# A PSEUDO STEREO VISION METHOD FOR UNSYNCHRONIZED CAMERAS

*Shoichi Shimizu*[†1], *Hironobu Fujiyoshi*[†1], *Yasunori Nagasaka*[†2], *Tomoichi Takahashi*[†3]

[†1] Dept. of Computer Science, Chubu University, Kasugai, Aichi, Japan
[†2] Dept. of Electronic Engineering, Chubu University, Kasugai, Aichi, Japan
[†3] Dept. of Information Science, Meijo University, Nagoya, Aichi, Japan

{shiyou,hf}@vision.cs.chubu.ac.jp, any@nn.solan.chubu.ac.jp, ttaka@ccmfs.meijo-u.ac.jp

**ABSTRACT**

Stereo vision systems have been proposed and used in various fields. They require cameras to synchronize with each other for tracking objects accurately measuring depth. In this paper, we present a vision system which uses two normal, unsynchronized cameras for calculating 3D positions of objects. The unsynchronization in shutter timing of cameras has a merit that it processes faster than a synchronized camera system. 3D positions of an object are measured as a crossing point of lines in 3D space through the detected position on the last frame and estimated 3D position calculated by past frames. This makes it possible for our system to consist of two unsynchronized cameras and to calculate the 3D position of the moving object about twice faster than a synchronized method with similar precisions. This can be useful for controlling a robot as a fast visual feedback system.

## 1. INTRODUCTION

In many practical applications of mobile robots, a visual feedback system is required for planning motions of robots based on environmental information obtained by sensors. In order to control robots precisely and rapidly using a visual sensor, a fast vision system with a high speed cycle of visual feedback is needed. Also, the vision system should detect the precise 3D positions of the robot.

There have been three approaches to implementing high speed vision systems. First, a vision board for detecting colors has been designed [1]. Second, fast vision algorithms using standard frame grabbers [2]. These two approaches work to reduce the processing time for vision, however they are not effective for implementing the high speed cycle of visual feedback. Because it depends on restriction by shutter cycle of the camera (normally 30 fps). Third, the use of commercial vision systems that have the ability to get high resolution with high speed have been used [3, 4]. The system process NTSC camera images at 60 fps rate with double buffering, but these systems are expensive.

Stereo vision is needed for measuring the 3D position of an object such as the robot. Stereo vision systems have been proposed and used in various fields. They require cameras to synchronize with each other for tracking objects accurately measuring depth. As an example of measuring the depth using unsynchronized cameras, human motion tracking system [5] and dynamic depth recovery method from unsynchronized video streams [6] have been reported. In the first paper, the system obtain 2D position of contact point between human and floor, and the cycle of visual feedback is on average 5 fps. In the later paper, the method creates a nonexisting image which is used for triangulation. The nonexisting image is created from estimated time delay between unsynchronized cameras and optical flow fields computed in each view. This method can output a depth map at $t-1$ (1 frame before), not current one.

We propose a method for calculating 3D positions of an object from unsynchronized cameras. A synchronized image is created from the past unsynchronized images. And current 3D position of the object is measured as a crossing point of the line in 3D space through a detected position on the image of a corresponding camera image. When calculating the corresponding point, its accuracy is improved by epipolar constraint. The unsynchronization of cameras also has a merit of processing more images per second than synchronized camera system.

The paper is organized as follows. In Section 2, we describe the method used for measuring 3D position using an unsynchronized camera system. In Section 3, we describe experimental results of recovering motion when an object moved in virtual 3D space, and show the effectiveness of proposed method. Finally, in Section 4, we describe some experimental results of an object moving at high speed and show the operability of proposed method.

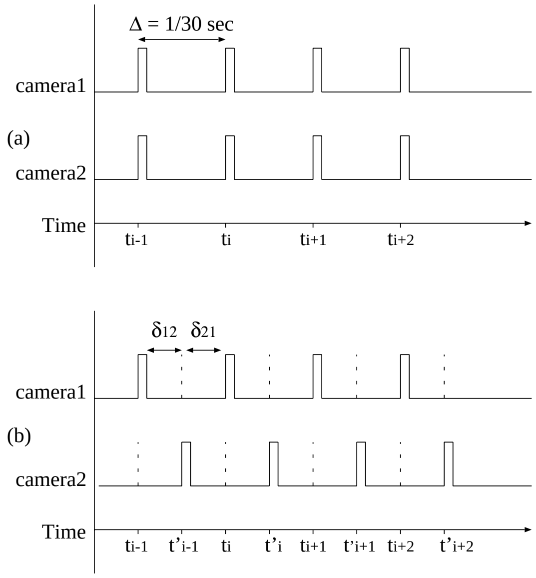



**Fig. 1**. Synchronized and unsynchronized camera timing

## 2. 3D POSITION MEASUREMENT WITH TWO UNSYNCHRONIZED CAMERAS

### 2.1. Model of unsynchronization in two cameras

Stereo vision method which measures 3D positions of the object requires two images captured at the same time to reduce error in the measurement. Using a general stereo vision system, 3D positions can be obtained at 30 fps maximum using normal 30 fps camera with fast vision algorithm described in [2].

Figure 1(a) shows the synchronized shutter timing of both cameras. When the cameras are not synchronized, there is a time lag between them (Figure 1(b)). The time delay $\delta$ depends on the shutter timing of each camera. When the last image is from camera2, stereo vision determines 3D position by $(u_t^1, v_t^1)$ and $(u_{t'}^2, v_{t'}^2)$ where $t' = t + \delta$ as corresponding points. So, the 3D position calculated from these corresponding points with the time delay may contain error as shown in Figure 2.

We propose a method for calculating the 3D position taking into account the time delay $\delta$ which occurs when using unsynchronized cameras. The 3D position in the last frame is estimated by using the time delay $\delta$ and the results from the previous two frames. The procedure of 3D position measurement using two unsynchronized cameras is as follows.

**Step1** Calculation of 3D positions at previous two frames

**Step2** Linear prediction of the 3D position

**Step3** Prediction using constraints from ray information

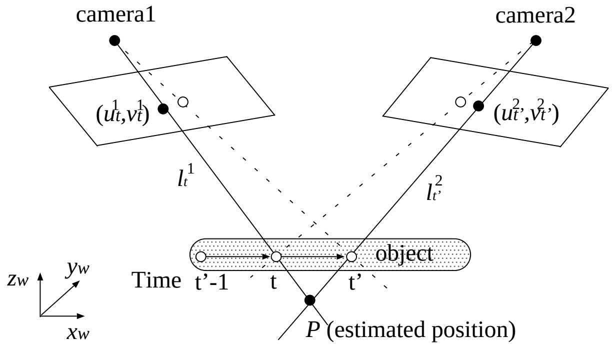



**Fig. 2**. Errors occured in stereo vision with time delay

First, 3D positions of previous two frames, which are used at next step of linear prediction are calculated. A method for calculating the 3D positions of previous two frames is described in Section 2.2. Second, the 3D position of lastest frame is calculated by linear prediction. Finally, the 3D position is calculated once more using constraint by viewing ray information in 3D space obtained from the last image. Since the 3D position is calculated at every shutter timing of each camera ($t_i$ and $t'_i = t_{i+\delta}$), it is possible that our vision system outputs the 3D positions twice faster than the synchronized one.

### 2.2. Calculation of 3D positions at previous two frames

In order to obtain accurate 3D position from the current frame by linear prediction, it is necessary to calculate the previous 3D positions $\boldsymbol{P}_{t-1}$, $\boldsymbol{P}_{t'-1}$ accurately. Since the shutter timings of two cameras are not synchronized, there is no corresponding point for depth when motion is introduced. The position of an object using image coordinates from each camera is obtained sequentially.

However, the position of an object using image coordinates from camera1 at frame $t$ $(u_t^1, v_t^1)$ does not have a corresponding point from camera2 at the same time, which is needed to calculate the 3D position using stereo vision. The predicted point of camera2 $(\hat{u_{t-1}^2}, \hat{v_{t-1}^2})$ corresponding to the observed point $(u_{t-1}^1, v_{t-1}^1)$ is generated by a basic interpolation as shown in Figure 3. The 3D position can be measured by the stereo vision using observed point $(u_{t-1}^1, v_{t-1}^1)$ and pseudo corresponding point $(\hat{u_{t-1}^2}, \hat{v_{t-1}^2})$. The algorithm of the 3D position calculation at $t-1$ is described as follows.

**StepI** Using two observed points in the frame $t'-1$ and $t'-2$, a pseudo-corresponding point from camera2 on frame $t-1$ $(\hat{u_{t-1}^2}, \hat{v_{t-1}^2})$ is interpolated by the following equation.

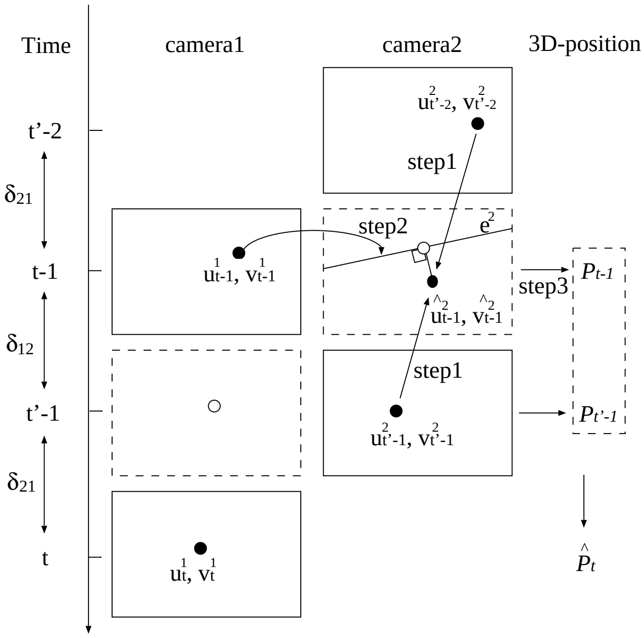



**Fig. 3**. Proposed calculation method of corresponding position

$$\hat{u^2_{t-1}} = \frac{\delta_{12} u^2_{t'-2} + \delta_{21} u^2_{t'-1}}{\delta_{21} + \delta_{12}} \tag{1}$$

$$\hat{v^2_{t-1}} = \frac{\delta_{12} v^2_{t'-2} + \delta_{21} v^2_{t'-1}}{\delta_{21} + \delta_{12}} \tag{2}$$

Note that the interpolated point is middle when the time interval $\delta_{12}$ is equal to $\delta_{21}$.

**StepII** Calculate the epipolar line $e^2$ on the image from camera2 using the corresponding point $(u^1_{t-1}, v^1_{t-1})$ from the image from camera1. Then, the nearest point $(u'^2_{t-1},\ v'^2_{t-1})$ from the interpolated point, calculated by stepI, can be set as corresponding point for $(u^1_{t-1}$, $v^1_{t-1})$.

**StepIII** A viewing ray of camera1 $\boldsymbol{l}^1_{t-1}$ passing through the point $(u^1_{t-1}, v^1_{t-1})$ and a viewing ray of camera2 $\boldsymbol{l}^2_{t-1}$ passing through the point $(u^2_{t-1}, v^2_{t-1})$ are obtained, so we can measure the 3D position $\boldsymbol{P}_{t-1}$ using triangulation as a crossing point of the two lines in 3D space.

Step I, II, III are repeated at the frame $t'-1$. The two previous 3D positions at $t'-1, t-1$ are calculated from previous and following frames and the epipolar constraint. These two previous points $\boldsymbol{P}_{t-1}$ and $\boldsymbol{P}_{t'-1}$ will be used and make lesser prediction error at the next step of linear prediction.

### 2.3. Linear prediction of 3D position

As shown in Figure 4, the predicted position $\hat{\boldsymbol{P}}_t = [x_w, y_w, z_w]^T$ of the last frame t is calculated by the following equation using already measured positions $\boldsymbol{P}_{t-1}$ and $\boldsymbol{P}_{t'-1}$ in the previous two frames. Let $\delta_{12}$ be the time interval of shutter timings from camera1 to camera2, $\delta_{21}$ be the time interval of shutter timings from camera2 to camera1.

$$\begin{aligned} \hat{\boldsymbol{P}}_t &= \boldsymbol{P}_{t'-1} + \delta_{21}\boldsymbol{v}_{t-1}, \\ \boldsymbol{v}_{t-1} &= (\boldsymbol{P}_{t'-1} - \boldsymbol{P}_{t-1})/\delta_{12} \end{aligned} \tag{3}$$

Note that Equation (3) is based on the analyzed image of the last frame from camera1. When an image from camera camera2 is being used, the suffixes of $\delta$ are cycled. For the position estimation, the Kalman filter [7, 8] and spline curve fitting have been proposed. In this paper, we just use the linear estimation because the time interval $\delta_{12}, \delta_{21}$ is less than $\frac{1}{30}$ second.

### 2.4. Prediction using constraints from ray information

In order to decrease the prediction error, the 3D position is calculated once more using a constraint of a viewing ray in 3D space obtained from the current image. Let $\boldsymbol{T}^1 = [T_x, T_y, T_z]$ be a translation matrix from the origin of the world coordinate to the foucus point of camera1, and $\boldsymbol{r}^1_t = [x_w, y_w, z_w]^T$ be a vector which denotes the direction of the viewing ray, $\boldsymbol{l}^1_t$, passing through the position on the image coordinate $(u^1_t, v^1_t)$ and the foucus point of the camera. The viewing ray shown in Figure 4 can be expressed by

$$\boldsymbol{l}^1_t = k\boldsymbol{r}^1_t + \boldsymbol{T}^1 \tag{4}$$

where $k$ is a real number. Although Equation (3) gives a good 3D position prediction , the position may not exist on the viewing ray $\boldsymbol{l}^1_t$ as shown in Figure 4 because of its prediction error. In order to solve this problem, the 3D position $\boldsymbol{P'}_t$ is calculated by the following equation as the nearest point on the ray $\boldsymbol{l}^1_t$.

$$\boldsymbol{P}'_t = \frac{(\hat{\boldsymbol{P}}_t - \boldsymbol{T}^1)\cdot \boldsymbol{r}^1_t}{|\boldsymbol{r}^1_t|^2}\boldsymbol{r}^1_t + \boldsymbol{T}^1 \tag{5}$$

Finally, $\boldsymbol{P}'_t$ will be the 3D position of the object at the last frame. In case camera2 is the current one, the image of the 3D position is calculated in the same way as mentioned above.

## 3. SIMULATION EXPERIMENTS

### 3.1. Recovery object motions

We evaluate the proposed method by simulation of recovering the object's motion with uniform and non-uniform mo-

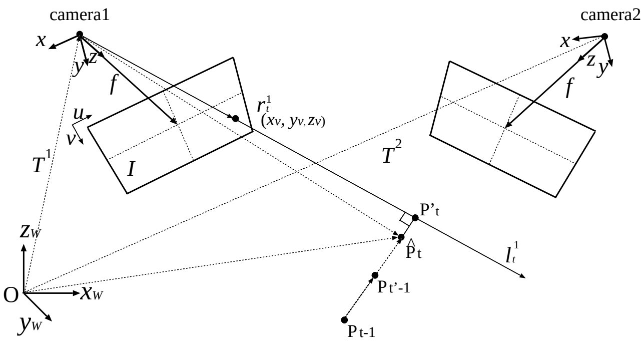



**Fig. 4**. Constraints in estimating 3D position

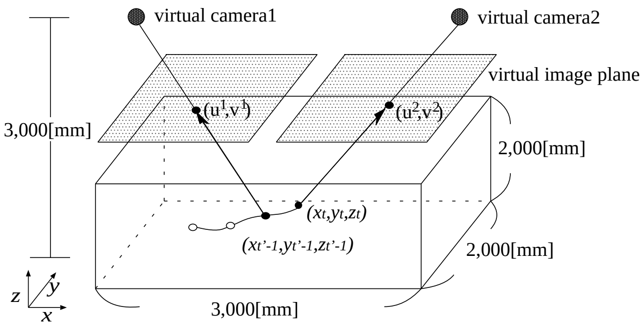



**Fig. 5**. Virtual camera placement in simulation experiment

tion in 3D space (3,000 $\times$ 2,000 $\times$ 2,000[mm]). In the simulation , we assume that two cameras mounted on the height of 3,000[mm] and virtual image planes of each camera as shown in Figure 5. The time delay of shutter timing by each camera was set as 1/60[sec] ($\delta_{12} = \delta_{21}$). The proposed method is evaluated by following three motion.

- uniform motion(straight): An object moves to $(x, y, z)$ = (3,000, 1,200, 0) from $(x, y, z)$ = (0, 1,200, 2,000) at velocity of 3,000[mm/sec]
- uniform motion(spiral) An object moves on spiral by radius of 620[mm] at velocity of 3,000[mm/sec] at center $(x, y)$ = (1,000, 1,000)
- non-uniform motion: An object falls from the height of 2,000[mm], then an object describe a parabola (gravitational acceleration:g=9.8[m/s$^2$])

The trajectory of the object is projected to the virtual image planes of each camera. A 3D position is estimated by the proposed method described in section 2 using the projected point on the virtual image plane $(u, v)$ of each camera.

### 3.2. Simulation results

Figure 6(a),(b) shows simulation results of recovering uniform motion of straight and spiral by the proposed method.

**Table 1**. Average of absolute errors in 3D positions [mm]

| method | fps | uniform | | non-uniform |
|---|---|---|---|---|
| | | straight | spiral | |
| unsynchronized | 30 | 23.2 | 21.4 | 16.2 |
| proposed | 60 | 1.1 | 2.2 | 1.8 |
| proposed* | 60 | 1.2 | 3.8 | 1.9 |
| synchronized | 30 | 0.06 | 0.19 | 0.08 |
| | 60 | 0.03 | 0.14 | 0.12 |

*$\delta_{12}$ = 1/90, $\delta_{21}$ = 2/90

Table 1 shows averages of estimation error with simulation experiments. The unsynchronized method in Table 1 means result of stereo vision by corresponding points in time-delay as shown in Figure 1 and the synchronized method means result of general stereo vision with no time-delay. It is clear that the proposed method has better result than the unsynchronized method and its accuracy is close to the synchronized method. The error of the proposed method is 3[mm] or less to all motions, which is very accurate considering the speed of the object. Figure 6(c) shows simulation result of recovering non-uniform motion by ball falling. At the moment of when an object bounds on the floor, the estimation error becomes large. This is caused by our simple linear motion predictor.

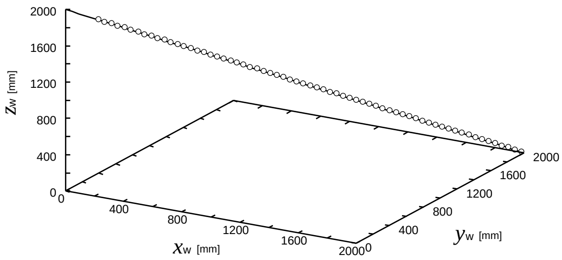
## 4. EXPERIMENTS USING REAL CAMERAS

We evaluated our method using real data in the same way as the simulation experiments.

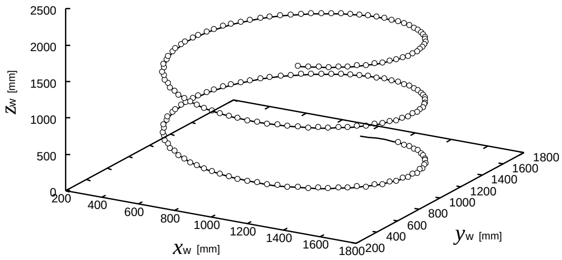
### 4.1. Configuration of vision system

Figure 7 shows the camera placement of our vision system that uses two cameras, camera1 and camera2 The cameras are mounted on either side at a height of 2,800[mm], and each camera has a view of an area of 2,000 $\times$ 3,000[mm]. Each camera is calibrated using corresponding points of world coordinates $(x_w, y_w, z_w)$ and image coordinate $(u, v)$ [9]. The frame grabbers for the two cameras are installed on a PC.

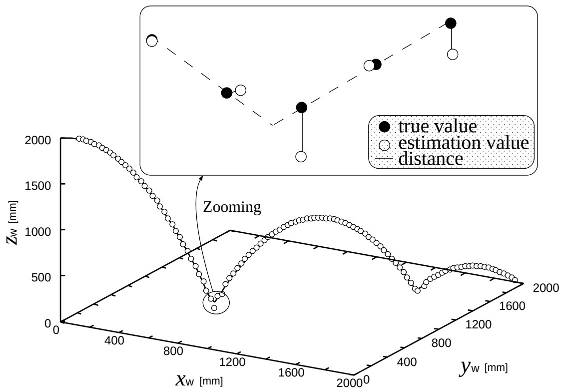
Process-1 and process-2 analyze images from camera camera1 and camera2 at every 1/30 second respectively. The analyzed results such as $(u^i, v^i)$ and the time instant at which the analyzed image was captured are sent via UDP interface to process-3 that calculates the 3D positions of the object. There is negliglible delay due to communications among processes because these process work on a same PC. Time delay of shutter timings $\delta_{12}$ and $\delta_{21}$ is calculated from the time instants sent by each processes.




(a) uniform straight motion (3,000[mm/s])




(b) uniform spiral motion (3,000[mm/s])




(c) non-uniform motion (max speed 5,800 [mm/s])

**Fig. 6**. Calculated object motions

### 4.2. Experiments

As an experiment of measuring 3D positions, we used a turntable and a ball as shown in Figure 8. A ball attached on the edge of ruler(1,000[mm] length) makes a uniform circular motion with a radius of 500[mm]. The turntable is placed on a box at the height of 500[mm], and the ball's height from the floor will be 650[mm]. The turntable rotates at a speed of 45[rpm], its rotation speed per second is $\frac{45}{60}2\pi$ = 0.478 radian.

### 4.3. Experimental results

Figure 9 shows the experimental results for the 3D position of a ball making a uniform circular motion. The radius of the circular motion is about 500[mm] at the height of 650[mm] from the floor. Table 2 shows the averages and variances of the radius measured by the projection into $x_w$-

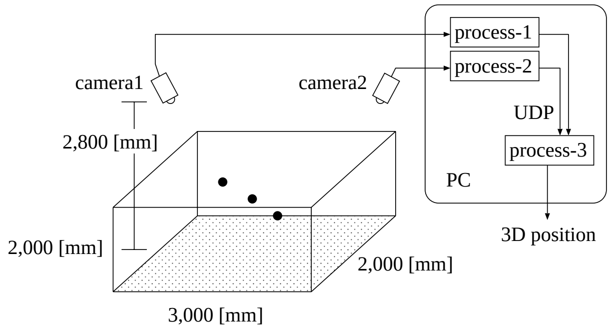
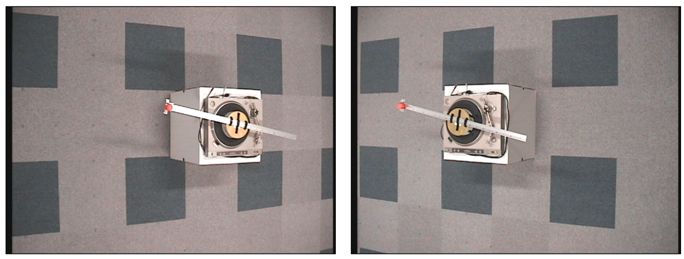



**Fig. 7**. Overview of our vision system



(a) image from camera1 (b) image from camera2

**Fig. 8**. Captured images of turntable

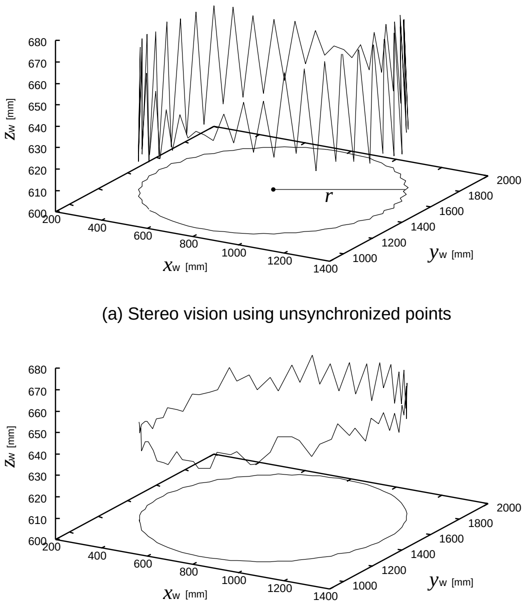
$y_w$ plane of the 3D circular motion We can see from Table 2 that there is no significant difference among the unsynchronized and proposed method. Table 3 shows the average and variance of the 3D position on the $z_w$ axis by the two methods. The average of the positions from the two methods was measured within 4[mm] from the actual height of 650[mm]. The variances are 482.3 with unsynchronized method and 57.7 with the proposed method. It is clear that the accuracy of unsynchronized method is not good because of the corresponding points for the stereo vision contain a time-delay of 16[ms]. We can see from Figure 9 that the result of proposed method is better. It shows a 30[mm] accuracy in position, because the 3D positions are calculated using constraints from ray information.

### 4.4. Processing time

This vision system is implemented on a PC with dual Xeon processors. In our implementation, it takes 0.0026[sec] for the vision process to calculate the ball position from an image, and 0.00018[sec] for the transmission process via UDP within one PC. Therefore, this system can determine the 3D positions of the object in 0.00413[sec] from the time the analyzed image was captured.

Our system does not have any synchronization mechanism to ensure that time interval $\delta_{12}$ and $\delta_{21}$ are equally separated. The timing depends on the timing of each camera shutter.

Figure 10 shows timing charts of the processes in the




(a) Stereo vision using unsynchronized points

(b) Proposed method

**Fig. 9**. Results of 3D position measurement

**Table 2**. Average and variance of z values of 3D positions

| | average[mm] | variance |
|---|---|---|
| unsynchronized method | 492.9 | 61.9 |
| proposed method | 494.3 | 54.9 |

case of ($\delta_{12} \simeq \delta_{21} \simeq \frac{1}{60}$). The timing for capturing and analyzing images, $\delta_v$, is required to be less than $\delta_{12}$ and $\delta_{21}$.

When this requirement is satisfied, this system can determine the 3D positions of the object at 60 times per second. When this requirement is not satisfied, the two camera images are regarded as having been captured simultaneously, and are used to the normal stereo vision at 30 fps.

## 5. DISCUSSION AND CONCLUSION

In this paper, we proposed a method for calculating the 3D position of an object with two normal unsynchronized cameras. The 3D position is measured as a crossing point of the line in 3D through the detected position from the last image frame as well as from past frames.

We showed that the calculation of the 3D position from the previous two frames for linear prediction more accurate. Our system, which consists of two unsynchronized cameras and a PC, determines the 3D position of a moving object at 60 fps making use of the time lags of the shutter timing between the two cameras.

**Table 3**. Average and variance of z axis

| | average[mm] | variance |
|---|---|---|
| unsynchronized method | 647.3 | 482.3 |
| proposed method | 647.5 | 57.7 |

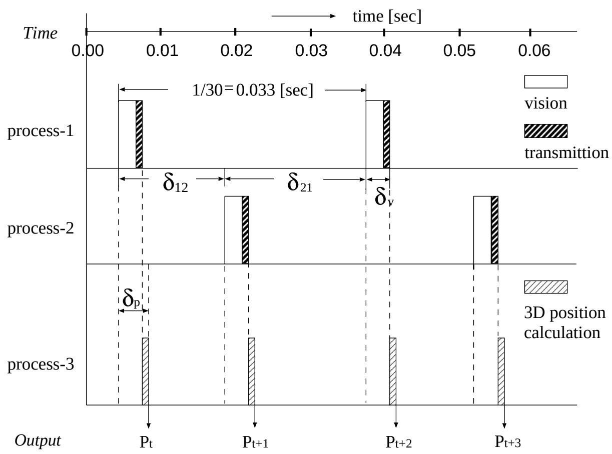



**Fig. 10**. time chart of processes

## 6. REFERENCES


[1] J. Akita, "Real-time color detection system using custom lsi for high-speed machine vision," *RoboCup-99: Robot Soccer World Cup III*, pp. 1–4, 1999.

[2] M. Veloso J. Bruce, T. Balch, "Fast and inexpensive color image segmentation for interactive robots," *IROS-2000*, vol. 3, pp. 2061–2066, 2000.

[3] R. D'Andrea, "http://robocup.mae.cornel.edu/," .

[4] Y. Kodama K. Murakami K. Katoh, S. Hibino and T. Naruse, "High-speed image processing method to extract small size robot for the robocup," *CVIM*, vol. CVIM-136, no. 16, pp. 115–122, 2003.

[5] J. Ohya H. Mori, A. Utsumi and M. Yachida, "Human motion tracking using non-synchronous multiple observations," *IEICE*, vol. J84-D-II, no. 1, pp. 102–110, 2001.

[6] C. Zhou and H. Tao, "Dynamic depth recovery from unsynchronized video streams," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 351–358, 2003.

[7] M. Bowling B. Browning and M. Veloso, "Improbability filtering for rejecting false positives," *IEEE International Conference on Robotics and Automation*, pp. 120–200, 2002.

[8] S. Kaneko K. Horiuchhi and T. Honda, "Object motion estimation based on multiple distributed kalman filters," *The IEICE Trans*, vol. J79-D-II, no. 5, pp. 840–850, 1996.

[9] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp. 323–344, 1987.