J Intell Robot Syst (2011) 64:543–560
DOI 10.1007/s10846-011-9554-8

Short-Baseline Binocular Vision System for a Humanoid Ping-Pong Robot

Jian-Dong Tian · Jing Sun · Yan-Dong Tang

Received: 20 September 2010 / Accepted: 9 February 2011 / Published online: 5 May 2011

Abstract We develop a short-baseline vision system for a humanoid ping-pong robot. The vision system can provide four-dimensional space-time information and can predict the future trajectory of a ball. Short baseline poses special challenges for achieving sufficient 3-D reconstruction and prediction accuracy within limited processing time. We propose two algorithms including direct calibration of projection matrix and Gaussian-fitting based ball-center location to guarantee the 3-D reconstruction accuracy; we propose algorithm of five-point based ball representation and utilize the constraint of ball detecting region to guarantee the processing speed; we also propose algorithm of smoothing-based trajectory prediction to improve the prediction accuracy. Experimental results show the accuracy and the speed of our vision system can meet the requirements of a humanoid ping-pong robot.

Keywords Humanoid robot · Short-baseline vision · Ping-pong ball detection · Trajectory prediction

J.-D. Tian (✉) · J. Sun · Y.-D. Tang
State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, 110016 Shenyang, People's Republic of China
e-mail: tianjd@sia.cn

J. Sun
e-mail: sunjing@sia.cn

Y.-D. Tang
e-mail: ytang@sia.cn

J.-D. Tian · J. Sun
Graduate School of the Chinese Academy of Sciences, 100039 Beijing, People's Republic of China

1 Introduction

In the recent decades, humanoid robotics is becoming an active research area. A ping-pong playing robot, as one of the prototypes of humanoid robot, is often used as a research tool. Therefore, designing such a robot is of broad interest in robotics. Andersson [1] firstly constructed a ping-pong robot capable of playing against humans and machines. The ping-pong robot designed by Miyazaki et al. can play against humans [2] or play against wall [3]. The ping-pong system described in [4, 5] is capable of returning an incoming ball to the opponent's court. Angel et al. [6, 7] describe a position-based visual serving system for a ping-pong robot. Brunnett et al. [8] model the collisions between a ball and a racket in their virtual table tennis platform. Rusdorf et al. [9] report the simulation of an immersive table tennis and the method of collision detection.

For a ping-pong playing robot, vision is an unavoidable component which provides the ball's position at each instant and the prediction of its future trajectory. In the vision system described in [1], four cameras are applied to achieve high accuracy. But synchronization may become difficult when they capture images in high speed. In [2], the stereo vision system named Quick MAG III is used to extract a ball every 1/60 s. However, the system excessively relies on ball's color. The vision system developed in [10] only needs a single CCD video camera, based on detecting a ball and the shadow it projects on the table. This system shows a great simplicity compared with stereoscopic vision systems. However, it requires a light located at a prefixed position over the table to generate ball's shadow, making this method sensitive to illumination variation.

A ball flying over the table only takes about 0.2 s to 0.5 s. Sometimes, a vision system must finish its work in 0.1 s in order that enough time can be left for the robot to make an action plan. To meet such a high speed requirement, most of the time resolutions in current systems are more than 60 Hz. Recently, Zhang et al. [11] developed a high speed vision system with two 250 fps cameras. However, their system is not suitable for a humanoid robot due to the long baseline.

The goal of the vision system is to predict the ball's future trajectory, so ball state including position, velocity, and acceleration must be calculated as accurately as possible. Among these factors the most fundamental one is the ball's position. The accuracy of 3-D reconstruction heavily depends on the placement of cameras. In [12, 13], two cameras are orthogonal placed (one side look and another one over look) to achieve high accuracy. In parallel placement, the accuracy has strong relationship with baseline length, i.e., longer baseline helps to obtain more accurate results [14]. This is the reason why most of the previous systems use long baseline stereo cameras. However, humanoid robots cannot be equipped with long baseline cameras, let along those orthogonal placed. This particular requirement brings a great challenge for system's accuracy.

The baseline lengths of the previous state-of-the-art ping-pong vision systems and our system are tabulated in Table 1. From the comparison one could find that our system owns the shortest baseline, thus may make it more suitable for the humanoid ping-pong robots. The contribution of this paper is that we develop a short baseline stereoscopic vision system for a humanoid ping-pong robot and propose some new methods to guarantee the system's accuracy. In detail, we propose several new algorithms including direct calibration of projection matrix, five-point based ball detection, and Gaussian fitting based ball center location to guarantee 3-D

Table 1 Baseline lengths of previous vision systems and our system

Developers	Ref. [1]	Ref. [2]	Ref. [11]	Ref. [25]	This paper
Baseline lengths	0.5 m	3.57 m	2.0 m	0.6 m	0.18 m

reconstruction accuracy, and we propose the method of smoothing-based trajectory prediction to improve prediction accuracy.

The rest of the paper is organized as follows. In Sections 2, 3, 4, 5 we present our methods of system configuration, projection matrix calibration, ball detection and center location, 3-D reconstruction, respectively. In Section 6, we describe our trajectory prediction method, followed by some experimental results in Section 7. We end this paper with a brief conclusion in Section 8.

2 System Configuration

2.1 Hardware System

Figure 1 illustrates our hardware system configuration. A ball is launched by a tennis serving machine. The ball is detected by the stereo cameras (IMPERX IPX-VGA210) with 0.18 m baseline. The two cameras are placed about 0.4 m behind the end of the table. The frame rates of the cameras are 110 frames per second, and image resolutions are 640 × 480 pixels. The cameras are equipped with two 8 mm focal length lens and are connected via camera link interfaces to an image capture card (DALSA X64-CL) plugged in HP workstation XW 8200. The two cameras are trigged by an external trigger for the synchronization. Both the table and the balls are on the international competition standard (Table: width: 1526 mm, length: 2740 mm, height: 730 mm. Balls: diameter: 40 mm, weight: 2.7 g).

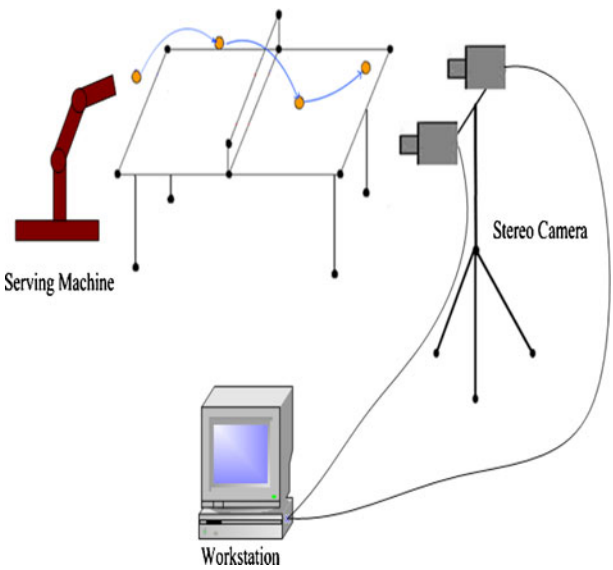


Fig. 1 Schematic diagram of the hardware configuration

2.2 Software System

Figure 2 shows the flowchart of our software system. It begins with detecting a ball in the captured frames. If the ball center in left image and that in right image satisfy the constraint of fundamental matrix, the ball can be reconstructed with the calibration

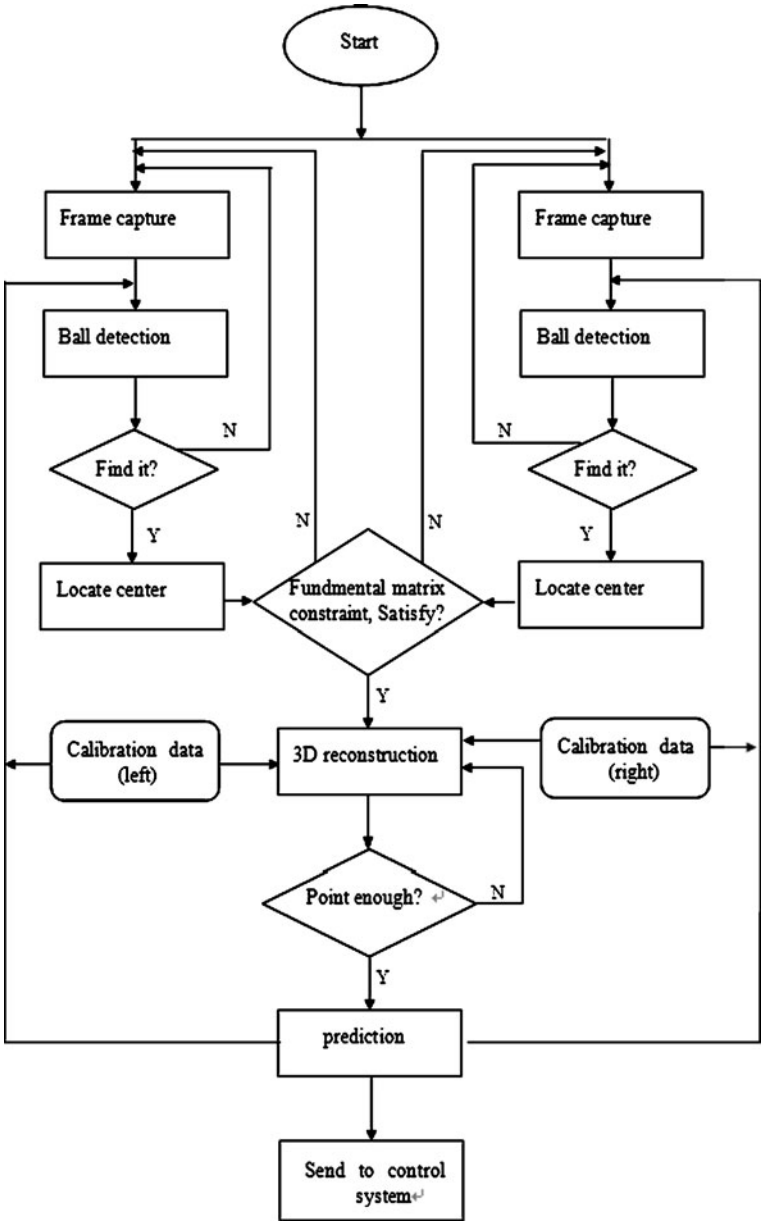


Fig. 2 Software flowchart

data. If they do not satisfy the constraint or the ball is not detected, the current stereo images will be discarded. When the system has gotten enough 3-D reconstructed points (at least two), the ball's future trajectory can be predicted. The predicted data are then sent to the control system to guide the robotic arm to return the ball, and also are fed back to provide information for the future ball detection. With the feedback of the prediction, the ball can be only detected in a window where the ball is expected to appear.

3 Camera Calibration

Camera calibration is necessary to reconstruct 3-D information from 2-D images. Furthermore, reconstruction accuracy highly depends on calibration accuracy. There are many good camera calibration methods; however most of them are designed to calibrate intrinsic parameters and extrinsic parameters of a camera. In contrast, our vision system only requires a projection matrix. Although projection matrix can be obtained through the intrinsic matrix and the extrinsic matrix, the accuracy usually cannot be guaranteed. In this paper, we propose a direct method for projection matrix calibration.

If the homogeneous coordinates of a 2-D point are denoted by $q = [u, v, 1]^T$ and the homogeneous coordinates of the corresponding 3-D point are denoted by $Q = [Z, Y, Z, 1]^T$, 2-D coordinates can be obtained from 3-D coordinates by,

$$sq = MQ \tag{1}$$

where M is a 3×4 projection matrix, and s is an arbitrary scale factor.

Our calibration method originates from [15], but aims to obtain more accurate result and meanwhile keep its easy-use quality. Our model plane and our coordinate system used for calibration are shown in Fig. 3. We first calculate the homography

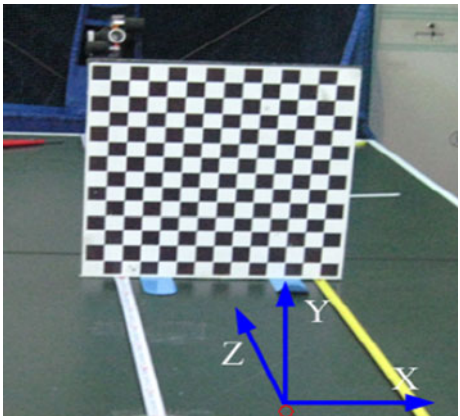


Fig. 3 The calibration model plane and the coordinate system used in our experiment

between the model plane and its image. Without loss of generality, we assume $Z = 0$ when we compute the homography H. Thus we have:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = [m_1\ m_2\ m_4]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = H\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2}$$

The reason we compute homography is that we can extract corners more easily and accurately. The method proposed in [15] is employed to calculate homography; the method proposed in [16] is used to extract corners.

The model plane will be moved along Z axis to generate 3-D points with different depth. Given a set of counterparts of corner coordinates in 2-D images and in 3-D world, the projection matrix can be calculated through the linear algebra methods. The result can be written as:

$$M = (A^T A)^{-1} A^T B \tag{3}$$

where A is a coefficient matrix, and B is a column vector. A and B can be easily gotten by rewriting the projection matrix into 3×4 entries and further by expanding Eq. 1. Usually, matrix A has large condition number, which makes the solution sensitive to noise. In [17], a normalized method is proposed to deal with this problem. Inspired by that, we also preprocess both 2-D and 3-D coordinates. In detail, if we use n points on the plane and the plane is moved with m steps along Z axis, letting $\widehat{q_m} = [q_{m1}, q_{m2}, ...q_{mn}]$ denote the 2-D n points generated by the mth movement and letting $\widetilde{q} = [\widehat{q_1}, \widehat{q_2}, ...\widehat{q_m}]$ denote all 2-D points, the normalized conversion matrix is defined as:

$$\Gamma = \begin{bmatrix} E(\widetilde{q}_u) & 0 & \dfrac{E(\widetilde{q}_u)}{D(\widetilde{q}_u)} \\ 0 & E(\widetilde{q}_v) & \dfrac{E(\widetilde{q}_v)}{D(\widetilde{q}_v)} \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $E(\bullet)$ is mean value calculation and $D(\bullet)$ is variance calculation. For the model plane, letting $\widehat{Q_m} = [Q_{m1}, Q_{m2}, ...Q_{mn}]$ denote the 3-D n points generated by the mth movement and letting $\widetilde{Q} = [\widehat{Q_1}, \widehat{Q_2}, ...\widehat{Q_m}]$ denote all generated 3-D points, the normalized conversion matrix is:

$$\Phi = \begin{bmatrix} E(\widetilde{Q}(X)) & 0 & 0 & \dfrac{E(\widetilde{Q}(X))}{D(\widetilde{Q}(X))} \\ 0 & E(\widetilde{Q}(Y)) & 0 & \dfrac{E(\widetilde{Q}(Y))}{D(\widetilde{Q}(Y))} \\ 0 & 0 & E(\widetilde{Q}(Z)) & \dfrac{E(\widetilde{Q}(Z))}{D(\widetilde{Q}(Z))} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

The data actually used in computation are $\widetilde{q} = \Gamma q$ and $\widetilde{Q} = \Phi Q$. Substituting them into Eq. 1, we have:

$$s\widetilde{q} = \Gamma M \Phi^{-1} \widetilde{Q} \tag{6}$$

So the projection matrix in converted coordinates is $\widetilde{M} = \Gamma M \Phi^{-1}$. Correspondingly, the projection matrix in the original coordinates is,

$$M = \Gamma^{-1} \widetilde{M} \Phi \tag{7}$$

4 Ball Detection and Center Location

In the system, the ball detection and center location method in 2-D images plays an important role. Our ball detection and center location method consist of the following three steps:

- To restrict the ball's searching region;
- To detect the ball by using five-point method;
- To locate ball center by fitting Gaussian function.

4.1 Ball's Searching Region

Ball detection results are easily influenced by the surrounding environments, e.g., person movement. We restrict the searching region into table region to lessen the environment disturbance and meanwhile to shorten the processing time. We use a simple method to locate table edges rather than image processing ones. Because the two cameras have been calibrated, the projected image coordinates of a world point can be calculated. Since two points determine a straight line, we utilize the four corners of the table to determine the two side lines. As shown in Fig. 4, the searching region is restricted to the part between the two lines. When prediction trajectory is

Fig. 4 The extracted table region

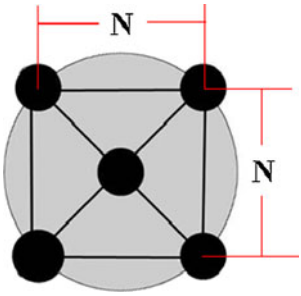


Fig. 5 Five key points are used to present a ping-pong ball

available, the searching region will be further restricted into the window where the ball expected to appear.

4.2 Ball Detection

Ball detection methods have been well studied in vision systems for ball sports such as ping-pong ball [11, 18], volleyball [19, 20], soccer ball [21, 22]. Tong et al. [21] employ an indirect ball detection method by eliminating non-ball regions using color and shape constraints. Yamada et al. [22] take white regions as the ball candidates after removing players and field lines. In our method, as shown in Fig. 5, the five key points are used to describe a ping-pong ball. The five-point representation actually considers both the size and the shape of a ball in images. The method is applied on frame-difference images. Discontinuous frames are used to avoid crescent-shaped region that may appear when ball's flight direction is parallel to camera's optical axis. In this step, the mean value of the difference image is simply chosen as the threshold to generate a binary image. When a pixel value is higher than the threshold, we take it as a possible upper left boundary point of the ball and then judge whether the other four points comply with the rules one by one. Only one parameter N that describes the current size of the ball needs to be determined. The current 3-D ball position determines the 2-D ball size based on the fact that the size of an object in image is inversely proportional to the distance from its location to camera.

4.3 Center Location

The ball center should be located once a ball is detected by the five-point method, and it should be located with high accuracy since the short baseline. Mouhamed et al. [23] compute ball centers using boundary pixels. This method needs a threshold to detect ball boundaries, but the threshold often is sensitive to illumination changes. Some works like [10] simply locate ball center as centroid. The center located by the centroid-based method may be not accurate enough because the brightness of a ball in image is not uniform. Therefore, in this paper, we locate ball centers directly on original images rather than on binarized images. We observe that the brightness of a ball can be well approximated by Gaussian function, as shown in Fig. 6.

We use two-dimensional Gaussian function defined as follows to describe the brightness of a ball.

$$I(u, v) = A \cdot \exp(B \cdot [(u - u_0)^2 + (v - v_0)^2]) \tag{8}$$

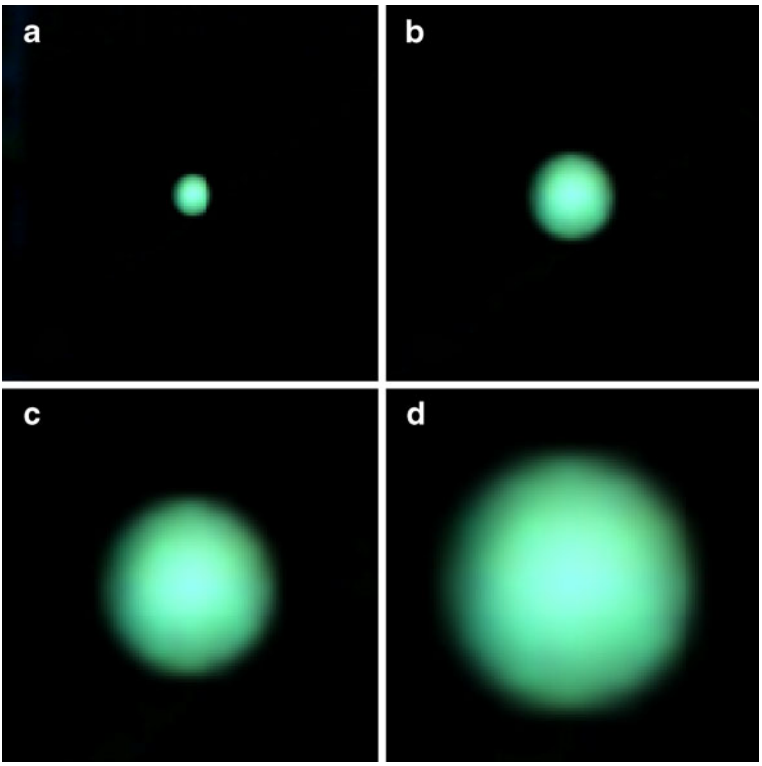


Fig. 6 The brightness of a ball in image is not uniform. It decreases with the distance increasing from the center. In particular, there exist aliasing effects around the ball edge. **a**, **b**, **c**, and **d** show a ball in different scale for close-up view purpose

where u_0, v_0, A, B are the parameters of the function, and particularly (u_0, v_0) is the center.

Taking logarithm of both sides of the equation, we get that:

$$\ln[I(u,v)] = \ln(A) + B(u^2+v^2) + B(u_0^2+v_0^2) - 2Buu_0 - 2Bvv_0 \tag{9}$$

Let:

$$\begin{cases} a_3 = B \\ a_2 = -2Bu_0 \\ a_1 = -2Bv_0 \\ a_0 = B(u_0^2+v_0^2) + \ln(A) \end{cases} \tag{10}$$

Equation 9 can be rewritten as:

$$(u^2+v^2)a_3 + ua_2 + va_1 + a_0 = \ln[I(u,v)] \tag{11}$$

It is a linear equation about $a_0 \sim a_3$. Given a set of counterparts $[(u_i, v_i), I(u_i, v_i)]$, the parameters can be easily solved in the least-squares sense and thereby the ball center can be located.

5 3-D Reconstruction

Letting q^l and q^r denote the homogeneous coordinates of the projected points from a 3-D point Q to the left image and right image respectively, they satisfy the following constraint:

$$(q^r)^T F q^l = 0 \tag{12}$$

where F is the fundamental matrix [17]. The equation cannot be strictly held due to noise. Thus Eq. 12 is usually written as the following approximate formula:

$$(q^r)^T F q^l < \varepsilon \tag{13}$$

where ε is a very small positive value.

Letting q_c^l, q_c^r denote the homogeneous coordinates of located ball centers on the left image and the right image respectively, and letting Q_c denote the homogeneous coordinates of their corresponding 3-D point, if q_c^l and q_c^r satisfy formula (13), based on Eqs. 14 and 15, the 3-D world coordinates of the ball center can be easily solved by the least-squares method.

$$s^l q_c^l = M^l Q_c \tag{14}$$

$$s^r q_c^r = M^r Q_c \tag{15}$$

where M^l and M^r are the projection matrices of the left and right camera, respectively. Only ball centers q_c^l, q_c^r that satisfy formula (13) will be reconstructed, which help to guarantee the accuracy of 3-D reconstruction. If the ball in left or in right image is wrongly detected or is inaccurately center located (though these phenomena rarely happen, they must be considered for robustness of the whole system), it can be diagnosed by formula (13).

Deriving fundamental matrix from projection matrix is described and proved in Appendix.

6 Trajectory Prediction

Obtaining the impact location and the impact time for the paddle to return a ball is most important in performing the table tennis task. For a robot to complete this task, the trajectory of a moving ball must be predicted in advance. We use the physical-based method rather than the regression one [2, 24] or the quadratic fitting one [1] to predict ball's trajectory.

As shown in Fig. 7, when a ball is flying in air, mainly three forces[1] act on it: gravity F_g, buoyant force F_b, and resistance force F_r. Air resistance can be calculated by:

$$F_r = -\frac{1}{2} C \rho S V^2 \tag{16}$$

[1]The order of magnitude of gravity, resistance force, buoyant force, and spin caused magnus force are about 10^{-2}, 10^{-3}, 10^{-4}, and 10^{-5}, respectively. Magnus force is neglected in this paper.

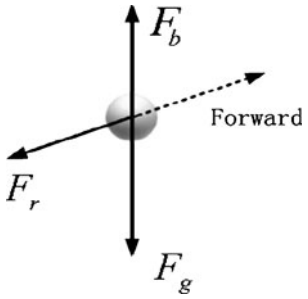


Fig. 7 Force analysis on a flying ball

where C is air resistance coefficient; ρ is air density; S is cross-sectional area of the ball; V is velocity norm. Buoyant force can be calculated by:

$$F_b = \frac{1}{6}\pi\rho d^3 g \tag{17}$$

where d is the ball diameter and g is the acceleration of gravity.

Position, velocity, and acceleration of a ball in the world coordinates are denoted as X, $\dot{\mathrm{X}}$, and $\ddot{\mathrm{X}}$, respectively. All of them are three-dimensional vectors. Letting $k = \frac{C\rho S}{2m}$, the acceleration caused by air resistance is:

$$\ddot{\mathrm{X}}_r = -k \left\|\dot{\mathrm{X}}\right\|_2 \dot{\mathrm{X}} \tag{18}$$

where m denotes ball mass. The motion of the ball in X and Z axes are only influenced by air resistance, but in Y axis, not only the air resistance but also gravity and buoyant force are the influencing factors. The resultant acceleration of the ball can be denoted by:

$$\ddot{\mathrm{X}}_r = -k \left\|\dot{\mathrm{X}}\right\|_2 \dot{\mathrm{X}} - \overrightarrow{g'} \tag{19}$$

where $\overrightarrow{g'} = [0\ \mathrm{g}'\ 0]^T$ and g′ is the acceleration caused by gravity and buoyant force.

From $\begin{cases} \dot{\mathrm{X}} = d\mathrm{X}/dt \\ \ddot{\mathrm{X}} = d\dot{\mathrm{X}}/dt \end{cases}$, we can get:

$$\begin{bmatrix} \mathrm{X} \\ \dot{\mathrm{X}} \end{bmatrix} = \begin{bmatrix} \dot{\mathrm{X}} \\ \ddot{\mathrm{X}} \end{bmatrix} dt \tag{20}$$

Ball trajectory can be iteratively archived by:

$$\begin{bmatrix} \mathrm{X} \\ \dot{\mathrm{X}} \end{bmatrix}_n = \begin{bmatrix} \mathrm{X} \\ \dot{\mathrm{X}} \end{bmatrix}_{n-1} + \begin{bmatrix} \dot{\mathrm{X}} \\ \ddot{\mathrm{X}} \end{bmatrix}_{n-1} dt \tag{21}$$

The corresponding time can be calculated by:

$$T_n = n \cdot dt \tag{22}$$

When the ball rebounds on the table, the velocity vector will be changed as:

$$\dot{\mathrm{X}}' = \eta \dot{\mathrm{X}} \tag{23}$$

where η is the vector of rebound coefficient which is estimated by simulation, and its element equals to 0.8, −0.96, and 0.8 in X, Y, and Z direction, respectively.

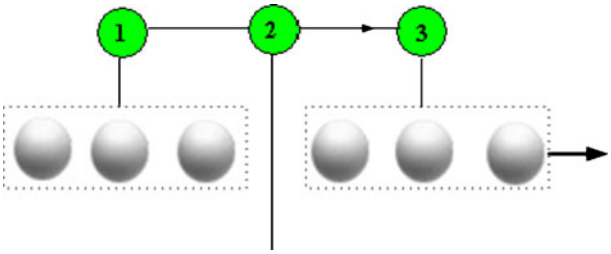


Fig. 8 To smooth ball position. Green point ① is the mean value of the first part; green point ③ is the mean value of the second part; green ball ② is the mean value of ① and ③

We need an initial position and an initial velocity to start the numerical iteration. Using two adjacent reconstructed ball positions to calculate velocity is very sensitive to noise. Small deviations from the real 3-D positions will produce large errors in the calculations of velocity due to the high speed of the ball. A smoothing technique is employed here to address the problem. As shown in Fig. 8, if we have a number of points, we cut it into two parts at middle. The middle green ball ② is chosen as the initial position; green point ① and green point ③ are used to calculate the initial velocity.

Trajectory prediction takes place each time when a new data is available, so that the robot planning can be continually updated.

7 Experiments and Evaluations

7.1 Calibration Results

Figure 9 shows the comparison of our calibration result with that by Zhang's method [15]. In the experiment, the model plane is moved with four steps, at $Z = 200$ mm, $Z = 400$ mm, $Z = 600$ mm, and $Z = 800$ mm. The corners of the model plane at $Z = 1{,}700$ mm are projected onto image to test the calibration accuracy. The experimental results show that our projected corners are of more accuracy than that of Zhang's method, which indicates that our calibration result is more accurate.[2]

7.2 Ball Detection, Center Location, and 3-D Reconstruction Results

Figure 10 shows our ball detection and center location results. It shows that balls are correctly detected and their centers are accurately located. The accuracy of the vision system's static 3-D reconstruction is tested by placing ping-pong balls at measured locations in the workspace. Twenty positions are uniformly chosen on the table, and then over the table at height of 20 cm and 40 cm respectively, i.e., totally 60 positions are used for testing. The mean 3-D reconstruction accuracy is 4 mm.

[2]Zhang's method is a very famous one and is widely used for camera calibration. Zhang's method can be used to calibrate intrinsic matrix and extrinsic matrix while our method is only a specific one for projection matrix calibration.

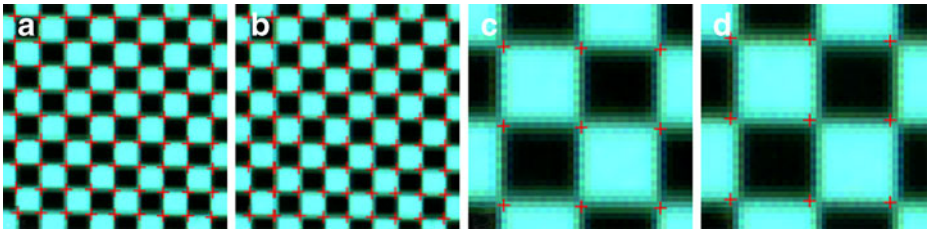


Fig. 9 Comparison of our calibration result with that by Zhang's method proposed in [15]. **a** is our result and **b** is the result obtained by Zhang's method; **c** is the close-up view of **a**; **d** is the close-up view of **b**

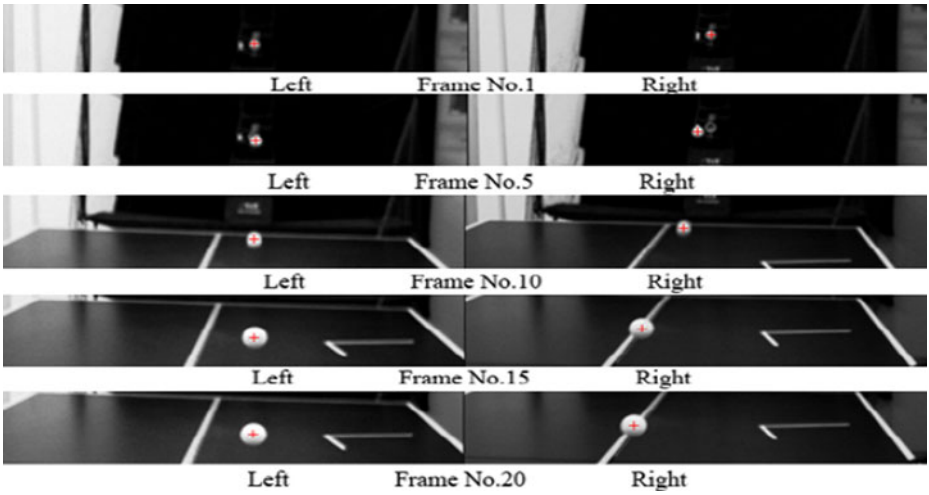
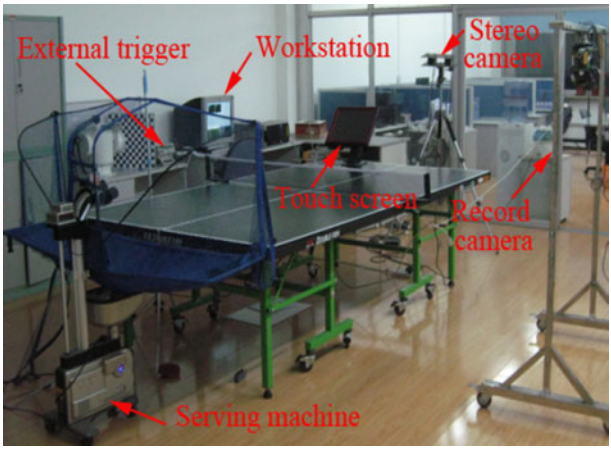


Fig. 10 Ball detection and center location results

Fig. 11 The picture of our experimental configuration



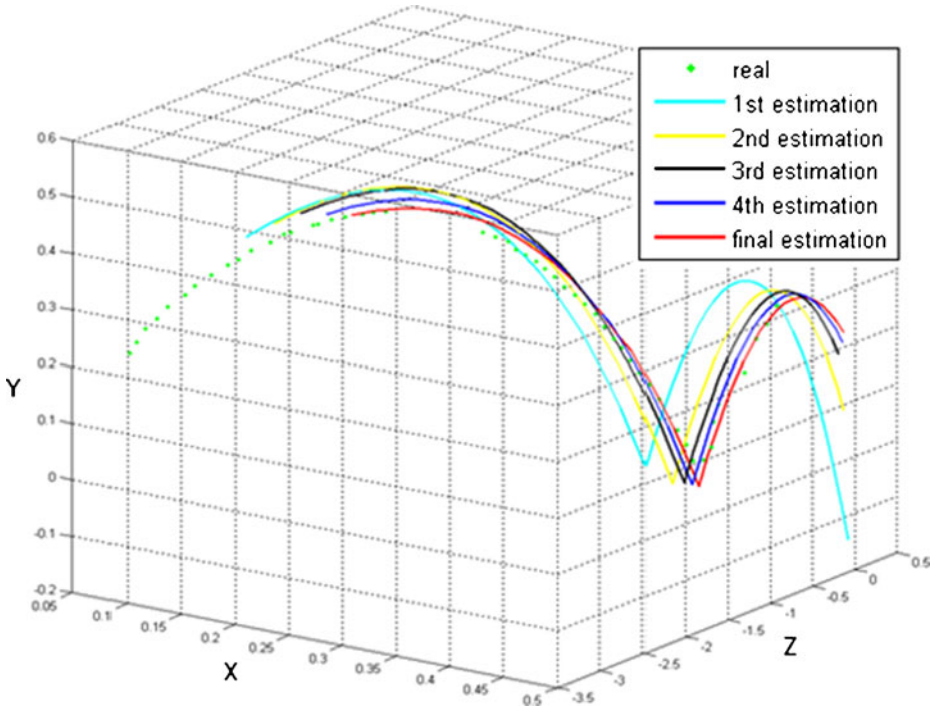


Fig. 12 Trajectory prediction results

7.3 Prediction Results

Figure 11 shows our experimental setup. We use a touch screen to record hit locations and use another high speed camera to record hit time. To make a robot have enough

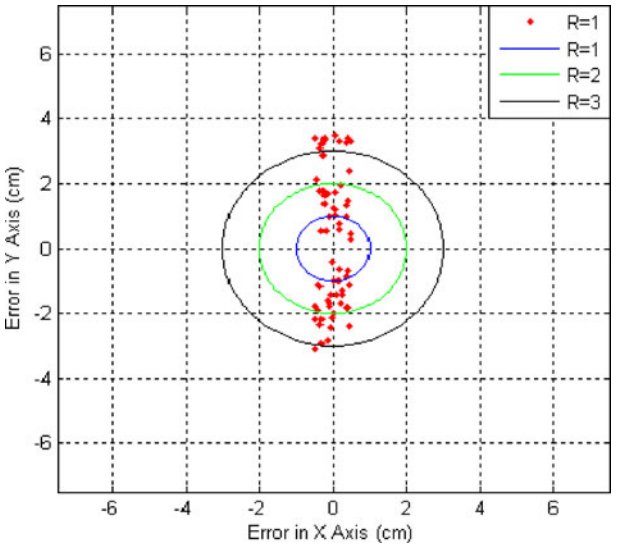


Fig. 13 Error distribution of predicted hit positions

Table 2 Comparisons of vision systems for ping pong playing robots

Developer (s)	Vision element	Resolution	Frame rate (Hz)	Ball velocity	Baseline (m)
Andersson [1]	Four cameras	256 × 240	60	5–6 m/s	0.5
Michiya [2]	Stereo cameras	640 × 416	60	Not mentioned	3.57
Acosta [10]	Single camera	768 × 576	40	5 m/s	No
Z. Zhang [11]	Stereo cameras	640 × 480	250	19 m/s	2.0
C. Smith [25]	Stereo cameras	320 × 240	50	5–6 m/s	0.6
This paper	Stereo cameras	640 × 480	110	15 m/s	0.18

time to react, our method begins the prediction once two ball's 3-D information are available. The robot can start to move and prepare to return the ball once the first prediction information received. The prediction is updated quickly when one new ball's information is calculated. Figure 12 shows that predicted trajectories can approach the actual one gradually. We generate 11 launch directions from the serving machine's control card, and for each direction we chose three different velocities. Each trajectory is repeated twice in the experiment, i.e., totally 66 different trajectories are used for testing. The error distribution of predicted hit positions is shown in Fig. 13. We can find that the error in Y direction is larger than that in X direction. This phenomenon may be caused by three forces take effects in Y direction but only one in X direction. In the experiment, the mean error of predicted hit positions is 2.07 cm and that of hit time is 20 ms. The prediction accuracy of the hit positions and the hit time meet the requirements of the control system for action plan.

7.4 Comparisons with Previous Vision Systems

As shown by the comparisons in Table 2, to our best knowledge, the vision system proposed in this paper owns the shortest baseline. The major advantage of our system is it may be more suitable for a humanoid ping-pong robot due to the short baseline. Besides, our system is able to handle balls with the velocity up to 15 m/s. The constraint of detecting region and the simple five-point ball representation are the main reasons for the fast-processing performance.

Unlike some other vision systems [2, 10, 12] that rely on color information to detect a ball, our system mainly use the motion feature and our five-point feature to detect a ball. We consider that color feature is not robust enough since a ball shows different appearances under different illumination conditions. As we do not utilize color information, both yellow and white balls can be used in our system.

8 Conclusion

It is well known that 3-D reconstruction accuracy depends on application configurations including cameras placement, baseline length, focal length, image resolution, and distance between object to camera. An orthogonal placed, long baseline, long-focal-length lens, high-resolution cameras, and close object-to-camera distance configurations help to obtain higher reconstruction accuracy. However, for a vision system of a humanoid ping-pong robot, like human eyes, the two cameras

must be parallel placed and the baseline length should be short. Furthermore, long-focal-length lens cannot be used in order that the ping-pong table is in view; high-resolution cameras cannot be used to ensure the processing speed; object-camera distance in our experiment is not close (about 4 m when a ball is launched). All these factors bring more difficulties to our system than other long baseline ones. In this paper, several algorithms are proposed to guarantee the accuracy and the processing speed of the short baseline vision system. Besides, some techniques including the constraint of ball-searching region, the restriction of fundamental matrix, and the smoothing of prediction data are employed to improve the robustness of the system.

Although the methods involved in our system are specific to humanoid ping-pong robots, they may be of some interest to broader applications. The limitation of our system is when light is strong, if white balls are used, the white wall opposite the cameras will bring unfavorable influence on the ball detection because there is little difference when a white ball is flying in white environments, which is still a difficult problem.

Acknowledgements This work is supported by Natural Science Foundation of China (Grant Number: 60805046 and 60835004).

Appendix

Proposition *If a projection matrix is decomposed into:*

$$M = \left[\begin{array}{ccc|c} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{array}\right] = [M_1 \ M_2],$$

the fundamental matrix can be presented by:

$$F = [M_2^r - M_1^r (M_1^l)^{-1} M_2^l]_\times M_1^r (M_1^l)^{-1}.$$

Proof Given a vector $d = (d_x, d_y, d_z)$, the skew symmetric matrix $[d]_\times$ defined by d is denoted by:

$$[d]_\times = \left[\begin{array}{ccc} 0 & -d_z & d_y \\ d_z & 0 & -d_x \\ -d_y & d_x & 0 \end{array}\right] \tag{24}$$

Projection matrix M can be decomposed into:

$$M = K[R \ T] \tag{25}$$

where K is called camera intrinsic matrix, and $[RT]$ is called camera extrinsic matrix. Suppose that the origin of world coordinates is at the first camera's optical center and define that the rotation matrix and the translation vector between two cameras are $\widehat{R}$ and $\widehat{T}$, respectively, thus we have:

$$P^l = K^l \ [\mathrm{I}|0] \tag{26}$$

$$P^r = K^r \ [\widehat{R}|\widehat{T}] \tag{27}$$

In [26], the following equation is given,

$$F = [K^r \widehat{T}]_\times K^r \widehat{R} K^{-1} \tag{28}$$

Fundamental matrix can be easily computed if both intrinsic and extrinsic parameters are known. However, we only know two projection matrices here. Defining the external parameters of the two cameras in our vision system are $[R^l \quad T^l]$ and $[R^r \quad T^r]$ respectively, we have:

$$M^r = K^r[R^r|T^r] = K^r[\widehat{R}R^l|\widehat{R}T^l + \widehat{T}] \tag{29}$$

From Eq. 5, we have:

$$\begin{aligned}
F &= [K^r \widehat{T}]_\times K^r R^r (R^l)^{-1} (K^l)^{-1} \\
&= [K^r \widehat{T}]_\times K^r R^r (K^l R^l)^{-1} \\
&= [K^r (T^r - \widehat{R}T^l)]_\times K^r R^r (K^l R^l)^{-1} \\
&= [K^r T^r - K^r R^r (R^l)^{-1} T^l]_\times K^r R^r (K^l R^l)^{-1} \\
&= [K^r T^r - K^r R^r (K^l R^l)^{-1} K^l T^l]_\times K^r R^r (K^l R^l)^{-1} \\
&= [M_2^r - M_1^r (M_1^l)^{-1} M_2^l]_\times M_1^r (M_1^l)^{-1}
\end{aligned}$$

End of proof. □

References

1. Andersson, R.L.: Dynamic sensing in a ping-pong playing robot. IEEE Trans. Robot. Autom. **5**(6), 728–739 (1989)
2. Matsushima, M., Hashimoto, T., Takeuchi, M., Miyazaki, F.: A learning approach to robotic table tennis. IEEE Trans. Robot. Autom. **21**(4), 767–771 (2005)
3. Takeuchi, M., Miyazaki, F., Matsushima, M., Kawatani, M., Hashimoto, T.: Dynamic dexterity for the performance of 'wall-bouncing' tasks. IEEE Int. Conf. Robot. Autom. **2**, 1559–1564 (2002)
4. Matsushima, M., Hashimoto, T., Miyazaki, F.: Learning to the robot table tennis task-ball control and rally with a human. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pp. 2962–2969 (2003)
5. Muelling, K., Peters, J.: A computational model of human table tennis for robot application. In: Proceedings of Autonome Mobile Systeme, vol. 3, pp. 1309–1314 (2009)
6. Angel, L., Sebastian, J.M., Saltaren, R., Aracil, R., SanPedro, J.: RoboTenis: optimal design of a parallel robot with high performance. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Canada, pp. 2134–2139 (2005)
7. Angel, L., Traslosheros, A., Sebastian, J.M., Pari, L., Carelli, R., Roberti, F.: Vision-based control of the robotenis system. Lect. Notes Control Inf. Sci. **370**, 229–240 (2009)
8. Brunnett, G., Rusdorf, S., Lorenz, M.: V-Pong: an immersive table tennis simulation. IEEE Comput. Graph. Appl. **26**(4), 10–13 (2006)
9. Rusdorf, S., Brunnett, G., Lorenz, M., Winkler, T.: Real time interaction with a humanoid avatar in an immersive table tennis simulation. IEEE Trans. Vis. Comput. Graph. **13**(1), 15–25 (2007)
10. Acosta, L., Rodrigo, J.J., Mendez, J.A., Marichal, G.N., Sigut, M.: Ping-pong-player prototype—a PC-based, low-cost ping-pong robot. IEEE Robot. Autom. Mag. **10**(4), 44–52 (2003)
11. Zhang, Z., Xu, D.: Design of high-speed vision system and algorithms based on distributed parallel processing architecture for target tracking. In: Proceedings of the 7th Asian Control Conference, pp. 1638–1643 (2009)
12. Sabzevari, R., Shahri, A.: 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)

13. Sorensen, V., Ingvaldsen, R.P., Whiting, H.T.: The application of co-ordination dynamics to the analysis of discrete movements using table tennis as a paradigm skill. Biol. Cybern. **85**(1), 27–38 (2001)
14. Klarquist, W.N., Bovik, A.C.: Fovea: a foveated vergent active stereo vision system for dynamic three-dimensional scene recovery. IEEE Trans. Robot. Autom. **14**(5), 755–770 (1998)
15. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
16. Harris, C., Sttephens, M.: A combined corner and edge detector. In: Proceeding of 4th Alvey Vision Conference, pp. 147–151 (1988)
17. Hartley, R.I.: In defense of the eight-point algorithm. IEEE Trans. Pattern Anal. Mach. Intell. **19**(6), 580–593 (1997)
18. Wong, P.K.C.: Developing an intelligent table tennis umpiring system: identifying the ball from the scene. In: Second Asia International Conference on Modelling & Simulation, pp. 445–450 (2008)
19. Chen, H., Chen, H.S., Lee, S.: Physics-based ball tracking in volleyball videos with its applications to set type recognition and action detection. In: IEEE International Conference on Pattern Recognition, vol. 1, pp. 1097–1100 (2007)
20. Zhang, P., Lu, T.: Real-time motion planning for a volleyball robot task based on a multi-agent technique. J. Intell. Robot. Syst. **49**(4), 355–366 (2007)
21. Tong, X.F., Lu, H.Q., Liu, Q.S.: An effective and fast soccerball detection and tracking method. In: Proc. IEEE Int. Conf. Pattern Recognit., Cambridge, U.K., vol. 4, pp. 795–798 (2004)
22. Yamada, A., Shirai, Y., Miura, J.: Tracking players and a ball in video image sequence and estimating camera parameters for 3-D interpretation of soccer games. In: Proc. IEEE Int. Conf. Pattern Recognit., Québec, QC, Canada, vol. 1, pp. 303–306 (2002)
23. Mouhamed, M., Toker, O., Harthy, A.: A 3-D vision-based man–machine interface for hand-controlled telerobot. IEEE Trans. Ind. Electron. **52**(1), 306–319 (2005)
24. Miyazaki, F., Matsushima, M., Takeuchi, M.: Learning to dynamically manipulate: a table tennis robot controls a ball and rallies with a human being. In: Advances in Robot Control, pp. 317–341. Springer (2005)
25. Smith, C., Bratt, M., Christensen, H.I.: Teleoperation for a ball-catching task with significant dynamics. Neural Netw. **24**, 604–620 (2008) (Special Issue on Robotics and Neuroscience)
26. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)