Robotics and Autonomous Systems 101 (2018) 34–44

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Accurate real-time ball trajectory estimation with onboard stereo camera system for humanoid ping-pong robot

Yong Liu *, Liang Liu

State Key Lab of Industrial Technology, Zhejiang University, Hangzhou 310027, China
Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

HIGHLIGHTS

- We present a real-time ball trajectory estimation approach for ping-pong robot.
- The approach is under the asynchronous observations with ball's motion consistency.
- The approach can achieve the performance as hardware triggered synchronizing method.

ARTICLE INFO

Article history:
Received 2 October 2017
Received in revised form 21 November 2017
Accepted 11 December 2017
Available online 22 December 2017

Keywords:
Humanoid ping-pong robot
Onboard vision
Trajectory estimation

ABSTRACT

In this paper, an accurate real-time ball trajectory estimation approach working on the onboard stereo camera system for the humanoid ping-pong robot has been presented. As the asynchronous observations from different cameras will great reduce the accuracy of the trajectory estimation, the proposed approach will main focus on increasing the estimation accuracy under those asynchronous observations via concerning the flying ball's motion consistency. The approximate polynomial trajectory model for the flying ball is built to optimize the best parameters from the asynchronous observations in each discrete temporal interval. The experiments show the proposed approach can performance much better than the method that ignores the asynchrony and can achieve the similar performance as the hardware-triggered synchronizing based method, which cannot be deployed in the real onboard vision system due to the limited bandwidth and real-time output requirement.

1. Introduction

The task to build the onboard vision system for the humanoid Ping-Pong robot,[1] shown in Fig. 1, is a challenge, as the vision system equipped on the robot will be constant vibration when the arm hitting the ball. Thus the vision system needs to estimate its 6 DOF pose related to the table quickly and localizes the ball's coordinates related to the table by the triangulation and then estimates the trajectory of the ball to further predict the ball's arriving time, velocity and position for the visual servo planning of the arm. In the designated vision system, the multiple-camera pose estimation algorithm [1] is used to estimate the pose in real-time and a Kalman filter [2,3] based estimation method to predict the status of the ball. Then the accurate real-time ball trajectory estimation becomes the critical point for the onboard stereo vision system.

In the normal rallying, the processing of the ball flying through the table only takes less 600 ms. The arm needs to occupy about 400 ms to start its motion and move to the hit point, and the prediction will cost 50 ms, there are only less 150 ms left for the ball's trajectory estimation. Thus two difficulties for the trajectory estimation come up.

The first difficulty is to design the optimal capture software and hardware system that can consider both the accuracy and the capability of real-time performance. In the designated vision system, two cameras[2] working at a resolution of 640 × 480 pixel, 60 frame/s are used. Although a larger frame rate and higher resolution will lead to more dense or accurate observations for the trajectories, it will also slow down the output of the estimation results of the trajectory due to the limitations of the computation and transferring bandwidth. In the designated vision system, it is

* Corresponding author at: Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China.
E-mail addresses: yongliu@iipc.zju.edu.cn (Y. Liu), leonliuz@zju.edu.cn (L. Liu).

[1] The video of our Ping-Pong robot working with onboard vision system is attached in the submission system.

[2] There is a rigid constraint among these two cameras when mounting, the constraint can be calibrated offline.

https://doi.org/10.1016/j.robot.2017.12.004
0921-8890/

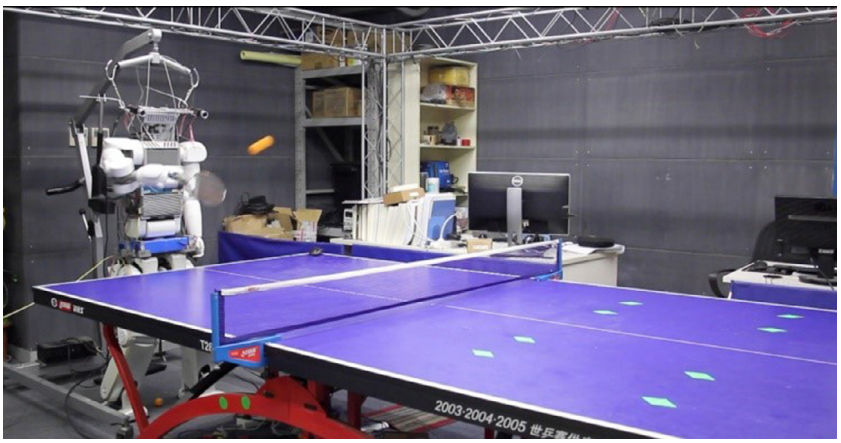


Fig. 1. Our Humanoid Ping-Pong Robot equipped with two onboard cameras with a baseline of 34 cm on the robot's head. It can walk by two legs and rally to human only with its onboard stereo vision system.

difficult to use the hardware triggers to synchronously control the capturing of the stereo images, as the hardware triggered mode needs to synchronize the capturing time strictly and will interrupt the data transferring from the camera to the principal computer on the robot, thus the time of the image pairs processed in the computer will severe lag their captured time and the prediction for the ball's arriving time will be intractable, although it can guarantee the synchronism of the image pairs from different cameras.

Then the second difficulty comes up, how to reduce the errors in trajectory estimation caused by the asynchrony stereo image pairs. Although these two cameras are set at the frame rate of 60 HZ, their real frame rates will be wave around their setting rates and there is also a time interval between those image pairs as the hardware-triggered mode is not available. This small time gap such as an interval less than 1/60 s will also lead to large estimation errors to the trajectories, as there will be a remarkable motion for the fast ball between two asynchronous observations from stereo cameras. Fig. 2 shows the result of a simulation experiment, which can illustrate that the different time gaps of the two cameras will induce significant affections of localization accuracies. The results show that the estimated localization errors from the first 10 pairs of observations[3] may rough close to be the ball's radius, those errors thus will be amplified in the prediction processing and then the robot's arm will fail to hit the ball back to the expected position.

In the following, we will address on the second difficulty and propose a ball trajectory estimation method, which can output accurate trajectories for the humanoid ping-pong robot in real-time, the proposed approach will consider both the asynchrony caused by the software trigger and the ball's motion model simultaneously.

2. Related works

In practical real-time vision tasks [4–9] such as accurate detecting [5] or tracking [6] fast moving targets, the temporal asynchrony problem among the cameras usually is non-ignorable as the tiny temporal intervals among asynchronous cameras may lead to large estimation errors especially when the velocities of the targets are quite large or even the worst condition that one of the camera's frame rate is unknown. Thus almost all those multi-camera vision systems [10–12] need to concern the asynchronism among the cameras for accurate results.

[3] As mentioned previously, only 150 ms, which can be approximated into 10 frames of the camera working with 60 fps, is left to our vision system to capture the images of the ball and estimate the trajectory of the ball, so we only concern the localization errors of the first 10 pairs of the ball's images.

There are three categories of the methods to synchronize multi-camera system: *hardware-triggered synchronization*, *software-triggered synchronization*, and *motion consistency based synchronization*.

The hardware-triggered methods [23–27] use special hardware to connect all cameras physically and control their capturing with a synchronous signal. Then the time gap between different cameras can be reduced to the level of microsecond, which can be suitable for most of the synchronous observations in fast motions. The drawbacks of these approaches are also obvious, the physical connection for those cameras may not be available in some practical applications. In addition, the strict physically synchronization will seriously affect the real-time capability of the image frames, as the hardware signal is treated as a higher priority that will interrupt and delay the transferring of the images to the processing devices. Then these methods are not suitable for those systems, such as the onboard vision system of the ping-pong robot, with the requirement of real-time capability.

Instead of using hardware triggers to send signals, the software-triggered methods [13–15] will use some software synchronization commands to trigger the multi-camera or estimate the time intervals among cameras. The binary light source based synchronization [13] is a typical software-triggered method, which uses a random on–off light source to generate a binary valued signal that is captured by the video cameras, and then the captured binary-valued sequences are matched to estimate the time intervals among cameras. Moreover, some systems [14,15] may use the network messages to synchronize the clocks of the computers directly connected with the cameras and the network latency is also concerned during the synchronizing. Although this kind of approach does not require that all the cameras should be connected physically with a triggering control unit, it also requires additional devices or special connection architectures, e.g. the client/server architecture in [14], to produce the software commands for the synchronizing of cameras.

The motion consistency based methods [16–19] can be regarded as post-processing synchronizations; these methods will utilize the consistency of the motions observed by different cameras in both time and space. These methods need to capture enough image frames of the same motion from different cameras and thus estimate the time gaps among those cameras based on the fact that the timeline of the motion is unique and all the observations from different cameras should be consistent with the unique motion. There are varied consistency features that may be used to estimate the time gaps, such as the dynamic silhouettes of objects [16], the distribution of the correlating space–time interest point [17], the

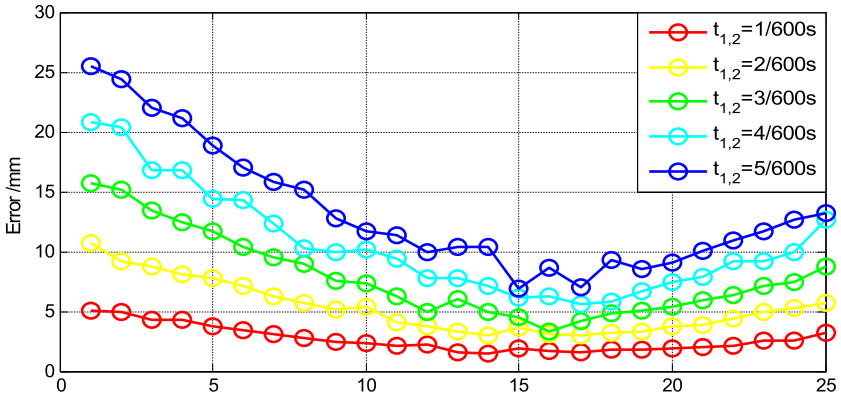


Fig. 2. Simulation example of the impact relations between the capturing time gap of two cameras and the trajectory estimation errors. In this simulation, two cameras are setting with a resolution of 640 × 480, working at 60 frames/s and use the intrinsic and external parameters as same as the real cameras. The *x*-coordinate is the number of the frame pair, which contains two frames from different cameras with a time gap of t_{12}. The *y*-coordinate is the distance error between the estimated position from asynchronous observation and the ground true value.

similarity of the action features [18], the photogrammetric features [19] and the motion model of the object [20] etc. The motion consistency based methods do not require additional synchronization devices, thus can be more flexible comparing to the previous two kinds of methods. The proposed synchronizing approach in this paper may be categorized into the motion consistency based synchronizations. As there are fewer image features, normally the observations can only obtain the ball and the reference points in the Ping-Pong table by color segmentation, the flying ball's physical model will be employed as the motion consistency.

3. Trajectory estimation with asynchronous observations

3.1. Camera model used in the proposed approach

In the proposed onboard vision system, the intrinsic parameters and external parameters of those two cameras are already calibrated, then the ball center can be recovered by the following perspective projection model [20]:

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

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A_{3\times 4}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{2}$$

$(X_w, Y_w, Z_w)^{\mathrm{T}}$ and $(X_c, Y_c, Z_c)^{\mathrm{T}}$ denote point P in world coordinate and camera coordinate respectively, and P's image is denoted as $(u, v)^{\mathrm{T}}$. $\mathbf{A_{3\times 4}}$ is camera's intrinsic matrix. $\mathbf{R}$ and $\mathbf{t}$ denote the camera's external parameters.

Based on formula (2):

$$\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A_{3\times 4}} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \left(\begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \right)^{-1} \mathbf{A}_{3\times 4} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{4}$$

Assuming $\mathrm{H} = \mathbf{A}_{3\times 4} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$, $\mathrm{K} = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}$, using $\tilde{m} = (u, v, 1)^{\mathrm{T}}$ and $\tilde{M} = (X_w, Y_w, Z_w, 1)^{\mathrm{T}}$ to denote the homograph coordinate of the image point and the world homograph coordinate of the point respectively, then:

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

3.2. Motion model of the flying ping-pong ball

As the proposed onboard vision system needs to estimate the accurate ball trajectory quickly before the ping-pong ball flies over a quarter length of the table. Thus the motion model of the flying ball should be able to obtain accuracy trajectory as well as costing with tiny computation complexity. The forces [21,22] acting on the flying ball is shown in Fig. 3.

The world coordinate is located at the center of the table, there are four forces, i.e. the gravity (F_g), the air resistance (F_s), the air buoyancy (F_b), and the Magnus force (F_m) acting on the flying ball. As the ping-pong bat used by our robot is pure wooden, the ball will be barely spinning when hitting, then the Magnus force can be regarded as zero. F_g and F_b are always along the vertical direction and have the opposite direction and constant magnitude, thus it can be denoted as an unify $F_{Vertical} = F_g - F_b$. The air resistance is assumed to be always contrary to the ball's flying direction and proportional to the ball's velocity. In the following approximated motion model, the air resistance is not involved directly; however, the air resistance can be implanted during the parameters estimation.[4]

Using $g, \ddot{Q}(t), \dot{Q}(t), Q(t)$ to denote the gravity acceleration, the ball's acceleration, velocity, and position at moment t, then the ball's approximated motion model in the world coordinate is presented:

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

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

[4] The proposed approach uses an approximate model to fit the motion in a short interval, in this condition, these estimated parameters fitted for the approximated motion model already contain the affection of the air resistance.

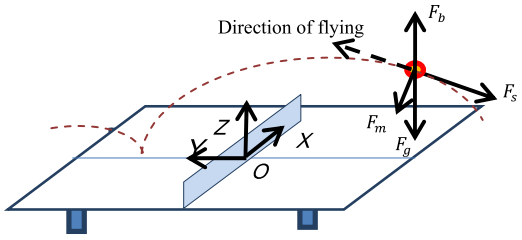


Fig. 3. The world coordinate and forces on the flying ball.

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

Obviously, it is a polynomial approximated model for the flying ping-pong ball. Here t_0 is the initial moment. It only needs to estimate seven parameters, i.e., the initial position of the ball $X(t_0), Y(t_0), Z(t_0)$, the initial velocity of the ball $\dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0)$ and the gravity acceleration g.

3.3. Trajectory estimation

In the proposed method, we assume the capturing cycles of both cameras are stable although the capturing cycles may be varied and unknown. The time intervals (or capturing cycle) among two successive frames are denoted as t_1 and t_2 for the left camera and right camera respectively. In the proposed onboard vision system, it is difficult to use the hardware-triggered method to synchronize the capturing time of both cameras due to the bandwidth limitation to output and process the frames in real-time, then these two cameras will work asynchronously, $t_{1,2}$ is used to denote the time gap between two cameras. Furthermore, the time gaps will be shifty with the variations of the overload conditions in the operating system and the overload of the CPU and Bus etc. Thus $t_{1,2}$ is not a stable constant in practice and will be changed related to the system overloads. In the proposed approach, the image capturing time gaps, $t_{1,2}$, is assumed as a constant in a short time quantum, which is reasonable in practice. Thus $t_{1,2}$ can be regarded as a stable value in a very short observation.

The flying ball's homograph coordinate based on the motion model can be given as follow:

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

m_1^i and $\tilde{Q}(t_{m_1^i})$ $(i = 1, 2, 3, \ldots, k)$ denote the image point sequences and their corresponding world coordinate from the left camera, and use m_2^j and $\tilde{Q}(t_{m_2^j})(j = 1, 2, 3, \ldots, q)$ to denote the image point sequences and their corresponding world coordinate from the right camera. Although the images from different cameras are not synchronous, the capturing sequence for the same camera can be in a right temporal order, which means the capturing time of m_1^i will be earlier than the capturing time of m_1^{i+1}.

Then the following formula that can transform the world coordinate of the ball to its corresponding image coordinate can be obtained, based on formula (5):

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

In the above formula, $\tilde{m}_1^i$ and $\tilde{m}_2^j$ are denoted as the left and right cameras' re-projections from the ball's world coordinates respectively, and the time moment at the first frame coming from left camera is t_0. The trajectory estimation can be regarded as a discrete optimization problem shown in formula (11), which is to use many parameterized sub-trajectories in a short time quantum to represent the whole trajectory, and once the parameter sets for every sub-trajectories are obtained, the whole trajectory of the flying ball can be estimated.

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

Based on the formula (6)–(11), the optimization need to solve nine parameters, which are denoted with a parameter set E $(E = \{X(t_0), Y(t_0), Z(t_0), \dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0), g, t_2, t_{1,2}\})$. In the proposed approach, the Levenberg–Marquardt (LM) optimization method is used to solve these parameters, and the initial settings for these nine parameters are also discussed in the following section.

As the onboard vision system for ping-pong robot needs to process the camera observations and output predictable results for the flying ball in real-time, the trajectory cannot be optimized until all the observations obtained. Following the idea of estimating the discrete sub-trajectories to estimate the whole trajectory, we present an algorithm to estimate the parameter set E for each sub-trajectories and also iterate to optimize the sub-trajectory's parameters with a slider window policy.

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

3.4. Discussing and setting on the algorithm

As each slider window in algorithm 1 will confirm a position state for the ball's trajectory, it is easy to generate the ball's trajectory, which can be represented by a sequenced discrete position states located in the timeline of both cameras.

As mentioned in previous section, the formula (11) in the SWRTEA is solved by the LM optimization method, thus the reasonable initial values for those parameters are required. There are nine parameters in E. The initial values of the gravity accelerate is set as $g = 9.8\ \mathrm{m/s^2}$, and $t_{1,2} = \frac{t_1}{2}$, which means the initial value of time gap is half cycle of the left camera, t_2' initial value is set the same as t_1. As the time beginning from the first iteration, which means $\tilde{Q}(t_0) = \tilde{Q}(0)$ at the observation of m_1^1. The proposed approach will first assume the observations from both cameras are synchronous, thus the initial values of $X(t_0), Y(t_0), Z(t_0)$ can be calculated from m_1^1 and m_2^1. Then the initial values of $\dot{X}(t_0), \dot{Y}(t_0), \dot{Z}(t_0)$ can be obtained by derivation the positions of two successive observations from both cameras which are assumed to be synchronous. After the first iteration, those new estimated parameters are used as the initial values in the next optimization iteration.

There are two additional parameters, i.e. H and K ($\mathrm{H}_1^i, \mathrm{K}_1^i$ for the left camera, $\mathrm{H}_2^j, \mathrm{K}_2^j$ for the right camera), which need to be mentioned. These two parameters are consisting with the cameras' intrinsic matrix and their corresponding external parameters at each observation. The intrinsic matrix of camera is calibrated offline, while the external parameters should be updated for each observation. In the proposed onboard vision system for the ping-pong robot, there are eight landmark points with known coordinates placed on the table and the *Perspective-n-Point* method is used to estimate the external parameters of the camera in real-time.

As there are less than 15 points in the short time interval when the balls fly over 1/4 of the table, the slider window size s will also be less than 15, thus there are only dozens of dimensions in the optimization, and the temporal computation for SWRTEA is

Algorithm 1: SWRTEA

Input: 1. Slider window size S, Successive images of the ball's central points $M_{left} = \{m^i_1, i = 1,2,3 \ldots\}, M_{right} = \{m^j_2, j = 1,2,3 \ldots\}$

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

1. CB=1, $n_{left} = 1, n_{right} = 1$;
2. Sort M_{left}, M_{right} by timestamp to obtain $M = \{m^k, k = 1,2,3 \ldots\}$;
3. *While*$(CB < |M| - S)$
4. Pop S *Successive images* $(m^{CB}, \ldots, m^{CB+S-1})$ from M, there are p images from M_{left} and q images from M_{right}, $p + q = S$;
5. $E_{CB} = \arg Min_E \left(\sum_{i=n_{\text{left}}}^{n_{\text{left}}+p-1} ||\widetilde{m}^i_{left} - m^i_{left}||^2 + \sum_{j=n_{\text{right}}}^{n_{\text{right}}+q-1} ||\widetilde{m}^j_{right} - m^j_{right}||^2\right)$;
6. CB++;
7. *if* $m^{CB-1} \in M^i_{left}$
8. n_{left}++;
9. *else* n_{right}++;
10. *end while*

also quite low as the parameters that need to be estimated can be initialized very close to the optimal values. Thus step 5 in the algorithm can reach the extremum with only several iterations, and guarantee the real-time performance.

Furthermore, the time-consuming of the LM optimization is not significantly relying on the window size, as there are only nine parameters for every point pairs from different cameras, thus the optimization temporal costs for window size 1 to 15 are almost the same. Then the time-costs of varied window sizes are not the main issue that should be concerned to choose the window size. In the above algorithm, the size of the slide window is used as a parameter to adjust the fitting results. To achieve real-time estimation results, our model for the ball's flying trajectory is an approximation version of the real model, which simplifies many complex parameters. Then it needs to fit proper parameters for that simplified model with the real data. The slide window size may be regarded as an important parameter in model fitting to decide in which time interval the simplified model can be most approximated to the real observation results, as the size of the slide window is equal to the length of the time interval used in the optimization. That is why the results with slide window size of 5 will be better than that with window size of 10 in the experiments of Fig. 4.

4. Experiments and discussion

This section will present comprehensive experiments in both simulation and real ping-pong robot system to evaluate the performance of the proposed method and other state-of-the-art method. This section will compare the proposed method with the trajectory estimation method[5] that directly calculates the positions of the balls from a pair of images captured by two different cameras without concerning the asynchrony between the cameras. This method [22] is denoted as SA (Synchronizing on Asynchronous condition) in the following experiments.

As the continuous trajectory of the ball is hard to be quantitatively evaluated, a discrete metric named *timeline error* to evaluate the accuracy of the estimation is defined. And a discrete representation, $Q(t) = [X(t), Y(t), Z(t)]^{\mathrm{T}}$, is used to denote a piece of the trajectory. The corresponding ground true trajectory is denoted as $Q'(t) = [X'(t), Y'(t), Z'(t)]^{\mathrm{T}}$. Then the errors can be calculated as follows:

$$
\begin{aligned}
E_x &= \left(\frac{1}{n}\sum_{i=1}^{n} |x_i(t_i) - x'_i(t_i)|^2\right)^{\frac{1}{2}} \\
E_y &= \left(\frac{1}{n}\sum_{i=1}^{n} |y_i(t_i) - y'_i(t_i)|^2\right)^{\frac{1}{2}} \\
E_z &= \left(\frac{1}{n}\sum_{i=1}^{n} |z_i(t_i) - z'_i(t_i)|^2\right)^{\frac{1}{2}} \\
E_m &= \frac{1}{n}\sum_{i=1}^{n} \left|\left|Q(t_i) - Q'(t_i)\right|\right|_F
\end{aligned}
\tag{12}
$$

Where n is the number of points in that piece of trajectory, if the points number is equal to the size of the slider window, that is $n = s$, then the quantitative error metric for each sub-trajectories output by algorithm 1 can be obtained.

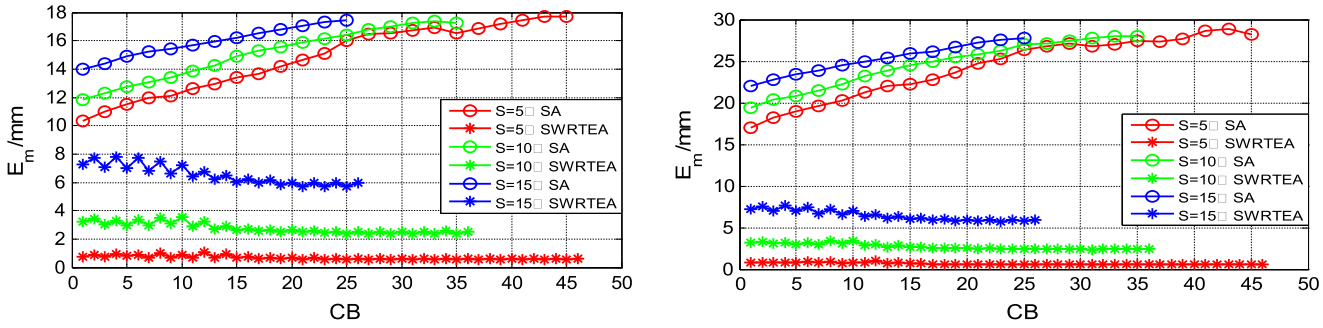
4.1. Simulation experiments

In the simulation experiments, the ground true trajectories with the model in [21] are first generated, the model concerns almost all the possible factors when the Ping-Pong ball flying. The experiments also simulate two cameras working at 60 HZ with varied time gaps, and obtain their estimation results with the proposed approach and SA method. Then the errors can be calculated with formula (12) for both methods working on different time gaps. Fig. 4 shows the experimental results of trajectory estimation errors comparing with the ground true trajectories.

In Fig. 4, the results indicate that the performances of SWRTEA are always better than the performances of SA. The experiment of Fig. 4 has employed three window sizes, i.e., 5, 10, and 15, and the results show the size of 5 can obtain the best performance, results of SWRTEA working at window size 5 on $t_{1,2} = 3/600$ s and $t_{1,2} = 5/600$ s can be lower than 2 mm comparing with the ground truth.

From the results in Fig. 4, it also proves the analysis of Section 3.4, that the slider window's size will also be related to the accuracy of estimation, and the larger size will not lead to better accuracy in our model as it is an approximate model. To achieve real-time estimation results, the proposed approach simplifies many

[5] The same ball motion model for both methods is used to compare fair.



(a) $t_{1,2} = 3/600$ s. (b) $t_{1,2} = 5/600$ s.

Fig. 4. The comparison results of the simulation experiments on trajectory errors estimated by SA and SWRTEA with varied window sizes. Here CB is the current beginning of the slider window. The circles are denoted as the E_m of SA, and stars are denoted as the E_m of SWRTEA.

Table 1
The frame rate settings.

Experimental group I	Left camera	60 HZ
	Right camera	80 HZ
Experimental group II	Left camera	60 HZ
	Right camera	40 HZ
Experimental group III	Left camera	60 HZ
	Right camera	60 HZ

Table 2
Average errors for SA and SWRTEA under different $t_{1,2}$.

		Average $E_{trajectory}$/mm	STD/mm
$t_{1,2} = 1/600$ s	SA	3.42	1.54
	SWRTEA	0.830	0.122
$t_{1,2} = 2/600$ s	SA	6.74	3.14
	SWRTEA	0.803	0.122
$t_{1,2} = 3/600$ s	SA	10.06	4.72
	SWRTEA	0.793	0.141
$t_{1,2} = 4/600$ s	SA	13.21	6.23
	SWRTEA	0.783	0.145
$t_{1,2} = 5/600$ s	SA	16.42	7.75
	SWRTEA	0.767	0.130

complex nonlinear parameters, thus the model used to estimate the ball's flying trajectory is reduced to an approximation linear version of the real nonlinear physical model. Then it needs to fit proper parameters for that simplified linear model from the real data. The slide window size may be regarded as an important parameter in model fitting to decide in which time interval the simplified linear model can be most approximated to the real nonlinear observation results, as the size of the slide window is equal to the length of the time interval used in the optimization. Obviously, a shorter time interval fitting with the approximate linear model will be closer to the real nonlinear model. That is why the results with slide window size of 5 will be better than that with window size of 10 in the experiments of Fig. 4 (if the window size is less than 5, there will be less constraint for the optimization formula (11) and it will also lead to a worse results). So the window size is set as 5 in the following experiments.

In the second simulation experiment, the slider window size is set as 5, and concerning the detailed performances of these two approaches on different $t_{1,2}$. The results are shown in Fig. 5. The results in Fig. 5 illustrate that although $t_{1,2}$ is varied, the errors of SWRTEA will converge to a small value, while the errors of SA will increase significantly with the value of $t_{1,2}$ increasing.

The third simulation experiment will evaluate SWRTEA working on the condition that one of the camera's frame rate is unknown. The simulation experiment sets three experimental groups shown in Table 1, and the experiment will assume the frame rate of right camera is unknown.

The experimental results on the unknown frame rate camera are given in Fig. 6, and the trajectory estimation results are shown in Fig. 7. The results show the estimation errors on frame rate are quite low, even in the worst condition of group II, the maximal estimation error is slightly larger than 1%. And the corresponding trajectory estimation errors are also suitable small, that the maximal error is less than 3 mm in all the groups.

The overall performance of the proposed approach will be further concerned. The average trajectories estimation errors under different $t_{1,2}$ are calculated. In our task, there is only less 150 ms left for the vision system to estimate the trajectory, and the ball just can fly over a quarter of the table within such an interval. So only the trajectory's section that the ball flies into the table and flies over a quarter of the table, i.e. $y \in [-1375, -700]$, is calculated. For each time gap, the experiment simulates 110 trajectories with the constraints of $x \in [-500, 500], z \in [80, 400]$. For each trajectory, the corresponding error is averaged on each slider window E_m to obtain the average trajectory error $E_{trajectory}$ for each single trajectory, and then average the $E_{trajectory}$ of those 110 trajectories. The results are shown in Table 2. And the results show that SWRTEA can achieve impressive better performance than SA when considering their average estimation error on the same trajectories.

4.2. Experiments on practical vision system

The experiments on our ping-pong robot hardware system are also carried out, shown in Fig. 8, which having two cameras with a rigid baseline of 34 cm, both cameras work at 60 HZ and output the image with a resolution of 640×480. An external stereo vision system is also deployed to obtain the ground true trajectories for evaluation; the external vision system has two high speed cameras put on the ceiling of the table. Both external cameras work at 120 HZ, and also are synchronized by the hardware-triggered signals, thus the ground true trajectories can be calculated offline from those image pairs obtained by the external vision system.

In the experimental scene of Fig. 8, it is almost impossible to align the observations from the external and onboard vision systems. Thus the calculation for formula (12) is not available. So this experiment aligns the continuous trajectory[6] generated by the observations from low rate onboard vision system with the discrete observations obtained from the fast rate external vision system in the Y direction. That is to choose the positions, which have

[6] The continuous trajectory can be generated with interpolation based on the parameterized formula (8).

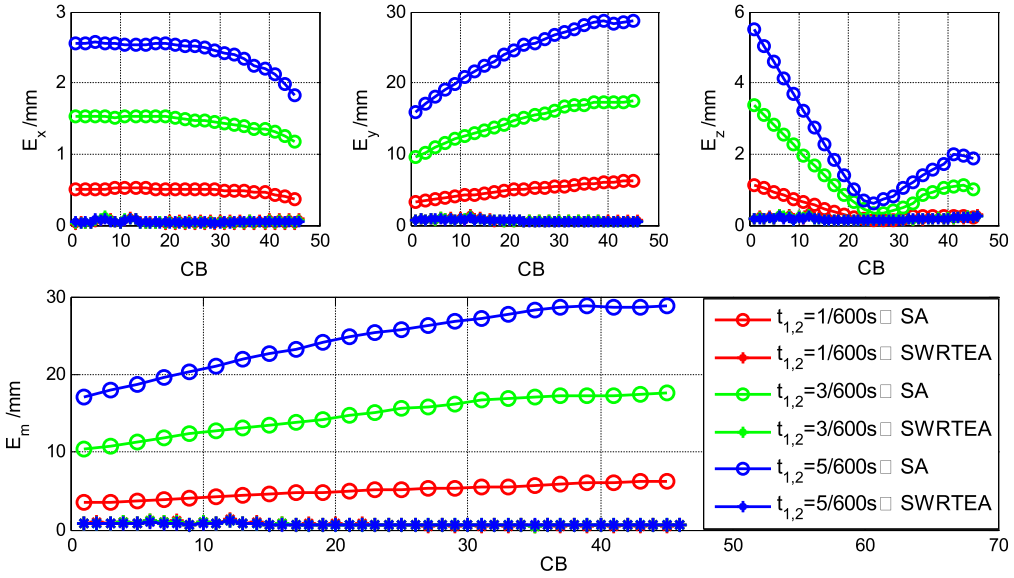


Fig. 5. The comparison results of the simulation experiments on trajectory errors estimated by SA and SWRTEA with different $t_{1,2}$. The circles are denoted as the E_m of SA, and stars are denoted as the E_m of SWRTEA.

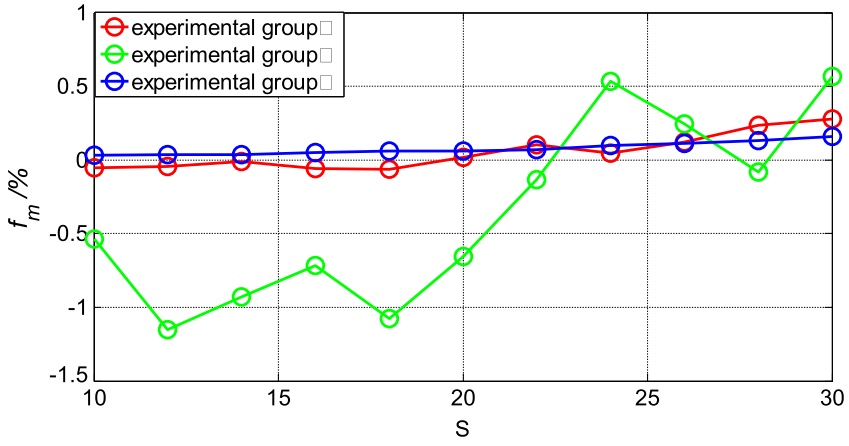


Fig. 6. Simulation comparison results of the frame estimation errors on SWRTEA with different window sizes. Here $f_{\mathrm{m}} = \frac{f_{\mathrm{estimated}} - f_{\mathrm{real}}}{f_{\mathrm{real}}} \times 100\%$.

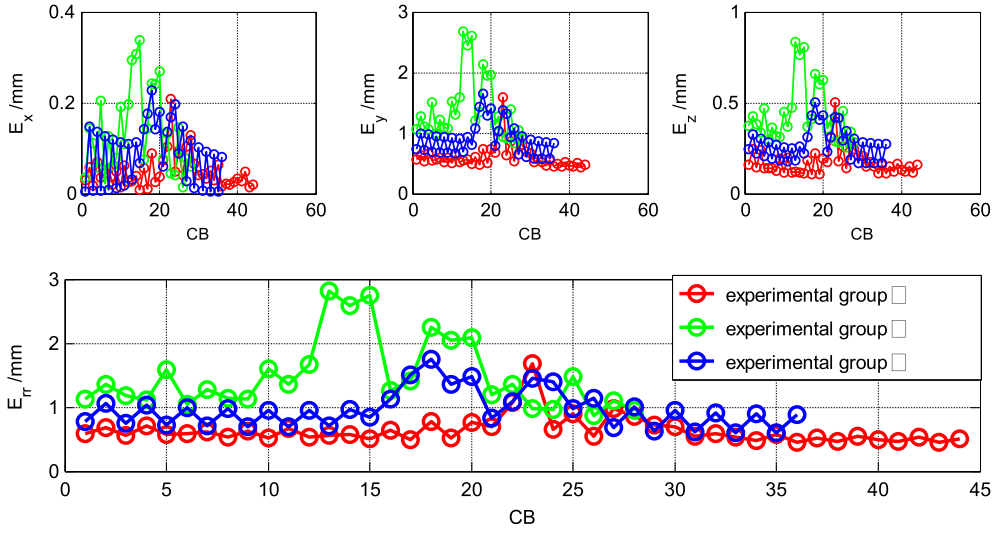


Fig. 7. Simulation comparison results of the trajectory estimation errors on SWRTEA for the three experimental groups.

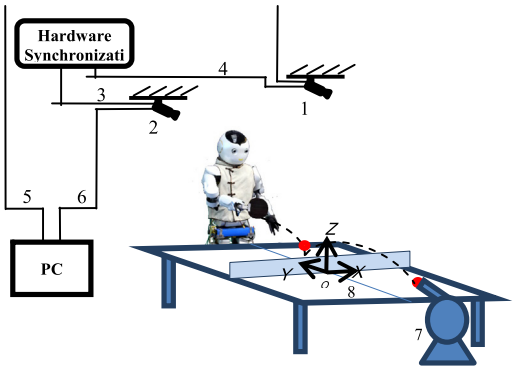


Fig. 8. The experimental setting in our practical vision system for the ping-pong robot. 1, 2: Two external cameras working at the frame rate of 120 HZ installed on the ceiling; 3, 4: The hardware trigger to synchronize those two external cameras; 5, 6: Image data transferring from cameras to computer for offline processing; 7: Pitching machine launching repeatable trajectories of the flying ball; 8: World coordinate in the center of the table.

the same Y coordination with the discrete observations obtained from the fast rate vision system, from the continuous trajectory estimated by the low rate vision system, and then compare their biases on X and Y directions. Then the error on a piece of trajectory can be calculated as follow:

$$E'_x = \left(\frac{1}{n}\sum_{i=1}^{n} |x_i(Y_i) - x'_i(Y_i)|^2\right)^{\frac{1}{2}}$$
$$E'_z = \left(\frac{1}{n}\sum_{i=1}^{n} |z_i(Y_i) - z'_i(Y_i)|^2\right)^{\frac{1}{2}} \quad (13)$$
$$E'_m = \frac{1}{n}\sum_{i=1}^{n} \left||Q(Y_i) - Q'(Y_i)|\right|_F$$

Here Y_i is sampled from the ground true trajectory that observed by the external stereo vision system, there are n sampled points in that ground true trajectory. And the error calculated by formula (13) is called *unified Y error*.

In the first real experiment, the real observation image sequences are employed, and these images are captured by the external stereo vision system working at the frame rate of 120 HZ and strictly synchronized. In the experiment, the stagger frames from both external cameras are selected. Thus these stagger frames construct two asynchronous image sequences whose frame rates are all 60 HZ and have a time gap of 5/600 s between each other. Both SA and SWRTEA methods are then executed on those two constructed image sequences, and calculate their *unified Y errors* and *timeline errors* shown in Fig. 9 and Fig. 10 respectively.

The results in Fig. 9 are calculated by formula (13) as the ground truth at the sample point Y_i can be calculated by the synchronized frames of the external stereo system in 120 HZ. In both error metrics, these two methods under varied window sizes, i.e., 5, 10, and 15 are also be evaluated. As the time stamp for each observation in the asynchronous image sequences can be corresponding to the observation used in the ground truth, the *timeline error* can also be calculated with formula (12) shown in Fig. 10.

According to Fig. 9 and Fig. 10, the results show the proposed method can achieve much better performance than SA on varied window sizes. Although the results on Fig. 9, Fig. 10 also show the absolute error calculated by the *unified Y error* is less than the error calculated by the *timeline error* on the numerical value,[7] the results on both figures indicate the same consistent performance for those two methods, SWRTEA and SA. Comparing the results on Fig. 10 and Fig. 4, it can be observed that the curves on both Fig. 10 and Fig. 4 are almost similar, which also indicate the consistency between the simulation results and the real experimental results. The *timeline errors* of the SWRTEA in Fig. 10 are all less than 3 mm on the condition of window size 5, thus the proposed method can satisfy the accuracy requirement of the onboard vision system for humanoid ping-pong robot, as the error is only 1/10 of the ball's radius.

In the second real experiment, the onboard vision system of our ping-pong robot is employed to obtain the trajectories of the ball, and the offline processing results are used from the captured images captured by the external stereo vision system as the ground truths. In that experiment, the slider window size is set as $s = 5$, and each sample point Y_i involving in the error calculation is extracted from the external vision system' observations based on formula (13). The results are given in Fig. 11.

Fig. 12 gives out the estimated trajectories by SA and the proposed method compared with the ground truth trajectory obtained from the external vision system as shown in Fig. 8. Here Fig. 12 only plots the Y-Z projections of the estimated trajectories. The results indicate that the proposed method can perform much better than SA and is almost overlapped with the ground true trajectory.

In the third experiment, the camera system working on varied frame rates will be evaluated. Two groups of camera settings, which assume the frame rate of the right camera in each group is unknown, are used. The trajectories generated by these two groups of camera settings are compared with the ground truth trajectory obtained from the external vision system as shown in Fig. 8. The trajectories estimated by SWRTEA and SA methods are given in Fig. 13. The corresponding trajectory estimation errors are given in Fig. 14. The results show that the SWRTEA can perfect approach the ground truth and much better than the results of SA, although the frame rate of right camera is unknown.

There is also a further experiment to evaluate the overall performances of varied method working in real onboard vision system. This experiment only considers the section of the trajectory from the start point that the ball flies into the table to the end point that the ball flies over 1/4 of the table. 48 trajectories with our ping-pong robot's onboard vision system are captured firstly; all those trajectories are launched by a ping-pong ball pitching machine. The *unified Y errors* for these 48 trajectories on SA and SWRTEA can be calculated respectively. And the average error for each method is also calculated. The results are given in Table 3. Here the ground true trajectories are captured by the external stereo vision system, which use two high-speed cameras with a frame rate of 120 HZ. The performance of the proposed approach is then compared with the hardware triggered synchronization method, which cannot provide the results online and needs to be processed offline. The same pitching machine is used to repeat another 36 trajectories with the exactly same setting as the previous 48 trajectories. These new 36 trajectories are observed by the same humanoid robot vision system, while the captured image frames from these two cameras are synchronized by hardware triggered control signals with a frame rate of 60 HZ. The same *unified Y error* based evaluation metric from those synchronous frames are calculated offline, and HS (hardware triggered synchronization) is used to denote this method's average errors in Table 3.

The results in above table indicate that the proposed approach can achieve much better overall performance than the SA method. The results also show the performance of the SWRTEA can approach to the performance of the hardware-triggered synchronization method, which cannot output estimation results in real-time and requires offline calculation.

[7] As the *unified Y error* involves only the errors on the directions of X and Z, the error on direction Y is indirectly coupled. While the *timeline error* concerns all the errors in directions of X, Y and Z.

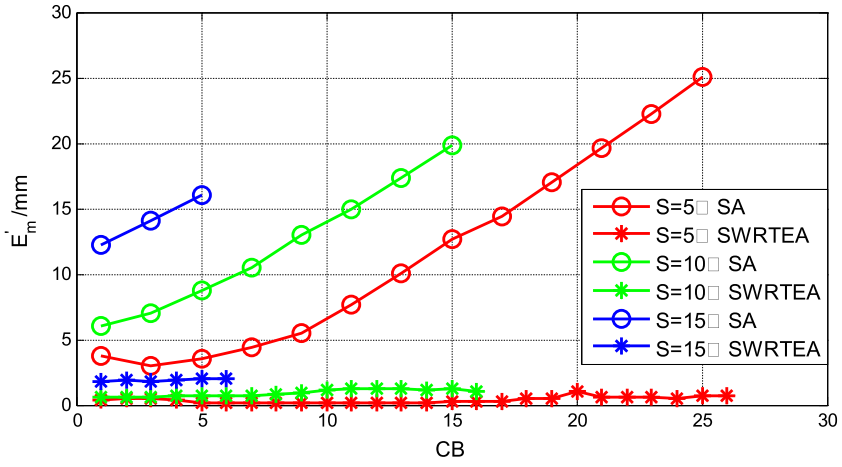


Fig. 9. Real experimental results for both trajectory estimation methods evaluated by *unified Y errors* under different slider window sizes.

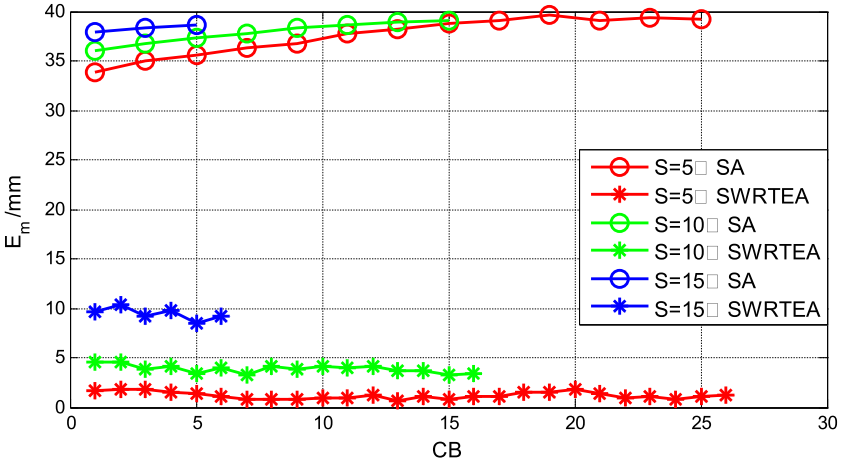


Fig. 10. Real experimental results for both trajectory estimation methods evaluated by *timeline errors* under different slider window sizes.

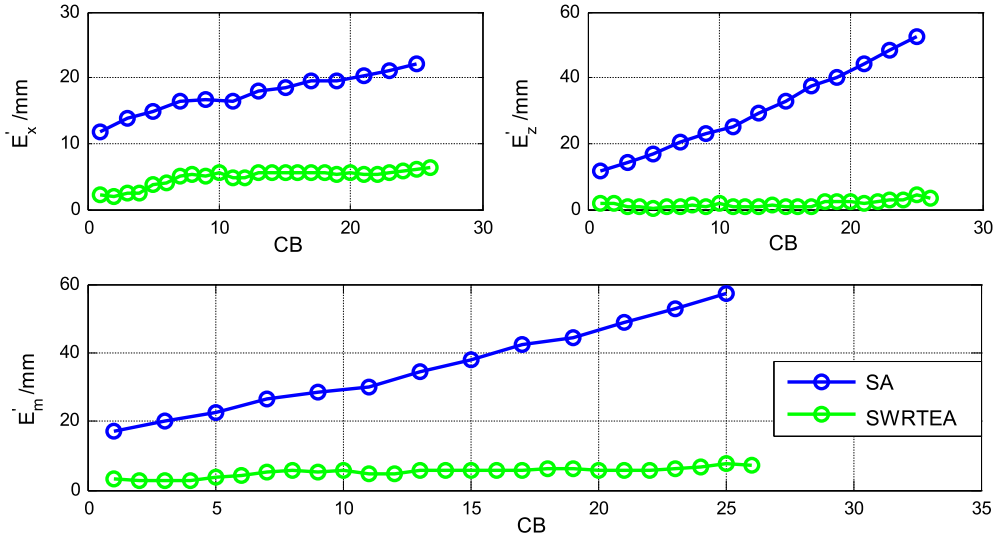


Fig. 11. Real experimental results of the trajectory estimation errors with onboard vision system, window size is 5.

5. Conclusion

This paper presents an accurate real-time ball trajectory estimation approach, which can solve the problem of asynchronous observations among different cameras by concerning the flying ball's motion consistency, with the onboard stereo camera system equipped on the humanoid ping-pong robot. Both simulation experiments and practical experiments are designed to

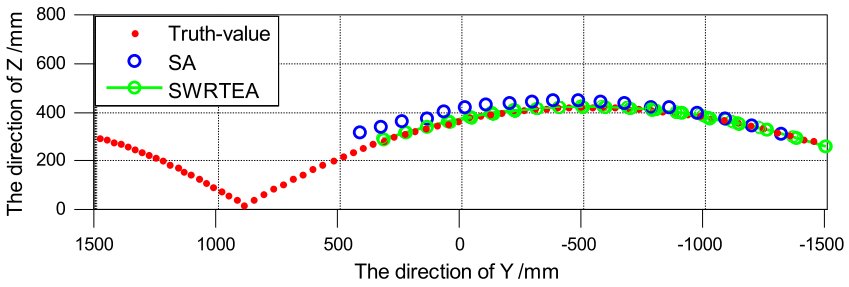
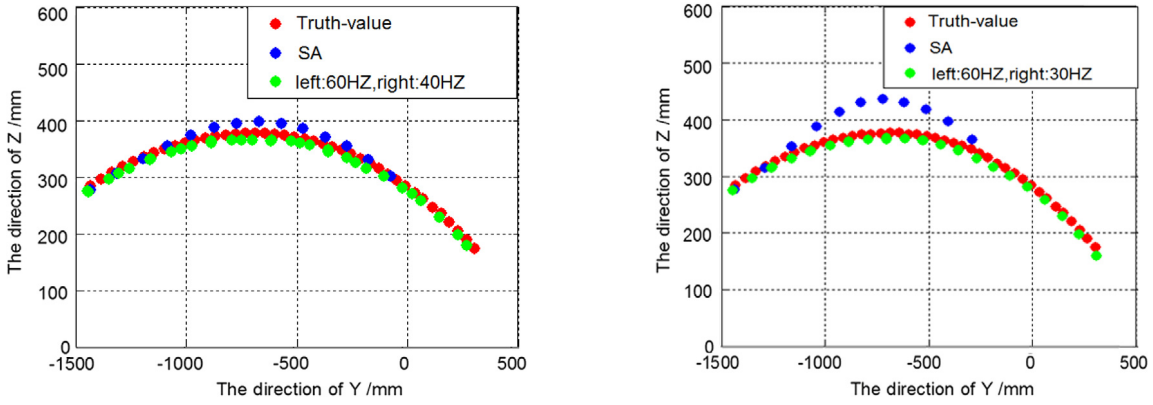
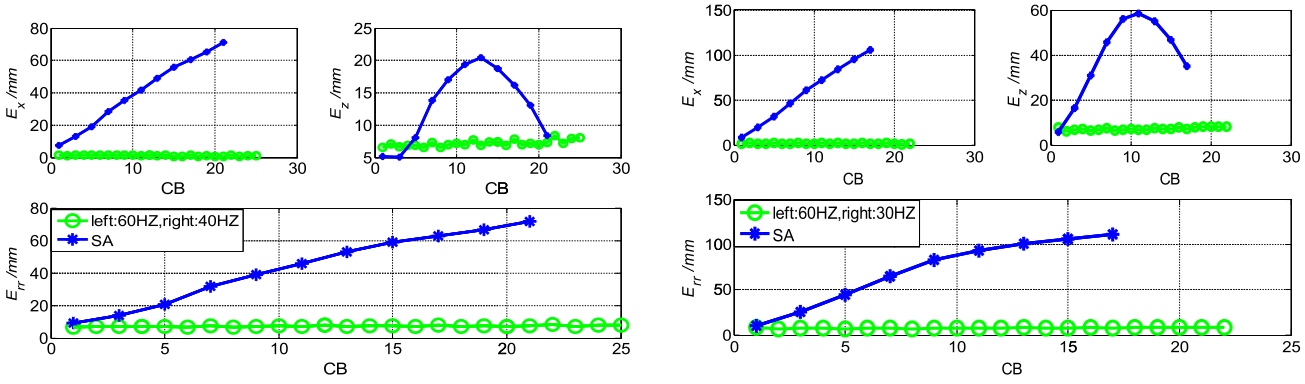


Fig. 12. Comparison results of the trajectories obtained by SA, SWRTEA and ground truth. The red dots denote the ground truths, blue circles denote trajectory estimated by SA, and green circles denote trajectory estimated by SWRTEA. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) Group I: left 60 HZ, right 40 HZ. (b) Group II: left 60 HZ, right 30 HZ.

Fig. 13. Trajectory comparison results on SWRTEA, SA and ground truth.



(a) Group I: left 60 HZ, right 40 HZ. (b) Group II: left 60 HZ, right 30 HZ.

Fig. 14. Real comparison results of the trajectory estimation errors on SA and SWRTEA for the two groups.

Table 3
The average estimation results of SA, HS, and SWRTEA on unified Y errors.

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

evaluate the performance of the proposed approach comparing with other state-of-the-art methods. The experimental results show the proposed method can perform much better than the method that ignores the asynchrony and can achieve the similar performance as the hardware-triggered synchronizing based method.

The proposed approach is built on the framework of optimization, thus it is able to be implemented to the cases with more asynchronous cameras. Furthermore, the optimization framework of the proposed approach can also support the conditions that only one of the camera's frame rate is known, which means the frame rates of other cameras are unknown, as long as there are enough of observations from all the cameras.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant U1509210.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.robot.2017.12.004.

References

[1] Y. Liu, R. Xiong, Y. li, Robust and accurate multiple-camera pose estimation toward robotic applications, Internat. J. Adv. Robot. Syst. 11 (9) (2014) 153.
[2] Y. Liu, R. Xiong, et al., Stereo visual-inertial odometry with multiple Kalman filters ensemble, IEEE Trans. Ind. Electron. 63 (10) (2016) 6205–6216.
[3] F. Auger, M. Hilairet, et al., Industrial applications of the Kalman filter: A review, IEEE Trans. Ind. Electron. 60 (12) (2013) 5458–5471.
[4] S. Zhao, Z. Hu, et al., A robust real-time vision system for autonomous cargo transfer by an unmanned helicopter, IEEE Trans. Ind. Electron. 62 (2) (2015) 1210–1219.
[5] I. Mario, G.D. Sergio, An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems, IEEE Trans. Ind. Electron. 59 (8) (2012) 3286–3298.
[6] B. Wu, C.C. Kao, et al., An adaptive neural-fuzzy approach for object detection in dynamic backgrounds for surveillance systems, IEEE Trans. Ind. Electron. 61 (8) (2014) 4228–4237.
[7] Z.Q. Cao, X.L. Liu, N. Gu, S. Nahavandi, D. Xu, C. Zhou, M. Tan, A fast orientation estimation approach of natural images, IEEE Trans. Syst. Man Cybern. 46 (11) (2016) 1589–1597.
[8] J.H. Du, C. Mouser, W.H. Sheng, Design and evaluation of a teleoperated robotic 3-D mapping system using an RGB-D sensor, IEEE Trans. Syst. Man Cybern. 46 (5) (2016) 718–724.
[9] Ç. Aytekin, Y. Rezaeitabar, S. Dogru, İ. Ulusoy, Railway fastener inspection by real-time machine vision, IEEE Trans. Syst. Man Cybern. 45 (7) (2015) 1101–1107.
[10] M. Matsushima, T. Hashimoto, M. Takeuchi, F. Miyazaki, A learning approach to robotic table tennis, IEEE Trans. Robot. 21 (2005) 767–771.
[11] X. Zhou, L. Xie, Q. Huang, S.J. Cox, Y. Zhang, Tennis ball tracking using a two-layered data association approach, IEEE Trans. Multimedia 17 (2) (2015) 145–156.
[12] H. Su, Z. Fang, D. Xu, M. Tan, Trajectory prediction of spinning ball based on fuzzy filtering and local modeling for robotic ping–pong player, IEEE Trans. Instrum. Measur. 62 (2013) 2890–2900.
[13] Q. Zhao, Y.Q. Chen, High-precision synchronization of video cameras using a single binary light source, J. Electron. Imaging 18 (2009) 040501.
[14] 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.
[15] L. Ahrenberg, I. Ihrke, 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.
[16] S.N. Sinha, M. Pollefeys, Synchronization and calibration of camera networks from silhouettes, in: 17th Int. Conf. Pattern Recognition (ICPR'04), vol. 1, Aug. 2004, pp. 116–119.
[17] J. Yan, M. Pollefeys, Video synchronization via space–time interest point distribution, in: Proc. Advanced Concepts for Intelligent Vision Systems (ACIVS), 2004.
[18] L. Zini, A. Cavallaro, F. Odone, Action-based multi-camera synchronization, IEEE J. Emerg. Sel. Top. Circuits Syst. 3 (2013) 165–174.
[19] K. Raguse, C. Heipke, Photogrammetric analysis of asynchronously acquired image sequences, in: A. Grün, H. Kahmen, (Hrsg.): Optical 3-D Measurement Techniques VII, Band II, 2005, pp. 71–80.
[20] Q. Xie, Y. Liu, et al., Real-time accurate ball trajectory estimation with asynchronous stereo camera system for humanoid ping-pong robot, in: IEEE International Conference on Robotics & Automation (ICRA), May 31 - June 7, 2014. Hong Kong, China, pp. 6212–6217.
[21] 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.
[22] Y.F. Zhang, R. Xiong, Real-time vision system for a ping-pong robot, Sci. Sin. Inf. 42 (2012) 1115–1129.
[23] N.W. Liu, Y.F. Wu, X.X. Tan, G.J. Lai, Control system for several rotating mirror camera synchronization operation, in: Dennis L. Paisley, ALan M. Frank (Eds.), 22nd Int. Congress on High-Speed Photography and Photonics, vol. 2869, 1997, pp. 695–699.
[24] B. Holveck, H. Mathieu, Infrastructure of the Grimage Experimental Platform: the Video Acquisition Part, Tech. Rep. RT-0301, INRIA, Number RT-0301, Nov. 2004.
[25] T. Svoboda, H. Hug, L.V. Gool, Viroom–low cost synchronized multicamera system and its self-calibration, in: Pattern Recognition, 24th DAGM Symposium, Springer-Verlag, London, UK, 2002, pp. 515–522.
[26] G. Litos, X. Zabulis, G. Triantafyllidis, Synchronous image acquisition based on network synchronization, in: IEEE Workshop on Three-Dimensional Cinematography (conj. CVPR), 2006.
[27] T. Kanade, H. Saito, 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.

Yong Liu received his B.S. degree in computer science and engineering from Zhejiang University in 2001, and the Ph.D. degree in computer science from Zhejiang University in 2007. He is currently a professor in the institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. He has published more than 30 research papers in machine learning, computer vision, information fusion, robotics. His latest research interests include machine learning, robotics vision, information processing and granular computing. He is the corresponding author of this paper.

Liang Liu received his B.S. degree in communications engineering from Zhejiang University of Technology in 2015. He is currently a Ph.D. candidate of the institute of Cyber Systems and Control, Department of Control Science and Engineering, Zhejiang University. His latest research interests include machine learning, robotics vision.