventional Analysis

Table tennis matches were analyzed with conventional analysis in two steps with the GUI computer software we developed: (1) analysis that provided brief results based on data related to rallies and (2) analysis that provided detailed results, based on data collected at (1) and data related to shots. We collected data related to rallies in analysis (1), such as those from the server, the scoring player, and the shot number of the scoring shot[1]. After the data were collected, we computed statistics that could be used as indicators of (a) the scoring possibility depending on serves or returns, (b) transitions in scores, and (c) the scoring possibility depending on the shot number. Many data were obtained automatically with this software, based on the assumption that a shot was played alternately by players. Input operation was therefore just to record the times of shots and the shot number of the scoring shots, and we could conduct this analysis in real time. We collected data related to shots in analysis (2), such as the types of shots and the direction (position from which the ball bounced and the direction in which it moved) of a shot. After the data were collected, we computed the statistics that could be used as indicators of (a) the scoring possibility depending on the type of shot and (b) the scoring possibility depending on the direction of the shot. We mainly analyzed the spatial factor of tactics in this analysis. It was difficult to manually record the precise coordinates of shots, but it was possible to record the data on a scale according to a 3 x 3 divided area of a half court. Spatial tactics in table tennis are generally considered based on the 3 x 3 divided area. We could therefore use the direction data if they were obtained on a 3 x 3 scale of accuracy. This analysis involved an enormous workload, while analysis (2) provided useful results for performance analysis.

---

[1]Shot number denotes the ordinal number of shots in a rally. For example, making the serve is the first, and receiving it is the second.

Conventional analysis was used in performance analysis at the London Olympics in 2012. We analyzed 136 matches based on (1) and 59 matches based on (2) as a result of this analysis, which provided advanced performance analysis in table tennis. However, the problem where we needed a great deal of time to collect shot data remained because this could not be solved by manually collecting data conventionally, and we urgently needed a method of automatically collecting data.

3. Trajectory reconstruction

The biggest problem with conventional analysis is the enormous workload imposed by collecting shot data. We propose a method of reconstructing the 3D trajectories of table tennis balls to solve this problem. 3D trajectories provide the spatial features of shots (*e.g.* the positions from which balls bounced, their maximum height, and the direction and velocity of movements). However, because we cannot set up optimal environments for performance analysis in practice, it is difficult to reconstruct the 3D trajectories of table tennis balls. This section first introduces technical issues and related work. We then propose our method that overcame these issues.

3.1. Technical issues

First, let us consider usable devices in practical scenario (*e.g.* competition venues and training rooms). The setup to operate devices for performance analysis would be difficult to achieve if it required many devices. Therefore, the smaller the number of devices, the better. We can minimize the number of devices by using only one device that can detect depth, like an RGB-D camera. However, since RGB-D cameras cannot be set to have either high frame rate or short exposure times, we cannot target fast moving balls with them. We can target a fast moving balls by using two RGB cameras, although usability would decrease. We used two RGB cameras or an RGB-D camera to reconstruct trajectories for this reason.

Next, we considered the technical issues we faced in reconstructing the 3D trajectories of a table tennis ball. First, it is a difficult to detect a table tennis ball. We took a video far from the court at a competition venue. We could therefore not take images of the ball at high resolution. We would encounter the same problem when we used an RGB-D camera even in a training room because the resolution of standard RGB-D cameras is around 640 x 480 pixels. The diameter of a table tennis ball is less than 10 pixels in this situations and it is difficult to discriminate this from many other objects. Although there has been much research that has tried to detect actual balls from images, no researchers have detected balls accurately[3, 14]. Second, the 3D position of balls cannot be measured robustly because of the restrictions with usable devices. We cannot measure the

3D coordinates of balls with two RGB cameras when ball-to-objects, usually ball-to-player, occlusion occurs at one viewpoint. We encountered more difficult situations using an RGB-D camera because there were many missing values in the depth images we took with it. The ratio, expressed as a percentage, where we successfully measured the 3D positions of a ball was about 20% in all frames including the ball in our experiments. We needed a method that could reconstruct the 3D trajectories of table tennis balls, even if we had few 3D positions for the balls.

3.2. Related work

Poliakov *et al*. [10] proposed a physics-based method of tennis ball tracking based on knowledge that the ball was traveling from a racket. This method avoided the difficulty of ball detection using racket positions. However, it was also difficult to detect rackets, especially in table tennis, because they were small and were often occluded by the players. We would have faced other difficult issues if we had used their method.

Fei *et al*. [1] proposed a method of reconstructing the 2D trajectories of a tennis ball from cluttered data. They approximated a motion model of balls with constant acceleration over short periods of time and linked short trajectories by solving the shortest path problem. Their experimental results indicated this method could reconstruct a trajectory from detection results that including many false positives. However, if we failed to measure ball positions in successive frames, the approximation used in their method would be inappropriate. We often encountered such situations because of restrictions with usable devices, as we mentioned in the previous subsection. Therefore, their method could not deal with the problem.

Ren *et al*. [11] proposed a method of reconstructing the 3D trajectory of a soccer ball. They approximated the trajectory of a soccer ball with planar curves on vertical planes. Because they could obtain the 3D positions of the ball by projecting its 2D position, this method could be used to reconstruct the 3D trajectories of the ball when there were few 3D positions of balls. However, when a ball spinning fast and changed direction after being affected by lift force, the approximation that the ball will travel on a vertical plane is inappropriate. We could not use this method because table tennis balls are light and can be severely affected by lift force.

As a result, there were technical problems that related work could not solve. The problems were attributed to the dimensions of trajectories and the weights of the balls. We therefore needed a new method of reconstructing the 3D trajectories of table tennis balls.
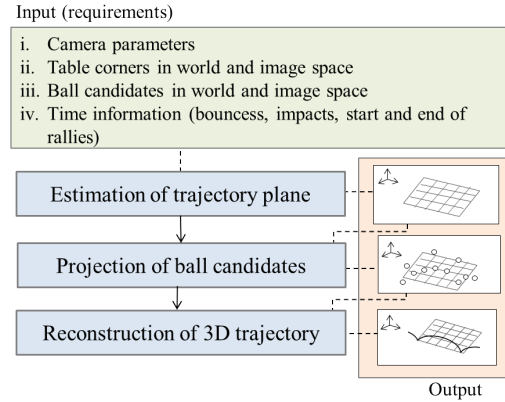

Figure 1. Flowchart to reconstruct trajectory for table tennis.

3.3. Our method

Figure 1 shows a flowchart for our method whose main feature was to approximate a ball traveling on tilted planes (trajectory planes). The planes were not assumed to be vertical to the ground, and the method could be applied to table tennis balls. We could reconstruct the 3D trajectories of balls from a minimum of three points by projecting ball positions in image space using the approximation. The problem with the few 3D positions of balls was solved. The problem of ball detection was solved by a method of reconstructing the 3D trajectories of balls from candidates including false positives. The approach that was used to estimate the trajectory model was based on the method proposed by Fei *et al*. [1].

The proposed method required six items of information: camera parameters, table corners in world and image space, ball candidates in world and image space, bounce time, impact time, and the start and end times of rallies. We assumed the impact times, and start and end times of rallies would be recorded with the conventional method. As it is not a problem to record data with the conventional method, these assumptions were appropriate. We have described methods of obtaining other previous information in Sections 4 or 5 because the methods depend on devices. We have described methods of reconstructing the 3D trajectories of table tennis balls in this section.

3.4. Estimating trajectory planes

We approximated that a ball was traveling on a tilted plane between ball-to-racket or ball-to-court collisions. We also obtained the bounce times and impact times. We therefore need to separately take each periods of time between collisions into consideration after this.

First, outliers are detected and eliminated by fitting the planar model to ball candidates in world space. Next, three optimal ball candidates are chosen. Let us denote the num-

ber of ball candidates in world space as n_c, the jth candidates as c_j, the plane constructed by three chosen ball candidates as P, and the Euclidean distance between plane P_i and candidate c_j as $D(P_i, c_j)$. The candidates that minimize the following cost function are chosen as:

$$C = \sum_{j=0}^{n_c} D(P, \boldsymbol{c}_j) \tag{1}$$

This computation may provide poor results when we have so few candidates because of the errors included in true positives. The trajectory plane is robustly estimated using bounce points as anchors. The number of candidates in a period can be too small to reconstruct a plane even using a bounce position. We set the threshold for the number of candidates required to reconstruct a trajectory plane with the method. The candidates in the time trajectory plane are not reconstructed are projected to another plane, which is in the closest planes in the future or past and provides smaller C.

3.5. Reconstruction of trajectory model

First, we project ball candidates in image space to a trajectory plane and obtain ball candidates in world space. Let us denote the normal of a trajectory plane as $\boldsymbol{n}$, the model of the trajectory plane as $ax + by + cz + d = 0$, and the physical position of the ball on the image as x_{image}. A ball position in world space x_{plane} is computed as:

$$\boldsymbol{x}_{world} = -\left(\frac{d}{\boldsymbol{n} \cdot \boldsymbol{x}_{image}}\right)\boldsymbol{x}_{iamge} \tag{2}$$

where the optical center of the camera that takes the projected ball candidate is defined as the origin of world space. The physical position of the ball on the image is computed by using camera parameters.

Next, we reconstruct the 3D trajectory model from the 3D positions of balls obtained before processes. We used the method proposed by Fei *et al.* [1] to estimate the trajectory model with extensions for 3D. First, false positives are eliminated based on the consecutive candidates in adjacent frames, as described in Fig. 2. Every candidate is examined in the process to check whether it has consecutive candidates, which is defined by the distance from the candidate, in adjacent frames. The distance that defines successive candidates is the maximum distance the ball can travel between frames. A candidate is eliminated when no other candidates are detected in either or both adjacent frames. If there is more than one candidates in a frame, all candidates in the frame are eliminated. Almost all false positives are eliminated after the process.

Next, trajectory models are estimated. We approximate that the motion model of the ball has constant acceleration in the specific period of time, n_c. Let us denote the positions
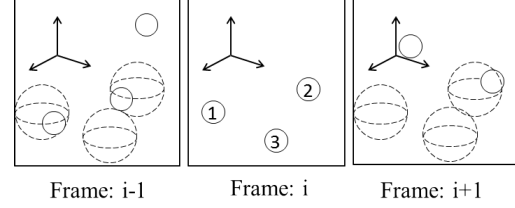

Frame: i-1   Frame: i   Frame: i+1

Figure 2. Elimination of false positives. Candidates 1 and 3 are eliminated in frame i, because they have no consecutive candidates within the threshold range (dashed sphere) in frame i-1 and/or frame i+1. Candidate 2 is not eliminated, because it has consecutive candidates in both adjacent frames.
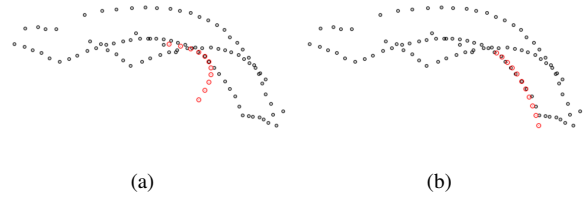

(a)   (b)

Figure 3. Estimation of trajectory model by iterative optimization. (a) is model before optimization and (b) is model after tree optimizations. Black circles are detected ball candidates. Red circles are estimates given by trajectory model.

of candidates as $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, and $\boldsymbol{p}_3$, their time as t_1, t_2, and t_3 ($t_1 < t_2 < t_3$), $t_2 - t_1$ as Δt_{21}, $t_3 - t_1$ as Δt_{31}, constant acceleration as $\boldsymbol{a}$, and velocity at time t_1 as $\boldsymbol{v}_1$. The constant acceleration model can be solved as:

$$\boldsymbol{v}_1 = \frac{\boldsymbol{p}_2 - \boldsymbol{p}_1}{\Delta t_{21}} - \frac{\Delta t_{21}\boldsymbol{a}}{2} \tag{3}$$

$$\boldsymbol{a} = 2\frac{\Delta t_{21} \times (\boldsymbol{p}_3 - \boldsymbol{p}_2) - \Delta t_{32}(\boldsymbol{p}_2 - \boldsymbol{p}_1)}{\Delta t_{21}\Delta t_{32}(\Delta t_{21} + \Delta t_{32})} \tag{4}$$

Models could be inaccurate when they are reconstructed from successive candidates in close frames. We solved the problem with the trajectory model by using three points, *i.e.* the earliest, the latest, and the middle ones in inliers, which were consistent with the trajectory. This iterative optimization stops when the three points do not change or the sum of errors starts to increase. As we can see from Fig. 3, the distance between the trajectory model that is estimated from the three successive points (Fig. 3 (a)) is longer than that of the trajectory model that is estimated after three iterations of the optimization process (Fig. 3 (b)). Next, we link reconstructed trajectories in time-series order. We then smooth the trajectories with a Savitzky-Gollay filter[12]. As a result, we can finally reconstruct the 3D trajectories of a table tennis ball.
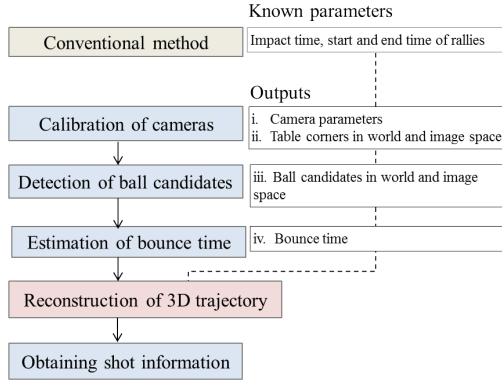
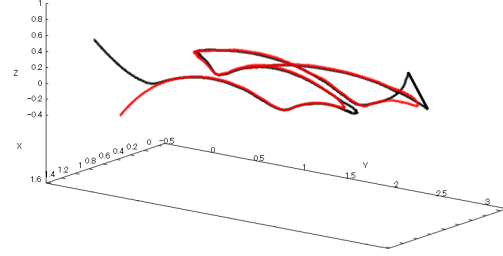Figure 4. Flowchart to obtain data using multiple camera system.

Figure 5. Reconstructed trajectory with system using two RGB cameras. Red curve was reconstructed using trajectory plane and black curve was reconstructed without trajectory plane.

4. System using two RGB cameras

4.1. System architecture

We described the system to reconstruct the 3D trajectory of a ball based on the proposed method in the previous section. This system was composed of two RGB cameras, whose exposures were synchronized, and an ordinary PC. It was operated in collaboration with conventional analysis and provided spatial data of shots. We expected collaboration to solve problems in conventional analysis. Figure 4 has the flowchart to obtain data with the system. The steps are described in detail in the following subsections.

4.2. Camera calibration

We used the method proposed by Zhang[15] to compute intrinsic camera parameters. Extrinsic camera parameters were obtained from planar transformation matrix (Homography) H[13]. Let us define the end lines and side lines of the court as the X-Y plane, the vertical direction of the court as the Z axis in world space, and the coordinates of the four corners as (0 m, 0 m, 0 m), (1.525 m, 0 m, 0 m), (0 m, 2.74 m, 0 m), and (1.525 m, 2.74 m, 0 m). Based on this definition, H can be computed [2]. The four corners were manually recorded. Optimal extrinsic parameters cannot be expected with this method because we only used four points on the same plane and independently calibrated each of the cameras. However, this method is easy to operate and can be used in many cases. Its results are also sufficiently accurate to be used for performance analysis. We used the calibration method in the system for this reason.

4.3. Detection of ball candidates

First, we eliminate large segments by using contour based segmentation and area based elimination. We used findContours, which is an OpenCV function, to obtain contours. The main aim of the process was to eliminate player segments. Player segments contain that have a similar ap-

pearance to balls and they can generate false positives. Ball candidates were then detected by extracting moving objects through inter-frame subtraction, and extracting high intensity segments and those that were circularity. Candidates without consecutive candidates in adjacent frames were eliminated (the same as the algorithm in Fig. 2) after this process. Next, we computed the 3D positions of the ball by triangulation. If there were multiple candidates in a frame, no candidates were computed in the frame. Finally, we could obtain ball candidates in world space.

4.4. Estimation of bounce time

We detected bounces from the 3D positions of the balls. First, we reconstructed the trajectories by using the method proposed by Fei *et al*. [1] and smoothed them with a Savitzky-Gollay filter[12]. Next, we looked for the time when the vertical component of the direction of movement changed from going downward to going horizontally or upward. This method could not detect bounces when a player hit a ball right after it bounced and the ball was occluded by a racket or a player. However, this had no affect on reconstructing the trajectories, since the time between bounces and shots was short.

4.5. Experiments

We performed experiments to verify whether our method could reconstruct 3D trajectories and provide performance indicators. The subjects were experienced table tennis players who executed ten rallies. The resolution of the video was 640 x 480 pixels, and the frame rate was 60 fps.

The reconstructed trajectory of a table tennis ball is shown in Fig. 5. The results for estimating the bounce positions are summarized in Table 1 where manually measured data were used as the reference data.

If we only use Fei *et al*. [1]'s method, the shape of the trajectory could be unnatural around impact due to the occlusion between players and the ball (the black curve in Fig. 5). Since our method is robust against occlusion by one viewpoint because the approximation of the trajectory of a ball is a planar curve, the shape of the trajectory is natural

Table 1. Results for estimating bounces. Errors are differences from ground truth recorded manually.

Indicator	Value
Minimum error [mm]	7.3
Maximum error [mm]	298.3
Average error [mm]	56.5
Standard deviation [mm]	50.4
Recall [%]	91.9
Precision [%]	97.1

(the red curve in Fig. 5). The average error in bounce positions, *i.e.* 56.5 ± 50.4 mm (Table 1), is attributed to inaccuracy in calibration because ball detection and the trajectory plane were accurately estimated according to the shape of the trajectory. This error was acceptable in actual practical use of performance analysis. As we discussed in Section 2, coaches and players who consider bounce positions generally refer to a 3 x 3 separated area of the half court and conventional analysis provides accurate data. The short side of the separated area is 460 mm, which is much greater than the error in our method. All the bounce positions were correctly predicted in this experiment on the 3 x 3 scale. Therefore, the results using the system were sufficiently accurate to solve the problem with conventional analysis.

However, we need to note that this system used two RGB cameras whose exposure times synchronized. Although it could be acceptable and effective in many cases, setup and calibration operations were required and usability was not optimal. The next section describes a system that makes it possible to solve the issue of usability.

5. System using RGB-D camera

5.1. System architecture

It is optimal to use an RGB-D camera, which can detect depth in a scene, from the aspect of usability as we mentioned in Section 3. Although current RGB-D cameras cannot track fast moving balls because of functional issues(*e.g.* low frame rates and long exposure times), these would not be problems if we only targeted services[2]. Moreover, developing the skill of service is regarded as an important issue in table tennis. It would be useful for players if there were a system to reconstruct the trajectory of services and provide their quantitative features. In addition, there is the possibility that the capabilities of RGB-D cameras will be improved and they will be able to track all kinds of balls in the future.

We developed a system that provided the features of service using an RGB-D camera. Fewer devices were required

---

[2]Players making serves in table tennis need to bounce the ball in their own court before they bounce it in the opposite court. It is therefore difficult to score by increasing the velocity of the ball. This is why many players mainly consider tactics with rotation and direction when serving.
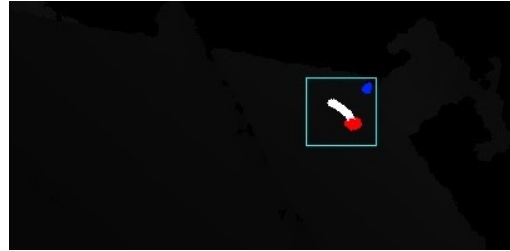

Figure 6. Detection of ball candidates in a depth image. White curve is 2D trajectory reconstructed based on RGB images. Rectangle is the range of positions where ball can be, which was defined with the trajectory. Red segment was selected as ball, and blue segment was rejected.

and it was easy to use. The flowchart for the system is nearly the same as the one that used two RGB cameras (Fig. 4). However, the algorithms that were adopted in each process were different. The steps are described in detail in the following subsections.

5.2. Detection of Table corners

First, we reconstructed a planar model of a court using the 3D positions of all pixels in the depth images. The RGB-D camera was placed above the table, and the major part of the image was a court. We could reconstruct the planar model using the random sample consensus (RANSAC) algorithm for this reason. We then detected white lines on the reconstructed plane by using a Hough transform. Finally, we selected four optimal points from the intersections of the white lines based on the area of the quadrangle that was comprised of selected points.

5.3. Detection of ball candidates

We needed to detect the ball in depth images to compute their 3D positions. However, it was difficult to detect the table tennis ball in depth images because there were many missing values and there was a great deal of noise in the depth images taken with the RGB-D camera. We therefore detected the ball in the RGB images, defined the range of positions where it could be in a depth images, and detected it. The position in a depth image calculated from the position in an RGB image using known geometric relations could be inaccurate because the exposure of the RGB camera and the depth camera were asynchronous. We therefore needed to detect the ball in RGB images and a depth images.

The method of detecting balls in RGB images was the same as that used in the system using two RGB cameras: the elimination of large segments, the extraction of moving objects, the extraction of high intensity segments, and extraction based on circularity.

We then reconstructed a 2D trajectory in RGB images using the method proposed by Fei *et al.* [1]. As we can see

Table 2. Results of recall ratio (%) of successfully measured 3D positional data for balls with RGB-D camera. This table indicates RGB-D camera could not robustly obtain data from moving balls.

Indicator	Value
Minimum [%]	10.3 (4/39)
Maximum [%]	24.4 (10/41)
Average [%]	20.7(7.8/37.8)
Standard deviation [%]	6.4

from Fig. 6, the trajectory defines the range of positions where the ball can be in a depth image. For example, when we look for ball candidates in the ith frame of depth images, the range is defined as the circumscribed rectangle of the 2D trajectory from the $i-1$th to the $i+1$ frames, which include specific size margins. Ball candidates are detected based on the area and the circularity of segments. The closest one in the detected candidates is chosen. The ball positions in the depth images provide the 3D coordinates of balls[6].

However, the ratio, expressed as a percentage, successfully used to measure the 3D position of a ball was about 20% in all frames that included the ball (Table 2). The ratio was low because there were many missing values in the depth images taken with the RGB-D camera and the table tennis ball was small. The results indicated that the RGB-D camera could not robustly obtain the depth values of moving table tennis balls.

5.4. Estimation of bounce time

There were few 3D positions for the ball as we mentioned in the previous subsection. We therefore detected bounces based on the 2D trajectory of the ball in RGB images, which had already been obtained when we detected the ball in depth images. The field of view could be assumed to be almost fixed. The bounce positions are the minimal values vertical to the direction of movement in the trajectories. The direction of movement in trajectories is determined according to the direction of movement of the ball when it bounces or peeks. The temporal differentiation of the inner product of the successive directions of movement of the ball takes a minimal value at peaks and bounces in the trajectory. We detected peaks and bounces in the 2D trajectory of the ball using this fact, and calculated the direction of movement in the trajectory by computing the average direction of movement by the ball at bounces and peaks. Let us denotes the direction vector of the ball at i as $\boldsymbol{v}_i$, and the second order differential of the inner product of adjacent directions of movement by the ball as $\boldsymbol{c}_{i,i-1}$, $\boldsymbol{c}_{i-1,i-2}$. The

direction of movement in 2D trajectory $\boldsymbol{d}$ is calculated as:

$$\boldsymbol{d} = \frac{\Sigma s_i}{|\Sigma s_i|} \quad (5)$$

$$\text{where} \quad \boldsymbol{s}_i = \begin{cases} \boldsymbol{0} & \text{if } c''_{i-1,i-2} = 0 \\ \boldsymbol{0} & \text{if } c''_{i,i-1} c''_{i-1,i-2} \geq 0 \\ \boldsymbol{v}_i & \text{otherwise} \end{cases} \quad (6)$$

We rotate the trajectory to fit the direction of movement in the trajectory on X axis. We can then detect bounces by seeking the minimal value for the Y components in the rotated trajectory. The service in table tennis bounces two times, $i.e.$ first in the serving player's own court and then in the receiver's court. As the second bounce position is important for tactical reasons. The second bounce position is used to estimate the trajectory plane.

5.5. Experiments

We carried out experiments to verify whether the system could reconstruct the 3D trajectory of a service and provide information. The subjects were experienced table tennis players who made six services. The resolution of the RGB and the depth images were 640 x 480 pixels, and the frame rate was 30 fps. Microsoft Kinect was used to take videos and was placed high, $i.e.$ 2.3 m, above the ground, opposite to the server.

The 3D trajectory of two services are shown in Fig. 7. The recall rates to measure the 3D positions of balls with the RGB-D camera are listed in Table 2. Five out of six trial shots were successfully reconstructed, the same as in Figs. 7 (a) and (b).

As we can see from Fig. 7, few 3D positions were measured with the RGB-D camera (white spheres) . The trajectory could not be reconstructed only by using the method proposed by Fei et $al.$ [1] from these positions. Unfortunately, the accuracy of results could not be quantitatively evaluated in these experiments, but we expected the trajectory to provide information that could be used for training purposes. However, one of the reconstructed trajectories could not be successfully reconstructed, as we can see from Figs 7 (c) and (d). There were few 3D positions of balls in the experiments. Therefore, all the ball positions in the trajectory were projected onto one trajectory plane. The projection was inappropriate for the service shown in Figs. 7 (c) and (d) because it rotated quickly and rapidly changed the direction of movement when it bounced. We may be able to solve problem by using RGB-D cameras whose frame rates are higher than that of Kinect because the problem just involves the amount of 3D positional data. We need to consider other RGB-D cameras for future systems.
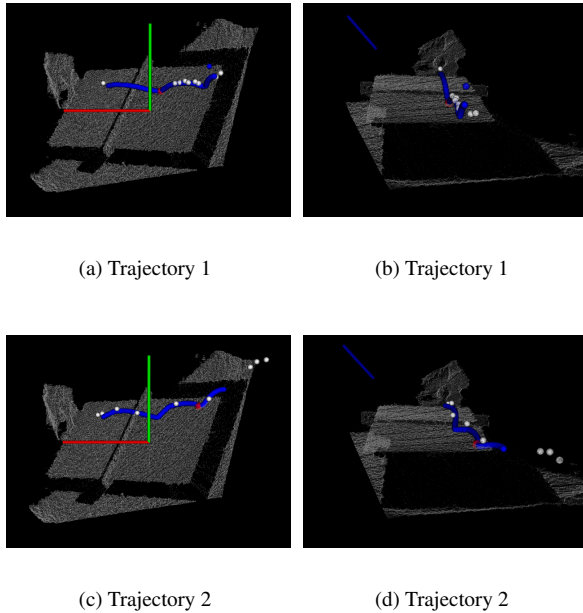
(a) Trajectory 1   (b) Trajectory 1

(c) Trajectory 2   (d) Trajectory 2

Figure 7. Reconstructed 3D trajectory. This figure indicates trajectory 1 was successful and trajectory 2 failed.

6. Conclusion

We introduced a method of reconstructing the 3D trajectories of a table tennis ball. The two key features of the proposed method were: (1) It was robust against failures in measuring 3D ball positions in some frames and (2) It could reconstruct trajectories from ball candidates including false positives. A system using two RGB cameras was developed based on the new method and it was experimentally verified that it could provide accurate information for match analysis. We also developed a system using an RGB-D camera to optimize usability for practitioners. The system experimentally verified that it could provide accurate information for service analysis. Although the system using the RGB-D camera cannot currently track fast moving or rotating balls, this will not be a problem in the future because this issue depends on the capability of current RGB-D cameras. In conclusion, it was possible to collect shot data from table tennis matches without imposing heavy workloads by using the systems. Future issues involve using the systems in practical scenarios that will contribute to making performance analysis more efficient through upgrades.

Acknowledgements

References

[1] Y. Fei, K. Alexey, C. William, and K. Josef. A novel data association algorithm for object tracking in clutter with application to tennis video analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 634–641. IEEE Computer Society, 2006. 1, 3, 4, 5, 6, 7

[2] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2004. 5

[3] Y. Huang, J. Llach, and C. Zhang. A method of small object detection and tracking based on particle filters. In *Proceedings of 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 2

[4] M. D. Hughes and R. M. Bartlett. The use of performance indicators in performance analysis. *Journal of Sports Sciences*, 20(10):739–754, 2002. 1

[5] Z. Hui, Y. Lijuan, and H. Jinju. Computer-aided game analysis of net sports in preparation of chinese teams for beijing olympics. *The International Journal of Computer Science in Sport*, 9(3):53–69, 2010. 1

[6] K. Khoshelham and S. O. Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012. 7

[7] M. Lanzoni Ivan, D. M. Rocco, and M. Franco. Performance indicators in table tennis: a review of the literature. *Proceedings of the 12th ITTF Sports Science Congress*, pages 71–75, 2011. 1

[8] A. Lees. Science and the major racket sports: a review. *Journal of sports sciences*, 21(9):707–732, 2003. 1

[9] A. Nevill, G. Atkinson, and M. Hughes. Twenty-five years of sport performance research in the journal of sports sciences. *Journal of sports sciences*, 26(4):413–426, 2008. 1

[10] A. Poliakov, D. Marraud, L. Reithler, and C. Chatain. Physics based 3d ball tracking for tennis videos. In *Proceedings of International Workshop on Content-Based Multimedia Indexing*, pages 1–6. IEEE, 2010. 3

[11] J. Ren, J. Orwell, G. A. Jones, and M. Xu. A general framework for 3d soccer ball estimation and tracking. In *Proceedings of International Conference on Image Processing*, volume 3, pages 1935–1938. IEEE, 2004. 3

[12] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964. 4, 5

[13] G. Simon, A. W. Fitzgibbon, and A. Zisserman. Markerless tracking using planar structures in the scene. In *Proceedings of International Symposium on Augmented Reality*, pages 120–128. IEEE, 2000. 5

[14] C. Wei and Z. Yu-Jin. Tracking ball and players with applications to highlight ranking of broadcasting table tennis video. In *Proceedings of Computational Engineering in Systems Applications*, volume 2, pages 1896–1903, 2006. 2

[15] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000. 5