SICE Annual Conference 2007
Sept. 17-20, 2007, Kagawa University, Japan

On-line Stable Evolutionary Recognition by Quaternion Motion-Feedforward Compensation

Wei Song[1], Mamoru Minami[1], Yasushi Mae[1] and Seiji Aoyagi[2]

[1]Graduate school of Engineering, University of Fukui, Fukui, Japan
(Tel : +81-776-27-8527;Email: {songwei,minami,mae}@rc.his.fukui-u.ac.jp)
[2]Faculty of Engineering, Kansai University, Kansaio, Japan
(E-mail: aoyagi@iecs.kansai-u.ac.jp)

Abstract: This paper presents a pose measurement method of a 3D object. The proposed method utilizes an evolutionary search technique of a genetic algorithm (GA) and a fitness evaluation based on a matching stereo model whose pose is expressed by unit quaternion. To improve the dynamics of recognition, a motion-feedforward compensation method is proposed for the hand-eye system. The effectiveness of the proposed method is confirmed by simulation experiments.

Keywords: 3D pose measurement, GA, motion-feedforward compensation

1. INTRODUCTION

In recent years, object recognition, visual tracking and servoing using a stereo camera system have been studied intensively in the field of robotics and in other research areas. For a robot to be much smarter than just a mechanical device, vision is required so that it can adapt itself to a changing working environment and recognize objects that exist in its surroundings. Tasks in which visual information are used to direct the end-effector of a manipulator toward a target object are referred to visual servoing. This field is the fusion of many areas, such as kinematics, dynamics, image recognition, and control theory. This paper deals with problems of the real-time 3D pose (both position and orientation) measurement of a target.

Since a visual servoing system incorporates the vision sensor in the feedback loop, a changing of the sensing unit will cause direct influence to the output motion of the robot manipulator. So it is important to improve the dynamics of the sensing unit which may cause the feedback system unstable. Here, we define the recognition dynamics as a phenomenon that the sensed variables (the 3D pose of the target object) can be detected with time delay because sensing mechanism generally be governed by differential equations in time domain. Recently, several researches deal with the problem of recognition dynamics. Hashimoto and Kimura [1] propose a nonlinear controller and a nonlinear observer for the visual servo system to estimate the object velocity and predict the object motion. Theoretically, prediction without error can be obtained when time is infinity using nonlinear observer. However, the initial error exists and it is possible to cause the visual servoing system unstable. The same method is also used by Luca [2] to estimate the distance between the object to the camera. However, prediction makes no effort at the beginning of the estimation. And the convergence to the true value is obtained during the motion of the camera, that is, the method does not work if the camera is static. As we know, there is a big difference between the sampling rate of the camera 33[ms] and that of the joint controller 1[ms], which also cause the time delay of the sensing unit. Nakabou and Ishigawa [3] use a vision chip whose sampling rate is about 1[ms] to perform high-speed image processing. It has been shown that high-speed moving object can be tracked by using vision chip without any prediction or compensation. However, such a high-speed vision chip system is so expensive that can not be applied popularly.

In this paper, we proposed a motion-feedforward method to improve the recognition dynamics of a hand-eye robot system that has two cameras mounted on the end-effctor. Since the cameras are mounted on the robot end-effector, it is important for the robot to distinguish what is the real motion of the target object and what is a fictional motion just coming from the cameras. Motion-feedforward method is to predict the target's 3D pose based on the motion of the end-effector to compensate the target's fictional motion coming from the cameras. When the fictional motions are compensated during recognizing the target object using hand-eye cameras, it seems that the recognition is performed by using just fixed cameras, so the recognition will become easier and the recognition dynamics will be improved. Contrast to the nonlinear observer method, the proposed motion-feedforward method can give effective prediction as soon as the camera starts to move. So the stability of visual servo system can be guaranteed from the beginning.

Here, we use model-based method to deal with problems of the real-time 3D pose measurement. Unit quaternion is used to represent the orientation of the target object, which has a advantage that can represent the orientation of a rigid body without singularities. Solid models are defined to search the target object in the image and their orientations are also represented by unit quaternion. The matching degree of the model to the target can be estimated by a fitness function, whose maximum value represents the best matching and can be solved by GA. An advantage of our method is that we use a 3D solid model which enables it possesses six-DOF. In other methods like feature-based recognition, the pose of the target object should be determined by a set of image points, which makes it need a very strict camera calibration. Moreover, searching the corresponding points in Stereo-vision camera images is also complicated and time consuming, e.g., [4]. There is another approach of 3D pose measurement, named appearance-based method, e.g. [5]. The image is compared with various templates, which are made based on the object in different views beforehand, resulting in wasting time to recognize.

The GA-based scene recognition method described here can be designated as an “evolutionary recognition method”, since for every step of the GA’s evolution, it struggles to perform the recognition of a target in the input images. To recognize a target input by CCD camera in Real-Time, and to avoid time lag waiting for the convergence to a target, we used GA in such manner that only one generation is processed to a newly input image, which we called “1-Step GA” [8]. In this way, the GA searching process and the convergence to the target is not completed in one image but the recognition is achieved by the sequence of the input images, where the GA converges on the target in the continuously input images.

2. QUATERNION REPRESENTATION OF 3D POSE

In this paper, unit quaternion is used to represent the orientation of the target object, which has a advantage that can represent the orientation of a rigid body without singularities (singularities exist in other orientation representations, like three Euler angles, angle/axis representation, et al.). Recently, unit quaternion has been successfully used for attitude control of rigid bodies [6] and control of robot manipulator [7].

The unit quaternion, viz. Euler pparameters, defined as

$$\boldsymbol{Q} = \{\eta, \boldsymbol{\epsilon}\}, \tag{1}$$

If an object is rotated by an angle θ around an axis $\boldsymbol{k}$ by angle/axis representation, then the orientation defined by quaternion is where

$$\eta = cos\frac{\theta}{2}, \qquad \epsilon = sin\frac{\theta}{2}\boldsymbol{k}. \tag{2}$$

η is called the scalar part of the quaternion while ϵ is called the vector part of the quaternion. They are constrained by

$$\eta^2 + \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} = 1 \tag{3}$$

hence the name unit quaternion. It is worth remarking that, differently from the angle/axis representation, a rotation by an angle $-\theta$ about an axis $-\boldsymbol{k}$ gives the same quaternion as that assosiated with a rotation by θ about $\boldsymbol{k}$ which solves the nonuniqueness problem. Also, no singularity occurs.

The rotation matrix corresponding to a given quaternion is

$$\boldsymbol{R}(\eta, \boldsymbol{\epsilon}) = (\eta^2 - \boldsymbol{\epsilon}^T\boldsymbol{\epsilon})\boldsymbol{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T + 2\eta\boldsymbol{S}(\boldsymbol{\epsilon}) \tag{4}$$

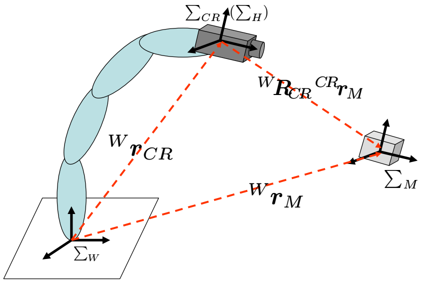

$\Sigma_{CR}(\Sigma_H)$
${}^{W}\boldsymbol{R}_{CR}\,{}^{CR}\boldsymbol{r}_M$
${}^{W}\boldsymbol{r}_{CR}$
Σ_M
${}^{W}\boldsymbol{r}_M$
Σ_W

Fig. 1 Hand-eye coordinate system

On the other hand, the quaternion corresponding to a given rotation matrix $\boldsymbol{R}$ is

$$\eta = \frac{1}{2}\sqrt{1 + R_{11} + R_{22} + R_{33}} \tag{5}$$

$$\boldsymbol{\epsilon} = \frac{1}{4}\begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix} \tag{6}$$

The relations between the time derivative of the Euler parameters and the body angular velocity $\boldsymbol{\omega}$ is established by the so-called propagation rule:

$$\dot{\eta} = -\frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\omega} \tag{7}$$

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2}(\eta\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}))\boldsymbol{\omega} \tag{8}$$

With reference to the problem of describing mutual orientation between two frames, the quantity

$$\boldsymbol{Q}_{21} = \{\eta_{21}, \boldsymbol{\epsilon}_{21}\}, \tag{9}$$

denotes the quaternion that can be extracted directly from ${}^1\boldsymbol{R}_2$.

The composition in terms of quaternions corresponding to $\boldsymbol{R}_2 = \boldsymbol{R}_1{}^1\boldsymbol{R}_2$ is defined by the operator “$*$” as

$$\boldsymbol{Q}_2 = \boldsymbol{Q}_1 * \boldsymbol{Q}_{21}, \tag{10}$$

with

$$\eta_2 = \eta_1\eta_{21} - \boldsymbol{\epsilon}_1^T\boldsymbol{\epsilon}_{21}, \tag{11}$$

$$\boldsymbol{\epsilon}_2 = \eta_1\boldsymbol{\epsilon}_{21} + \eta_{21}\boldsymbol{\epsilon}_1 + \boldsymbol{S}(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_{21}, \tag{12}$$

Note that in Eq. (10) $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$ are the quaternions that can be extracted from $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$, respectively, with $\boldsymbol{Q}_{21}$ as in Eq. (9).

Here, the target pose based on Σ_{CR} is defined by unit quaternion, as ${}^{CR}\boldsymbol{Q}_M = \{{}^{CR}\eta_M, {}^{CR}\boldsymbol{\epsilon}_M\}$. Since ${}^{CR}\eta_M$ can be determined by only ${}^{CR}\boldsymbol{\epsilon}_M$ (Eq. (3)), we use only three parameters ${}^{CR}\boldsymbol{\epsilon}_M$ to express the target's orientation. So the position/orientation of the target can be expressed by a six-parameter representation

$${}^{CR}\boldsymbol{\psi}_M = \begin{bmatrix} {}^{CR}\boldsymbol{r}_{CR,M} \\ {}^{CR}\boldsymbol{\epsilon}_M \end{bmatrix}, \tag{13}$$

where ${}^{CR}\boldsymbol{r}_{CR,M} = [t_x, t_y, t_z]^T$, ${}^{CR}\boldsymbol{\epsilon}_M = [\epsilon_1, \epsilon_2, \epsilon_3]^T$.

The target's position/orientation volecity is defined as

$${}^{CR}\dot{\boldsymbol{\psi}}_M = \begin{bmatrix} {}^{CR}\dot{\boldsymbol{r}}_{CR,M} \\ {}^{CR}\dot{\boldsymbol{\epsilon}}_M \end{bmatrix}. \tag{14}$$

3. MOTION-FEEDFORWARD COMPENSATION

First, we establish relations among relative velocities of three moving frames, world coordinate system Σ_W, target coordinate system Σ_M and camera coordinate systems as Σ_{CR}, shown in Fig. 1. Take Σ_W as the reference frame. Denote the vector from O_W (the origin of Σ_W) to O_{CR} expressed in Σ_W as ${}^W\boldsymbol{r}_{CR}$, the vector from O_W to O_M expressed in Σ_W as ${}^W\boldsymbol{r}_M$, and the vector from Σ_{CR} to Σ_M expressed in Σ_{CR} as ${}^{CR}\boldsymbol{r}_{CR,M}$. We define the robot's end-effector coordinate system as Σ_H, which is considered the same as Σ_{CR} since the camera is mounted on the robot's end-effector, so the rotation matrix ${}^W\boldsymbol{R}_{CR}$ is a function of the joint vector $\boldsymbol{q}$. The following relations hold:

$$^{CR}\boldsymbol{r}_{CR,M} = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{r}_M - {}^W\,\boldsymbol{r}_{CR}(\boldsymbol{q})). \quad (15)$$

Differentiating Eq. (15) with respect to time

$$^{CR}\dot{\boldsymbol{r}}_{CR,M} = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\dot{\boldsymbol{r}}_M - {}^W\dot{\boldsymbol{r}}_{CR}) + \boldsymbol{S}({}^{CR}\boldsymbol{\omega}_W)$$
$$^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{r}_M - {}^W\,\boldsymbol{r}_{CR}(\boldsymbol{q})). \quad (16)$$

where $\boldsymbol{S}(\cdot)$ is the operator performing the cross product between two (3×1) vectors. Given $\boldsymbol{\omega} = [\omega_x, \omega_y, \omega_z]^T$, $\boldsymbol{S}(\boldsymbol{\omega})$ takes on the form

$$\boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \quad (17)$$

Similarly, the angular velocities of Σ_{CR} and Σ_M with respect to Σ_W, are ${}^W\boldsymbol{\omega}_{CR}$ and ${}^W\boldsymbol{\omega}_M$, and the angular velocity of Σ_M with respect to Σ_{CR} is ${}^{CR}\boldsymbol{\omega}_{CR,M}$. Then the following relations hold:

$$^{CR}\boldsymbol{\omega}_{CR,M} = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{\omega}_M - {}^W\,\boldsymbol{\omega}_{CR}). \quad (18)$$

The relation between the time derivative of ${}^{CR}\boldsymbol{\epsilon}_M$ and the body angular velocity ${}^{CR}\boldsymbol{\omega}_{CR,M}$ is given by

$$^{CR}\dot{\boldsymbol{\epsilon}}_M = \frac{1}{2}({}^{CR}\eta_M \boldsymbol{I} - \boldsymbol{S}({}^{CR}\boldsymbol{\epsilon}_M)){}^{CR}\boldsymbol{\omega}_{CR,M}(19)$$

${}^{CR}\boldsymbol{\omega}_{CR,M}$ is given by Eq. (18).

Moreover, the camera velocity, which is considered as the end-effector velocity, can be expressed using the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}) = [\boldsymbol{J}_p{}^T(\boldsymbol{q}), \boldsymbol{J}_o{}^T(\boldsymbol{q})]^T$,

$$^W\dot{\boldsymbol{r}}_{CR} = \boldsymbol{J}_p(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad (20)$$

$$^W\boldsymbol{\omega}_{CR} = \boldsymbol{J}_o(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad (21)$$

$$\boldsymbol{S}({}^{CR}\boldsymbol{\omega}_W) = -{}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{J}_o(\boldsymbol{q})\dot{\boldsymbol{q}}){}^W\boldsymbol{R}_{CR}(\boldsymbol{q}). \quad (22)$$

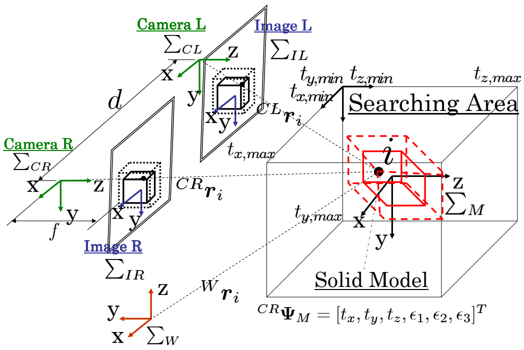


Fig. 2 Coordinate systems

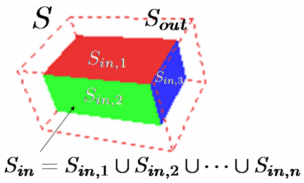


Fig. 3 Solid model searching for a block

Substituting Eq. (20), Eq. (21), Eq. (22) to Eq. (16), Eq. (19), the target velocity ${}^{CR}\dot{\boldsymbol{\psi}}_{CR,M}$ can be described by a mathematical formulation using $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$, that is, $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = -\boldsymbol{S}(\boldsymbol{b})\boldsymbol{a}$:

$$^{CR}\dot{\boldsymbol{\psi}}_{CR,M} = \begin{bmatrix} {}^{CR}\dot{\boldsymbol{r}}_{CR,M} \\ {}^{CR}\dot{\boldsymbol{\epsilon}}_M \end{bmatrix}$$

$$= \begin{bmatrix} -{}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{J}_p(\boldsymbol{q}) + {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) \\ \boldsymbol{S}({}^W\boldsymbol{R}_{CR}(\boldsymbol{q}){}^{CR}\boldsymbol{r}_{CR,M})\boldsymbol{J}_o(\boldsymbol{q}) \\ -\frac{1}{2}({}^{CR}\eta_M \boldsymbol{I} - \boldsymbol{S}({}^{CR}\boldsymbol{\epsilon}_M)){}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{J}_o(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}}$$

$$+ \begin{bmatrix} {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) & 0 \\ 0 & {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} {}^W\dot{\boldsymbol{r}}_M \\ {}^W\dot{\boldsymbol{\epsilon}}_M \end{bmatrix}$$

$$= \boldsymbol{J}_m(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{J}_n(\boldsymbol{q}){}^W\dot{\boldsymbol{\psi}}_M. \quad (23)$$

The relationship $\boldsymbol{J}_n(\boldsymbol{q})$ given by Eq. (23) describes how target pose change in Σ_{CR} with respect to the pose changing of itself in real word. The relationship $\boldsymbol{J}_m(\boldsymbol{q})$ given by Eq. (23) describes how target pose change in Σ_{CR} with respect to changing manipulator pose which influences the recognition from the relative motion of the camera to the object.

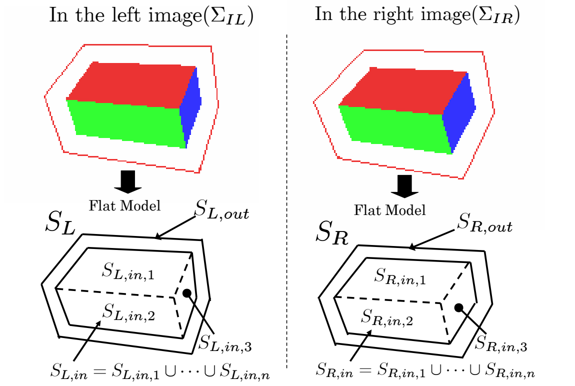
In this paper, we do not deal with the prediction of the target's motion in the real world, and we take account of the prediction of the target velocity in Σ_{CR} based on the joint velocity of manipulator $\dot{\boldsymbol{q}}$, so we can rewrite Eq. (23) as

$$^{CR}\dot{\boldsymbol{\psi}}_{CR,M} = \boldsymbol{J}_m(\boldsymbol{q})\dot{\boldsymbol{q}}. \quad (24)$$

Then the 3D pose of the target in time $t + \Delta t$ can be predicted from the current end-effector motion, presented by

$$^{CR}\hat{\boldsymbol{\psi}}_M(t + \Delta t) = {}^{CR}\,\boldsymbol{\psi}_M(t) + {}^{CR}\dot{\boldsymbol{\psi}}_M \Delta t. \quad (25)$$

${}^{CR}\dot{\boldsymbol{\psi}}_M \Delta t$ is the changing extent from the current pose to the next. We consider that the recognition ability will be improved by using Eq. (25) to predict the future pose of the target based on the relative motion from the camera to the object. And the recognition will be robust to the motion of manipulator itself.


In the left image(Σ_{IL})
In the right image(Σ_{IR})
Flat Model
S_L
$S_{L,out}$
$S_{L,in,1}$
$S_{L,in,2}$
$S_{L,in,3}$
$S_{L,in} = S_{L,in,1} \cup \cdots \cup S_{L,in,n}$
Flat Model
S_R
$S_{R,out}$
$S_{R,in,1}$
$S_{R,in,2}$
$S_{R,in,3}$
$S_{R,in} = S_{R,in,1} \cup \cdots \cup S_{R,in,n}$

(a) Left searching model (b)Right searching model

Fig. 4 Searching model

4. 3D MEASUREMENT METHOD

4.1 Kinematics of Stereo-Vision

We utilize perspective projection as projection transformation. The coordinate systems of left and right cameras in Fig. 2 are Σ_{CR} and Σ_{CL}, and image coordinate systems Σ_{IR} and Σ_{IL}. A point i on the target can be described using these coordinates and homogeneous transformation matrices. At first, a homogeneous transformation matrix from Σ_{CR} to Σ_M is defined as ${}^{CR}\boldsymbol{T}_M$. And an arbitrary point i on the target object in Σ_{CR} and Σ_M is defined ${}^{CR}\boldsymbol{r}_i$ and ${}^{M}\boldsymbol{r}_i$. Then ${}^{CR}\boldsymbol{r}_i$ is,

$$ {}^{CR}\boldsymbol{r}_i = {}^{CR}\boldsymbol{T}_M \ {}^{M}\boldsymbol{r}_i. \tag{26}$$

Where ${}^{M}\boldsymbol{r}_i$ is predetermined fixed vectors. Using a homogeneous transformation matrix from Σ_W to Σ_{CR}, i.e., ${}^{W}\boldsymbol{T}_{CR}$, then ${}^{W}\boldsymbol{r}_i$ is got as,

$$ {}^{W}\boldsymbol{r}_i = {}^{W}\boldsymbol{T}_{CR} \ {}^{CR}\boldsymbol{r}_i. \tag{27}$$

The position vector of i point in right image coordinates, ${}^{IR}\boldsymbol{r}_i$ is described by using projection matrix $\boldsymbol{P}$ of camera as,

$$ {}^{IR}\boldsymbol{r}_i = \boldsymbol{P} \ {}^{CR}\boldsymbol{r}_i. \tag{28}$$

By the same way as above, using a homogeneous transformation matrix of fixed values defining the kinematical relation from Σ_{CL} to Σ_{CR}, ${}^{CL}\boldsymbol{T}_{CR}$, ${}^{CL}\boldsymbol{r}_i$ is,

$$ {}^{CL}\boldsymbol{r}_i = {}^{CL}\boldsymbol{T}_{CR} \ {}^{CR}\boldsymbol{r}_i. \tag{29}$$

As we have obtained ${}^{IR}\boldsymbol{r}_i$, ${}^{IL}\boldsymbol{r}_i$ is described by the following Eq. (30) through projection matrix $\boldsymbol{P}$.

$$ {}^{IL}\boldsymbol{r}_i = \boldsymbol{P} \ {}^{CL}\boldsymbol{r}_i \tag{30}$$

Then position vectors projected in the Σ_{IR} and Σ_{IL} of arbitrary point i on target object can be described ${}^{IR}\boldsymbol{r}_i$ and ${}^{IL}\boldsymbol{r}_i$. Here, position and orientation of Σ_M based on Σ_{CR} has been defined as ${}^{CR}\boldsymbol{\psi}_M$. Then Eq. (28), Eq. (30) are rewritten as,

$$ \left\{ \begin{array}{l} {}^{IR}\boldsymbol{r}_i = f_R({}^{CR}\boldsymbol{\psi}_M, \ {}^{M}\boldsymbol{r}_i) \\ {}^{IL}\boldsymbol{r}_i = f_L({}^{CR}\boldsymbol{\psi}_M, \ {}^{M}\boldsymbol{r}_i). \end{array} \right. \tag{31}$$

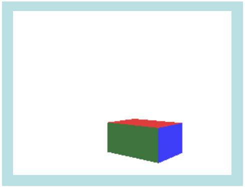


Fig. 5 Target object

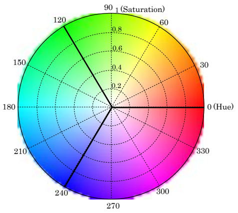


Fig. 6 Hue Circle

This relation connects the arbitrary points on the object and projected points on the left and right images with the variables ${}^{CR}\boldsymbol{\psi}_M$. This measurement problem of ${}^{CR}\boldsymbol{\psi}_M(t)$ in real time will be solved by consistent convergence of a matching model to the target object by a "1-Step GA".

The 3-D model for the target object of a rectangular block is shown in Fig. 3. The set of coordinates inside of the block is depicted as S_{in}, which is composed of each surfaces $S_{in,k}(k = 1, 2, \cdots, n)$, the outside space enveloping S_{in} is denoted as S_{out}, and the combination is named as S. Then, the set of the points of solid searching model S consisted of S_{in} and S_{out}, which are projected onto the two dimensional coordinates of left camera are expressed as,

$$ S_{L,in}({}^{CR}\boldsymbol{\psi}_M) = \sum_{k=1}^{m} S_{L,in,k} = \sum_{k=1}^{m} \{ {}^{IL}\boldsymbol{r}_i \in \Re^2 \mid {}^{IL}\boldsymbol{r}_i = f_L({}^{CR}\boldsymbol{\psi}_M, {}^{M}\boldsymbol{r}_i), {}^{M}\boldsymbol{r}_i \in S_{in,k} \in \Re^3 \} \tag{32}$$

$$ S_{L,out}({}^{CR}\boldsymbol{\psi}_M) = \{ {}^{IL}\boldsymbol{r}_i \in \Re^2 \mid {}^{IL}\boldsymbol{r}_i = f_L({}^{CR}\boldsymbol{\psi}_M, {}^{M}\boldsymbol{r}_i), {}^{M}\boldsymbol{r}_i \in S_{out} \in \Re^3 \} \tag{33}$$

where $m<n$ denotes the number of the visible surfaces. The left searching model projected to left camera coordinates is shown in Fig. 4(a). The area composed of $S_{L,in}$ and $S_{L,out}$ is named as S_L. The above defines only the left-image searching model, the right one is defined in the same way and the projected searching model is shown in Fig. 4(b).

4.2 Model Definition

Here, we define evaluation function to estimate how much the moving solid model S defined by ${}^{CR}\boldsymbol{\psi}_M$ lies on the target being imaged on the left and right cameras. The input images will be directly matched by the projected moving models S_L and S_R, which are located by only ${}^{CR}\boldsymbol{\psi}_M$ as described in Eq. (33) that includes the kinematical relations of the left and right camera coordinates. Therefore, if the camera parameters and kinematical relations are completely accurate, and the solid searching model describes precisely the target object shape, then the $S_{L,in}$ and $S_{R,in}$ will be completely lies on the target reflected on the left and right images, provided that true value of ${}^{CR}\boldsymbol{\psi}_M$ is given.

To search for the target object in the images, the surface-strips model shown in Fig. 4 and its color information are used. It is easy to understand that the

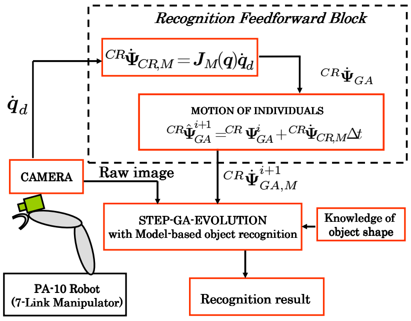

Recognition Feedforward Block
$^{CR}\dot{\Psi}_{CR,M} = J_M(q)\dot{q}_d$
$^{CR}\dot{\Psi}_{GA}$
$\dot{q}_d$
MOTION OF INDIVIDUALS
$^{CR}\hat{\Psi}^{i+1}_{GA} = ^{CR}\Psi^{i}_{GA} + ^{CR}\dot{\Psi}_{CR,M}\Delta t$
CAMERA
Raw image
$^{CR}\dot{\Psi}^{i+1}_{GA,M}$
STEP-GA-EVOLUTION
with Model-based object recognition
Knowledge of object shape
PA-10 Robot (7-Link Manipulator)
Recognition result

Fig. 7 Feedforward recognition system

color can be limited only by hue value of the HSV color system as shown in Fig. 6. Let therefore $b_k, (k = 1, 2, \cdots, n)$ denote the hue value of the color in $S_{in,k}$ surface of the target object. An example of the target object is shown in Fig. 5 which is a rectangular solid block($100mm \times 150mm \times 200mm$) with symmetrical colored surfaces. So here S_1 is red then we have $b_1 = 0$, in the same way, S_2 is green then $b_2 = 120$, S_3 is blue then $b_3 = 240$, the other surfaces are symmetrical so $b_4 = 0, b_5 = 120, b_6 = 240$.

Let $h(^{IL}\boldsymbol{r}_i)$ (or $h(^{IR}\boldsymbol{r}_i)$) denote the hue value at the image position $^{IL}\boldsymbol{r}_i$(or $^{IR}\boldsymbol{r}_i$). Then the evaluation function of the left moving surface-strips model is given as,

$$F_L(^{CR}\boldsymbol{\psi}_M) = \frac{1}{H}\sum_{k=1}^{m}\{ \sum_{^{IL}\boldsymbol{r}_i \in S_{L,in,k}(^{CR}\boldsymbol{\psi}_M)} \delta(h(^{IL}\boldsymbol{r}_i) - b_k) - \sum_{^{IL}\boldsymbol{r}_i \in S_{L,out}(^{CR}\boldsymbol{\psi}_M)} \delta(h(^{IL}\boldsymbol{r}_i) - b_k)\}$$

where δ is the Kronecker delta function defined as

$$\delta(n) = \begin{cases} 1, & (n = 0) \\ 0, & (n \neq 0). \end{cases} \tag{34}$$

Here $H = \sum_{k=1}^{m} n_k$, n_k represents the number of the searching points in $S_{L,in,k}$. It is a scaling factor that normalized $F_L(^{CR}\boldsymbol{\psi}_M) \leq 1$. In the case of $F_L(^{CR}\boldsymbol{\psi}_M) < 0$, $F_L(^{CR}\boldsymbol{\psi}_M)$ is given to zero. The first part of this function expresses how much each color area of $S_{L,in}$ defined by $^{CR}\boldsymbol{\psi}_M$ lies on the target being imaged on the left and right cameras. And the second part is the matching degree of its contour-strips. The difference between the internal surface and the contour-strips of the surface-strips model can make the estimation more sensible, especially in distance recognition between the target to the cameras which determine the size of the flat models. The right one is defined in the same way. Then the whole evaluation function is given as

$$F(^{CR}\boldsymbol{\psi}_M) = (F_L(^{CR}\boldsymbol{\psi}_M) + F_R(^{CR}\boldsymbol{\psi}_M))/2. \tag{35}$$

Equation (35) is used as a fitness function in GA process. When the moving searching model fits to the target object being imaged in the right and left images, then the fitness function $F(^{CR}\boldsymbol{\psi}_M)$ gives maximum value.

Therefore the problem of finding a target object and detecting its 3D pose can be converted to searching

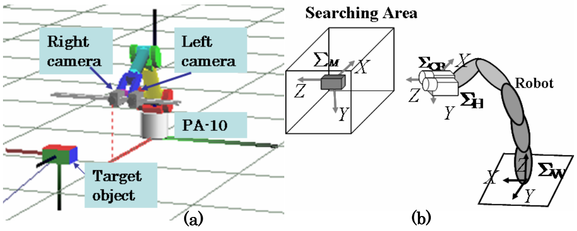


Fig. 8 (a)Simulation experiment system created by OpenGL. (b) coordinate systems of simulation experiment

$^{CR}\boldsymbol{\psi}_M$ that maximizes $F(^{CR}\boldsymbol{\psi}_M)$. We solve this optimization problem by GA whose gene representing possible pose solution $^{CR}\boldsymbol{\psi}_{GA}$.

Using Eq. (25), the pose of the individuals $^{CR}\boldsymbol{\psi}_{GA}$ in the next generation can be predicted based on the current pose, presented by

$$^{CR}\hat{\boldsymbol{\psi}}^{i+1}_{GA} = ^{CR}\boldsymbol{\psi}^{i}_{GA} + ^{CR}\dot{\boldsymbol{\psi}}_{CR,M}\Delta t. \tag{36}$$

The recognition system of the proposed method is shown in Fig. 7. We consider that the recognition ability will be improved by Eq. (equ18) to move all the individuals to compensate the influence of the motion of the camera. So the recognition will be robust to the motion of robot itself.

5. SIMULATION EXPERIMENT OF RECOGNITION

To verify the effectiveness of the proposed motion-feedforward recognition method, we have conducted the simulation experiment to recognize a rectangular solid block with colored surfaces introduced in section 3.

The simulation experiment is performed under a software "Open GL". Here, we create a manipulator which is the same as an actual 7-link manipulator namely "PA-10" robot, shown in Fig. 8. Two cameras are mounted on the robot end-effector.

5.1 Simulation experiment

In this simulation experiment, two kinds of motion are given to the robot end-effector while recognizing a moving target object's 3D pose (6DOF), as shown in Fig. 9. The motion of the target object given by Eq. (2), (13) is

$$\theta = 15sin(\omega_o t), \qquad \boldsymbol{k} = [0, 0, 1]^T, \tag{37}$$

$$^{W}\boldsymbol{\epsilon}_M = sin(\theta/2)\boldsymbol{k}, \tag{38}$$

and the other parameters of the target object is unchanged. Here ω_o represents the frequency of target object's motion. In this experiment, we fixed $\omega_o = 0.125[rad/s]$. Let ω represents the frequency of end-effector's motion. If the robot is static, then $\omega = 0[rad/s]$. Fig. 10 shows the recognition result of the rotating target object when $\omega = 0$. Fig. 10(a) shows the recognition result of position x, y, z compared with the desired position (in Σ_{CR}), where the desired position is

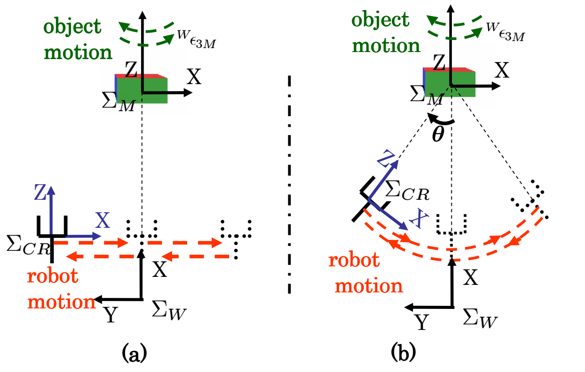
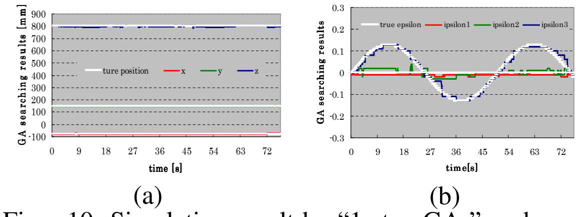


Fig. 9 (a) motion "A": given position changing of end-effector. (b) motion "B": given orientation changing of end-effector



(a) (b)

Fig. 10 Simulation result by "1-step GA " under $\omega = 0[rad/s]$ (no motion). (a) Recognition result of position x, y, z compared with the desired position (in Σ_{CR}). (b) Recognition result of orientation $\epsilon_1, \epsilon_2, \epsilon_3$ compared with the desired position (in Σ_{CR}).

depict in white lines. Fig. 10(b) shows recognition result of orientation $\epsilon_1, \epsilon_2, \epsilon_3$ compared with the desired orientation(in Σ_{CR}). In the same way, the desired orientation is depict in white lines. It can be found that only "1-step GA " is enough to track it well since the motion of the target object is slow and not complicated.

To evaluate the effectiveness of the proposed motion-feedforward recognition method, we compare the recognition results using "1-step GA " with that using "1-step GA + FF. " under two robot's motions respectively as follows ("FF. "represents the motion-feedforward recognition method).

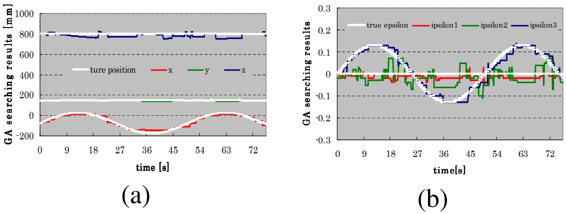
(1) Recognition under motion "A": given position changing of end-effector (shown in Fig. 9(a)).

In this case, the shuttle motion in y axis of Σ_W is given to the robot end-effector with frequency $\omega = 0.125[rad/s]$. The initial pose of the end-effector is shown in Fig. 8(b) defined as ${}^{W}\boldsymbol{\psi}_{e0} = [x_{e0}, y_{e0}, z_{e0}, {}^{W}\boldsymbol{Q}_{e0}]^T$, where ${}^{W}\boldsymbol{Q}_{e0} = [\epsilon_{1e0}, \epsilon_{2e0}, \epsilon_{3e0}]$. The desired motion track is given as

$${}^{W}y_d = y_{e0} + 0.1sin(\omega t), \tag{39}$$

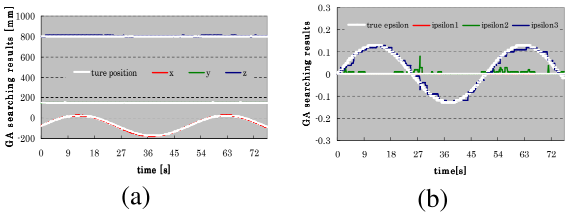
and the other parameters keep their initial values. The motion of the end-effector starts with known initial target's pose. It means the searching model possesses true values at $t = 0$.

When the end-effector is moving, the motion of the target object in the image includes both the real motion of the target and the relative motion with respect to the camera's. So the motion become complicated and the



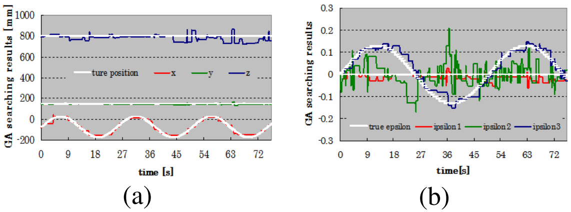
(a) (b)

Fig. 11 Simulation result under motion "A" by "1-step GA " under $\omega = 0.125[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



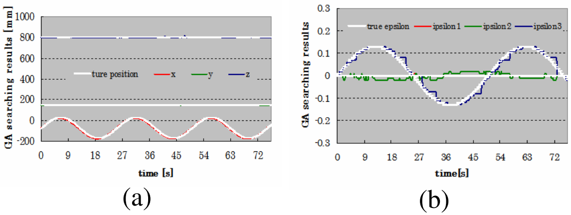
(a) (b)

Fig. 12 Simulation result under motion "A" by "1-step GA + FF. " under $\omega = 0.125[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



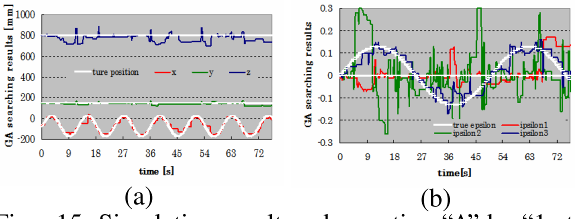
(a) (b)

Fig. 13 Simulation result under motion "A" by "1-step GA " under $\omega = 0.25[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



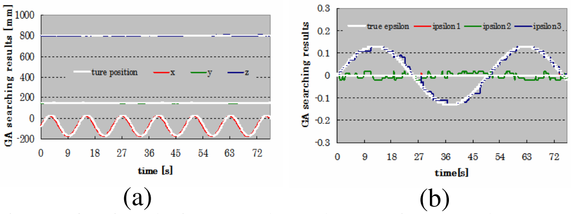
(a) (b)

Fig. 14 Simulation result under motion "A" by "1-step GA + FF. " under $\omega = 0.25[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



(a) (b)

Fig. 15 Simulation result under motion "A" by "1-step GA " under $\omega = 0.5[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

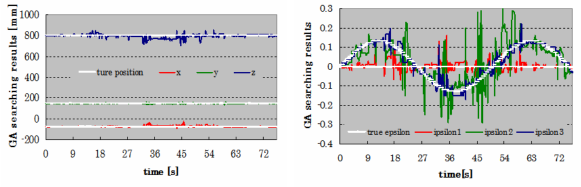


(a) (b)

Fig. 16 Simulation result under motion "A" by "1-step GA + FF. " under $\omega = 0.5[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

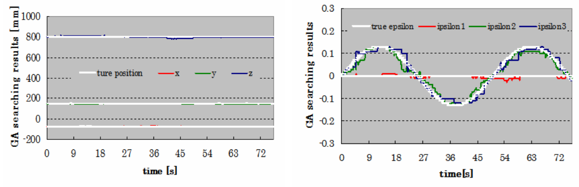
$\bar{F}, \tilde{\Delta\phi}$ \ ω		$\omega = 0$	$\omega = 0.125$	$\omega = 0.25$	$\omega = 0.5$
1-step GA	$\bar{F}$	0.9525	0.7047	0.5794	0.4378
	$\tilde{\Delta x}$ [mm]	1.238529	15.43175	16.53167	30.21539
	$\tilde{\Delta y}$ [mm]	3.963243	18.58162	26.21253	32.96989
	$\tilde{\Delta z}$ [mm]	5.544477	25.84152	36.72746	44.76831
	$\tilde{\Delta\epsilon_1}$	0.011146	0.018499	0.021081	0.059064
	$\tilde{\Delta\epsilon_2}$	0.015599	0.041138	0.054877	0.123591
	$\tilde{\Delta\epsilon_3}$	0.014914	0.019545	0.023279	0.030351
FF. + 1-step GA	$\bar{F}$		0.9473	0.9556	0.9590
	$\tilde{\Delta x}$ [mm]		2.128153	1.316153	1.923763
	$\tilde{\Delta y}$ [mm]		6.43577	2.329286	3.173726
	$\tilde{\Delta z}$ [mm]		6.873411	3.128897	4.313114
	$\tilde{\Delta\epsilon_1}$		0.000485	0.001283	0.00084
	$\tilde{\Delta\epsilon_2}$		0.00952	0.013151	0.013249
	$\tilde{\Delta\epsilon_3}$		0.014976	0.015944	0.013401

Fig. 17 Conclusion of simulation under motion "A"



(a) (b)

Fig. 18 Simulation result under motion "B" with "1-step GA " under $\omega = 0.125[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

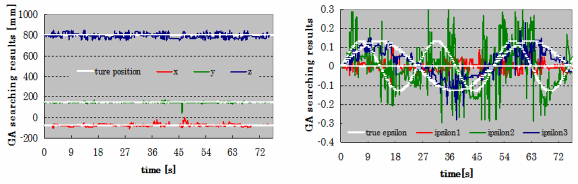


(a) (b)

Fig. 19 Simulation result under motion "B" with "1-step GA + FF. " under $\omega = 0.125[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

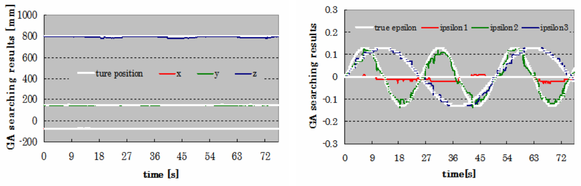
target object will be difficult to recognize. Fig. 11 shows the simulation result only by "1-step GA " under position shuttle motion $\omega = 0.125[rad/s]$. Without the feedforward compensation, the "1-Step GA" can not track all the six variables precisely, even though the correct initial values are given. Compared with Fig. 11, Fig. 12 shows the simulation results by "1-step GA + FF. " under the same motion of the robot end-effector. It shows that the simulation result in Fig. 12 always overlap the real 3D pose which verifies that the motion-feedforward method works well.

The same simulation is conducted under $\omega = 0.25[rad/s]$. The recognition result without using feedforward recognition method is shown in Fig. 13, tracking of the target for GA became more difficult when the speed of the end-effector gets quicker, which caused GA's convergence speed is not faster than the target speed relative to the camera. However, using feedforward recognition method, the data shown in Fig. 14 indicates the models kept matching the target well. When the velocity of the end-effector gets more quicker, $\omega = 0.5[rad/s]$, GA lost the target soon, as shown in Fig. 15. On the other hand, it can be found in Fig. 16 that the recognition result always overlap the real 3D pose even under such a high-speed moving of the robot manipulator.



(a) (b)

Fig. 20 Simulation result under motion "B" with "1-step GA " under $\omega = 0.25[[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



(a) (b)

Fig. 21 Simulation result under motion "B" with "1-step GA + FF. " under $\omega = 0.25[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

Here, we use the mean value of the fitness function $\bar{F}$ and the root-mean-square value of the error of 3D pose $\tilde{\Delta\psi}$ from 0s to 72s to evaluate the recognition ability. $\bar{F}$ is given by

$$\bar{F} = \frac{1}{n}(F(\psi_{t_1}) + F(\psi_{t_2}) + \cdots + F(\psi_{t_n})). \quad (40)$$

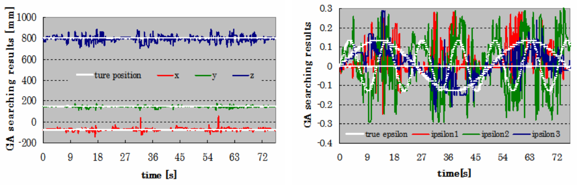
Let $\Delta\psi$ describes the error of 3D pose, which is defined as the differece between the desired value of the recognized value, $\Delta\psi = [\Delta x, \Delta y, \Delta z, \Delta\epsilon_1, \Delta\epsilon_2, \Delta\epsilon_3]^T$. Then the root-mean-square value of $\Delta\psi$ is given by

$$\begin{aligned}\tilde{\Delta\psi} &= [\tilde{\Delta x}, \tilde{\Delta y}, \tilde{\Delta z}, \tilde{\Delta\epsilon_1}, \tilde{\Delta\epsilon_2}, \tilde{\Delta\epsilon_3}]^T \\ &= \sqrt{\frac{1}{n}({\Delta\psi_{t_1}}^2 + {\Delta\psi_{t_2}}^2 + \cdots + {\Delta\psi_{t_n}}^2)}.\end{aligned} \quad (41)$$

It is obvious that high value of $\bar{F}$ and small value of $\tilde{\Delta\psi}$ represent good recognition. Here, we use milimeters to measure position. When using quaternion to express the orientation of an object, no unit, just values. Suppose the object rotates 1[deg] around x axis, we can calculate $\epsilon_1 = 0.008, \epsilon_2 = 0, \epsilon_3 = 0$ based on the quaternion definition, corresponding to the same orientation. Like this, we can estimate approximately that around which axis, how much [deg] the error of orientation is.

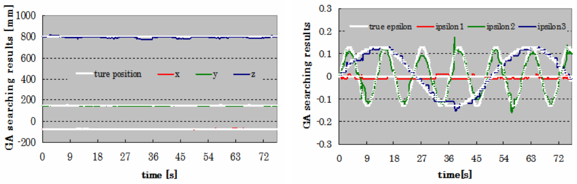
Fig. 17 shows $\bar{F}$ and $\tilde{\Delta\psi}$ of each situation we have discussed above. We can see that using only "1-step GA ", $\bar{F}$ gets lower and $\tilde{\Delta\psi}$ gets bigger (to about 35[mm],12[deg]) along with ω changing from 0 to 0.5[rad/s]. By "1-step GA + FF.", the end-effector's motion has been compensated completely, even the end-effector moves faster and faster, both $\bar{F}$ and $\tilde{\Delta\psi}$ are not changed much (about 3[mm],1[deg]). It is confirmed that using motion-feedforward recognition method, the recognition in a hand-eye system is the same with that in a fixed-camera system.

(2) Recognition under motion "B": given orientation changing of end-effector (shown in Fig. 9(b)). Here, the orientation changing of end-effector is defined as the mo-



(a) (b)

Fig. 22 Simulation result under motion “B” with “1-step GA ” under $\omega = 0.5[rad/s]$. (a), (b) is the same meaning as that in Fig. 10



(a) (b)

Fig. 23 Simulation result under motion “B” with “1-step GA + FF. ” under $\omega = 0.5[rad/s]$. (a), (b) is the same meaning as that in Fig. 10

$\bar{F}, \tilde{\Delta\phi}$ \ ω		$\omega=0$	$\omega=0.125$	$\omega=0.25$	$\omega=0.5$
1-step GA	$\bar{F}$	**0.9525**	**0.7445**	**0.5089**	**0.3987**
	$\tilde{\Delta x}$ [mm]	1.238529	7.084036	12.50464	15.94141
	$\tilde{\Delta y}$ [mm]	3.963243	15.10137	19.43937	24.79264
	$\tilde{\Delta z}$ [mm]	5.544477	21.11374	26.54229	33.70501
	$\tilde{\Delta\epsilon_1}$	0.011146	0.025746	0.030625	0.089354
	$\tilde{\Delta\epsilon_2}$	0.015599	0.073936	0.106551	0.118325
	$\tilde{\Delta\epsilon_3}$	0.014914	0.026867	0.042807	0.062964
FF. + 1-step GA	$\bar{F}$		**0.9556**	**0.9682**	**0.9443**
	$\tilde{\Delta x}$ [mm]		1.212835	2.475826	2.069171
	$\tilde{\Delta y}$ [mm]		5.841326	3.708801	5.811863
	$\tilde{\Delta z}$ [mm]		3.082948	4.88349	4.00924
	$\tilde{\Delta\epsilon_1}$		0.006454	0.012079	0.008623
	$\tilde{\Delta\epsilon_2}$		0.016637	0.02126	0.024202
	$\tilde{\Delta\epsilon_3}$		0.017566	0.012186	0.013883

Fig. 24 Conclusion of simulation under motion “B”

tion in a circle with a fixed distance to the target and keeping the eye-line (z axis of Σ_{CR}) passes the center of the target. The shuttle motion looking the target from the left side to the right side is given to the end-effector under $\omega = 0.125[rad/s]$. The desired motion track is given as

$$\theta = 15sin(\omega t), \tag{42}$$

$$^W x_d = x_{e0} + d * (1 - cos\theta), \tag{43}$$

$$^W y_d = -(y_{e0} + d * sin\theta), \tag{44}$$

$$^W z_d = z_{e0}, \tag{45}$$

$$^{e0}\boldsymbol{Q}_d = \{^{e0}\eta_d, ^{e0}\boldsymbol{\epsilon}_d\} = \{cos\frac{\theta}{2}, 0, sin\frac{\theta}{2}, 0\} \tag{46}$$

$$^W\boldsymbol{Q}_d = {}^W\boldsymbol{Q}_{e0} * {}^{e0}\boldsymbol{Q}_d. \tag{47}$$

The change of $^W x_d, ^W y_d$ is very small, almost 0. The motion of the end-effector also starts with known initial target's pose.

In the same way as the previous experiment, Fig. 18 shows the recognition result under orientation shuttle motion of end-effector using only “1-step GA ” . Fig. 19 shows the simulation result using “1-step GA + FF.” under the same motion of the robot end-effector. Also, It can be found the motion-feedforward method gave good prediction of the target's 3D pose in Σ_{CR}.

The same simulation is also conducted under $\omega = 0.25[rad/s]$ (Fig. 20, 21) and $\omega = 0.5[rad/s]$ (Fig. 22, 23) respectively. Tracking of the target for GA became more difficult when the speed of the end-effector gets quicker. In the case of $\omega = 0.5[rad/s]$, the target is lost soon. However, using feedforward recognition method, in both cases the target is recognized well all the time. Also, Fig. 24 shows a conclusion of simulation under motion “B”, which verifies the proposed method leads to a robustly accurate recognition even under high-speed motion.

6. CONCLUSION

We have proposed a 3D pose measurement method which utilizes an evolutionary recognition technique of GA and a fitness evaluation based on a matching stereo model whose pose is expressed by unit quaternion. A proposed motion-feedforward compensation method is confirmed that can improve the dynamics of recognition. Simulation results have been verified the effectiveness of the proposed method to recognize the pose of a target object along with two kinds of motion of the end-effector.

As future research, we will apply this method to visual servoing task. We are looking forward to see the stability of visual servo system can be improved since the robot is able to distinguish what is the real motion of the target and what is a fiction motion just comes from the camera.

REFERENCES

[1] K.Hashimoto and H.Kimura, “Visual Servoing - Nonlinear Observer Approach”, Journal of the Robotics Society of Japan (in Japanese), Vol.13, No.7, pp986-993, 1995

[2] A. De Luca, G. Oriolo and P. R. Giordano, “On-line Estimation of Feature Depth for Image-Based Visual Servoing Schemes”, IEEE Int. Conf. on Robotics and Automation (ICRA2007).

[3] Y.Nakabo and M.Ishikawa, “Visual Servoing using 1ms High-Speed Vision”, Journal of the Society of Instrument and Control Engineers (in Japanese), Vol.40, No.9, pp636-640, 2001

[4] Y.Maeda and G.Xu “Smooth Matching of Feature and Recovery of Epipolar Equation by Tabu Search”, IEICE, Vol.J83-D-2, No.3, pp.440-448, 1999.

[5] L.Masson, M. Dhome and F.Jurie, “Robust Real Time Tracking of 3D Objects” Proc.17th IEEE Int.Conf.on Patten Recognition, 2004.

[6] J.T.-Y.Wen and K.Kreutz-Delgado, “The attitude control problem”, IEEE Trans. on Robotics and Automation, vol. 39, pp. 1148-1163 1991.

[7] S.Caccavale, C.Natale, B.Siciliano and L.Villani, “Six-DOF Impedance Control BAsed on Angle/Axis Representations”, IEEE Trans.on Robotics and Automation, vol. 15, no. 2, APRIL 1999.

[8] M.Minami, H.Suzuki, J.Agbanhan and T.Asakura, “Visual Servoing to Fish and Catching Using Global/Local GA Search”, 2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics Proc., pp.183-188, 2001.