Fast 3D Position Measurement with Two Unsynchronized Cameras

Hironobu Fujiyoshi
Dept. of Computer Science
Chubu University
www.vision.cs.chubu.ac.jp

Shouichi Shimizu
Dept. of Computer Science
Chubu University

Takayuki Nishi
Dept. of Computer Science
Chubu University

Yasunori Nagasaka
Dept. of Electronic Engineering
Chubu University

Tomoichi Takahashi
Dept. of Information Science
Meijo University

Abstract

Stereo vision systems have been proposed and used in various fields. They require cameras to synchronize with each other for tracking objects. In this paper, we present a vision system which uses two normal, unsynchronized cameras for calculating the 3D position of objects. The unsynchronization in shutter timing of cameras is used to process more images per second than a synchronized camera system. 3D positions of an object are measured as a crossing point of lines in 3D through the detected position on the latest frame and estimated 3D position calculated by past frames. This makes it possible for our system to consist of two unsynchronized cameras and allows for a PC to calculate the 3D position of the moving object at 60 fps. 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 the motion of the robot 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 [3]. 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. Third, the use of commercial vision systems that have the ability to get high resolution with high speed have been uses [5]. The system process NTSC allows camera images at 60 fps rate with double buffering, but these systems are expensive.

Furthermore, stereo vision is needed for measuring the 3D position of an object such as a robot. Stereo vision systems have been proposed and used in various fields. They require cameras to synchronize with each other for tracking objects.

This paper describes a vision system with two normal unsynchronized cameras for calculating the 3D position of objects at the rate of 60 fps. The unsynchronization in shutter timing of cameras is used to process more images per second than a synchronized camera system. The 3D position is measured as a crossing point of the line in 3D through the detected position on the image at the latest frame and a 3D position is estimated by calculating past frames.

The paper is organized as follows. In section 2, we present the vision system using two unsynchronized cameras. In section 3, we describe the method used for measuring 3D position using an unsynchronized camera system. Finally, in section 4, we describe some experimental results of an object moving in at high speed and show the effectiveness of our proposed method.

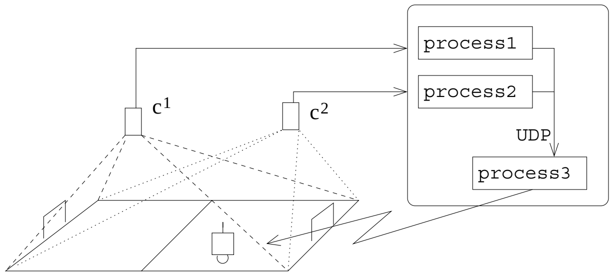


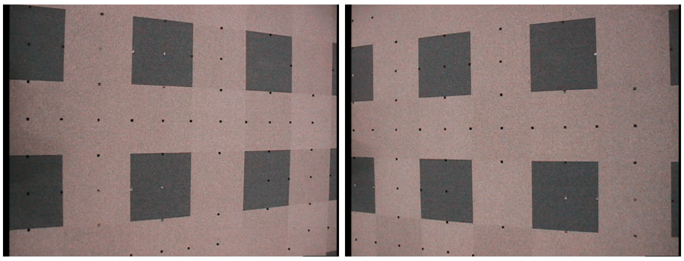
Figure 1: Overview of our vision system

2 Vision system with two unsynchronized cameras

2.1 Configuration of vision system

Figure 1 shows the camera placement of our vision system that uses two cameras, c^1 and c^2 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 × 3,000[mm]. The frame grabbers for the two cameras are installed on a PC. Figure 2 shows the images captured by each camera.

Process-1 and process-2 analyze images from camera c^1 and c^2 at every 1/30 second respectively (Figure 1). The analyzed results are positions (u^i, v^i) of objects in the images at times when it was captured. The positions are sent via UDP interface to process-3 which calculates the 3D position of objects. These three processes work on the same PC, so there is no delay during UDP transport.



(a) image from c^1 (b) image from c^2

Figure 2: Images from each camera

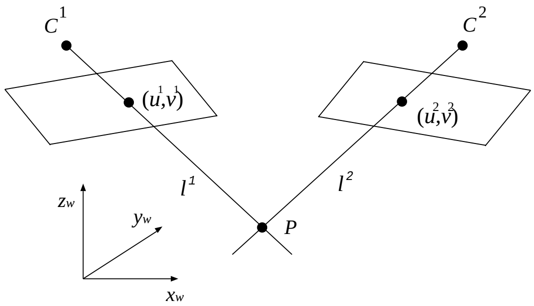


Figure 3: stereo

2.2 Shutter timing of each camera

Each camera is calibrated using corresponding points of world coordinates (x_w, y_w, z_w) and the image coordinate (u, v) [6]. Then, the rays l^1 and l^2, shown in Figure 3, which pass through points (u^1, v^1) and (u^2, v^2) on each image plane and are then calculated using the intrinsic parameters and extrinsic parameters of each camera. The intersecting point $\boldsymbol{P}$ of two lines l^1 and l^2 is able to be used to calculate the 3D position of the object in world coordinates.

In the case of stereo vision, each camera's images have to be taken at the same time. Figure 4(a) shows the synchronized shutter timing of both cameras. When the cameras are not syndhronized, there is a time lag between them (Figure 4(b)). The time delay δ depends on the shutter timing of each camera.

3 3D position measurement with two un-synchronized cameras

We propose a method for calculating the 3D position taking into account the time delay δ which occurs when using un-synchronized cameras. The 3D position in the latest frame is estimated bu using the time delay δ and the results from in the previous two frames. Moreover, it is possible that our vision system processes 3D positions as fast as 60 fps camera as a result of each cameras every shutter timing (t_i and $t'_i = t_{i+\delta}$).

3.1 3D position estimation

Since the shutter timings of two cameras are not synchronized, a general stereo vision will not work. So, the 3D position of an object is estimated as a crossing point of the ray in 3D space from the latest image and a predicted 3D position is calculated from the previous two frames.

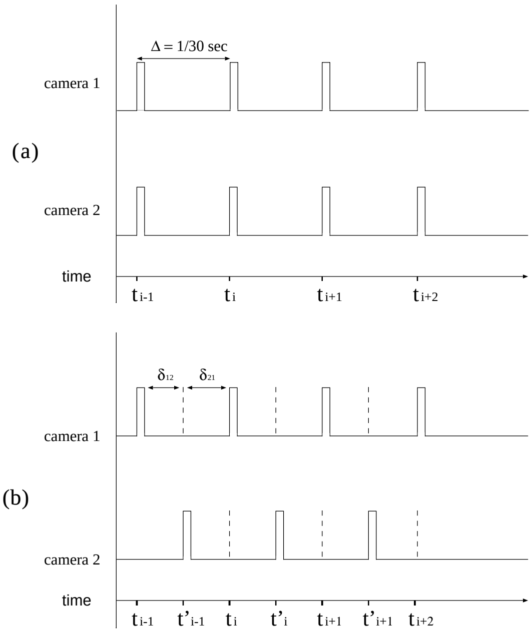


Figure 4: Shutter timing of each camera

Linear prediction of 3D position

As shown in Figure 5, the predicted position $\hat{\boldsymbol{P}}_t = [x_w, y_w, z_w]^T$ of the latest frame 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 δ_{12} be the time interval between the start of process-1 and the start of process-2 and δ_{21} be the time interval between the start of process-2 and the start of the next iteration of process-1.

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

Note that Equation (1) is based on the analyzed image of the latest frame from camera1. When an image from camera c^2 is being used, the suffixes of δ are cycled. For the position estimation the Kalman filter [2, 4] and spline curve fitting have been proposed. In this paper, we just use the linear estimation because the time interval δ_{12}, δ_{21} is less than $\frac{1}{30}$ second.

Prediction using constraints from ray information

In order to decrease the prediction error, the 3D position is calculated once more using a restraint of a ray in 3D space obtained from the latest image. Let $\boldsymbol{T}^1 = [T_x, T_y, T_z]$ be a translation matrix from the origin of the world coordinate to the optical center of camera1, and $\boldsymbol{r}_t^1 = [x_w, y_w, z_w]^T$ be a vector which denotes the direction of the ray, $\boldsymbol{l}_t^1$, passing through the position on the image coordinate (u_t^1, v_t^1) and the optical center of the camera. The ray shown in Figure 5 can be expressed by

$$\boldsymbol{l}_t^1 = k\boldsymbol{r}_t^1 + \boldsymbol{T}^1, \tag{2}$$

where k is a real number. Although Equation (1) predicts the 3D position accurately, the position may not exist on the ray $\boldsymbol{l}_t^1$ as shown in Figure 5 because of its

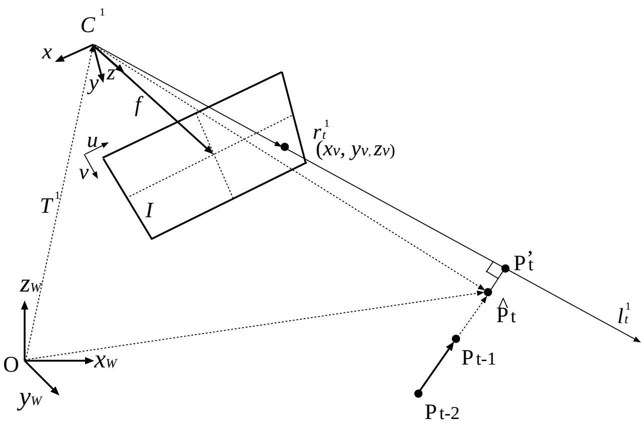


Figure 5: 3D position estimation

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 $\boldsymbol{l}_t^1$.

$$\boldsymbol{P}'_t = \frac{(\hat{\boldsymbol{P}}_t - \boldsymbol{T}^1) \cdot \boldsymbol{r}_t^1}{|\boldsymbol{r}_t^1|^2} \boldsymbol{r}_t^1 + \boldsymbol{T}^1 \tag{3}$$

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

3.2 Re-calculation of 3D position at previous two frames

In order to obtain a more accurate 3D position $\hat{\boldsymbol{P}}_t$ from the latest frame, it is necessary to calculate the previous 3D positions $\boldsymbol{P}_{t-1}$, $\boldsymbol{P}_{t'-1}$ accurately using the calculation of the velocity $\boldsymbol{v}_{t-1}$ in Equation (1). There have been two way of calculating $\boldsymbol{P}_{t-1}$ and $\boldsymbol{P}_{t'-1}$ as follows.

i. A method using 3D position $\boldsymbol{P}'_{t-1}$ and $\boldsymbol{P}'_{t'-1}$ calculated by Equation (3) at every frame.

ii. A method be used on the re-calculation of the previous 3D position $\boldsymbol{P}_{t-1}$, $\boldsymbol{P}_{t'-1}$ at the latest frame t.

In the case of the prior method (i), it is necessary to set a initial value for the 3D position. Also, the prediction error of $\boldsymbol{P}'_{t-1}$ and $\boldsymbol{P}'_{t'-1}$ may be propagated into the latest position $\hat{\boldsymbol{P}}_t$ in Equation (1).

Whereas, the latter method (ii) re-calcurating the previous 3D position $\boldsymbol{P}_{t-1}$ and $\boldsymbol{P}_{t'-1}$ is more accurate in the latest frame for decreasing the propagatable prediction error. The algorithm of this re-calculation is described as follows.

The predicted point of camera2 $(u^{\hat{2}}_{t-1}, u^{\hat{2}}_{t-2})$ corresponding to the observed point (u^1_{t-1}, v^1_{t-2}) is generated by a basic interpolation as described in Figure 6.The position of an object using image coordinates from each camera is obtained sequentialy. However, the position of an object using image coordinates from camera1 at frame t (u^1_t, v^1_t) does not have a corresponding point from camera2 at the same time needed to calculate the 3D position using stereo vision because of the unsynchronization of the two cameras. Further, the 3D position can be measured by the stereo vision using corresponding points of $(u^{\hat{2}}_{t-1}, v^{\hat{1}}_{t-1})$ and (u^2_{t-1}, v^2_{t-1}).

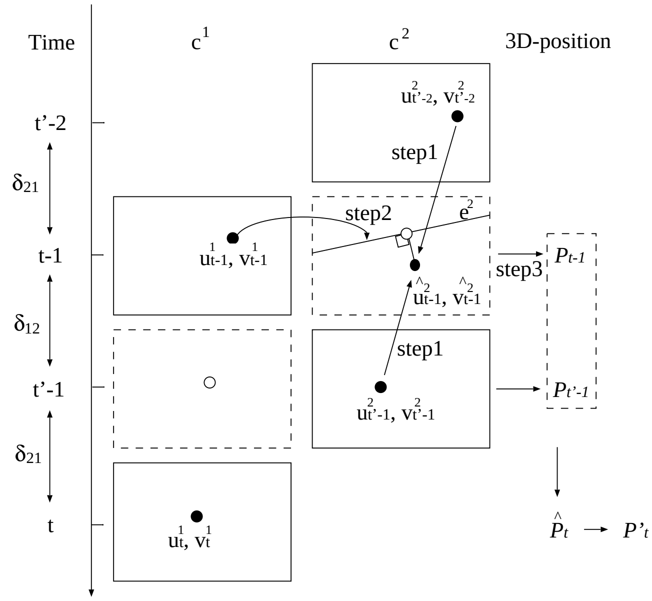


Figure 6: re-calculation of position

Step1 Using two observed points in previous and the following frame from $t-1$, a pseudo-corresponding point from camera2 on frame $t-1$ ($u^{\hat{2}}_{t-1}$, $v^{\hat{2}}_{t-1}$) is interpolated by the following equation.

$$u^{\hat{2}}_{t-1} = \frac{\delta_{21}}{\delta_{21} + \delta_{12}}(u^2_{t'-2} + u^2_{t'-1}) \tag{4}$$

$$v^{\hat{2}}_{t-1} = \frac{\delta_{21}}{\delta_{21} + \delta_{12}}(v^2_{t'-2} + v^2_{t'-1}) \tag{5}$$

Note that the intepolated point is middle when the time interval δ_{12} is equal to δ_{21}.

Step2 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 step1, can be set as a new corresponding point for the image of camera2 at the frame $t-1$.

Step3 A line in 3D $\boldsymbol{l}^2_{t-1}$ and $\boldsymbol{l}^2_{t-2}$ is obtained, so we can measure the 3D position $\boldsymbol{P}_t$ using stereo vision as a crossing point of the two lines in 3D space.

These steps are repeated at the frame $t'-1$. Using a re-calculated 3D position $\boldsymbol{P}_{t-1}, \boldsymbol{P}_{t'-1}$, the predicted 3D position $\hat{\boldsymbol{P}}_t$ from Equation (1) is calculated. The accuracy of the predicted 3D position $\hat{\boldsymbol{P}}_t$ may have a lesser prediction error because the previous two positions used to predict are re-calculated from the positions of the previous and following frames and the epipolar restraint.

4 Experiments

4.1 Measuring positions circular motion

As an experiment of measuring 3D positions, we used a turntable and a ball as shown in Figure 7. A ball attached on the edge of ruler(100[cm] length) makes a uniform circular motion with a radius of 50cm. The turntable is placed on a box at the height of 500[mm],

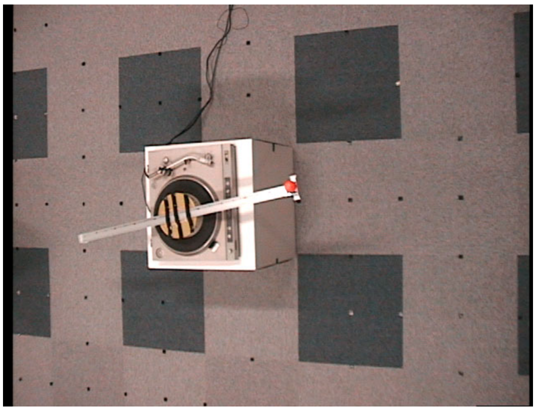


Figure 7: A scene of the experiment with a turntable

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.

The three methods of 3D position calculation are evaluated how precisely the 3D motion of the ball is measured.

(a)Stereo vision by corresponding points in time-delay

When the latest image is from camera1, method A determines stereo vision by (u_t^1, v_t^1) and $(u_{t'-1}^2, v_{t'-1}^2)$ as corresponding points. $(u_{t'-1}^2, v_{t'-1}^2)$ is the point on the image coordinated from camera2 at the frame previous to the latest one. Therefore, the corresponding points may contain the time-delay when the object moves fast.

(b)Proposed method 1

This method is described in section 3.2 (i). Method 2 uses the 3D points $\boldsymbol{P'}_{t-1}$ and $\boldsymbol{P'}_{t'-1}$ for linear prediction of the 3D position using Equation (1).

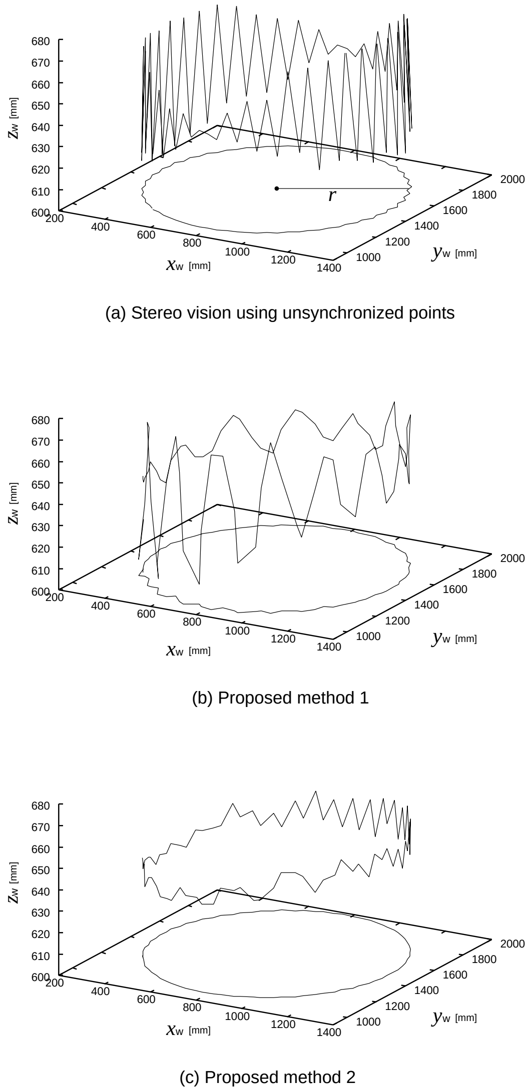
(c)Proposed method 2

Method c is that recalculation of the 3D positions of the previous two frames in order to reduce a prediction error as described in section 3.2 (ii).

4.2 Experimental results

Figure 8 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.1 shows the averages and variances of the radius measured by the projection into x_w-y_w plane of the 3D circular motion. We see from Table 1 that there is no significant difference among the three method because the radius of the circular motion is 500[mm]. Table 2 shows the average and variance of the 3D position on the z_w axis as measured by the three methods. The average of the positions from the three methods was measured within 4[mm] from the actual height of 650[mm]. The variances are 482.3 with method (a), 234.0 with method (b) and 57.7 with method (c).

It is clear that the accuracy of method (a) is not good because of the corresponding points for the stereo vision contain a time-delay of 16[ms] using the unsynchronization of the two cameras. From methods (b) and (c), the pseudo corresponding point is generated from the prediction in order to synchronize the two cameras, the variances of methods (b) and (c) were less than the one of method (a). We see from Figure 8 that method (c) is the best. It shows a 30[mm] accuracy in position, because the positions for linear prediction are re-calculated in order to make reduce prediction errors.



(a) Stereo vision using unsynchronized points

(b) Proposed method 1

(c) Proposed method 2

Figure 8: Results of 3D position measument

4.3 Time interval of cameras

Figure 9 shows the projected points into x_w-y_w plane of the 3D circular trajectories of the ball's motion measured by method (c). Points in each circle are labeled from 0, 1, ... in time sequence. Theoretically, it takes 1.33 sec to rotate one turn and therefore $1.33 \times 60 \approx 80$ points are plotted in the turn for a 60 fps vision system. We see from Figure 9 that the plotted points from 1.33 sec is 80. This indicates the speed is the same as the fast (60 fps) camera.

The rotation angle between two consecutive points, $\boldsymbol{P}_i$ and $\angle \boldsymbol{P}_i\boldsymbol{O}\boldsymbol{P}_{i+1}$ is 0.0785 [rad] when they are captured at 60 fps. Figure 10 shows the distributions of $\angle \boldsymbol{P}_i\boldsymbol{O}\boldsymbol{P}_{i+1}$. White triangles indicate a result of the 3D position measurement when both camera's shutter tim-

Table 1: Average and variance of radius

	average[mm]	variance
(a)	492.9	61.9
(b)	493.5	101.2
(c)	494.3	54.9

Table 2: Average and variance of z axis

	average[mm]	variance
(a)	647.3	482.3
(b)	646.7	234.0
(c)	647.5	57.7

ings are equally separated ($\delta_{12} \simeq \delta_{21} \simeq \frac{1}{60}$). Black triangle indicates a result when two cameras shutter timing are almost the same ($\mid \delta_{12} - \delta_{21} \mid = \frac{1}{30}$). In the case of the time intervals of the shutter timings of two cameras δ_{12} and δ_{21} are the same ($\delta_{12} \simeq \delta_{21} \simeq \frac{1}{60}$), and the average of the rotation angle is nearly equal to theoretical value (0.0785[rad]).

From this, our proposed two camera vision system has almost the same speed as 60 fps.

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 δ_{12} and δ_{21} are equally separated. The timing depends on the timing of each camera shutter.

Figure 11 shows timing charts of the processes in the case of ($\delta_{12} \simeq \delta_{21} \simeq \frac{1}{60}$). The timing for capturing and analyzing images, δ_v, is required to be less than δ_{12} and δ_{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 normal the stereo vision at 30 fps.

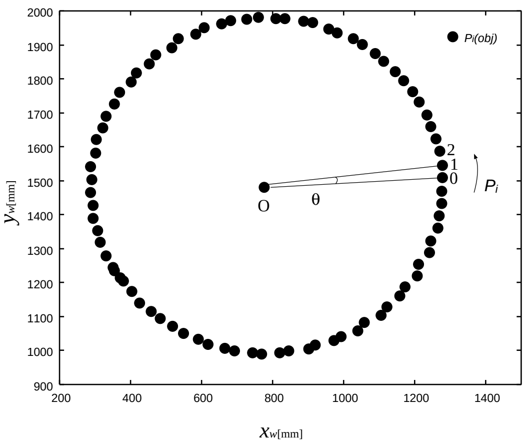


Figure 9: Projected points on x_w-y_w plane

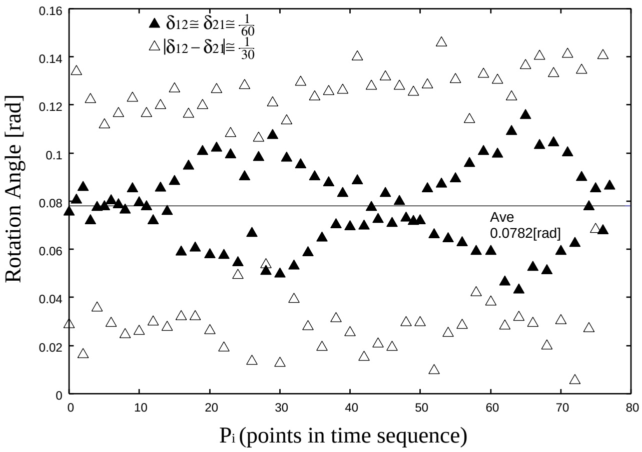


Figure 10: Distribution of rotation angle

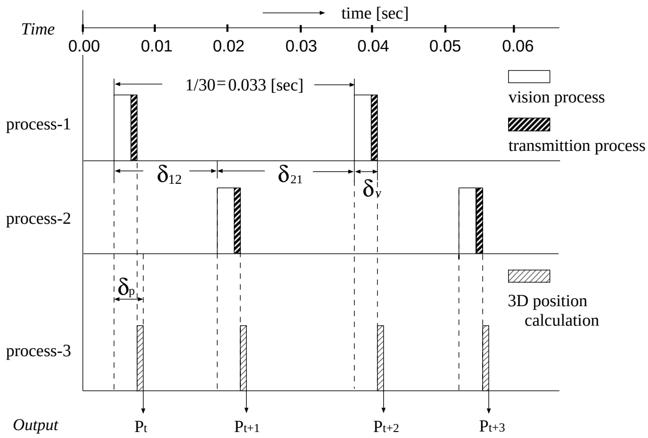


Figure 11: timing chart

5 Conclusion

In this paper, we propose 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 latest image frame as well as from past frames.

We show that the re-calculation of the 3D position from the previous two frames for linear prediction more accurate. Our system, consisting 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. This system can be useful for controlling a robot using a fast visual feedback system.

References

[1] J. Akita. Real-time color detection system using custom lsi for high-speed machine vision. In *RoboCup-99: Robot Soccer World Cup III*, pages 129–135, 2000.

[2] M. Bowling B. Browning and M. Veloso. Improbability filtering for rejecting false positives. In *In Proceedings of 2002 IEEE International Conference on Robotics and Automation*, Washington, DC, May 2002.

[3] T. Balch J. Bruce and M. Veloso. Fast and inexpensive color image segmentation for interactive robots. In *IROS-2000*, pages 2061–2066, Japan, October 2000.

[4] S. Kaneko K. Horiuchhi and T. Honda. Object motion estimation based on multiple distributed kalman filters. volume J79-D-II, pages 840–850, 1996.

[5] M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Trans. PAMI*, 15(4):353–363, 1993.

[6] 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*, RA-3(4):323–344, August 1987.