2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Real-time Accurate Ball Trajectory Estimation with "Asynchronous" Stereo Camera System for Humanoid Ping-Pong Robot*

Qi Xie, Yong Liu *Member IEEE*, Rong Xiong *Member IEEE*, and Jian Chu, *Senior Member, IEEE*

***Abstract*— Temporal asynchrony between two cameras in the vision system is a usual problem in practice. In some vision task such as estimating fast moving targets, the estimation error caused by the tiny temporal asynchrony will become non-ignorable essentials. This paper will address on the asynchrony in the stereo vision system of humanoid Ping-Pong robot, and present a real-time accurate Ping-Pong ball trajectory estimation algorithm. In our approach, the complex Ping-Pong ball motion model is simplified by a polynomial parameter function of time *t* due to the limited observing time interval and the requirement of real-time computation. We then use the perspective projection camera model to re-project the ball's parameter function on time *t* into its image coordinates on both cameras. Based on the assumption that the time gap of two asynchronous cameras will maintain a const during very short time interval, we can obtain the time gap value and also the trajectory parameters of the Ping-Pong ball in a short time interval by minimizing the errors between the images of the ball in each camera and their re-projection images from the modeled parameter function on time *t*. Comprehensive experiments on real Ping-Pong robot cases are carried out, the results show our approach is more proper for the vision system of humanoid Ping-Pong robot, when concerning the accuracy and real-time performance simultaneously.**

I. INTRODUCTION

Building the vision system equipped on the body of a humanoid robot is a challengeable task. In previous work [1], we have built two humanoid robots, see figure 1, which are 160cm in height, 55kg in weight and contain 30 degrees of freedom (DOF), and can rally to each other with 114 rounds at most and rally to human with 145 rounds at most[1]. Generally speaking, there are three main successive tasks for the vision system of a humanoid robot, i.e. pose estimation for the onboard vision system, real-time accurate ball trajectory estimation, prediction for the arriving time, position, velocity of the Ping-Pong ball. The pose estimation, which can be used to identify the ball's world coordinate and the initial pose of the arm in the world coordinate, requires to obtain the robot's 6DOF pose in the world coordinate in real-time. Based on the world pose of the vision system, the trajectory estimation will use stereo cameras to localize the ball and fit its motion model to estimate the trajectory of the ball. Then the last task of the vision system will focus on predicting the ball's status (including arriving time, position, velocity), which can be used as visual servo for the arm of robot to hit the ball back. Obviously, the accuracy of prediction is highly relying on the results of trajectory estimation.

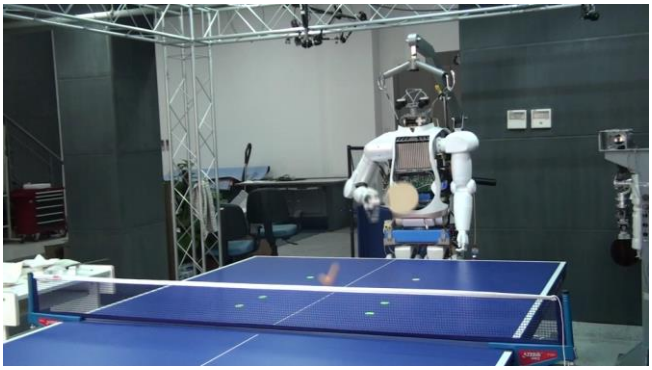

Figure 1. Humanoid Ping-Pong Robot with two cameras on the head.

This paper will focus on the problem of estimating the trajectories of Ping-Pong ball accurately in real-time. Basically, there are two main difficulties in estimating the trajectories of the ball for a humanoid robot.

The first difficulty is how to reach an optimal compromise between the accuracy of estimation and the capability of real-time output the trajectory results. Obviously, increasing the capturing frame rate of the vision system or increasing the number of cameras will promote the accuracy of trajectory estimation, but add heavy burdens and delay the output of the trajectory results due to the strict limitations on the transferring bandwidth and the computation capability of the onboard computer. In fact, it will cost less 600ms that the Ping-Pong ball flies over the table in the normal rallying. And the robot need reserve more than 400ms to move the arm to the planned hit point. Besides the computation on prediction, there are only about 150ms left for the trajectory estimation. Thus, in our vision system, we use two cameras[2] to compose a stereo camera system working at a resolution of 640x480, 60 frame/s, which can provide enough accuracy on the ball's localizations while keeping the real-time output of the trajectory results.

The second difficulty is how to reduce the trajectory estimation errors caused by the asynchrony in the stereo camera system. In our system, those two single cameras are

*Rearsch supported by National Natural Science Foundation of China (61173123). Zhejiang Provincial Natural Science Foundation of China (LR13F030003).

Q. Xie is with the Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027 China (e-mail: qxie@iipc.zju.edu.cn).

Y. Liu is with Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027 China (He is the corresponding author of this paper, e-mail: yongliu@iipc.zju.edu.cn).

R. Xiong is with the Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027 China (e-mail: rxiong@iipc.zju.edu.cn).

J. Chu is with Cyber-Systems and Control, Zhejiang University, Zhejiang, 310027 China (He is the corresponding author of this paper, e-mail: chuj@iipc.zju.edu.cn).

[1] The demo for our Ping-Pong robot can be found at: http://www.youtube.com/watch?v=t_qN3dgYGqE

[2] These two cameras are mounted with a rigid constraint, which can be calibrated in previous.

setting to work at the same frame rate (60HZ). It can be regarded as a synchronous stereo camera system when proceeding normal tasks without fast motions. However, in our case of Ping-Pong robot, the tiny time gap of the interval $[0,1/60s)$ between the two cameras will also affect the estimation accuracy of trajectories greatly. The estimation errors will be amplified by the prediction and lead to failure when the robot's arm hits ball back to the specific position.

The reason of asynchrony is that the capturing triggers are controlled by the operation system, which cannot guarantee the absolute synchrony of both cameras. We call this vision system as "asynchronous" stereo camera system.

In this paper, our approach will address on reducing the estimation errors caused by the "asynchrony" in humanoid robot vision system. We present a real-time accurate Ping-Pong ball trajectory estimation algorithm, which concerns the variable asynchrony of the cameras and the motion model of the Ping-Pong ball simultaneously. The accuracy of the proposed approach is demonstrated by comprehensive experiments in real Ping-Pong robotcases.

II. Related Works

As the vision systems of Ping-Pong robot need localize the positions of ball in real-time accurately, the popular vision systems [2-3] are almost adopting two or more than two cameras to construct a stereo camera system, localize the positions of the ball, and estimate the trajectories of the ball in real-time. Thus the synchronization of different cameras is important when the same motion is captured by different devices from different viewpoints which are used to localize the position of the ball accurately. To our best knowledge, our system is the first Ping-Pong robot vision system concerning the asynchrony between those two cameras.

Generally speaking, the methods to synchronize multiple-camera system can be divided into three categories, i.e. *physical-connection based synchronization*, *cyber-connection based synchronization*, and *motion consistency based synchronization*.

The physical-connection based synchronizations are intuitive approaches [4-8], which use special synchronous-purpose hardware to connect those cameras physically and control them by a microcomputer unit to trigger the cameras with synchronization signals. The accuracy of this synchronization can be guaranteed by the hardware systems, but it requires that all the cameras should have the function of capturing images controlled by the signals. Besides, the requirement of physical connection of the cameras will also limit the applicability of this approach. Furthermore, this kind of approach also needs to attach additional complex hardware devices, this may be not acceptable in our humanoid Ping-Pong robot vision system, which is quite sensitive to the size and weight of the vision system.

The cyber-connection based synchronization methods [9-11] try to employ some cyber signals to trigger the cameras or estimate the time gap of the cameras. The typical systems of this kind of approach may use the binary light source based synchronization [9] or the network messages [10, 11]. The main drawback of this kind of method is the unacceptable accuracy due to the assumption of the constant latency, especially, in our humanoid robot vision system, the heavy capturing and calculation burden of the onboard computer obviously cannot guarantee the constant latency.

The motion consistency based synchronizations [12-20] can be regarded as post-processing methods, which rely on the temporal-spacial consistency of the motions observed by different cameras. These methods do not require additional synchronization devices, but cannot be applied in real-time and almost need the assumption of constant time gaps among the cameras. Furthermore, those methods are also sensitive to occlusion which may be quite usual in practice. Those reasons prevent those approaches implemented in our vision system.

III. Trajectory Estimation Algorithm

A. Perspective Projection Model for Camera

In our robot system, two cameras are used to estimate the trajectory of the balls, and we use perspective projection model [21] to recover the position of the ball's center:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{1}$$

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{2}$$

Here $(X_w, Y_w, Z_w)^{\mathrm{T}}$ is the world coordinate of point P, $(X_c, Y_c, Z_c)^{\mathrm{T}}$ is P' corresponding camera coordinate, $(u, v)^{\mathrm{T}}$ is the image coordinate of P and $\mathbf{A}$ is a 3x4 intrinsic matrix of camera. $\mathbf{R}, \mathbf{t}$ are external parameters of the camera.

Rewriting formula (2) as following:

$$\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{3}$$

We then have:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \left(\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \right)^{-1} \mathbf{A} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4}$$

Let $\mathrm{H} = \mathbf{A} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$, $\mathrm{K} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$, and the homograph coordinate of the image point denoted as $\tilde{m} = (u, v, 1)^{\mathrm{T}}$, and the world homograph coordinate of the point denoted as $\tilde{M} = (X_w, Y_w, Z_w, 1)^{\mathrm{T}}$, then formula (4) can be rewritten as:

$$\tilde{m} = \left(\mathrm{K}\tilde{M}\right)^{-1} \mathrm{H}\tilde{M} = \frac{\mathrm{H}\tilde{M}}{\mathrm{K}\tilde{M}} \tag{5}$$

B. Modeling the Flying Motion of the Ping-Pong Ball

As mentioned previously, the vision system of our Ping-Pong robot needs to obtain the accurate ball trajectory in real-time before the ball flying over a quarter of the table. Thus we should use a proper flying trajectory model which can consider both the accuracy and computation complexity simultaneously. Figure 2 shows the forces [22-23] during the ball flying.

Figure 2 presents the world coordinate of our robot system, F_g, F_b, F_s, F_m are the gravity, air buoyancy, air resistance, and

Magnus force respectively. To simplify the computation, we ignore the spin of the ball[4], thus the Magnus force is zero. As the mass of the ball will be much larger than the mass of the air with the same volume, the air buoyancy can be ignored comparing with the gravity. The air resistance is proportional to the velocity of the ball and its force direction is contrary to the flying direction of the ball. In our model, air resistance can also be ignored due to the short time sample interval[5].

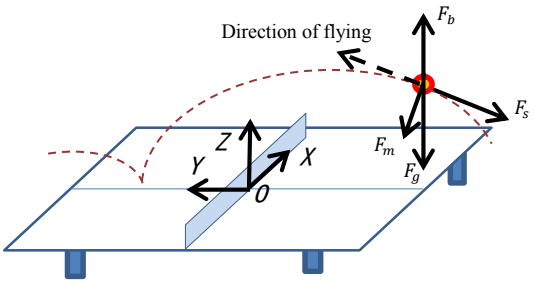

Figure 2. Force analysis of flying ball

Assuming the gravity acceleration is $g, \ddot{Q}(t), \dot{Q}(t), Q(t)$ are the acceleration, velocity, and position of the ball in time t, and the motions of the ball can be presented as follow:

$$\ddot{Q}(t) = \begin{bmatrix} \ddot{X}(t) \\ \ddot{Y}(t) \\ \ddot{Z}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{6}$$

$$\dot{Q}(t) = \begin{bmatrix} \dot{X}(t) \\ \dot{Y}(t) \\ \dot{Z}(t) \end{bmatrix} = \begin{bmatrix} \dot{X}(t_0) \\ \dot{Y}(t_0) \\ -g(t-t_0) + \dot{Z}(t_0) \end{bmatrix} \tag{7}$$

$$Q(t) = \begin{bmatrix} X(t) \\ Y(t) \\ Z(t) \end{bmatrix} = \begin{bmatrix} \dot{X}(t_0)(t-t_0) + X(t_0) \\ \dot{Y}(t_0)(t-t_0) + Y(t_0) \\ -\frac{g}{2}(t-t_0)^2 + \dot{Z}(t_0)(t-t_0) + Z(t_0) \end{bmatrix} \tag{8}$$

Thus we use a polynomial model to present the motion of the flying Ping-Pong ball. In this model, t_0 is the time stamp of the initial moment. And there are seven parameters: the gravity acceleration g, the ball's initial position $X(t_0), Y(t_0), Z(t_0)$, the ball's initial velocity $\dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0)$, which are needed to be estimated in this model.

C. Ball Trajectory Estimation on "Asynchronous" Stereo Camera System

Assuming the left camera's time interval among two successive frames is t_1, and the right camera's time interval among two successive frames is t_2. The time gap between the left camera and the right camera is $t_{1,2}$. In our Ping-Pong robot system, the capturing cycles t_1 and t_2 are easy to obtain. However, the vision system is not able to guarantee both cameras capturing images at the exact same time, furthermore, the time gaps are also varied affected by the overload condition of the operation system, CPU and Bus etc. Thus, $t_{1,2}$ is not zero in practice and will be varied randomly. In our approach, we make a reasonable assumption that the image capturing time gaps between two cameras will be a const in a short time quantum. That means $t_{1,2}$ will keep steady state value in a very short observation.

[4] In our robot, the Ping-Pong bat is bare wooden, which will reduce the spin of the ball greatly when hitting.

[5] In our following algorithm, the model will only concern at most 5 frames of the observations, thus the time window is less than 90ms, in this condition, the air resistance can be ignored.

Based on the motion model of the ball, we can obtain the homograph coordinate of the flying ball as follow:

$$\tilde{Q}(t) = [X(t), Y(t), Z(t), 1]^{\mathrm{T}} \tag{9}$$

In our vision system, although the accurate capturing time of each frame in both cameras is not available, the captured successive sequence of each camera is known[8]. That means we can know the true sequence of each frames from the cameras. In a short time quantum, we assume the left camera can obtain the sequenced image points of the ball, denoted as m_1^i $(i = 1,2,3,\cdots,k)$ and the corresponding world coordinate of those image points are denoted as $\tilde{Q}(t_{m_1^i})$. Similarly, the sequenced image points of the right camera and their world coordinate can be denoted as m_2^j and $\tilde{Q}(t_{m_2^j})$ $(j = 1,2,3,\cdots,q)$.

Recalling the formula (5), we can re-project the ball's world coordinates into their corresponding image coordinates:

$$\begin{cases} \tilde{m}_1^i = \frac{\mathrm{H}_1^i \tilde{Q}((i-1)*t_1)}{\mathrm{K}_1^i \tilde{Q}((i-1)*t_1)}, (i = 1,2,3,\cdots,k) \\ \tilde{m}_2^j = \frac{\mathrm{H}_2^j \tilde{Q}(t_{1,2}+(j-1)*t_2)}{\mathrm{K}_2^j \tilde{Q}(t_{1,2}+(j-1)*t_2)}, (j = 1,2,3,\cdots,q) \end{cases} \tag{10}$$

Here, $\tilde{m}_1^i$ and $\tilde{m}_2^j$ are the re-projections of the ball's world coordinates from the left and right cameras respectively, and the start time of the first point obtained from the left camera is t_0. Then estimating the trajectory of the ball in such short time quantum can be regarded as an optimization problem:

$$\arg Min_E \left(\sum_{i=1}^{k} ||\tilde{m}_1^i - m_1^i||^2 + \sum_{j=1}^{q} ||\tilde{m}_2^j - m_2^j||^2\right) \tag{11}$$

In formula (11), there are eight parameters ($E = \{X(t_0), Y(t_0), Z(t_0), \dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0), g, t_{1,2}\}$) that need to be estimated during the optimization. This optimization problem can be easily solved by the Levenberg-Marquardt (LM) Optimization [24] method. The choosing of the initial values for those eight parameters will be discussed later.

According to formula (11), estimating the whole track can be divided into a series of sub-trajectory estimation problems. That is using few pairs of the ball's image points to estimate a set of parameters to represent that sub-trajectory and then a slider window based iterative optimization policy is adopted to obtain the successive parameters of sub-trajectories.

The Slider Window based Real-time Trajectory Estimation Algorithm (SWRTEA) is presented as follow:

Algorithm 1: SWRTEA

Input: Slider window size S, successive images of the ball's central points $M_{left} = \{m_1^i, i = 1,2,3 \ldots\}, M_{right} = \{m_2^j, j = 1,2,3 \ldots\}$

Output: parameters sets, $E_1, E_2, \ldots$, for every sub-trajectories

1. $CB \leftarrow 1$; // set current beginning of the slider window
2. **while** $(CB < |M_{left}| - S \text{ or } CB < |M_{right}| - S)$
3. $\quad E_{CB} = \arg Min_E \left(\sum_{i=CB}^{CB+S} ||\tilde{m}_1^i - m_1^i||^2 + \sum_{j=CB}^{CB+S} ||\tilde{m}_2^j - m_2^j||^2\right)$;
4. $\quad CB++$;
5. **end while**

[8] In the condition that the images from different cameras are captured asynchronous purely controlled by software, the image sequences can keep the same order as their temporal sequence when capturing.

D. Further Analysis for the SWRTEA

In the SWRTEA, the LM optimization method is used to solve formula (11).

Normally, the initial values for the estimated parameters $E = \{X(t_0), Y(t_0), Z(t_0), \dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0), g, t_{1,2}\}$ are required. Obviously, in the first iteration of the SWRTEA, the gravity accelerate can be set as g =9.8 m/s^2, and the time gap between left and right cameras can be set as $t_{1,2} = \frac{t_1}{2}$. As the time beginning from the first iteration, which means $\tilde{Q}(t_0) = \tilde{Q}(0)$ at the observation of m_1^1. We can assume the observations from both cameras are synchronous, and calculate the initial values of $X(t_0), Y(t_0), Z(t_0)$ based on m_1^1 and m_2^1. And the initial values of $\dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0)$ can also be obtained by derivation the positions of two successive observations from both cameras which are assumed to be synchronous. After the first iteration, the following iteration can use the parameters estimated from its previous optimization processing as its initial values.

When solving the optimization, there are two other parameters, i.e. H and K ($\mathrm{H}_1^i, \mathrm{K}_1^i$ to the left camera, $\mathrm{H}_2^j, \mathrm{K}_2^j$ to the right camera) which should be clarified. Based on formula (4), these two parameters are related with the intrinsic matrix of camera and its corresponding external parameters in the time of observations. The intrinsic matrix of camera can be calibrated offline [21], while the external parameters need update for each capturing image. In our Ping-Pong robot system, we place eight green points with known structures on the tables and use *P*n*P* method [25] to estimate the external parameters of the camera in real-time.

The dimension of the optimization problem in our SWRTEA is quite low. The slider window size *s* is almost less than 10, which means only no more than 20 points involving in the optimization. So the temporal computation for SWRTEA is quite low. In practice, the parameters that need to be estimated can be initialized very near to the optimal values, this will further reduce the computation in optimization. Thus step 3 in SWRTEA only needs to execute a few of iterations to guarantee the process in real-time.

IV. EXPERIMENTS

In this section, comprehensive experiments in real Ping-Pong robot system are carried out in order to evaluate the proposed approach. We compare our SWRTEA with the trajectory estimation method [9] [23] denoted as SA (Synchronizing on Asynchronous condition) in the following experiments, whtch directly calculates the positions of the balls from a pair of images captured by two different cameras without concerning the asynchrony between the cameras.

Figure 3 shows the scene of experiments on practical Ping-Pong robot vision system, which equips with two cameras working at 60HZ with a resolution of 640x480. To capture the ground true trajectories of the ball, we add a verified external vision system with two high speed cameras mounted on the ceiling over the table, these two cameras are working at 120HZ, and synchronized by a hardware triggering system. The trajectories obtained from the external vision system are carefully calculated offline and can be regarded as the ground truths.

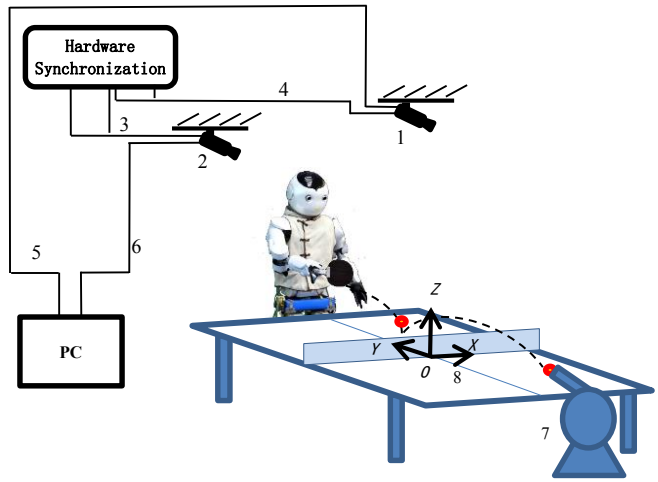

Figure 3. The scene of Experiments on Practical Ping-Pong Robot Vision System. 1,2: Two color CCD cameras working at 120HZ are mounted on the ceiliing; 3,4: The trigger line transmitting synchronization signal to guarantee those two camera are capturing synchronously; 5,6: The 1394 data line transmitting capturing images to computer ; 7: Pitching machine giving out repeatable ball flying trajectory; 8: The world coordinate.

As we need to evaluate the accuracies of the continuous trajectories of the balls, a new evaluation metric is desired. We introduce a *timeline error* to estimate the accuracy of the trajectory estimation. Assuming the observation of a piece of the trajectory can be represented as $Q(t) = [X(t), Y(t), Z(t)]^{\mathrm{T}}$ and the true trajectory is $Q'(\mathrm{t})$=$[X'(t), Y'(t), Z'(t)]^{\mathrm{T}}$. Then the errors on a piece of trajectory can be defined as follows:

$$E_x=\left(\frac{1}{n}\sum_{i=1}^{n}|x_i(t_i)-x_i'(t_i)|^2\right)^{\frac{1}{2}}$$

$$E_y=\left(\frac{1}{n}\sum_{i=1}^{n}|y_i(t_i)-y_i'(t_i)|^2\right)^{\frac{1}{2}} \tag{12}$$

$$E_z=\left(\frac{1}{n}\sum_{i=1}^{n}|z_i(t_i)-z_i'(t_i)|^2\right)^{\frac{1}{2}}$$

$$E_m=\frac{1}{n}\sum_{i=1}^{n}||Q(t_i)-Q'(t_i)||_F$$

Here, *n* is the sample points on the ground true trajectory. Then we can evaluate the errors of those sub-trajectories in the slider windows and we also set the sample points number equal to the size of windows, that is *n*=*s*.

In the first experiment, we use the data captured from the external verified vision system, and choose the stagger frames from each camera. As we can obtain the accurate time stamp of each observation in the external verified vision system with the hardware synchronization. So those selected frames from both cameras, which are working at 120HZ, can construct perfect real observations from a vision system working at 60HZ and with a time gap of 5/600s. We then apply both SA and SWRTEA on those data, the results of trajectory timeline estimation errors are shown in figure 4.

The experimental results in figure 4 show that SWRTEA can achieve better performances than the SA. It also shows the size of the slider window will also affect the accuracy of

[9] To compare fair, both methods use the same ball motion model.

estimation. In our experiment, we use three window sizes, 5, 10, and 15, and the size of 5 can achieve the best results.

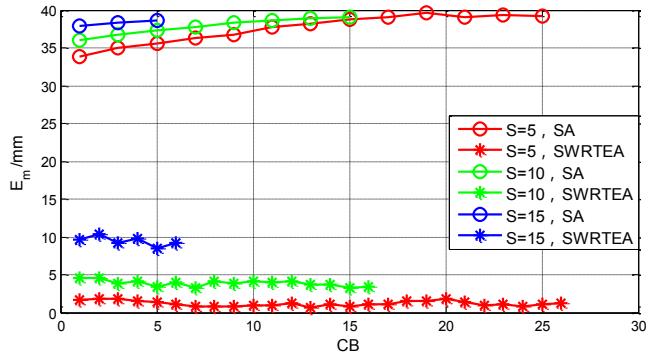

Figure 4. Comparison results of the trajectory estimation *timeline errors* on SA and SWRTEA under different window sizes on $t_{1,2} = 5/600$s. CB is the current beginning of the slider window. The circles in this figure represent the E_m of SA, and the stars represent the E_m of SWRTEA.

In the second experiments, we compare the methods SWRTEA and SA by using the data captured from both vision systems. As it is impossible to synchronize the time clock of these two vision systems accurately, we cannot use the formula (12) to calculate trajectory errors. We then try to unify the y-coordinate and consider the trajectories from observation and ground truth as two functions $Q(Y) = [X(Y), Y, Z(Y)]^{\mathrm{T}}$ and $Q'(Y) = [X'(Y), Y, Z'(Y)]^{\mathrm{T}}$. Then the error on a piece of trajectory can be calculated as follow:

$$E'_x = \left(\frac{1}{n}\sum_{i=1}^{n} |x_i(Y_i) - x'_i(Y_i)|^2\right)^{\frac{1}{2}}$$

$$E'_z = \left(\frac{1}{n}\sum_{i=1}^{n} |z_i(Y_i) - z'_i(Y_i)|^2\right)^{\frac{1}{2}} \qquad (13)$$

$$E'_m = \frac{1}{n}\sum_{i=1}^{n} ||Q(Y_i) - Q'(Y_i)||_F$$

Here Y_i is the sample point on the ground true trajectory, and n is the number of these sample points. And we call the error calculated by formula (13) as *unified Y error*.

In figure 5, we use formula (13) to calculate the errors and the sample point Y_i is obtained from the observation of the external vision system. And we also consider three kinds of window size, 5, 10, and 15.

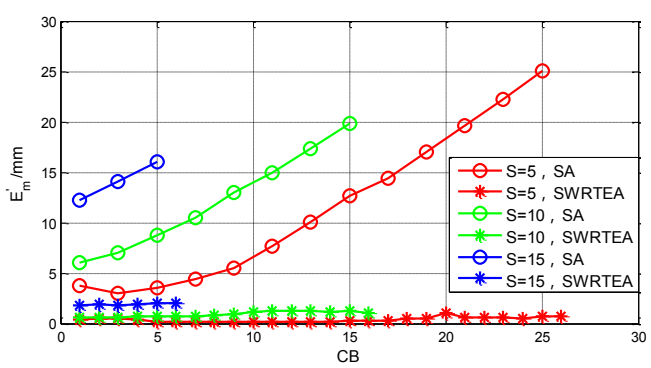

Figure 5. Comparison results of the trajectory estimation *unified Y errors* on SA and SWRTEA under different window sizes.

The results in both figure 4 and figure 5 show SWRTEA will perform much better than SA under those three window sizes. Although the error results presented by the *unified Y errors* is less than the error results presented by the *timeline errors* on the numerical values[10], these two types of errors can represent the same performance relations among those evaluated methods. We observe that the results in figure 5 show that the unified Y error of the SWRTEA on size 5 are all less than 3*mm*, which is almost less than 1/10 of the ball's radius.

In the third experiment, we use the Ping-Pong robot vision system to estimate the trajectories of ball, and the ground truths are obtained from the external verified vision system offline. In our experiment, the window size s=5, and each sample point Y_i is obtained from the observation of the external vision system when calculating the errors by formula (13). The results are shown in figure 6.

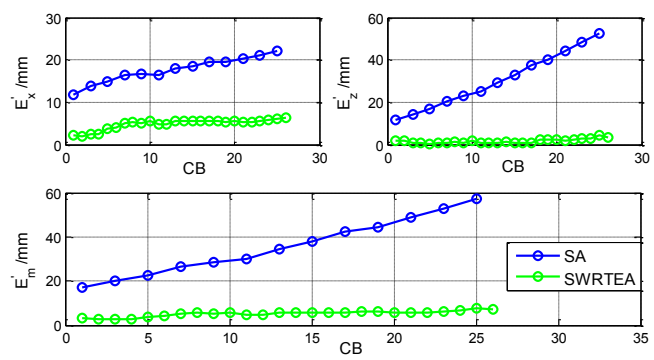

Figure 6. Real comparison results of the trajectory estimation errors on SA and SWRTEA.

The trajectories estimated by SWRTEA and SA are shown in figure 7, which is the projections of the trajectories on Y-Z plane. The results show that the SWRTEA can perfect approach the ground truth.

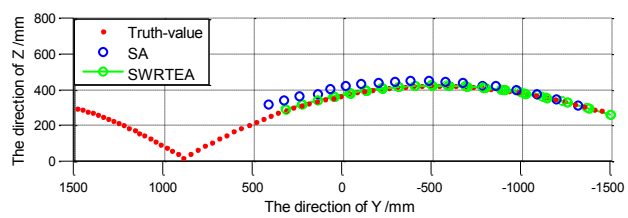

Figure 7. Trajectory comparison results on SWRTEA, SA and ground truth. The red dots are the ground truths, blue circles are the results estimated by SA, and green circles are the results estimated by SWRTEA.

We also carried out the experiment to consider the overall performances of both approaches in real vision system. This experiment only considers the trajectory's section, which the ball flies into the table and flies over a quarter of the table. We first use our Ping-Pong robot's onboard vision system to capture 48 trajectories, which are launched by a Ping-Pong ball pitching machine. And the *unified Y errors* for these 48 trajectories are calculated by SA and SWRTEA respectively. We then average the total errors on each estimation method.

[10] The reason is the *unified Y error* only concerns the errors on X and Z, and the errors on Y is only indirectly coupled in the *unified Y error*.

The results are shown in Table I. To evaluate the performance of our approach by comparing with other synchronization method, we also use the same pitching machine to launch another 36 trajectories, which are launched from the same position, angle and parameters as the previous 48 trajectories. These 36 trajectories are also observed by our humanoid robot vision system, the only difference is that both cameras are connected with a hardware synchronization triggering which can provide accurate control signals to capture synchronous frames from both cameras at a frame rate of 60. We then estimate the *unified Y error* from those synchronous frames offline and also obtain the average errors, shown in Table I, denoted as HS (hardware synchronization). In both observations, the ground truths are obtained from the external verified vision system working at a frame rate of 120.

TABLE I. AVERAGE TRAJECTORIES UNIFIED Y ERRORS ON SA, HS, AND SWRTEA

	HS	SA	SWRTEA
Average $E'_{trajectory}$ /mm	4.19	21.30	5.56
STD/mm	0.397	2.19	0.804

The results in table I show that our SWRTEA's overall performance will be much better than the SA and can approach to the performance of hardware synchronization based method, which needs to calculate offline.

V. CONCLUSION AND FUTURE WORKS

We presented a novel approach to accurately estimate ball trajectory with concerning the asynchrony between two cameras on humanoid Ping-Pong robot in real-time. The accuracy of our approach has been verified by practical experiments in this paper. The results obtained through those experiments also demonstrated that the performance of our SWRTEA is able to well approach to hardware synchronization based method. In some cases which need the cameras synchronous strictly while the hardware connection is not available, our SWRTEA will be usually helpful.

As our SWRTEA is estimated under the framework of optimization, it is easy to be extended to the conditions with more than two cameras that need to be synchronized.

Future work will be focused on a further integration of the ball trajectory estimation and the pose estimation of the onboard cameras. It will then include the rigid constraint of the stereo cameras equipped on the robot into the optimization of the estimation of the ball trajectories. Thus all the visual servo information, i.e. ball trajectories, pose of the cameras, required by the moving humanoid robot to play Ping-Pong will be obtained simultaneously.

REFERENCES

[1] R. Xiong, Y. Liu, H. B. Zheng, "A humanoid robot for table tennis playing," in *2011 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO),* New York, Oct. 2011, pp. 66–67.

[2] M. Matsushima, T. Hashimoto, M. Takeuchi, and F. Miyazaki, "A learning approach to robotic table tennis," *IEEE Trans. Robotics*, Aug. 2005, vol. 21, pp. 767-771.

[3] Z. T. Zhang, D. Xu, and M. Tan, "Visual measurement and prediction of ball trajectory for table tennis robot," *IEEE Trans. Instrument and Measurement*, Dec. 2010, vol. 59, pp. 3195-3205.

[4] N. W. Liu, Y. F. Wu, X. X. Tan, G. J. Lai, "Control system for several rotating mirror camera synchronization operation," *22nd Int. Congress on High-Speed Photography and Photonics*, Dennis L. Paisley, ALan M. Frank; Eds., Proc. SPIE, May. 1997, vol. 2869, pp. 695-699.

[5] B. Holveck, H. Mathieu, "Infrastructure of the grimage experimental platform: the video acquisition part," *Tech. Rep. RT-0301*, INRIA, Number RT-0301, Nov. 2004.

[6] T. Svoboda, H. Hug. and L. V. Gool, "ViRoom-- low cost synchronized multicamera system and its self-calibration," *In Pattern Recognition, 24th DAGM Symposium*, London, UK: Springer-Verlag, 2002, pp. 515-522.

[7] G. Litos, X. Zabulis, and G. Triantafyllidis, "Synchronous image acquisition based on network synchronization," *IEEE Workshop on Three-Dimensional Cinematography (conj. CVPR)*, 2006.

[8] T. Kanade, H. Saito, and S. Vedula, "The 3D room: Digitizing time-varying 3d events by synchronized multiple video streams," *Tech. Rep. CMU-RI-TR-98-34*, Carnegie Mellon Univ. Robotics Inst., 1998.

[9] Q. Zhao, and Y. Q. Chen, "High-precision synchronization of video cameras using a single binary light source," *J. Electron. Imaging*, 18, 040501, Apr. 2009.

[10] P. K. Rai, K. Tiwari, P. Guha, A. Mukerjee, "A cost-effective multiple camera ision system using firewire cameras and software synchronization," in *Proc. of the 10th Int. Conf. High Performance Computing* (HiPC 2003), Hyderabad, India, Dec. 17-20, 2003.

[11] L. Ahrenberg, I. Ihrke, and M. Magnor, "A mobile system for multi-video recording," in *IEE 1st European Conf. Visual Media Production (CVMP)*, London, UK, Mar. 2004, pp.127-132.

[12] S. N. Sinha, M. Pollefeys, "Synchronization and calibration of camera networks from silhouettes," in *17th Int. Conf. Pattern Recognition (ICPR'04)*, Aug. 2004, vol. 1, pp. 116-119.

[13] J. Yan, M. Pollefeys, "Video synchronization via space-time interest point distribution," in *Proc. Advanced Concepts for Intelligent Vision Systems* (*ACIVS*), 2004.

[14] G. P. Stein, "Tracking from multiple view points: Self-calibration of space and time," in *IEEE Proc. Comput. Soc. Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 1999, vol. 1, pp. 1521–1527.

[15] T. Tuytelaars, and L. V. Gool, "Synchronizing video sequences," in *Proc. IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2004, vol. 1, pp. I: 762-768.

[16] M. Ushizaki, T. Okatani, and K. Deguchi, "Video synchronization based on co-occurrence of appearance changes in video sequences," in *IEEE Int. Conf. Pattern Recogniti. (ICPR)*, 2006, vol. 3, pp. 71–74.

[17] X. Y. Lin, V. Kitanovski, Q. Zhang, and E. Izquierdo, "Enhanced multi-view dancing videos synchronisation," in *Proc. IEEE 13th Int. Workshop on Image Analysis Multimedia Interactive Services (WIAMIS)*, May. 2012, pp. 1–4.

[18] A. Elhayek, C. Stoll, K. I. Kim, H. -P. Seidel, and C. Theobalt, "Feature-based multi-video synchronization with subframe accuracy," in *Proc. Pattern Recognit.*, 2012, pp. 266–275.

[19] L. Zini, A. Cavallaro, F. Odone, "Action-based multi-camera synchronization," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, Jun. 2013, vol. 3, pp. 165-174.

[20] K. Raguse, C. Heipke, "Photogrammetric analysis of asynchronously acquired image sequences," In: Grün, A.; Kahmen, H. (Hrsg.): *Optical 3-D Measurement Techniques VII*, Band II, 2005, pp. 71-80.

[21] Z. Y. Zhang, "A flexible new technique for camera calibration", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Nov. 2000, vol. 22, pp. 1330–1334.

[22] L. Sun, J. T. Liu, Y. S. Wang, L. Zhou, Q. Yang, S. He, "Ball's flight trajectory prediction for table-tennis game by humanoid robot," in *Proc. IEEE Int. Conf. Robotics and Biomimetics (ROBIO)*, Dec. 2009, pp. 2379-2384.

[23] Y. F. Zhang, R. Xiong, "Real-time vision system for a ping-pong robot," *Scientia Sinica Informationis*, May. 2012, vol. 42, pp. 1115-1129.

[24] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems," *Tech. Rep.*, *Informatics and Mathematical Modelling (IMM)*, Technical University of Denmark, Apr. 2004.

[25] C.-P. Lu, G. D. Hager, E. Mjolsness, "Fast and globally convergent pose estimation from video images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Jun. 2000, vol. 22, pp. 610–622.