sensors

MDPI

Article

Development of an Active High-Speed 3-D Vision System [†]

Akio Namiki [1,*], Keitaro Shimada [1], Yusuke Kin [1] and Idaku Ishii [2]

[1] Department of Mechanical Engineering, Chiba University, 1-33 Yayoi-cho, Inage-ku, Chiba 263-8522, Japan; smd6ktr8@gmail.com (K.S.); yusukekin70278965@gmail.com (Y.K.)

[2] Department of System Cybernetics, Hiroshima University, 1-4-1 Kagamiyama, Higashi-Hiroshima, Hiroshima 739-8527, Japan; iishii@robotics.hiroshima-u.ac.jp

* Correspondence: namiki@faculty.chiba-u.jp; Tel.: +81-43-290-3194

[†] This paper is an extended version of the conference paper: High-speed 3-D measurement of a moving object with visual servo. In Proceedings of 2016 IEEE/SICE International Symposium on System Integration (SII), Sapporo, Japan, 13–15 December 2016 and High-speed orientation measurement of a moving object with patterned light projection and model matching. In Proceedings of 2018 IEEE Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018.

Received: 1 February 2019; Accepted: 22 March 2019; Published: 1 April 2019

Abstract: High-speed recognition of the shape of a target object is indispensable for robots to perform various kinds of dexterous tasks in real time. In this paper, we propose a high-speed 3-D sensing system with active target-tracking. The system consists of a high-speed camera head and a high-speed projector, which are mounted on a two-axis active vision system. By measuring a projected coded pattern, 3-D measurement at a rate of 500 fps was achieved. The measurement range was increased as a result of the active tracking, and the shape of the target was accurately observed even when it moved quickly. In addition, to obtain the position and orientation of the target, 500 fps real-time model matching was achieved.

Keywords: high-speed vision; spatial coded pattern projection; depth image sensor; active tracking

1. Introduction

In recent years, depth image sensors capable of obtaining 3-D information with a single sensor unit have shown remarkable progress, and various types of depth image sensors have been mainly used in robot vision research to obtain the 3-D environment of a workspace in real time. One type of depth sensor in widespread use employs a spatial coded pattern method in which the space is represented by binary codes by projecting multiple striped patterns [1]. In this method, no special function is required for the camera and projector, and it is easy to increase the spatial resolution via a simple calculation. However, there is a drawback that it takes time to recover the 3-D shape from multiple projection images. Therefore, this method is not fast enough for real-time visual feedback, and it is also difficult to use it for dynamically changing environments.

On the other hand, 1 kHz high-speed vision systems have been studied by a number of research groups [2,3]. Such a system is effective for controlling a robot at high speed, and various applications of high-speed vision have been developed [4,5]. The performance of such a target tracking system has improved [6]. One such application is a high-speed 3-D sensing system in which a 1 kHz high-speed vision system and a high-speed projector are integrated, and this system achieved 500 Hz high-speed depth sensing [7]. However, because of the limited range of the projection, the measurement range is narrow, and it is difficult to apply this system to a robot that works over a wide area.

In this paper, we propose a 500 Hz active high-speed 3-D vision system equipped with a high-speed 3-D sensor mounted on an active vision mechanism [8]. The concept is shown in Figure 1.

By controlling the orientation of the camera and tracking the center of gravity of a target, this system realizes a wide measurement range and fast and highly accurate 3-D measurement. Since object tracking is performed so that the object is kept within the imaging range, the system compensates for a large motion of the object, and the state becomes nearly stationary, enabling accurate 3-D measurement. In order to use the obtained depth information for real-time robot control, real-time model matching was performed, and the position and orientation of a target object could be extracted at 500 Hz [9]. Finally, the validity of the active high-speed 3-D vision system was verified by actual experiments.

This paper is an extension of our previous conference papers [8,9]. In those papers, a basic system configuration and a basic method were proposed. However, the system details were not fully explained and there were not enough results to evaluate its performance. We have added detailed explanations and new results in this paper.

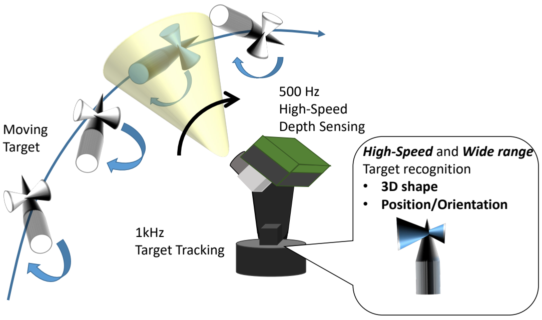


Figure 1. System concept.

2. Related Work

Three-dimensional depth sensors can acquire the 3-D shape of an object in real time more easily than conventional stereo cameras. Therefore, they are increasingly used in research on robot manipulation. In most of these sensor systems, 3-D point cloud data is obtained, and model matching is performed [10–12].

Concerning active measurement by projecting light patterns, many methods have been proposed [13,14]. The previous methods can be divided into one-shot measurement and multi-shot measurement. In one-shot methods, a 3-D shape can be measured with just a single projection by devising a suitable pattern to be projected. Therefore, it is easy to increase the speed [15–17].

On the other hand, in multi-shot methods, a 3-D image is acquired by projecting multiple patterns. These methods can obtain higher-resolution images than those of the one-shot methods; however, there is the disadvantage that they are not suitable for measurement of moving objects. Examples of multi-shot methods are the spatially coded pattern method [1,18] and the phase-shift method [19–22]. The spatially coded pattern method is easy to implement; however, in order to increase the resolution, the number of projection images needs to be increased. In the phase-shift method, high-resolution images can be obtained with three projected patterns. One problem, however, is the indeterminacy of corresponding points.

Research on increasing the speed by adopting a multi-shot method in order to measure moving subjects has been conducted. As examples of the phase-shift method, a phase-decoding method that considers movements of an object [23,24] and a method that integrates the phase-shift method with Gray-code pattern projection [25] have been proposed. Hyun and co-workers have proposed a system using a mechanical projector [26]. Lohry have proposed a system using 1-bit dithered patterns [27]. An example of the spatially coded pattern method is a 500 Hz measurement system using high-speed vision [7]. In order to measure a moving object, a method in which it is assumed that the movement is in one direction [28] and a method in which the movement of an object is constrained in advance [29] have been proposed. Although these previous methods have realized sufficient speed increases, they have difficulty in measuring a target that moves quickly in a large area. In the present study, by introducing a visual servoing technique, we have solved this problem, and we propose a method suitable for robot control [8,9].

Our group have been conducting research on high-speed vision systems [2,3]. These high-speed vision systems are based on a high-frame-rate image sensor and a parallel processing unit. By utilizing parallel computing techniques, these systems can achieve real-time image processing at rates of 500∼1000 Hz or higher. This is effective for controlling a robot at high speed, and various applications of high-speed vision have been developed [4,5]. Examples include a ball juggling robot [30] and a *kendama* robot [31]. In particular, the *kemdama* robot needs to handle an object with a complicated shape and measure its 3-D shape accurately. However, accurate 3-D measurement is difficult when using only 2-D stereo vision. In this paper, therefore, we propose a system that can quickly measure the 3-D shape of a target so as to achieve dexterous quick manipulation [9].

3. System

The configuration of the developed system is shown in Figure 2. A camera and a projector are mounted on a two-axis active vision system. First, pattern projection and image capturing are conducted by synchronization with a trigger from a computer for visual processing (Vision PC). Second, a spatial code, a depth map and the centroid of the object are calculated from the captured images by image processing with a Graphical Processing Unit (GPU). Third, the active vision system is controlled to track the target using a real-time control system, on the basis of the data about the centroid of the target. The real-time control system is constructed using a controller manufactured by dSPACE.

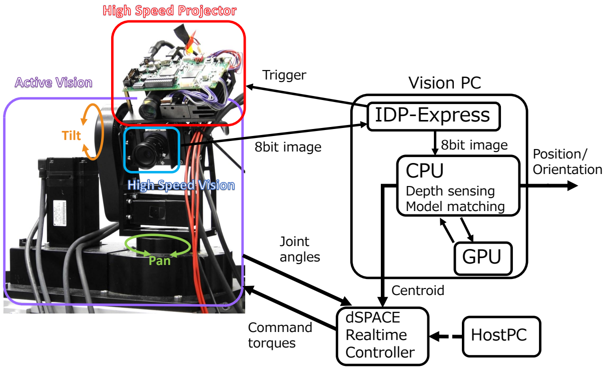


Figure 2. System configuration.

3.1. High-Speed Depth Sensor

The projector is a DLP Light Crafter 4500 manufactured by Texas Instruments (Dallas, TX, USA). The specifications are shown in Table 1. It projects a 9-bit Gray-code pattern of 912 × 1060 pixels at 1000 fps. The projected patterns are explained in Section 4.1.

The high-speed vision system is based on IDP-Express manufactured by Photron Inc. (San Diego, CA, USA) The specifications are shown in Table 2. The camera captures 8-bit monochrome, 512 × 512 pixel images at 1000 fps. The size of the camera head is 35 mm × 35 mm × 34 mm, and the weight is 90 g. The compact size and low weight of the camera reduce the influence of inertia.

In the Vision PC, the main processor is an Intel Xeon(R) E5-1630v4 (3.70 GHz, 4core), and the GPU is an NVIDIA Geforce GTX1080Ti.

Table 1. Specifications of projector.

Resolution (pattern sequence mode)	912 × 1140
Pattern rate (pre-loaded)	4225 Hz
Brightness	150 lm
Throw ratio	1.2
Focus range	0.5–2 m

Table 2. Specifications of camera and lens.

Image	8-bit Monochrome
Resolution	512 × 512
Frame rate	∼2000 Hz
Image size	1/1.8
Focal length	6 mm
FOV	57.4° × 44.3°

3.2. Active Vision

The active vision system, called AVS-III, has two orthogonal axes (pan and tilt), which are controlled by AC servo motors. The system specifications are shown in Table 3. It is capable of high-speed movement because each axis has no backlash and a small reduction ratio owing to a belt drive. The coordinate system is shown in Figure 3.

The performance of the tracking depends on the weight of the head and the power of the motor. In previous research such as [2], only a camera head was mounted and there was no projector. Therefore, it is estimated that the tracking performance was better than our system. However, we are currently designing a new head to reduce the inertia of the system to improve the performance.

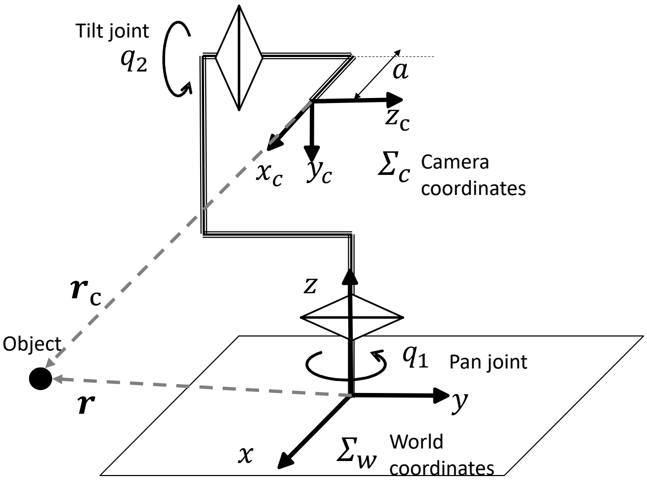


Figure 3. Active vision.

Table 3. Specifications in two axial directions.

	Pan	Tilt
Type of servo motor	Yaskawa SGMAS-06A	Yaskawa SGMAS-01A
Rated output [W]	600	100
Max torque [Nm]	5.73	0.9555
Max speed [rpm]	6000	
Reduction ratio	4.2	

3.3. Overall Processing Flow

A flowchart of the processing of the entire system is shown in Figure 4. It consists of three parts: high-speed depth image sensing, target tracking, and model matching. The system tracks a fast moving object by visual servoing, and a depth image is generated at 500 Hz while controlling the direction of the visual axis of the camera. Next, the generated image is converted to point cloud data, model matching is performed with a reference model, and the position and orientation of the target are detected at 500 Hz. A detailed explanation of each step is given in the next section.

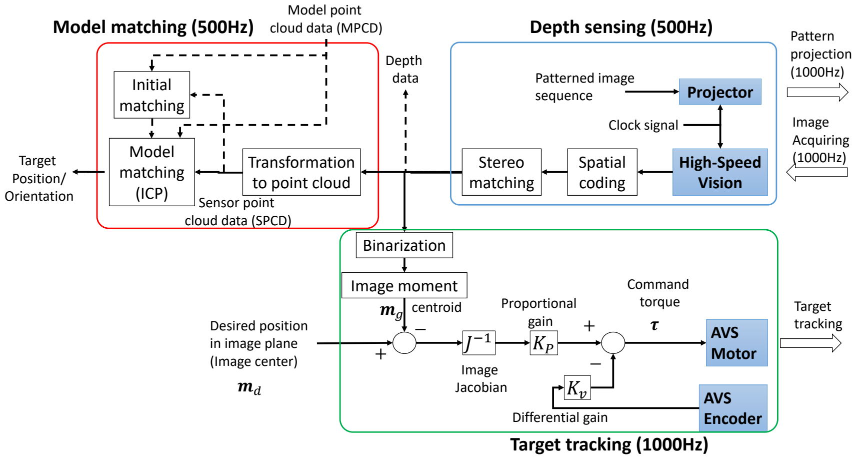


Figure 4. Block diagram.

4. High-Speed Depth Image Sensing

The system projects a 9-bit Gray-code pattern of 912 × 1060 pixels at 1000 fps. The projected pattern maps are shown in Figure 5a. After projection of each pattern, a black/white reversed image called a negative is projected in order to reduce errors at the borders between white and black areas. Therefore, a total of 18 image patterns are projected, and the actual measurement rate per pattern becomes 500 Hz. The reason why only 1060 of the total of 1140 lines are used is so that the projection angle corresponds with the angle of view.

The upper left corner is a space for recognizing the most significant bit pattern map, and it is impossible to measure a depth value in this area. In this system, the spatial code is composed of 265 rows (rows 0–264), which are mapped onto the 1060 projector rows while leaving enough space between rows for image adjustment because it is necessary to modify the pattern maps owing to the inclination of the optical axes of the camera and the projector.

4.1. Spatial Coding

In this system, we adopt the spatially coded pattern method [1] and a Gray-code pattern [32]. The reason for using a Gray-code pattern is that the code error decreases because the Hamming distance with a Gray-code is only one. When the spatial code is n bits, and the k-th projected binary image $P(x, y, k)$, $(0 \leq k \leq n-1)$, has a spatial resolution of $P_x \times P_y$ pixels, $P(x_p, y_p, k)$ can be expressed by the following equation:

$$P(x_p, y_p, k) = \left\lfloor \frac{y_p \times 2^k}{P_y} + \frac{1}{2} \right\rfloor \bmod 2, \tag{1}$$

where $\lfloor x \rfloor$ is the greatest integer less than or equal to x. To enhance the accuracy of the measurement, the projector alternately projects this pattern and its black/white inverted pattern. The projected pattern image $P(x_p, y_p, k)$ is called the "positive pattern", and its inversion is called the "negative pattern".

When a captured positive pattern image $I_{pk}(x_c, y_c)$ and a captured negative image $I_{nk}(x_c, y_c)$ are represented by 8-bit intensities, a binarized image $g_k(x_c, y_c)$ is obtained by the following equation:

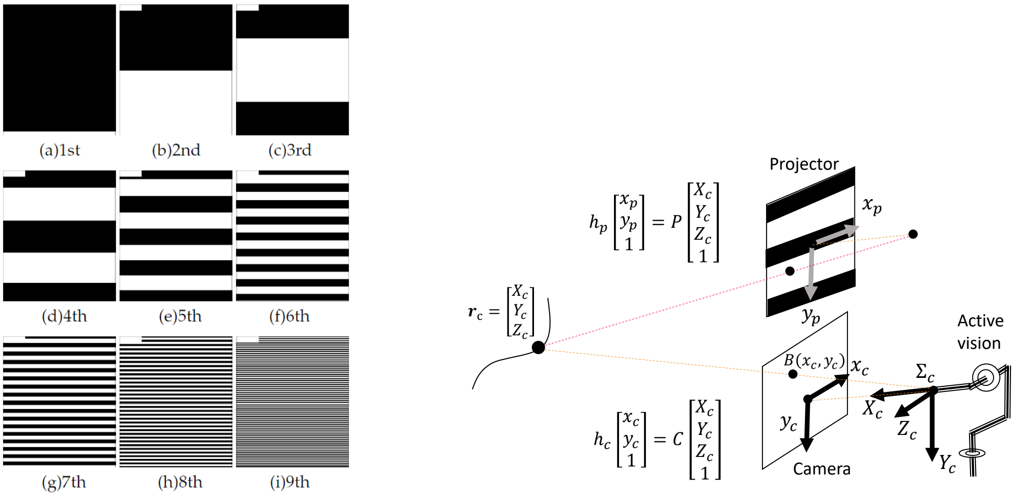
$$g_k(x_c, y_c) = \begin{cases} 1 & (I_{pk}(x_c, y_c) - I_{nk}(x_c, y_c) > \psi) \\ 0 & (I_{pk}(x_c, y_c) - I_{nk}(x_c, y_c) < -\psi) \\ \phi & \text{else} \end{cases} \tag{2}$$

where ψ is an intensity threshold for binarization, and ϕ is an ambiguous point because of occlusion.

The Gray code $G(x_c, y_c)$ is determined from n binary images:

$$G(x_c, y_c) = \{g_0(x_c, y_c), g_1(x, y), \cdots, g_{n-1}(x_c, y_c)\}. \tag{3}$$

Then, $G(x_c, y_c)$ is converted into the binary code $B(x_c, y_c)$.


(a)1st
(b)2nd
(c)3rd
(d)4th
(e)5th
(f)6th
(g)7th
(h)8th
(i)9th
Projector
$h_p \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = P \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$
x_p
y_p
$r_c = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$
Active vision
$B(x_c, y_c)$
x_c
Σ_c
X_c
Z_c
y_c
Y_c
Camera
$h_c \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = C \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$

(**a**) Pattern images (**b**) Active stereo

Figure 5. Depth sensing.

4.2. Stereo Calculation

By using the spatial code $B(x_c, y_c)$, the corresponding points on the projected image and the captured image can be detected. The relationship between the camera pixel coordinate, $\boldsymbol{u}_c = [x_c, y_c]^T$, the projector pixel coordinate, $\boldsymbol{u}_p = [x_p, y_p]^T$, and the 3-D position of the corresponding projected point, $\boldsymbol{r}_c = [X_c, Y_c, Z_c]^T$, is expressed as $h_c \bar{\boldsymbol{u}}_c = C \bar{\boldsymbol{r}}_c$ and $h_p \bar{\boldsymbol{u}}_p = P \bar{\boldsymbol{r}}_c$ where $C \in \boldsymbol{R}^{3 \times 4}$ and $P \in \boldsymbol{R}^{3 \times 4}$ are the camera parameter matrix and the projector parameter matrix obtained by calibration, and h_c and h_p are scale parameters. The operator $\bar{x}$ means a homogeneous vector $\boldsymbol{x}$.

Thus, $\boldsymbol{r}_c$ is obtained as

$$F \boldsymbol{r}_c = R, \quad F \triangleq \begin{bmatrix} C_{11} - C_{31} x_c & C_{12} - C_{32} x_c & C_{13} - C_{33} x_c \\ C_{21} - C_{31} y_c & C_{22} - C_{32} y_c & C_{23} - C_{33} y_c \\ P_{21} - P_{31} y_p & P_{22} - P_{32} y_p & P_{23} - P_{33} y_p \end{bmatrix}, \quad R \triangleq \begin{bmatrix} C_{34} x_c - C_{14} \\ C_{34} y_c - C_{24} \\ P_{34} y_p - P_{24} \end{bmatrix} \tag{4}$$

The results are calculated off-line and are used as a lookup table. For on-line measurements, using this look-up table improves execution speed.

4.3. Accuracy of Measurement

Consider the relationship between the frame rate and the delay. Although the frame rate of the camera and the projector is 1000 Hz, since one bit is obtained with a pair of images consisting of a positive image and its negative image in order to improve the accuracy, the frame rate for one-bit acquisition is actually 500 Hz. To generate a 9-bit depth image, 18 images are required. However, the depth image is updated every two images, that is, every 1-bit acquisition. Therefore, the update frame rate is 500 Hz, but its delay is $2 \times 18 = 36$ ms. This delay may degrade the accuracy of depth recognition for a moving target. However, if the target tracking is performed as described in the next section, the relative position error between the object and the camera becomes sufficiently small, and the target object can be regarded as if it has stopped.

Let us now analyze the relationship between measurement accuracy and target speed. The geometrical relationship in the active stereo vision is actually described by Equation (4). However, in order to simplify the problem, consider a simplified model shown in Figure 6, where it is assumed that the target is on a plane and the size of the projected plane is $W \times H$, the distance between the plane and the optical center is D, the distance between the projector and the camera is t, and $n = 9$ shows the number of the projected patterns.

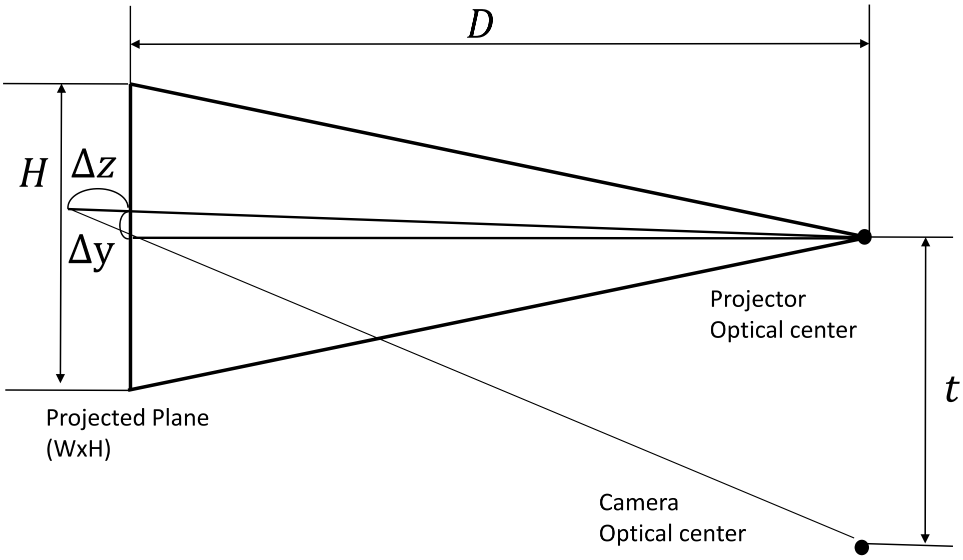


Figure 6. Analysis of accuracy.

Then, the maximum resolution in the horizontal direction is given by

$$\Delta y = \frac{H}{2^n}. \tag{5}$$

In our system, the throw ratio of the projector $D/W = 1.2$, and $W/H = 912/1060$, $t = 0.2$ m. Assuming that $D = 1.0$ m, the minimum resolution in the horizontal direction is given by $\Delta y = 1.4$ mm, and that in the depth direction is given by $\Delta z \simeq \Delta y/t \times D = 7$ mm.

The time to obtain one depth image is given by $\Delta t = 2n \times 10^{-3} = 18$ ms. Therefore, the maximum speed at which recognition can be performed is given by

$$v_x = \frac{\Delta y}{\Delta t} = 7.78 \text{ cm/s}, \quad v_z = \frac{\Delta z}{\Delta t} = 38.89 \text{ cm/s}. \tag{6}$$

Tis relative speed caused by the tracking delay of the camera can be compensated by adopting the visual tracking described in the next section. If the speed of the target is more than the speed shown above, the resolution will be reduced; however, its influence can be minimized by employing visual tracking.

4.4. Correction of Spatial Code

When only a high-order bit changes and lower bits remain unchanged, there is a possibility that the spatial code changes too much. Consider an example of a 4-bit Gray code, as shown in the Table 4. When $0010 = 3$ is obtained at step k and the most significant bit (MSB) changes to 1 at step $k+1$, the gray code become $1010 = 12$ because the lower bits in step k are used. This change is obviously too large compared to the actual change. In this case, the boundary is where the MSB switches between $0100 = 7$ and $1100 = 8$, and the farther from this boundary, the more the code changes. Therefore, we introduced the following correction method. A change of the MSB is suppressed when the code is away from the boundary. When the code is close to the boundary, the lower bits are forcibly set to

$100\cdots0$ to prevent a large change. The correction is executed in not only the MSB but also the lower bits. The process is described as follows (Algorithm 1).

Algorithm 1: Correction of spatial code.

```
if $g_k(i,j)$ is changed to $0 \to 1$ or $1 \to 0$ then
    $g_{k+1}(i,j) \to 1$
    $g_l(i,j) \to 0 \quad (l = k+2, k+3, \dots)$
end if
```

Table 4. 4-bit Gray code.

Digit	Gray Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001

5. Visual Tracking

5.1. Image Moment for Tracking

In our system, the direction of the camera–projector system is controlled according to the motion of the target. The desired direction is calculated by an image moment feature obtained from 3-D depth information. The $(k+l)$th image moment m is given by

$$m_{kl} = \sum_i \sum_j A(x_{c_i}, y_{c_j}) {x_{c_i}}^k {y_{c_j}}^l, \tag{7}$$

where $A(x_{c_i}, y_{c_j})$ is a binary image of the target at the pixel (x_{c_i}, x_{c_j}) on the camera image and is given by

$$A(x_{c_i}, y_{c_j}) = \begin{cases} 1 & (\phi_l < h_c(x_{c_i}, y_{c_j}) < \phi_h) \\ 0 & \text{else}, \end{cases} \tag{8}$$

where ϕ_l and ϕ_h are thresholds that show the depth range of the target object.

The image centroid $\boldsymbol{m}_g$ is calculated as

$$\boldsymbol{m}_g = \begin{bmatrix} \frac{m_{10}}{m_{00}} & \frac{m_{01}}{m_{00}} \end{bmatrix}^T, \tag{9}$$

and $\boldsymbol{m}_g$ is used as a desired position for target tracking control.

5.2. Visual Servoing Control

To realize a quick response, we adopt image-based visual servoing control [33]. In our system, the mechanism has only two degrees-of-freedom (2-DOF) joints, and the control method can be simplified as shown in [2,4]. The kinematic parameters are shown in Figure 3.

Assuming a pinhole camera model, the perspective projection between the image centroid $\boldsymbol{m}_g$ and the 3-D position of the target $\boldsymbol{r}_c = [X_c \; Y_c \; Z_c]^T$ in the camera coordinates can be expressed as

$$\boldsymbol{m}_g = \left[-\frac{fZ_c}{X_c}, \frac{fY_c}{X_c} \right]^T, \tag{10}$$

where f is the focal length. Differentiating both sides yields

$$\dot{\boldsymbol{m}}_g = \begin{bmatrix} \dfrac{fZ_c}{(X_c)^2} & 0 & -\dfrac{f}{X_c} \\ -\dfrac{fy_c}{(X_c)^2} & \dfrac{f}{X_c} & 0 \end{bmatrix} \dot{\boldsymbol{r}}_c \simeq \begin{bmatrix} 0 & 0 & -\dfrac{f}{X_c} \\ 0 & \dfrac{f}{X_c} & 0 \end{bmatrix} \dot{\boldsymbol{r}}_c, \tag{11}$$

where it is assumed that $X_c \gg Y_c$ and $X_c \gg Z_c$ when the distance between the object and the camera is large enough and when the object is kept around the image center. For image-based visual servoing, the depth data X_c may be an estimated value and need not be an accurate value. However, this affects the control system because of the change of the control gain due to changes of X_c.

The positional relationship between $\boldsymbol{r}$ in the world coordinates and $\boldsymbol{r}_c$ in the camera coordinates can be expressed as

$$\bar{\boldsymbol{r}} = \boldsymbol{T}_c \bar{\boldsymbol{r}}_c, \tag{12}$$

where $\boldsymbol{T}_c$ is the homogeneous transformation and the operator $\bar{\square}$ shows a homogeneous vector. Differentiating both sides using joint angle vector $\boldsymbol{q} = [q_1 \; q_2]^T$ yields:

$$\dot{\bar{\boldsymbol{r}}} = \dot{\boldsymbol{T}}_c \bar{\boldsymbol{r}}_c + \boldsymbol{T}_c \dot{\bar{\boldsymbol{r}}}_c = \left[\frac{\partial \boldsymbol{T}_c}{\partial q_1} \bar{\boldsymbol{r}}_c \; \frac{\partial \boldsymbol{T}_c}{\partial q_2} \bar{\boldsymbol{r}}_c \right] \dot{\boldsymbol{q}} + \boldsymbol{T}_c \dot{\bar{\boldsymbol{r}}}_c. \tag{13}$$

If the movement of the object is minute, it can be considered that $\dot{\bar{\boldsymbol{r}}} = 0$:

$$\dot{\bar{\boldsymbol{r}}}_c = -\boldsymbol{T}_c^{-1} \left[\frac{\partial \boldsymbol{T}_c}{\partial q_1} \bar{\boldsymbol{r}}_c \; \frac{\partial \boldsymbol{T}_c}{\partial q_2} \bar{\boldsymbol{r}}_c \right] \dot{\boldsymbol{q}}. \tag{14}$$

The relationship between $\dot{\boldsymbol{m}}_g$ and $\dot{\boldsymbol{q}}$ is obtained from Equations (11) and (14):

$$\dot{\boldsymbol{m}}_g = J\dot{\boldsymbol{q}}, \quad J \triangleq \begin{bmatrix} -\dfrac{f}{X_c}\left((a + X_c)\cos q_2 + Y_c \sin q_2\right) & 0 \\ -\dfrac{fZ_c}{X_c}\sin q_2 & -\dfrac{f}{X_c}(a + X_c) \end{bmatrix} \simeq \begin{bmatrix} -f\cos q_2 & 0 \\ 0 & -f \end{bmatrix} \tag{15}$$

where it is assumed that a, Y_c and Z_c are sufficiently small. The matrix J is called the "image Jacobian".

The active vision system tracks the target by torque control in accordance with the block diagram in Figure 4:

$$\boldsymbol{\tau} = K_P \boldsymbol{J}^{-1} \left(\boldsymbol{m}_d - \boldsymbol{m}_g \right) - K_v \dot{\boldsymbol{q}}, \tag{16}$$

where K_P is the positional feedback gain, and K_v is the velocity feedback gain. The vector $\boldsymbol{m}_d$ is the target position in the image, and it is usually set to $[0, 0]^T$. Notice that this control algorithm is based on the assumption that the frame rate of the vision system is sufficiently high. Because the vision obtains 3-D data at high-speed rate, it is not necessary to predict the motion of the target. Therefore, the control method is given as a simple direct visual feedback.

The target tracking starts when an object appears in the field of view. With regard to the model matching part, since it takes time for the initial processing, there is a possibility that the matching result may not be stabilized if the movement of the object is fast. On the other hand, the parts involving tracking and 3-D measurement work stably.

6. Realtime Model Matching

In this system, 3-D depth information can be output at high speed, but the data size is too large for real-time robot control. Therefore, the system is designed to output the target position and orientation by model matching [9]. The object model is given in the form of point cloud data (PCD), and the sensor information is also point cloud data. In this paper, we call the model point cloud data M-PCD, and the sensor point cloud data S-PCD. First, initial alignment is made, and then the position of the final model is determined by using the Iterative Closest Point (ICP) algorithm.

6.1. Initial Alignment

At each starting point of target tracking, initial alignment is performed by using a Local Reference Frame (LRF) [34]. For each point in the S-PCD and the M-PCD, the covariance matrix and its eigenvectors, as well as the normal of the curved surface, are calculated. Let N be the total number of points in the S-PCD and M-PCD, and $\boldsymbol{p}_i$ $(i = 1, 2, ..., N)$ be the position of each point in the camera coordinate system. The center $\boldsymbol{c}_i$ and the covariance matrix $\boldsymbol{C}_i$ of the neighborhood points of $\boldsymbol{p}_i$ can be obtained by the following expressions [35]:

$$\boldsymbol{c}_i = \frac{1}{n}\sum_{j=1}^{n} \boldsymbol{q}_j \tag{17}$$

$$\boldsymbol{C}_i = \frac{1}{n}\sum_{j=1}^{n} \left[(\boldsymbol{q}_j - \boldsymbol{c}_i)(\boldsymbol{q}_j - \boldsymbol{c}_i)^T\right], \tag{18}$$

where n is the total number of neighbors of $\boldsymbol{p}_i$, and $\boldsymbol{q}_j$ are the coordinates of the neighborhood of $\boldsymbol{p}_i$. Let $\lambda_0, \lambda_1, \lambda_2$ $(\lambda_0 \leq \lambda_1 \leq \lambda_2)$ be the eigenvalues of $\boldsymbol{C}_i$, and $\boldsymbol{e}_0, \boldsymbol{e}_1, \boldsymbol{e}_2$ be the corresponding eigenvectors. At this time, $\boldsymbol{e}_0$ coincides with the normal direction of the curved surface. Since the normal here is obtained from discrete neighbor points, it is actually an approximation of the normal.

It is necessary to correct the direction of the normal vector. The angle between the vectors $\boldsymbol{p}_i$ and $\boldsymbol{e}_0$ is given by $\theta_i = \cos^{-1}\left(\dfrac{\boldsymbol{p}_i \cdot \boldsymbol{e}_0}{\|\boldsymbol{p}_i\| \times \|\boldsymbol{e}_0\|}\right)$. If $\theta_i < \frac{\pi}{2}$, the direction of $\boldsymbol{e}_0, \boldsymbol{e}_1$ is reversed.

Next, points with large curvature are selected as feature points of the point clouds. The curvature is obtained by

$$k_i = \frac{2d_i}{\mu_i^2}, \tag{19}$$

where $d_i = ||\boldsymbol{e}_0 \cdot (\boldsymbol{p}_i - \boldsymbol{c}_i)||$, $\mu_i = \dfrac{1}{n}\sum_{j=1}^{n} \|\boldsymbol{q}_j - \boldsymbol{p}_i\|$. In order to find a feature point, the following normalized curvature $k_i' = \dfrac{k_i}{k_{max}}$ is actually used instead of k_i: where k_{max} is the maximum value of $k_i (i = 1, 2, ..., N)$.

Then, LRFs are generated at the selected feature points. In each LRF, the three coordinate axes are basically given as the corrected normal vector, the second eigenvector, and their outer product. For each feature point in the S-PCD and one feature point in the M-PCD, if the target object is moved and rotated so that the directions of the LRF coincide, and if feature points are selected in the same place, the target object and the model object match. However, since the M-PCD and the S-PCD are discrete data, it is not always possible to obtain feature points in the same place. To achieve robust matching, the second axis is replaced to use a voting method from the direction of the normal around the characteristic point [36]. Also, when the point cloud is symmetric when obtaining the second axis, two candidates of the second axis are detected. Therefore, the matching is performed for two kinds of second axes.

The homogeneous transformation T of the target object when matching an LRF of the M-PCD with an LRF of the S-PCD can be obtained by the following expression:

$$T \begin{bmatrix} \boldsymbol{l}_0 & \boldsymbol{l}_1 & \boldsymbol{l}_2 & \boldsymbol{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{l}'_0 & \boldsymbol{l}'_1 & \boldsymbol{l}'_2 & \boldsymbol{q} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

where $\boldsymbol{q}$ is the position and $\boldsymbol{l}'_0$, $\boldsymbol{l}'_1$, and $\boldsymbol{l}'_2$ are the axes of the LRF of the M-PCD, and $\boldsymbol{p}$ is the position and $\boldsymbol{l}_0$, $\boldsymbol{l}_1$, $\boldsymbol{l}_2$ are the axes of the LRF of the S-PCD.

To select the most suitable LRF, a threshold is set in advance, and only LRFs having a curvature exceeding the threshold are selected. Furthermore, the LRFs are down-sampled at a certain rate, which is set to 0.1 in this system. The sum of the distances between points in the S-PCD and points in the M-PCD is calculated for all combinations, and the LRF that minimizes the distance is selected.

6.2. ICP Algorithm

After the initial alignment, precise alignment is performed using the ICP algorithm [37]. In view of processing time and stability, we adopt the Point to Point method. ICP is a popular method in 3-D matching, so we briefly explain the processing flow in this section.

Let N_p be the number of points in S-PCD P and N_x be the number of points in M-PCD X. The closest point among X is associated with each data point $\boldsymbol{p}$ belonging to P. To speed up this association, we use a kd-tree to store the elements of the model. A point of X closest to $\boldsymbol{p}$ is determined as follows:

$$\boldsymbol{y} = \arg\min_{\boldsymbol{x} \in X} \|\boldsymbol{x} - \boldsymbol{p}\|, \tag{21}$$

and let $Y = \mathcal{C}(P, X)$ be the set of $\boldsymbol{y}$ by using the operator $\mathcal{C}$ for obtaining the nearest neighbor point.

Next, to calculate the correlation of the point $\boldsymbol{y}$ in X closest to $\boldsymbol{p}$, a covariance matrix is given by

$$\Sigma_{py} = \frac{1}{N_p} \sum_{i=1}^{N_p} \left[\left(\boldsymbol{p}_i - \mu_p\right) \left(\boldsymbol{y}_i - \mu_y\right)^{\mathrm{T}} \right], \tag{22}$$

where

$$\mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} \boldsymbol{p}_i, \mu_y = \frac{1}{N_p} \sum_{i=1}^{N_p} \boldsymbol{y}_i. \tag{23}$$

The singular value decomposition of the covariance matrix obtained by Equation (22) is:

$$\Sigma_{py} = U_{py} S_{py} {V_{py}}^{\mathrm{T}}. \tag{24}$$

Using U_{py} and V_{py}, the optimum rotation R can be expressed by the following equation [38]:

$$R = U_{py} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & \det(U_{py} {V_{py}}^{\mathrm{T}}) \end{bmatrix} {V_{py}}^{\mathrm{T}}. \tag{25}$$

The optimal translation vector $\boldsymbol{t}$ is obtained from this rotation matrix and the center of gravity:

$$\boldsymbol{t} = \mu_y - R\mu_p. \tag{26}$$

In the real-time model matching, the Equations (21), (25) and (26) are iteratively performed in one step. To speed up the convergence, the iterative calculation is divided into three stages. It is performed n_1 times for samples thinned out to a quarter of the S-PCD, n_2 times for samples thinned out to half, and n_3 times for all points. Hereinafter, the iteration count is described as $[n_1, n_2, n_3]$ times. For this time, convergence judgment based on the squared error between point groups is not performed, and the number of iterations is fixed.

Notice that the difference between the current S-PCD and the previous S-PCD is small because the rate of 3-D sensing is 500 Hz. For this reason, ICP can converge with a sufficiently small number of steps.

6.3. Down-Sampling

Although the high-speed 3-D sensor can obtain 3-D values with all 512 × 512 pixels, the processing time becomes too large if all of them are transformed to PCD. Therefore, it is necessary to reduce the number of points in the PCD.

First, the number i_{number} of target points is set. Next, the ratio C_p with respect to the target PCD number is calculated as follows using the number of pixels M from which the 3-D values can be obtained:

$$C_p = \begin{cases} \dfrac{M}{i_{number}} & \text{if } M > i_{number} \\ 1 & \text{else} \end{cases} \tag{27}$$

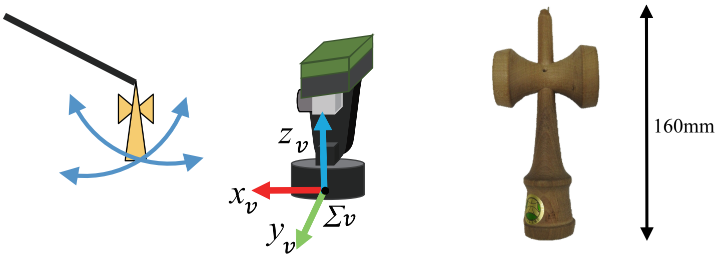
For $i = 1, 2, 3...$, only the $i \times C_p$th pixel ($i \times C_p < 512 \times 512$) is point clustered. Thereafter, down-sampling is performed to eliminate points within a cube of one side length r having the respective centers of gravity, and outlier points are removed when the average of 20 neighboring points is more than a certain value. Down-sampling is also performed for the M-PCD as well. In the experiments described in the next session, the final number of points was about 410.

7. Experiment

A video of the experimental results described in this section can be seen in [39].

7.1. High-Speed Depth Image Sensing

In the experiment, as the target object we used a "*ken*", which is a *kendama* stick, because it is composed of a complex curved surface, as shown in Figure 7b. In order to confirm that the sensor system could make 3-D measurements and could track the target, the *ken* at the tip of a rod was moved in front of the sensor. In this experiment, only 3-D measurement was performed without the real-time model matching.



(**a**) Layout (**b**)*Kendama stick*

Figure 7. Experimental setup.

7.1.1. Result with Target Tracking Control

In this section, we explain a result of 3-D measurement shown in [8]. The image centroid is shown in Figure 8. The system and the depth map expressed in color according to the 3-D value from the camera are shown in Figure 9. The *ken* is shown in the left image, and the depth map from the 3-D sensor is shown in the right image. The color of the depth map becomes redder as the object comes closer to the camera and bluer as the object goes farther away from the camera.

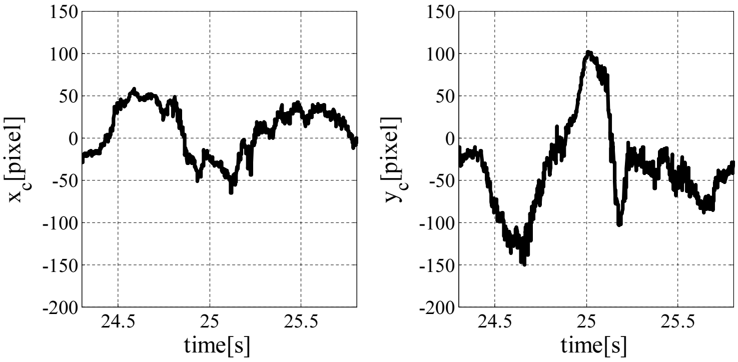


Figure 8. Temporal response of error in image plane [8].

The time taken to calculate the depth value was about 1.5 ms. This is shorter than the 2 ms taken to project 1-bit positive and negative pattern images. As seen in the depth map data in Figure 9, 3-D measurement seems to have been successful on the whole. Moreover, the sensor system could keep the target object within the field of view of the camera even when the *ken* moved dynamically, indicating that tracking of a moving object was also successful. However, when the *ken* moved violently, for example, at 25.0 s, the center of the image and the centroid of the *ken* were separated. This was due to the control method because there was no compensation for gravity and inertia. This issue will be addressed in future work. Regarding the depth maps in Figure 9, not only did the *ken* shape appear in the image, but also the inclination in the depth direction of the *ken* can be seen by the color gradation.

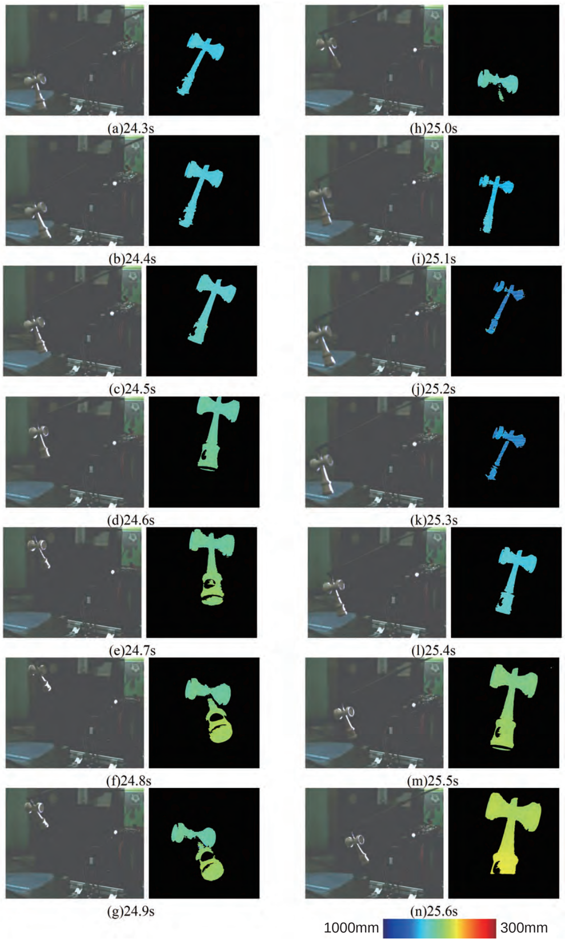
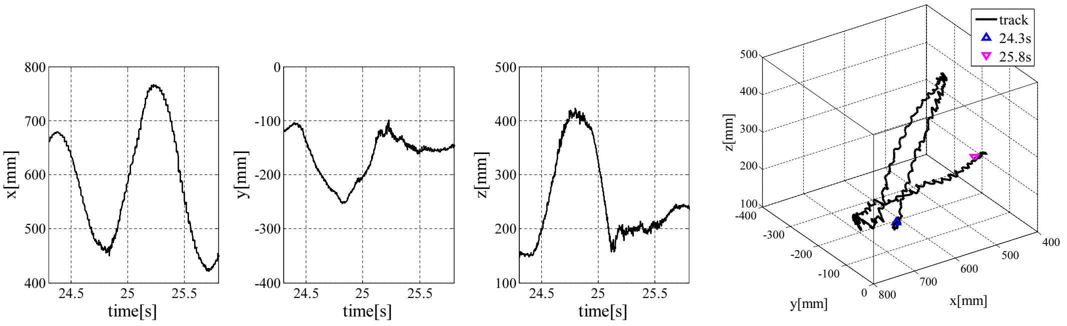


Figure 9. 3-D measurement with target tracking [8].

The temporal response for the position of the centroid of the object is shown in Figure 10a. The values in Figure 10a are expressed in the world coordinates. The 3-D trajectory of the center of gravity of the target is shown in Figure 10.

The trajectory of the centroid in Figure 10 was calculated offline by using the same model matching method. This is just to show the movement of the target for reference. Considering the size of the *ken* in the image and the trajectory, a wider 3-D measurement range could be achieved by tracking the target instead of fixing the high-speed 3-D sensor.



(**a**) Temporal response of centroid (**b**) 3-D trajectory of centroid

Figure 10. Temporal response of centroid [8].

7.1.2. Result without Tracking Tracking Control

For comparison, Figure 11 shows the depth map when target tracking was not performed. It can be seen that the grip of the *ken* disappeared when it was swung largely in the lateral direction. This is probably because the 3-D measurement did not catch up with the moving speed of the *ken*. On the other hand, when tracking was performed, such disappearance did not occur. This is probably because the relative target movement speed was reduced by target tracking.

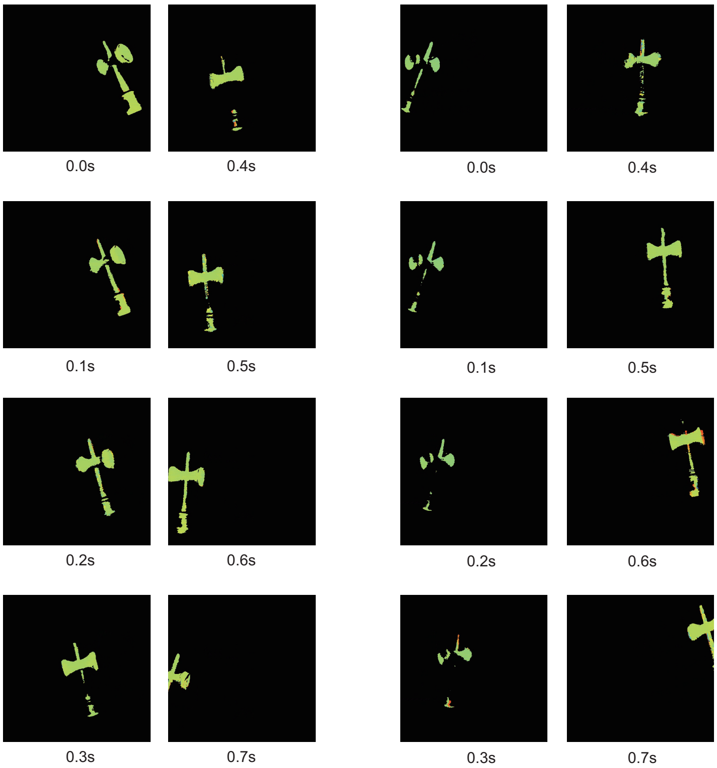
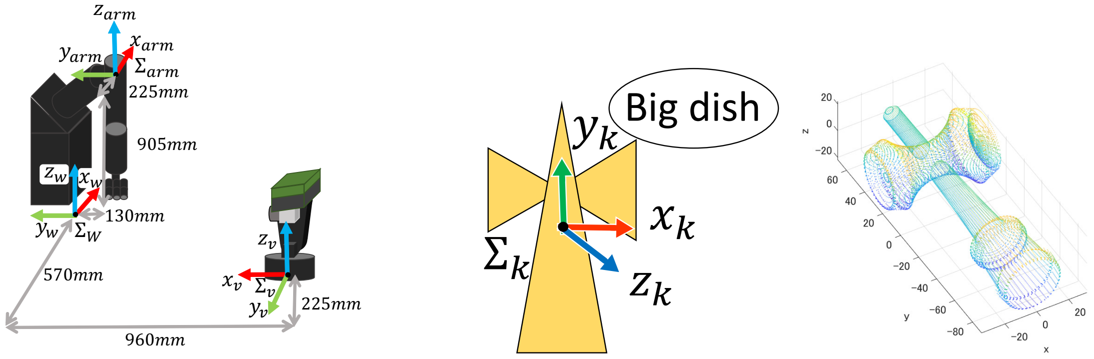


Figure 11. 3-D measurement without target tracking.

7.2. Model Matching

7.2.1. *Ken*

An experiment for the model matching was conducted using the *ken*. It was attached to a 7-DOF robot manipulator, and the true values of its position and orientation were calculated from the joint angles of the manipulator. These were compared with the values measured with the high-speed depth image sensor. The coordinate system fixed to the *ken* is shown in Figure 12a. Figure 12b shows the experimental setup of the manipulator and the high-speed depth image sensor. Figure 12c shows the M-PCD of the *ken*. The motion of the manipulator was such that it rotated about $\frac{\pi}{2}$ rad around the Z axis, $\frac{\pi}{4}$ rad around the Y axis, and $-\frac{\pi}{4}$ around the Z axis during about 2 s. In this trajectory, the maximum speed of the *ken* was about 1 m/s.



(**a**) Layout (**b**) Coordinate system (**c**) Model of ken

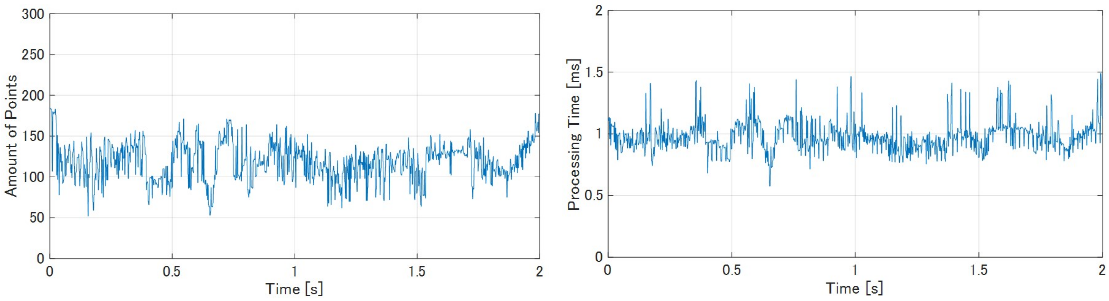
Figure 12. Experimental setup.

In the ICP, the number of iterations was set such that $[0,0,1]$, $i_{number} = 250$, and $r = 7.57$ mm. We set the number of points in M-PCD made from the CAD model by downsampling to 302.

The initial alignment was completed during a preparatory operation before the manipulator started to move. The time required for the initial alignment was 148 ms.

Figure 13a shows the temporal change of the number points in in S-PCD. It was adjusted depending on the parameter i_number, and it is shown that it was kept between 50 and 200 points.

Figure 13b shows the temporal change of the total processing time. It is shown that the processing time was about 1 ms and less than 1.5 ms. This means that the measurement could be achieved at a frame rate of 500 fps. In most of the measurement period, the observed S-PCD was a part of the overall body of the *ken*. However, by using the model matching, its position and orientation were accurately obtained within 2 ms.



(**a**) Number of points (**b**) Processing time

Figure 13. Result of model matching.

Figure 14 shows how the M-PCD and the S-PCD match. It can be seen that the M-PCD followed the motion of the *ken*. Figures 15 and 16 show the centroid and the orientation measured by the high-speed depth sensor and their true values calculated from the manipulator, respectively, in world coordinates. It is shown that there were no large errors in these measurements. It was difficult to distinguish between a large plate and a small plate in the *ken*, and it was reversed 180 degrees around the Y-axis sometimes. However, in most cases, the model matching was successful.

Notice that the cost of ICP is usually high, and it is not suitable for real-time high-speed matching in a conventional system. In our system, however, it is possible to reduce the number of iterations because the changes are sufficiently small as a result of the high frame rate.

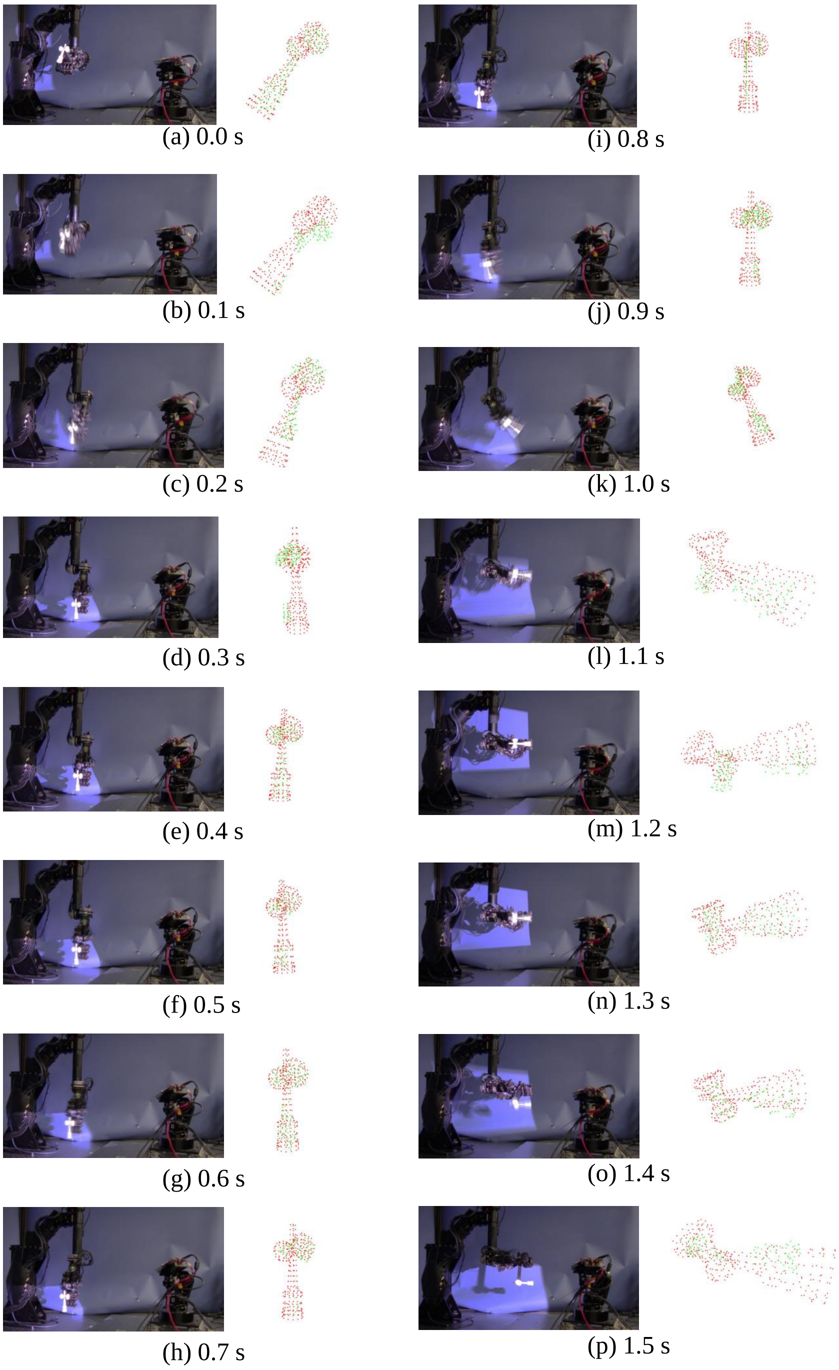
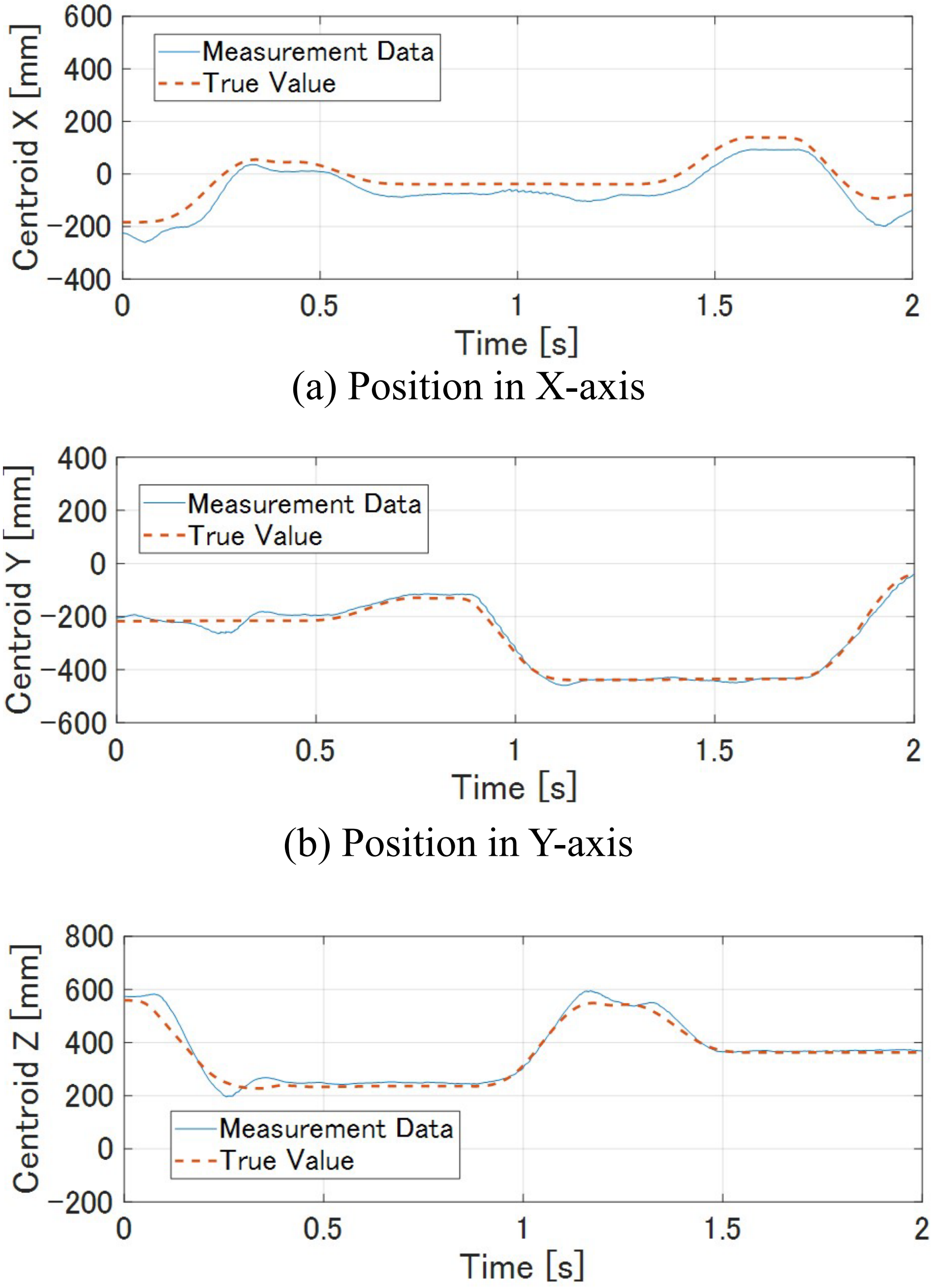

(a) 0.0 s
(b) 0.1 s
(c) 0.2 s
(d) 0.3 s
(e) 0.4 s
(f) 0.5 s
(g) 0.6 s
(h) 0.7 s
(i) 0.8 s
(j) 0.9 s
(k) 1.0 s
(l) 1.1 s
(m) 1.2 s
(n) 1.3 s
(o) 1.4 s
(p) 1.5 s

Figure 14. Continuous images of real-time model matching.

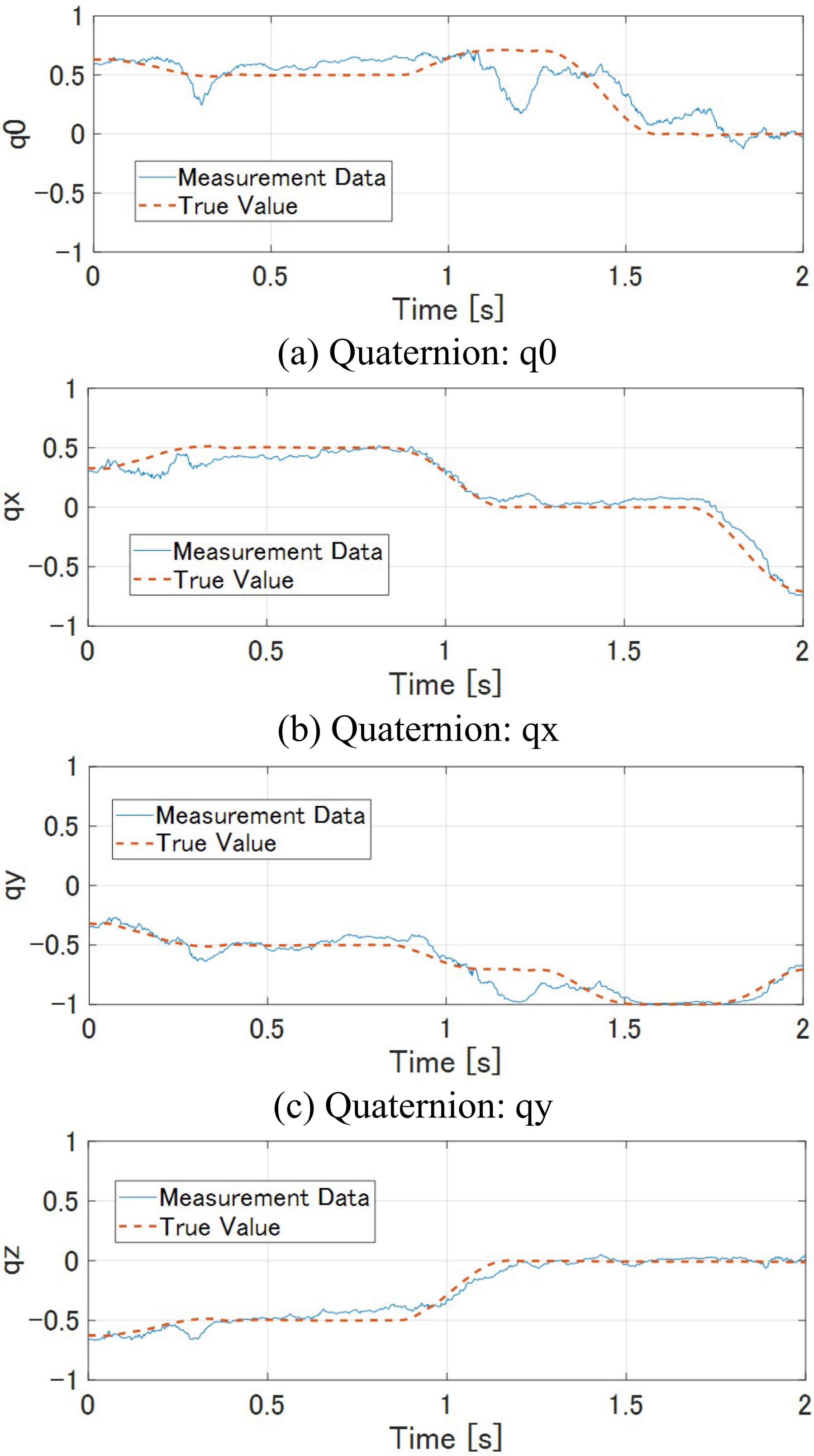


(a) Position in X-axis

(b) Position in Y-axis

(c) Position in Z-axis

Figure 15. Centroid of *ken*.



(a) Quaternion: q0

(b) Quaternion: qx

(c) Quaternion: qy

(d) Quaternion: qz

Figure 16. Quaternion of *ken*.

7.2.2. Cube

As a target, we used a 10 cm cube and comparing the results with the above results. Because the cube has a simpler shape than the *ken*, the model matching becomes more difficult. Therefore, we set the number of points in the M-PCD to 488 and i_{number} to 350 so as to increase the number of points.

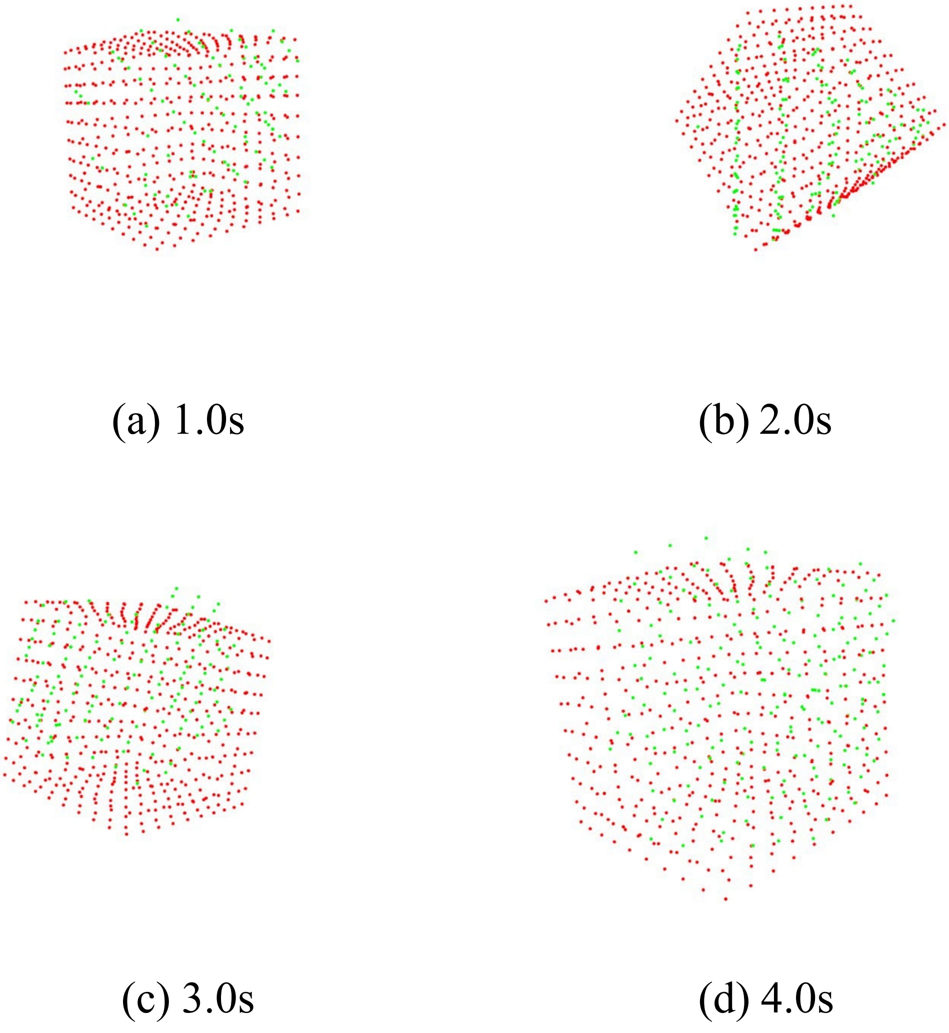
The time required for the initial alignment was 93.98 ms. The frequency of the ICP was 487 fps, which did not match the frame rate of 500 fps. Therefore, the frame rate of the ICP was asynchronous with the rate of the measurements in this experiment.

Figures 17 and 18 show the results of model matching for the cube and the time response of its centroid, respectively. It can be seen that the model matching and tracking succeeded, like the result for the *ken*.

7.2.3. Cylinder

We used a cylinder as a target. We set the number of points on the M-PCD to 248 and i_{number} to 300. The frequency of the ICP was 495 fps.

Figures 19 and 20 show the results of model matching for the cube and the time response of its centroid, respectively. It can also be seen that the model matching and tracking succeeded, like the results for the *ken* and the cube.

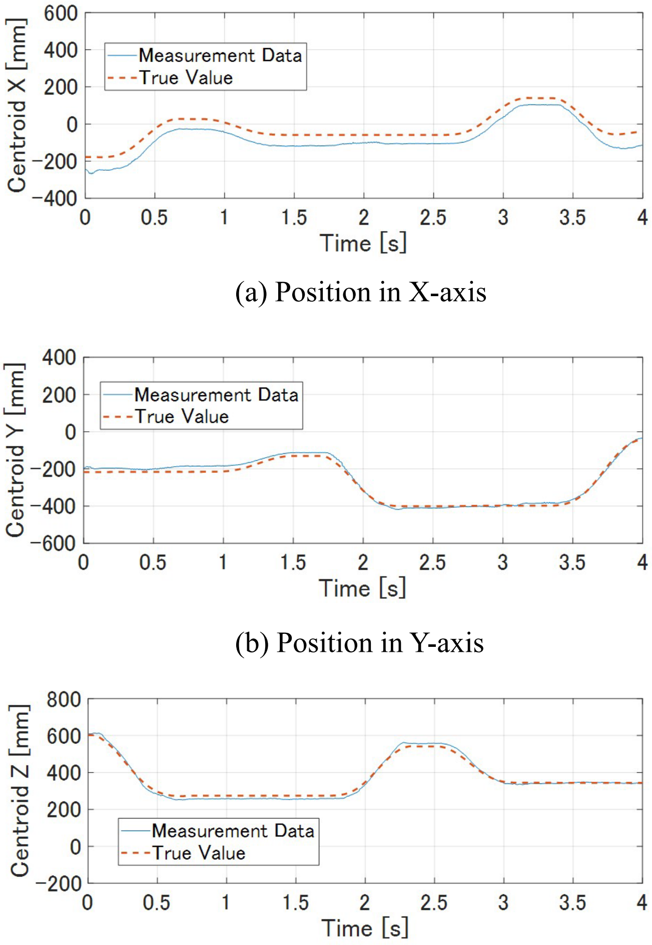

(a) 1.0s

(b) 2.0s

(c) 3.0s

(d) 4.0s

Figure 17. Point cloud of cube.

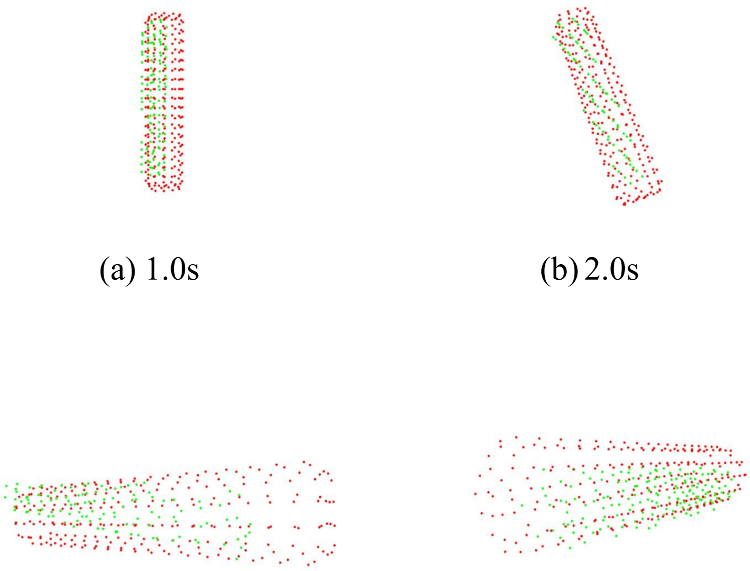


(a) Position in X-axis

(b) Position in Y-axis

(c) Position in Z-axis

Figure 18. Centroid of cube.



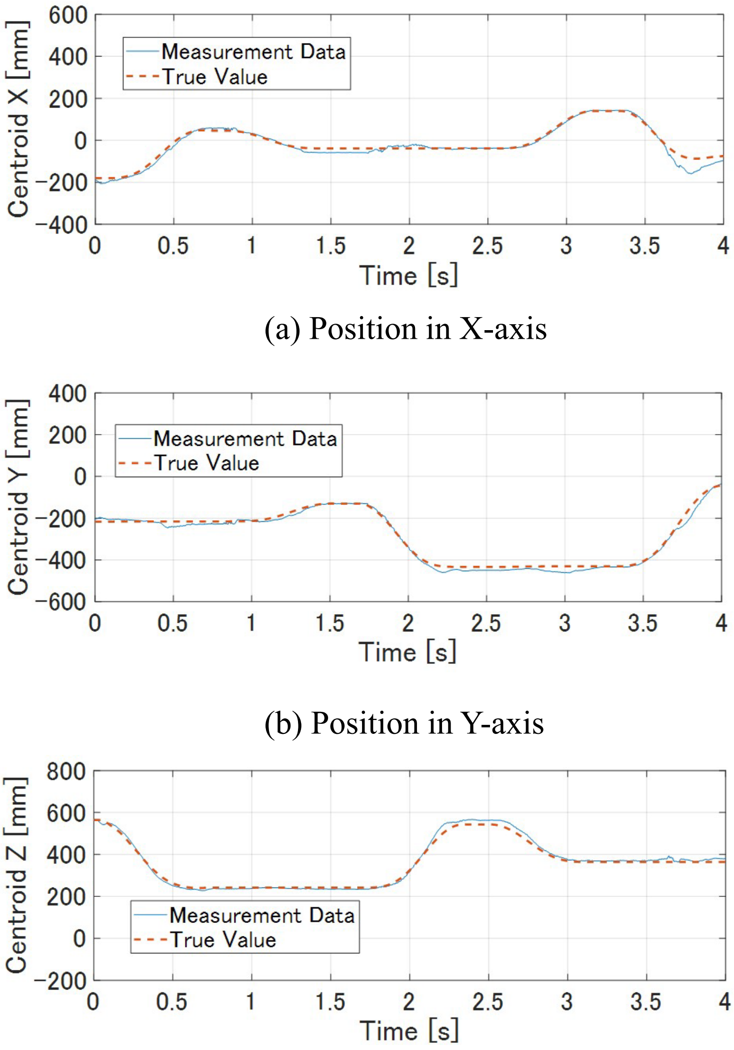
(a) 1.0s

(b) 2.0s

(c) 3.0s

(d) 4.0s

Figure 19. Point cloud data of cylinder.



(a) Position in X-axis

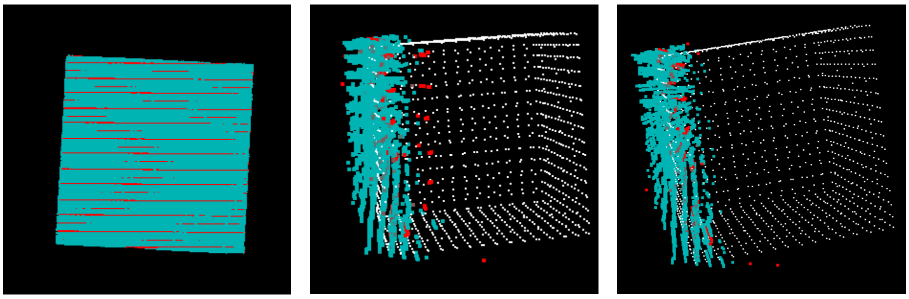
(b) Position in Y-axis

(c) Position in Z-axis

Figure 20. Centroid of cylinder.

7.3. Verification of Correction of Spatial Code

We conducted an experiment to verify the effect of the correction described in Section 4.4. Figure 21 (a) shows a camera image, (b) shows the point cloud before the correction, and (c) shows the point cloud after the correction. The blue points represent the observation point cloud, and the red points represent the points to be corrected.



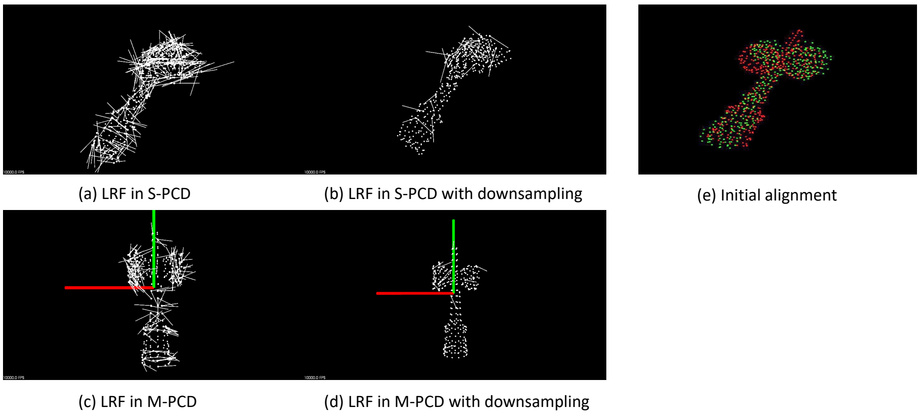
(a) A camera image (b) Non-compensated data (c) Compensated data

Figure 21. Correction of spatial code.

In (b), it can be seen that the point cloud to be corrected is in the wrong position. In (c), these points are moved around the correct position. This indicates that the proposed correction is valid.

7.4. Verification of Initial Alignment

In Figure 22a–d, the examples of LRFs are shown. Figure 22a shows the LRFs of the S-PCD, and Figure 22b shows the down-sampled LRFs, down-sampled to 0.1. Figure 22c,d are the original LRFs and the downsampled LRFs of the M-PCD. Figure 22e shows an example of the initial alignment. It can be seen that the M-PCD is roughly matched with the S-PCD.



(a) LRF in S-PCD

(b) LRF in S-PCD with downsampling

(e) Initial alignment

(c) LRF in M-PCD

(d) LRF in M-PCD with downsampling

Figure 22. Result of initial matching.

8. Conclusions

In this paper, we proposed a 500 Hz active high-speed 3-D vision system that has a high-speed 3-D sensor mounted on an active vision mechanism. By controlling the orientation of the camera and tracking the center of gravity of a target, it realizes both a wide measurement range and highly accurate high-speed 3-D measurement. Since object tracking is performed so that the object falls within the image range, a large motion of the object is compensated for, enabling accurate 3-D measurement. Real-time model matching was also achieved, and the position and orientation of the target object could be extracted at 500 Hz.

There are some problems to be addressed in the system. First, the performance of the tracking control is not sufficient. This will be improved by adopting improved visual servoing techniques taking account of the dynamics of the system. Also, it will be necessary to decrease inertia and reduce the size of the mechanism.

Second, although the coded pattern projection method was used in our current system, multiple image projections resulted in a delay. As a way to reduce the delay, we are considering acquiring 3-D information by other methods, such as one-shot or phase-shift methods. Moreover, due to the high frame rate and the active tracking, the target motion in each measurement step is very small. By utilizing this feature, it is possible to reduce the number of projection images, and we are currently developing an improved algorithm.

Author Contributions: Conceptualization, A.N.; methodology, A.N., K.S., Y.K., and I.I.; software, K.S. and Y.K.; validation, K.S. and Y.K.; formal analysis, A.N, K.S. and Y.K.; investigation, K.S. and Y.K.; resources, A.N. and I.I.; data curation, K.S. and Y.K.; writing—original draft preparation, A.N., K.S., and Y.K.; writing—review and editing, A.N.; visualization, A.N., K.S., and Y.S.; supervision, A.N.; project administration, A.N.; funding acquisition, A.N.

Funding: This work was supported in part by JSPS KAKENHI Grant Number JP17H05854 and Impulsing Paradigm Change through Disruptive Technologies (ImPACT) Tough Robotics Challenge program of Japan Science and Technology (JST) Agency.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Posdamer, J.L.; Altschuler, M.D. Surface measurement by space-encoded projected beam systems. *Comput. Graph. Image Process.* **1982**, *18*, 1–17. [CrossRef]
2. Nakabo, Y.; Ishii, I.; Ishikawa, M. 3-D tracking using two high-speed vision systems. In Proceedings of the 2002 IEEE/RSJ IEEE International Conference Intelligent Robots, Lausanne, Switzerland, 30 September–4 October 2002; pp. 360–365.
3. Ishii, I.; Tatebe, T.; Gu, Q.; Moriue, Y.; Takaki, T.; Tajima, K. 2000 fps real-time vision system with high-frame-rate video recording. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 1536–1541.
4. Namiki, A.; Nakabo, Y.; Ishii, I.; Ishikawa, M. 1 ms Sensory-Motor Fusion System. *IEEE Trans. Mech.* **2000**, *5*, 244–252. [CrossRef]
5. Ishikawa, M.; Namiki, A.; Senoo, T.; Yamakawa, Y. Ultra High-speed Robot Based on 1 kHz Vision System. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, 7–12 October 2012; pp. 5460–5461.
6. Okumura, K.; Yokoyama, K.; Oku, H.; Ishikawa, M. 1 ms Auto Pan-Tilt-video shooting technology for objects in motion based on Saccade Mirror with background subtraction. *Adv. Robot.* **2015**, *29*, 457–468. [CrossRef]
7. Gao, H.; Takaki, T.; Ishii, I. GPU-based real-time structured light 3-D scanner at 500 fps. *Proc. SPIE* **2012**, *8437*, 8437J.
8. Shimada, K.; Namiki, A.; Ishii, I. High-Speed 3-D Measurement of a Moving Object with Visual Servo. In Proceedings of the IEEE/SICE International Symposium on System Integration, Sapporo, Japan, 13–15 December 2016; pp. 248–253.
9. Kin, Y.; Shimada, K.; Namiki, A.; Ishii, I. High-speed Orientation Measurement of a Moving Object with Patterned Light Projection and Model Matching. In Proceedings of the 2018 IEEE Conference on Robotics and Biomimetics, Kuala Lumpur, Malaysia, 12–15 December 2018; pp. 1335–1340.
10. Klank, U.; Pangercic, D.; Rusu, R.B.; Beetz, M. Real-time cad model matching for mobile manipulation and grasping. In Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots, Paris, France, 7–10 December 2009; pp. 290–296.
11. Rao, D.; Le, Q.V.; Phoka, T.; Quigley, M.; Sudsang, A.; Ng, A.Y. Grasping novel objects with depth segmentation. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2578–2585.
12. Arai, S.; Harada, T.; Touhei, A.; Hashimoto, K. 3-D Measurement with High Accuracy and Robust Estimation for Bin Picking. *J. Robot. Soc. Jpn.* **2016**, *34*, 261–271. (In Japanese) [CrossRef]
13. Salvi, J.; Fernandez, S.; Pribanic, T.; Llado, X. A state of the art in structured light patterns for surface profilometry. *Pattern Recognit.* **2010**, *43*, 2666–2680. [CrossRef]
14. Geng, J. Structured-light 3-D surface imaging: A tutorial. *Adv. Opt. Photonics* **2011**, *3*, 128–160. [CrossRef]
15. Watanabe, Y.; Komuro, T.; Ishikawa, M. 955-fps realtime shape measurement of a moving/deforming object using high-speed vision for numerous-point analysis. In Proceedings of the IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007.
16. Tabata, S.; Noguchi, S.; Watanabe, Y.; Ishikawa, M. High-speed 3-D sensing with three-view geometry using a segmented pattern. In Proceedings of the IEEE/RSJ International Conference Intelligent Robots and Systems, Hamburg, Germany, 28 September–2 October 2015.
17. Kawasaki, H.; Sagawa, R.; Furukawa, R. Dense One-shot 3-D Reconstruction by Detecting Continuous Regions with Parallel Line Projection. In Proceedings of the IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
18. Gupta, M.; Agrawal, A.; Veeraraghavan, A.; Narasimhan, S.G. A practical approach to 3-D scanning in the presence of interreflections, subsurface scattering and defocus. *Int. J. Comput. Vis.* **2013**, *102*, 33–55. [CrossRef]
19. Wang, Y.; Zhang, S.; Oliver, J.H. 3-D shape measurement technique for multiple rapidly moving objects. *Opt. Express* **2011**, *19*, 8539–8545. [CrossRef] [PubMed]
20. Okarma, K.; Grudzinski, M. The 3-D scanning system for the machine vision based positioning of workpieces on the CNC machine tools. In Proceedings of the IEEE 17th International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, Poland, 27–30 August 2012; pp. 85–90.

21. Lohry, W.; Vincent, C.; Song, Z. Absolute three-dimensional shape measurement using coded fringe patterns without phase unwrapping or projector calibration. *Opt. Express* **2014**, *22*, 1287–1301. [CrossRef] [PubMed]
22. Hyun, J.S.; Song, Z. Enhanced two-frequency phase-shifting method. *Appl. Opt.* **2016**, *55*, 4395–4401. [CrossRef] [PubMed]
23. Cong, P.; Xiong, Z.; Zhang, Y.; Zhao, S.; Wu, F. Accurate dynamic 3-D sensing with fourier-assisted phase shifting. *IEEE J. Sel. Top. Signal Process.* **2015**, *9*, 396–408. [CrossRef]
24. Li, B.; Liu, Z.; Zhang, S. Motion-induced error reduction by combining fourier transform profilometry with phaseshifting profilometry. *Opt. Express* **2016**, *24*, 23289–23303. [CrossRef] [PubMed]
25. Maruyama, M.; Tabata, S.; Watanabe, Y.; Ishikawa, M. Multi-pattern embedded phase shifting using a high-speed projector for fast and accurate dynamic 3-D measurement. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Lake Tahoe, NV, USA, 12–15 March 2018; pp. 921–929.
26. Hyun, J.S.; Chiu, G.T.C.; Zhang, S. High-speed and high-accuracy 3-D surface measurement using a mechanical projector. *Opt. Express* **2018**, *26*, 1474–1487. [CrossRef] [PubMed]
27. Lohry, W.; Zhang, S. High-speed absolute three-dimensional shape measurement using three binary dithered patterns. *Opt. Express* **2014**, *22*, 26752–26762. [CrossRef] [PubMed]
28. Ishii, I.; Koike, T.; Gao, H.; Takaki, T. Fast 3-D Shape Measurement Using Structured Light Projection for a One-directionally Moving Object. In Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society, Melbourne, Australia, 7–10 November 2011; pp. 135–140.
29. Liu, Y.; Gao, H.; Gu, Q.; Aoyama, T.; Takaki, T.; Ishii, I. High-frame-rate structured light 3-D vision for fast moving objects. *J. Robot. Mech.* **2014**, *26*, 311–320. [CrossRef]
30. Kizaki, T.; Namiki, A. Two ball juggling with high-speed hand-arm and high-speed vision system. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 1372–1377.
31. Ito, N.; Namiki, A. Ball catching in kendama game by estimating grasp conditions based on a high-speed vision system and tactile sensors. In Proceedings of the 14th IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 November 2014; pp. 634–639.
32. Inokuchi, S.; Sato, K.; Matsuda, F. Range Imaging System for 3-D object recognition. In Proceedings of the International Conference on Pattern Recognition, Montreal, QC, Canada, 30 July–2 August 1984; pp. 806–808.
33. Hutchinson, S.; Hager, G.D.; Corke, P.I. A tutorial on visual servo control. *IEEE Trans. Robot. Autom.* **1996**, *12*, 651–670. [CrossRef]
34. Minowa, R.; Namiki, A. Real-time 3-D Recognition of a Manipulated Object by a Robot Hand Using a 3-D Sensor. In Proceedings of the 2015 IEEE Conference on Robotics and Biomimetics, Zhuhai, China, 6–9 December 2015; pp. 1798–1803.
35. Xu, F.; Zhao, X.; Hagiwara, I. Research on High-Speed Automatic Registration Using Composite-Descriptor-Could-Points (CDCP) Model. *JSME Pap. Collect.* **2012**, *78*, 783–798. (In Japanese) [CrossRef]
36. Akizuki, S.; Hashimoto, M. DPN-LRF: A Local Reference Frame for Robustly Handling Density Differences and Partial Occlusions. In Proceedings of the International Symposium on Visual Computing, Las Vegas, NV, USA, 14–16 December 2015; pp. 878–887.
37. Besl, P.J.; McKay, N.D. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1992**, *14*, 239–256. [CrossRef]
38. Eggert, D.W.; Lorusso, A.; Fisher, R.B. Estimating 3-D rigid body transformations: A comparison of four major algorithms. *Mach. Vis. Appl.* **1997**, *9*, 272–290. [CrossRef]
39. Active High-Speed 3-D Vision System, NamikiLaboratory Youtube Channel. Available online: https://www.youtube.com/watch?v=TIrA2qdJBe8 (accessed on 17 January 2019).