A Training Support System for Table Tennis Service using Kinect

Sho Tamaki[1], Jun Nakamura[1], Hideo Saito[1]

[1]Graduate School of Science and Technology, Keio University, Japan

Abstract: In this paper, a system which provides data about table tennis service is proposed. The system reconstructs 3D trajectory of a ball and obtains the features of the service. In addition, the system provides the video in which player served without any operations. The system is comprised of a Microsoft Kinect and an ordinary PC. Experimental results show that the system can provide videos and following kinds of information: bouncing position, velocity and maximum height of a service.

Keywords: Kinect, training support system, trajectory reconstruction

1. Introduction

In table tennis, developing service skills is important challenge for many players. Skillful players can confuse opponents in various rotations, velocities and courses. When we acquired such skills, service can increase the number of errors of opponents and make scoring easier.

However, it is hard to get quantitative understanding of improvement of the service skills. When we try to enhance own skills, we need analyze the features of own services. Conventionally, players manage to understand the feature of their services with only their subjective feelings or videos, which were taken by themselves or their teammate. This way to analyze has some problems. First, the analysis can be too subjective to compare multiple services. We should perform analysis with objective data. In addition, it takes a lot of time to playback service scene manually, repeatedly, and it would lessen the training efficiency.

We developed a vision-based system in order to solve the problem. The developed system reconstructs the 3D trajectory of a ball and obtains the features of the service. The system provides not only statistics, but also the videos in which a player is hitting a service without any operations. In addition, it is comprised of inexpensive devices: a Microsoft Kinect and an ordinary PC. In section 2, Kinect is briefly introduced. The developed system and experimental results are described in section 3, section 4 respectively. Finally, we conclude this paper in section 5.

2. Microsoft Kinect

Kinect is a device that can take RGB (color) images and depth images simultaneously. Since geometrical relation between the RGB camera, the depth camera and the infrared projector are known, 3D coordinate of every pixels in a RGB image are easily obtained [3]. It was primarily designed for natural interaction in a computer game environment. Kinect provides better depth values than conventional depth cameras and it is inexpensive [5]. Kinect has recently been used in a lot of research purposes.

In our experiment, however, Kinect has some problems as a capturing device for a moving table tennis ball. First, it is often impossible to obtain 3D position of moving object in a RGB image due to the asynchronous exposure of the RGB camera and the depth camera. The position of the moving object at the time RGB camera exposed is different from the position at the time depth camera exposed. In addition, the depth value of a table tennis ball is not obtained robustly, because there are many missing values and noises in depth images. When we use Kinect for ball tracking, we need solve the problems.

3. Developed System

The system is comprised of a Kinect and an ordinary PC. Kinect is set at high position of the opposite side to the server side.

Fig. 1 shows the flowchart of the obtaining service features using the system. First of all, the system detects court corners and reconstructs planar model of the court. The corners detection is done once before the beginning of training. Next, the system detects ball candidates in RGB images and depth images. To detect the candidates in a depth image, the system reconstructs 2D trajectory of the ball based on the RGB images. After the above processes, we have 3D position of balls in some frames. However, the amount of 3D positions of ball is too small to reconstruct 3D trajectory due to the problem of Kinect as mentioned in section 2. To deal with the problem, the system reconstructs 3D trajectory by estimating the plane on which ball travels (we call it ``trajectory plane'') and projecting candidates in RGB images to the trajectory plane. Finally, we obtain 3D features of a service from reconstructed trajectory. The algorithms of each process are described in the following sub sections.

3.1 Table Corners Detection

Kinect is placed above a court, and the major part of the image is a court. For this reason, we can reconstruct the planar model by fitting a plane to the 3D point clouds that are captured with Kinect. RANSAC [4] is adopted for the plane fitting procedure. At the next step, the system detects white lines on the reconstructed plane by Hough transform [1]. Finally, optimal 4 points are selected from intersections of white lines based on the area of the quadrangle comprised with selected points.

3.2 Ball Trajectory Reconstruction

Ball candidates in RGB images are detected by inter frame subtraction and segments extraction based on the color and the shape features. After the process, there would be ball segments and athlete segments. Since the exposure of Kinect cannot be set small value, we cannot discriminate between the balls and the white segments on athletes. The system eliminates big segments before the above processes to deal with the problem.

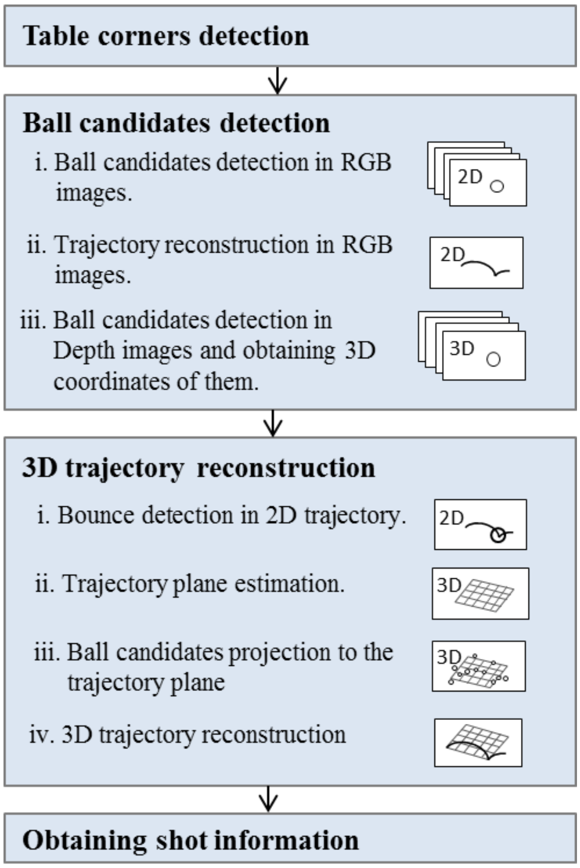


Figure 1. Flowchart of service features obtainment.

At the next step, ball candidates in depth images are detected. Since there are many noises and missing values in depth images, candidates are sought in the specific range where ball can exist. The seeking range is defined by 2D trajectories of the ball based on the RGB images. 2D trajectories are reconstructed by the method proposed in [2]. A seeking range of frame i is defined by the circumscribed rectangle of the 2D trajectory from frame i-1 to frame i+1 including arbitrary margins, as we can see in Fig. 2. After the process, we dispose candidates whose height (distance from court plane) is less than $0.05\ m$

Figure 2. Ball detection in a depth image. White curve is a reconstructed 2D trajectory which defines the seeking range (the rectangle in image). Red segment was selected as a ball candidate, blue segment was rejected.

or greater than *1 m* in order to lessen the number of false positives. Finally, we can obtain some 3D positions of ball.

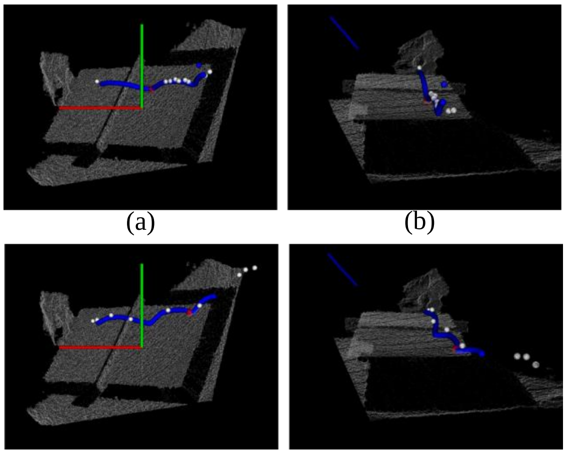
3.3 3D Trajectory Reconstruction

3D trajectories are reconstructed through 2 process; estimating the trajectory plane (the plane where ball travels) and projecting 2D trajectory to the plane. This way can reconstruct 3D trajectory, even if there are small amount of 3D positions of ball.

Trajectory plane is estimated using 2 ball positions and one bounce positions. Bounce position is estimated by seeking the point at which the ball direction changed from downward to upward against moving direction of 2D trajectory. 2 ball positions are selected based on the sum of distances between the trajectory plane reconstructed by the 3 positions (2 selected candidates and 1 bounce position) and every unselected 3D ball positions. Finally, we obtain 3D trajectory by projecting 2D trajectory to the plane.

3.4 Features and Videos Feedback

We can obtain bounce position, maximum height and velocity of balls from reconstructed 3D trajectory. Maximum height is defined as the maximum distance from court plane while a ball travels above the court. Velocity is calculated by averaging the velocities of a ball while a ball travels above the court. Video is provided after the 3D trajectory reconstruction. We defined the length of the video as 2 seconds, from 1.5 sec before the second bounce time to 0.5 sec after the second bounce time. This is empirically defined.



(a) (b) (c) (d)

Figure 3. Reconstructed 3D trajectory. (a) and (b), (c) and (d) are the same trajectories respectively, but different view point.

4. Experiment

We performed experiment to verify that the system can reconstruct 3D trajectory of services and provide bounce position, maximum height and velocity of a ball. The analysis subjects were 2 types of services performed by experienced table tennis players. 3 shots were performed on each type of service. The resolution of RGB and depth images were 640 x 480, frame rate was 30 fps.

Fig. 3 shows the reconstructed 3D trajectory of 2 trial services. 5 out of 6 trial shots were successfully reconstructed, like Fig.3 (a), (b). Unfortunately, quantitative evaluation about the accuracy of results cannot be done in the experiments, but we can expect the information provided by the trajectory which can be used for training purpose. However, 1 out of 6 trial shots was unsuccessfully reconstructed, as we can see in Fig.3 (c), (d). The reason of the fail in the reconstruction is that the approximation of whole trajectory onto one plane was inappropriate when a

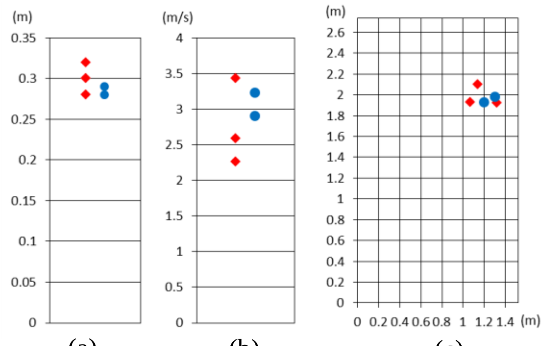


Figure 4. Features of five trial services. (a): Maximum height, (b): Velocity, (c): Bounced position. Red quadrangles are one kind of service, blue circles are another.

ball curves by its fast spin. We fitted only one trajectory plane for each entire service, because we were afraid that Kinect cannot obtain enough amounts of 3D positions to reconstruct multiple trajectory planes. We may solve this problem by fitting different planes for different bouncing trajectories.

Fig. 4 shows the features of five trial services except one service that was reconstructed unsuccessfully. We can evaluate services based on the data. For example, if an athlete has a technical challenge to shot 2 different spins with similar course or trajectory, we can check if there are some differences in the maximum height or bounced position. In addition, since the system provides videos, we can also evaluate the serving form as well as conventional method.

5. Conclusion

A training support system which supports table tennis service training was developed. The experiments showed the system can provide useful data and videos for service training.

There is a limitation of the system. The system cannot reconstruct trajectory of fast moving or fast rotating balls. For estimation of the trajectory plane, we need several positions of ball in a trajectory. If ball moves fast, the number of detected positions would be too small. If ball rotates fast, we need more amounts of 3D positions of ball because we should fit different planes for different bouncing trajectories. In the experiment, we could not achieve reconstructing multiple trajectory planes. The problem may be solved by using other RGB-D camera (Kinect-like device) whose frame rate is higher than Kinect's one, because the problem is just the number of 3D positions. We will consider other devices for the future system.

Acknowledgement

This work is supported by National Institute of Information and Communications Technology (NICT), Japan.

Reference

[1] Dana H. Ballard, Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition, 13(2), 111-122, 1981.

[2] Fei Yan, Alexey Kostin, William Christmas and Josef Kittler, A Novel Data Association Algorithm for Object Tracking in Clutter with Application to Tennis Video Analysis. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 634-641, 2006.

[3] Kourosh Khoshelham and Sander Oude Elberink, Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors, 12(2), 1437-1454, 2012.

[4] M. A. Fischler and R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6), 381-395, 1981.

[5] Smisek Jan, Jancosek Michal and Pajdla Tomas, 3D with Kinect. Computer Vision Workshops (ICCV Workshops), 1154-1160, 2011.