Ping-Pong Robotics with High-Speed Vision System

Li, Hailing; Wu, Haiyan; Lou, Lei; Kühnlenz, Kolja; Ravn, Ole

Published in:
12th International Conference on Control Automation Robotics & Vision (ICARCV), 2012

Link to article, DOI:
10.1109/ICARCV.2012.6485142

Publication date:
2012

Citation (APA):
Li, H., Wu, H., Lou, L., Kühnlenz, K., & Ravn, O. (2012). Ping-Pong Robotics with High-Speed Vision System. In *12th International Conference on Control Automation Robotics & Vision (ICARCV), 2012* (pp. 106 - 111). IEEE. https://doi.org/10.1109/ICARCV.2012.6485142

Ping-Pong Robotics with High-Speed Vision System

Hailing Li[1], Haiyan Wu[2], Lei Lou[1], Kolja Kühnlenz[1] and Ole Ravn[2]

[1]Institute of Automatic Control Engineering(LSR), Technische Universität München, Arcisstrasse 21, 80333 Munich, Germany
[2]Automation and Control, Department of Electrical Engineering, Elektrovej, building 326, 2800,Kgs. Lyngby, Denmark
Email: hailey0525@hotmail.com, hwua@elektro.dtu.dk, lou@tum.de, kolja.kuehnlenz@ieee.org, or@elektro.dtu.dk

***Abstract*—The performance of vision-based control is usually limited by the low sampling rate of the visual feedback. We address Ping-Pong robotics as a widely studied example which requires high-speed vision for highly dynamic motion control. In order to detect a flying ball accurately and robustly, a multi-threshold segmentation algorithm is applied in a stereo-vision running at 150Hz. Based on the estimated 3D ball positions, a novel two-phase trajectory prediction is exploited to determine the hitting position. Benefiting from the high-speed visual feedback, the hitting position and thus the motion planning of the manipulator are updated iteratively with decreasing error. Experiments are conducted on a 7 degrees of freedom humanoid robot arm. A successful Ping-Pong playing between the robot arm and human is achieved with a high successful rate of 88%.**

I. INTRODUCTION

Robotic table tennis has been fascinating the researchers all over the world since 1980s, as a challenging task both in machine vision and control fields. This task requires robust vision system and precise control of manipulator with fast movement to fulfil interaction with human. Robotic table tennis also servers as an ideal platform for different research directions, such as motion learning[1][2], strategy design for juggling[3] and wall bouncing[4] and so on.

For this attractive and challenging task, robust detection of table tennis ball is the basic but crucial step. As introduced in [5]-[9], most of the vision systems ran at low sampling rate, which leads to inaccuracy of ball trajectory prediction results and long image processing delay. Then, both influence the control performance of the robot system. Dedicated hardware may compensate for the deterioration of the control performance, but it increases complexity of design. Acosta et al. [5] employed a mono-camera running at 25 fps (frames per second). Sabzevari et al. [6] and Lampert et al. [7] both developed stereo-vision system working at 60 Hz and employed only color feather to detect the ball. The segmentation only based on color feather may lead to wrong detection, because ball-like objects with similar color, e.g. part of hands or face of human player, are easily to be recognized as the orange ball. Zhang et al. [8] constructed a stereovision system with a higher sample frequency of 100 fps. To detect the flying ball, it only adopted the adjacent frame difference of pixel gray value. But this segmentation method might often obtain incomplete ball contours, even it could not discern the table tennis ball when the ball flies slow. To construct a high-speed vision system, both high-speed cameras and efficient accurate ball detection algorithm are required.

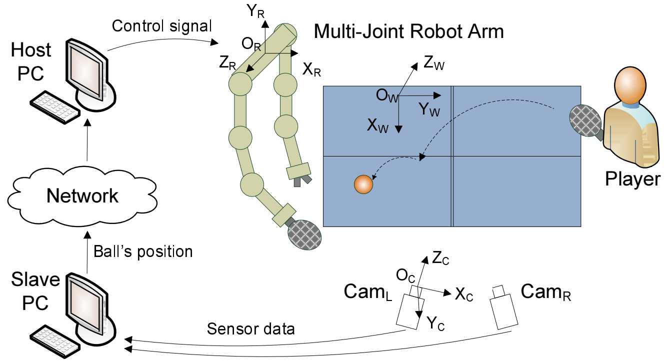

Fig. 1. System overview

In order to determine the hitting position, a trajectory prediction is required after obtaining adequate measured ball positions. Zhang et al. [8] utilized the iteration of dynamics model of the flying ball and collision model. However, the analysis of spinning ball was not involved. Miyazaki et al.[10] applied polynomial regression method to estimate a trajectory of the flying ball. Mülling et al. [11] exploited the extended Kalman filter, however, the velocity changes after bounce were only considered in vertical direction, which would result in unnegligible errors in two horizontal directions.

With regard to the control of the manipulator, the paddle fixed on the linear axis was mostly adopted as multi-DOF playing robot, e.g. [6], [8], [10], etc. The humanoid TOPIO (TOSY Ping Pong Playing Robot) in [12] has been developed since 2005 to realize playing table tennis against human beings, but it applies a sophisticated artificial intelligence system to learn playing skills. Two humanoid table tennis robots are developed at Zhejiang University[13], which are made of high-strength lightweight materials. The robot arm may be easier to be controlled for the stroke motion. Lai et al. [14] used a humanoid robot with 7-DOF arms, which learns ping pong skills with motion capture system to strike the coming ball. Mülling et al. [11] employed also a 7-DOF robot arm to return the ball to the opponent's court. However, in most studies the ball trajectories after bouncing off the table are not considered in trajectory prediction.

In this paper, a novel table tennis robot with a high-speed vision system is presented. The system structure is shown in Fig .1. The whole system consists mainly of two parts: a vision system with two high-speed cameras to capture and transmit the images to a slave PC, and an universal 7-DOF robot arm which has large inertia and is controlled by a host PC.

Benefiting from the high-speed vision system, the flying ball can be detected and localized efficiently in terms of the accuracy and sampling rate. In addition, the ball trajectory is estimated and updated successively based on the image processing results. Then, an appropriate stroke motion is generated to return the ball to the opponent's table. With an adaptive path planning, the robot arm can also adjust its swing to the new trajectory caused by unanticipated changes in the ball trajectory at the bounce point or the ball's spin.

The reminder of this work is organized as follows: The high-speed vision system for accurate ball detection and 3D localization is introduced in Section II. A novel two-phase ball trajectory prediction method is proposed in Section III. In Section IV an adaptive stroke motion generation for the robot arm is presented. The system configuration and evaluation of this ping-pong robot's performance are given with the experimental results in Section V.

II. HIGH-SPEED VISION SYSTEM

In order to drive the robot arm to hit the ball at the correct position and time, a well-predicted trajectory of the flying ball is necessary. However, the aerodynamic model of the flying ball is complex and sensitive to environmental noises. Therefore, more samplings of the fly trajectory based on the visual sensor are demanded for high-performance trajectory prediction. In this section, the novel high-speed image processing algorithm for ball tracking is presented.

A. Image Processing

The difficulties of image processing for the high-speed vision system are the processing speed and accuracy of ball detection, because the table tennis ball is small and easily disturbed by varying lighting and objects with approximate color. Consequently a multi-threshold segmentation, ball determination and precise data synchronization are especially designed for the high-speed vision system. The multi-threshold segmentation with the cooperation of motion detection method and thresholds of HSV color model and shape feathers is described as in Fig. 2.

1) Motion detection: For the motion detection, the very first frame without ball is defined as the base frame. In order to extract the moving objects from the background, a proper threshold $Diff_{Thres}$ is configured for the absolute difference of gray value of each pixel. This motion detection method is formulated as following

$$g_{diff}(x,y) = g_c(x,y) - g_b(x,y),$$

$$diff(x,y) = \begin{cases} 1, & if\,|g_{diff}(x,y)| \geq Diff_{Thres}, \\ 0, & otherwise, \end{cases}$$

where $g_c(x,y)$ and $g_b(x,y)$ are the gray value of pixel(x,y) in the current frame and base frame respectively, and $diff(x,y)$ is the binary value of the pixel(x,y) after the frame difference threshold.

Comparing to adjacent frame difference, this method wouldn't cause the partly missing of the ball, even if the ball flies relatively slowly.

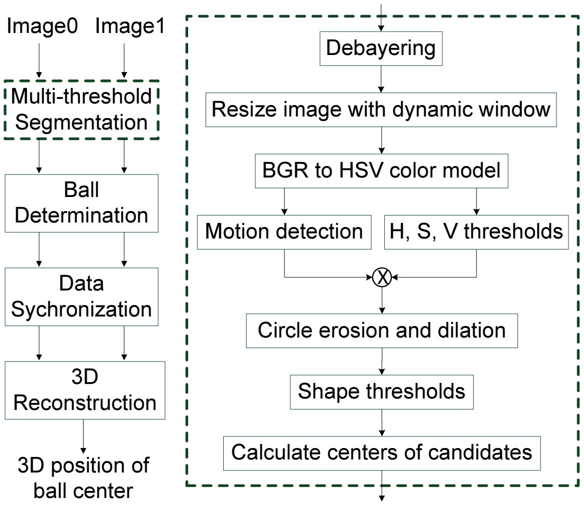

Fig. 2. Diagram of vision processing with multi-threshold segmentation

2) Threshold using color: Only using the frame difference threshold can not detect the ball among the images, because of the unstable lighting in the laboratory. Therefore, the color specification of the table tennis ball is exploited. Six range thresholds Hue_m, Hue_M, Sat_m, Sat_M, Val_m and Val_M are selected for the Hue, Saturation and Value of the HSV color model[15] according to the H, S, V features of uniform orange table tennis ball.

3) Threshold using shape features: Only the pixels satisfying the requirements in step 1) and 2) are considered and their consisting convex contours are then filtered based on the contour radius and area features described as

$$r_c \geq Radius_m,$$

$$Area_m \leq a_c \leq Area_M,$$

where r_c and a_c are the radius and the area of minimal enclosing circle of the segmented contour in pixels. $Radius_m$, $Area_m$ and $Area_M$ are the shape feature thresholds for the minimal radius, minimal and maximal area boundary conditions.

Moreover, a ball-like feature roundness is applied for the segmentation and following ball determination, which is defined as

$$round_c = \frac{a_c}{\pi r_c^2},$$

$$round_c \geq Round_m,$$

where $round_c$, $0 < round_c \leq 1$ is denoted as the roundness of segmented contour. It is a circle if $round_c = 1$. $Round_m$ is the selected minimal roundness value.

These three steps contribute to the improved robustness of the ball detection. The values of these thresholds can be tuned online for different environment or the tracking of other ball-like objects.

4) Dynamic window with ball movement estimation: To accelerate image processing, a dynamic window with the size of $WIN_W \times WIN_H$ is exploited. Once the ball is found in the previous frame, the dynamic window will be set in the current frame plane (frame i) and its center is defined as

$$c_i = b_{i-1},$$

where c_i denotes the dynamic window's center in the current frame, while b_{i-1} is the detected ball center in the previous frame.

When the ball is found both in the previous and current frames, the window's center in the next frame is determined based both on the current ball center b_i and the motion estimation in image plane, i.e. the distance d_i between the current and previous ball centers b_i and b_{i-1}

$$d_i = b_i - b_{i-1},$$

$$c_{i+1} = b_i + d_i.$$

If no ball is detected within the dynamic window of the current frame, the whole frame of image is processed for the next frame, which results in a dynamic switching between the dynamic window and the whole image frame.

5) Ball Determination: The convex contours through the above geometric feature filters are defined as the candidates of the ball. The ball's candidate with largest roundness is determined as the table tennis ball in current frame, and its minimal enclosing circle center will be used for further 3D ball position localization.

B. Parallel Computation

1) Software Trigger: In order to compute the 3D position of the ball, the 2D ball positions and capturing time stamps from two cameras should be synchronized. To decrease the estimation error caused by asynchronization of two cameras, the software trigger[16] is implemented for our vision system. The software trigger sends the capture commands to the two cameras at the same time. Since the shutter time and gain feature parameters of the two cameras are configured identically, the two frame images are captured at almost the same time and the trigger time is chosen as the time stamp.

2) Multi-Thread: For parallel image capturing and image processing multi-thread is exploited, as shown in Fig. 3. In thread 1 the clock-nanosleep function, which is required running under Linux RT Patch Kernel, is applied to trigger both cameras at a designed frequency. Thread 2 and 4 take charge of reading and saving the captured images from frame buffer, then inform thread 3 and 5 to process these two images respectively by using the boost interprocess message-queue[17]. The time interval from triggering left/right camera till finishing saving the relevant image is defined as τr_i /$\tau r'_i$, which is less than the soft trigger period.

As shown in thread 3 and 5, τp_i and $\tau p'_i$ are the image processing time. By using the dynamic window, the image processing is completed before the arrive of the next message. However, processing the whole image exceeds the trigger interval. In this case, the processing thread continues image processing while the next image is ignored, as shown in thread 5 at the time instant $t_{k+1} + \tau r'_{k+1}$. The latency of every visual result is τr_i + τp_i, which needs to be compensated in control system.

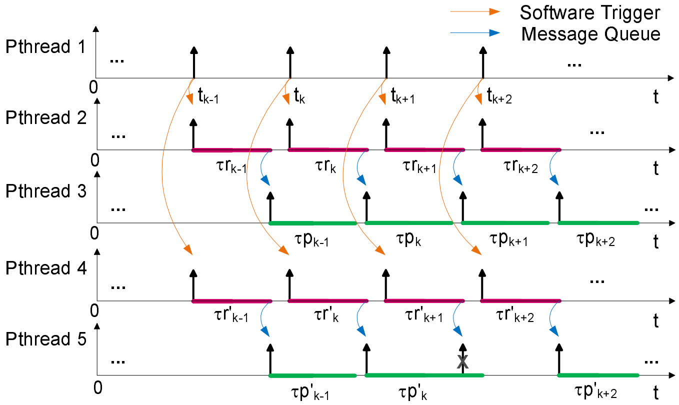

Fig. 3. Parallel processing of vision system

C. 3D Localization

To determine the 3D position of detected ball center in world coordinate and eliminate the distortion of captured images, the calibration of the camera system is required. With the synchronized measured data from two high-speed cameras, the 3D reconstruction is implemented by stereo camera calibration[18].

For the convenience of following ball's trajectory prediction, one point on the table surface is selected as the origin of the world coordinate, as shown in Fig .1.

III. BALL TRAJECTORY PREDICTION

According to the aerodynamics model of flying ball[19], the magnus force can not be neglected if the ball is spinning fast. However, the angular velocity of ball is extremely difficult to be measured.

In ball matches, players are suggested to keep their eyes on the ball until hitting it[20]. Pursuing the ball enhances a player's dynamic visual accuracy and helps to adjust the stroke motion by using cues, such as the unforeseen ball's movement affected by the ball's spin. Therefore, a novel ball trajectory prediction is proposed, which is based on frequent updating of measured ball positions, curve fitting method and bouncing model. For one stroke motion, the proposed ball trajectory prediction is comprised of two phases, which are introduced in the following.

A. Phase I: rough prediction before ball bouncing

For the ball trajectory prediction a curve fitting method is applied with N measured data. A second order polynomial curve fitting is exploited for z direction(gravity direction), and the first order polynomial curve fittings are appropriate for the trajectory prediction in x and y direction (horizontal plane)

$$x = a_x t + b_x, \quad y = a_y t + b_y, \quad z = a_z t^2 + b_z t + c_z.$$

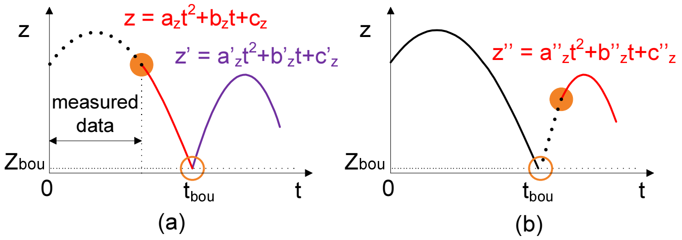

Fig. 4. (a)phase I: rough prediction with bouncing model before ball bouncing; (b)phase II: precise prediction after ball bouncing, the red curve z" is the predicted ball trajectory after bouncing

A collision model is required to compute the new polynomial coefficients of the position curves after the ball bounces on the table. Then the ball trajectory after bouncing can be predicted.

For z direction, the polynomial coefficients $[a'_z, b'_z, c'_z]$ of the estimated trajectory of the ball flight after bouncing, as shown in Fig. 4(a), can be calculated by following equations:

$$\begin{cases} a'_z t_{bou}^2 + b'_z t_{bou} + c'_z = Z_{bou}, \\ 2a'_z t_{bou} + b'_z = K_z(2a_z t_{bou} + b_z), \\ 2a'_z = -g, \end{cases}$$

where t_{bou} and Z_{bou} are the predicted bouncing time and z position of the ball on the table, and K_z is the parameter for ratio of ball velocities after and before collision in z direction. The equations for coefficients calculation in x and y direction are similar.

B. Phase II: precise prediction after ball bouncing

The first reasonable prediction results in Phase I enables the stroke motion of the robot arm. By means of the high-speed vision system, the prediction results are updated and thus utilized to adjust the motion trajectory of the robot arm. In order to improve the performance of trajectory prediction, the visual feedback after the ball bounced off the table is applied for a second-phase trajectory prediction. For example, the coefficients of the second-order polynomial in z direction after bouncing are re-calculated. This prediction phase in z direction is illustrated as in Fig. 4(b).

The advantage of the proposed trajectory prediction method is that only two simple steps of standard curve fitting methods are required. Compared with the algorithm presented in other works, such as iteration of dynamics model or extended Kalman filter without considering Magnus force, the proposed approach gives reliable prediction results in a simplified manner benefiting from the high-speed visual feedback.

IV. STROKE MOTION GENERATION

A. Hitting Point Determination

With the polynomial coefficients of the predicted ball flight trajectory after bouncing in three directions, the hitting point is determined by the shortest Euclidean distance from the trajectory in the configured hitting range of robot arm to the racket center at the robot arm initial position.

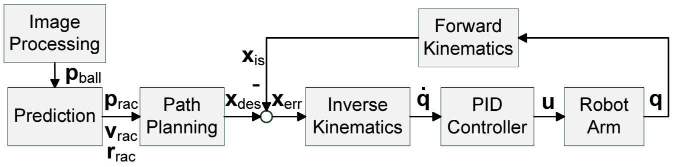

Fig. 5. Control diagram of humanoid 7-DOF robot arm

The velocity of the flying ball at hitting position is derived from the coefficients of predicted trajectory in three directions at the hitting time.

B. Racket Parameters Determination

With the prediction results of ball position and velocity at aimed hitting position, the required position, velocity and rotation of racket fixed on the robot arm are determined to return the ball to the opponent's court.

The y coordinate value of ball flying above the ball net is a constant. The height of ball at this moment and the aimed ball landing point at the opponent's court are freely chosen in possible situations. In this return flight only the gravity effect is considered, so that the required velocity of outgoing ball after hitting (vo_x, vo_y, vo_z) can be obtained by five nonlinear equations. The orientation of the racket at the hitting moment is specified by a normal vector of the racket $\boldsymbol{n}_{rac}$ = [n_{rx}, n_{ry}, n_{rz}][11].

C. Robot Arm Path Planning

With above racket parameters, we plan the trajectory of racket center in Cartesian space instead of in joint space, so that the trajectory is able to be recalculated and modified.

The path planning of racket center is implemented in robot coordinate, whose origin is on the robot arm base, consequently the desired racket center position $\boldsymbol{p}_{rac}$ = $[x_{hit}, y_{hit}, z_{hit}]$, velocity $\boldsymbol{v}_{rac}$ and norm vector $\boldsymbol{n}_{rac}$ for ball hitting are required to be translated from the world coordinate to the robot base's coordinate.

The Z-Y-X Euler angles (α, β, γ) are applied to denote the rotation of the racket norm along the three moving axis from initial to desired posture. The Euler angles are computed from the desired norm vector by

$$\alpha = 0,$$

$$\beta = Atan2(-n_{rx}, -n_{rz}),$$

$$\gamma = Atan2(n_{ry}, \sqrt{n_{rx}^2 + n_{rz}^2}).$$

As illustrated in Fig. 5, the desired racket position $\boldsymbol{p}_{rac}$, velocity $\boldsymbol{v}_{rac}$ and rotation vectors $\boldsymbol{r}_{rac}$ = [α, β, γ] for hitting the ball are exploited as the inputs of the path planning.

To create the manipulator's trajectory, six polynomials X_k = $\sum_{j=0}^{5} a_{kj}t^j$, $k = 1, \ldots, 6$, are used to represent the desired curves of the racket's position factors (x_{des}, y_{des}, z_{des}) and rotations factors (α_{des}, β_{des}, γ_{des}). And the polynomial coefficients vector $\vec{a}_k$ can be calculated through the boundary conditions of the racket position/rotated angles, velocities/angle velocities

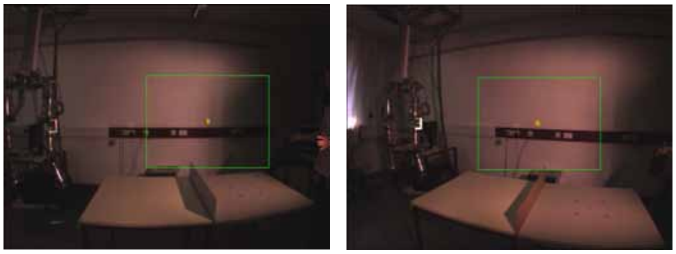

Fig. 6. Image processing results of frames from left and right cameras

and accelerations/angle acceleration at current and desired hitting time points.

After the racket reaches the hitting point at desired hitting time point t_{hit}, the racket will return to the initial position along a smooth replanned trajectory.

V. EXPERIMENTS AND EVALUATIONS

A. *Experimental Setup*

Two Dragonfly Express Cameras with wide angle lens are fixed on a camera stand consisting a stereo camera system. The software trigger is exploited to trigger these two cameras @150Hz to capture the images (640×480 pixels) simultaneously.

A 7-DOF humanoid robot arm[21] is utilized for fulfilling the appropriate stroke motion. The racket fixed on the end effector of robot arm is a round armor plate (radius 7.5 cm). The coordinates translation among the camera, table and robot arm base are illustrated as in Fig. 1.

The slave PC for capturing and image processing has 8GB RAM, and is connected to two cameras with the IEEE 1394b cables. The ADLINK PCIe-FIW64 card is exploited for multiple 1394b device connections with data transfer rates up to 3.2 Gb/s. The Host PC has Intel Q6600 Quad CPU @2.40GHz with 4GB RAM is used for ball trajectory prediction and robot arm control.

B. *Evaluations*

1) Ball Detection Results: To distinguish the flying orange table tennis ball from the background, the multi-threshold segmentation is implemented with the help of a 90×120 pixels dynamic window, which are denoted by the green rectangles as illustrated in Fig .6. The ball centers in the images of two cameras are indicated by the green points. For 500 pictures containing table tennis ball from left camera, the ball detection successful rate is 96.84%, while for the right one the successful rate is 98.98%.

Benefiting from the efficient ball detection algorithms, all of the image processing are easily implemented on CPU rather than on GPU. The overall latency consisting of the image capturing, processing and the network transmission is up to 15ms. And this latency is compensated in the path planning of the robot arm control.

2) Hitting Position Prediction: The measured ball trajectory, the predicted hitting positions, and time points for one stroke motion are illustrated in Fig .7.

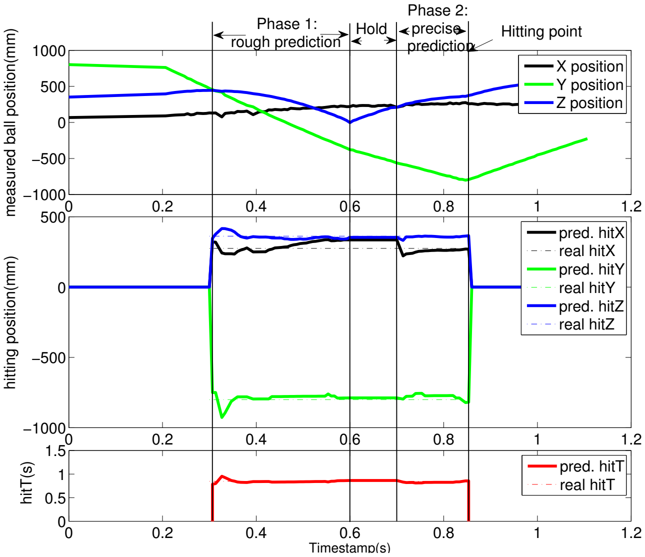

Fig. 7. Comparison of predicted hitting point with measured data

The first valid prediction results are available about 0.1 s after the ball flying towards the robot arm. In the prediction phase I, the prediction errors of hitting position and time are less than 5 cm and 0.1 s. When the ball bounces off the table, the prediction will restart and the prediction results in phase II are more accurate with maximal error 3 cm and 0.02 s. Although the initial results of predicted hitting position are not so accurate, these results enables the movement of the robot arm towards the hitting position before the ball flies over the net. With the iteratively updated prediction results, the humanoid robot arm continuously adjusts its strike position until the hitting time.

3) Robot Arm Control: The control error is calculated in Cartesian space, and then transformed into joint space through inverse kinematics. The PID controller is exploited to control the robot arm to playing table tennis. Fig. 8 shows the desired and real racket trajectory comparing with measured ball trajectory in x, y, z directions. The robot arm follows closely the desired trajectory and the ball is successfully returned to the opponent's court. In addition, it is observed that the measured ball position in x direction has larger noise than that in y and z directions due to the unavoidable stereo calibration error in the main axis (x direction) of cameras. The difference between the measured ball position and racket center at hitting instant is calculated for performance evaluation and the results are shown in Tab. 1. The mean difference is about 2.2cm, which is relatively small and allows a successful stroke.

The success rate of robot arm to hit and return the ball on the table in 100 trials is implemented in the experiments. The ratio of successfully return the ball to the opponent's court is 88%, which demonstrates a good performance achieved by the proposed two-phase strategy for trajectory prediction and the adaptive path planning algorithm based on high-speed visual feedback.

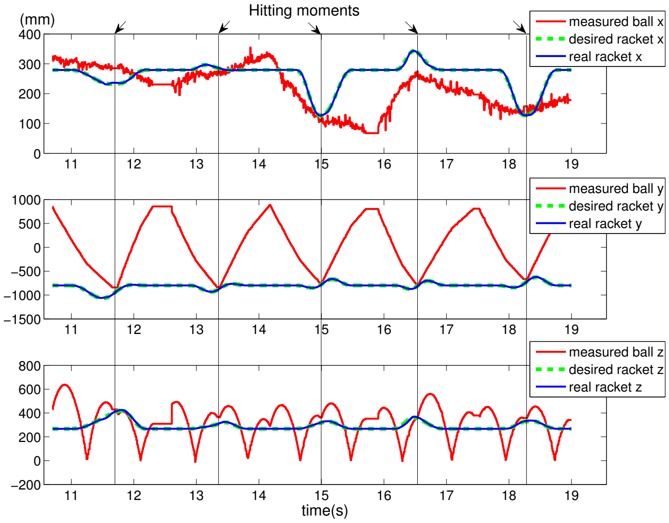

Fig. 8. Real and desired movement of the racket comparing with ball's flight trajectory in world coordinate for five sequential successful strikings

TABLE I
BALL VS. RACKET CENTER POSITIONS AT THE HITTING TIME IN WORLD COORDINATE

Index	Ball Position(x,y,z) at hitting time(mm)	Racket center(x,y,z) at hitting time(mm)	Errors (mm)
1	274.9, -668.6, 332.3	258.5, -679.2, 336.8	20.04
2	278.8, -647.7, 326.5	278.1, -662.2, 315.7	18.09
3	417.6, -666.6, 253.9	411.0, -676.1, 332.1	12.56
4	322.6, -657.0, 253.9	301.3, -678.1, 258.7	30.36
5	282.7, -671.0, 291.0	288.1, -694.1, 296.0	24.24
6	171.1, -699.5, 369.3	156.1, -720.1, 353.5	29.98
7	347.3, -675.8, 369.9	359.7, -695.1, 384.0	26.93
8	453.0, -649.4, 326.6	445.3, -653.3, 321.4	10.08
9	406.1, -648.2, 363.0	407.8, -653.6, 353.8	10.80
10	333.7, -632.5, 301.1	304.7, -659.1, 311.8	40.81

VI. CONCLUSIONS AND FUTURE WORK

In order to improve the performance of visual perception and stroke motion control in Ping-Pong robotics, a novel two-phase trajectory prediction algorithm based on high-speed visual feedback is proposed. A multi-thread segmentation algorithm is applied to detect the ball in images provided by a stereo vision system running at 150 Hz. The developed high-speed vision system achieves a good performance with a successful rate of 96% for ball detection. In addition, an adaptive path planning algorithm is designed for robot motion control based on the contentiously updated ball position. In particular the ball positions after bouncing off the table are utilized for motion estimation and path planning. Therefore, the error in the determinations of hitting position and hitting instant is efficient reduced, which enables stokes with high successful rate.

For the future work, the improvement of the playing skill and strategy of the table tennis robot can be considered.

ACKNOWLEDGMENT

This work is supported in part by the DFG excellence initiative research cluster *Cognition for Technical Systems - CoTeSys*, see also *www.cotesys.org*, the BMBF Bernstein Center for Computational Neuroscience Munich, see also *www.bccn-munich.de*, and the Institute for Advanced Study (IAS), Technische Universität München, see also *www.tum-ias.de*.

REFERENCES

[1] M. Matsushima, T. Hashimoto, M. Takeuchi and F. Miyazaki, "A Learning Approach to Robotic Table Tennis", *in Robotics, IEEE Transactions on*, vol. 21, 2005, pp 767-771.

[2] J. Kober, K. Muelling, O. Kroemer, C.H. Lampert, B. Schoelkopf and J. Peters, "Movement Templates for Learning of Hitting and Batting", *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.

[3] Nakashima, Akira, Kobayashi, Yosuke, Hayakawa, Yoshikazu, "Paddle Juggling of one Ball by Robot Manipulator with Visual Servo", *in Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[4] M. Takeuchi, F. Miyazaki, M. Matsushima, M. Kawatani, T. Hashimoto, "Dynamic dexterity for the performance of "wall-bouncing" tasks", *in Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, 2002, pp 1559.

[5] Acosta, L. and Rodrigo, J.J. and Mendez, J.A. and Marichal, G.N. and Sigut, M., "Ping-pong player prototype", *IEEE Robotics and Automation Magazine*, vol. 10, 2003, pp 44-52.

[6] Sabzevari, R. and Shahri, A. and Fasih, A. R. and Masoumzadeh, S. and Ghahroudi, M.R., "Object detection and localization system based on neural networks for Robo-Pong", *in Proceedings of the 5th International Symposium on Mechatronics and Its Applications*, 2008, pp 1-6.

[7] Christoph H. Lampert, Jan Peters, "Real-Time Detection of Colored Objects In Multiple Camera Streams With Off-the-Shelf Hardware Components", *Journal of Real-Time Image Processing*, vol. 7, 2010, pp 31-41.

[8] Zhang, Z. and Xu, D. and Tan, M., "Visual Measurement and Prediction of Ball Trajectory for Table Tennis Robot", *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, 2010, pp 3195-3205.

[9] H. Wu, L. Lou, C.-C. Chen, S. Hirche, K. Kühnlenz, "A framework of networked visual servo control system with distributed computation", *in the Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision*, 2010, pp 1466-1471.

[10] F. Miyazaki, M. Takeuchi, M. Matsushima, T. Kusano, T. Hashimoto, "Realization of the table tennis task based on virtual targets", *in Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, 2002, pp 3844-3849.

[11] K. Mülling, J. Kober, and J. Peters, "Simulating human table tennis with a biomimetic robot setup", *in Proceedings of the 11th international conference on Simulation of adaptive behavior*, 2010.

[12] TOSY, http://topio.tosy.com.

[13] Big Size Humanoid Robots Developed at Zhejiang University, http://www.zju.edu.cn/english/redir.php?catalog_id=279955&object_id =2036949.

[14] Lai, C.H. and Tsay, T.I.J., "Stroke Motion Learning for a Humanoid Robotic Ping-Pong Player Using a Novel Motion Capture System", *Journal of Computer Science*, vol. 6, 2010, pp 946-954.

[15] Wen Chen, Yun Q. Shi, Guorong Xuan, "Identifying Computer Graphics using HSV Color Model and Statistical Moments of Characteristic Functions", *in Proceedings of the IEEE International Conference on Multimedia and Expo*, 2007, pp 1123-1126.

[16] Damien Douxchamps, http://damien.douxchamps.net/ieee1394/libdc1394/v2.x/api/trigger/.

[17] Ion Gaztanaga, http://www.boost.org/doc/libs/1_38_0/doc/html/interprocess.html.

[18] Jean-Yves Bouguet, http://www.vision.caltech.edu/bouguetj/calib_doc/.

[19] Sun, L. and Liu, J. and Wang, Y. and Zhou, L. and Yang, Q. and He, S., "Ball's flight trajectory prediction for table-tennis game by humanoid robot", *in Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2009, pp 2379-2384.

[20] Spering M, Schütz AC, Braun DI, Gegenfurtner KR, "Keep your eyes on the ball: smooth pursuit eye movements enhance prediction of visual motion", *J Neurophysiol 105*, 2011, pp 1756-1767.

[21] B. Stanczyk, "Develpment and Control of an Anthroporphic Telerobotic System", PhD thesis, Technische Universität München, 2006.