Hand-eye Visual Servoing to 3D Pose Based on Quaternion with Feedforward Compensation

*Wei Song (Univ. of Fukui), Mamoru Minami (Univ. of Fukui)

Abstract— This paper deals with visual servoing to a 3D pose (6 degree of freedom) of a target object, whose pose is expressed by unit quaternion. We propose a motion-feedforward (MFF) method to improve visual recognition dynamics, which is worsened by being disturbed hand-eye motion during visual servoing of the robot manipulator. Simulation of visual servoing to a 3D pose will be performed to show the effectiveness of the proposed method.

Key Words: Pose measurement, Unit quaternion, GA, Motion-feedforward compensation

1. Introduction

In recent years, object recognition and visual tracking and servoing have been studied intensively in the field of robotics and in other research areas. Tasks in which visual information are used to direct a manipulator toward a target object are referred to visual servoing in [1]-[4]. 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 3-D pose (position and orientation) recognition of a target for visual servoing by a 7-link manipulator.

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 [5] 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 errors in the early stage need some time to decrease to nearly zero and may cause the visual servoing system unstable. The same method is also used by Luca [6] to estimate the distance z between the object to the camera. Also, correct prediction can not be achieved at the beginning of the estimation, the error of z is decreasing along with time passing. And the convergence of z is obtained by using the given motion of the camera since it uses single camera to recognize, 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 Ishikawa [7] 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.

Most visual servo systems use an hand-eye configuration, having the camera mounted on the robot's end-effector, so the dynamics of the manipulator will cause the recognition dynamics to deteriorate directly. It is common sense that the time-delay of recognition existing in feedback largely decreases the stability of whole control system. However, as we know, researches on improving the recognition dynamics have not been attracted by researchers. Thus it is important for the robot to improve the recognition dynamics by distinguishing the motion of the target object in real world from the motion detected through the camera by compensating the dynamic motion of the robot end-effector. In this paper, we proposed a motion-feedforward (MFF) method that 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 recognition, it seems that the recognition were performed by using just fixed cameras, so the recognition dynamics is separated from the dynamics of the manipulator. Thus the recognition can become easier and the recognition dynamics can 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.

We use model-based method to recognize 3-D pose in real-time. 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"1-Step GA" [8]. Unit quaternion is used

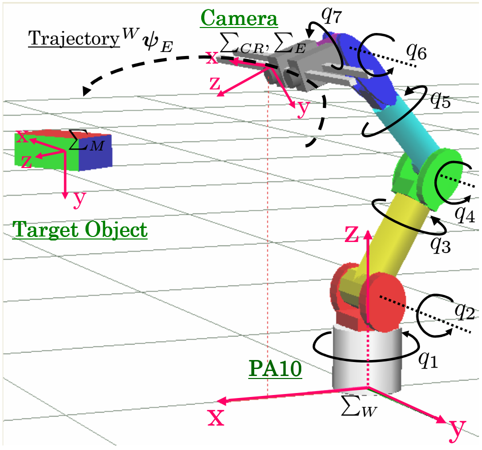


Fig.1 Visual servo system of PA-10

to represent the orientation of the target object, which has a advantage that can represent the orientation of a rigid body without singularities. singularities cause multi-solution for a given orientation that is difficult for GA to converge. An advantage of our method is that we use a 3-D solid model that possesses all 6 degree of freedom (DOF). Based on the 3D solid model, the robot can keep moving toward the target even it gets out of the images under high-speed action in visual servoing. On the other hand, visual servoing based on the 2-D image features has also been researched popularly. However, loosing visual features causes a big problem to control the robot stably.

2. Motion-Feedforward (MFF) Compensation

The motion of the target seeing from the camera will be affected by both the motion of the target in the real world and the motion of the camera in hand-eye system. Here we describe such a relationship by a mathematical function, which can distinguish these two motions.

The target coordinate system is represented as Σ_M (see 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 ${}^{CR}\boldsymbol{r}_M$. The following relations hold:

$$ {}^{CR}\boldsymbol{r}_M = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{r}_M - {}^W\boldsymbol{r}_{CR}(\boldsymbol{q})), \tag{1} $$

where ${}^{CR}\boldsymbol{R}_W$ is a rotation matrix determined by $\boldsymbol{q}$. Differentiating (1) with respect to time

$$ \begin{aligned} {}^{CR}\dot{\boldsymbol{r}}_M = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\dot{\boldsymbol{r}}_M - {}^W\dot{\boldsymbol{r}}_{CR}) + \boldsymbol{S}({}^{CR}\boldsymbol{\omega}_W) \\ {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{r}_M - {}^W\boldsymbol{r}_{CR}(\boldsymbol{q})). \end{aligned} \tag{2} $$

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

$$ \boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{3} $$

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

$$ {}^{CR}\boldsymbol{\omega}_M = {}^{CR}\boldsymbol{R}_W(\boldsymbol{q})({}^W\boldsymbol{\omega}_M - {}^W\boldsymbol{\omega}_{CR}). \tag{4} $$

In this paper, the target's orientation is also expressed by unit quaternion. The 3-D pose of the target is defined as ${}^{CR}\boldsymbol{\psi}_M = [{}^{CR}\boldsymbol{r}_M^T, {}^{CR}\boldsymbol{\epsilon}_M^T]^T$, where ${}^{CR}\boldsymbol{r}_M = [r_1, r_2, r_3]^T$, ${}^{CR}\boldsymbol{\epsilon}_M = [\epsilon_1, \epsilon_2, \epsilon_3]^T$.

The target's 3-D pose volecity is defined as

$$ {}^{CR}\dot{\boldsymbol{\psi}}_M = \begin{bmatrix} {}^{CR}\dot{\boldsymbol{r}}_M \\ {}^{CR}\dot{\boldsymbol{\epsilon}}_M \end{bmatrix}, \tag{5} $$

where the time derivation of target's position ${}^{CR}\dot{\boldsymbol{r}}_M$ is given by (2). The relation between the time derivative of ${}^{CR}\boldsymbol{\epsilon}_M$ and the body angular velocity ${}^{CR}\boldsymbol{\omega}_M$ is given by [10] and is rewrote as

$$ {}^{CR}\dot{\boldsymbol{\epsilon}}_M = \frac{1}{2}({}^{CR}\eta_M \boldsymbol{I} - \boldsymbol{S}({}^{CR}\boldsymbol{\epsilon}_M))^{CR}\boldsymbol{\omega}_M, \tag{6} $$

where ${}^{CR}\boldsymbol{\omega}_M$ is given by (4).

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

$$ {}^W\dot{\boldsymbol{r}}_{CR} = \boldsymbol{J}_P(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{7} $$

$$ {}^W\boldsymbol{\omega}_{CR} = \boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{8} $$

$$ \begin{aligned} \boldsymbol{S}({}^{CR}\boldsymbol{\omega}_W) &= -{}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{S}({}^W\boldsymbol{\omega}_{CR})^W\boldsymbol{R}_{CR}(\boldsymbol{q}) \\ &= -{}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{J}_O(\boldsymbol{q})\dot{\boldsymbol{q}})^W\boldsymbol{R}_{CR}(\boldsymbol{q}). \end{aligned} \tag{9} $$

Substituting (7), (8), (9) to (2), (6), the target velocity in ${}^{CR\ CR}\dot{\boldsymbol{\psi}}_M$ can be described by a mathematical formulation:

$$ \begin{aligned} {}^{CR}\dot{\boldsymbol{\psi}}_M &= \begin{bmatrix} {}^{CR}\dot{\boldsymbol{r}}_M \\ {}^{CR}\dot{\boldsymbol{\epsilon}}_M \end{bmatrix} \\ &= \begin{bmatrix} -{}^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{J}_P(\boldsymbol{q}) + {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) \\ \boldsymbol{S}({}^W\boldsymbol{R}_{CR}(\boldsymbol{q}){}^{CR}\boldsymbol{r}_M)\boldsymbol{J}_O(\boldsymbol{q}) \\ -\frac{1}{2}({}^{CR}\eta_M\boldsymbol{I} - \boldsymbol{S}({}^{CR}\boldsymbol{\epsilon}_M))^{CR}\boldsymbol{R}_W(\boldsymbol{q})\boldsymbol{J}_O(\boldsymbol{q}) \end{bmatrix} \dot{\boldsymbol{q}} \\ &+ \begin{bmatrix} {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) & 0 \\ 0 & {}^{CR}\boldsymbol{R}_W(\boldsymbol{q}) \end{bmatrix} \begin{bmatrix} {}^W\dot{\boldsymbol{r}}_M \\ {}^W\dot{\boldsymbol{\epsilon}}_M \end{bmatrix} \\ &= \boldsymbol{J}_M(\boldsymbol{q}, {}^{CR}\boldsymbol{r}_M, {}^{CR}\boldsymbol{\epsilon}_M)\dot{\boldsymbol{q}} + \boldsymbol{J}_N(\boldsymbol{q})^W\dot{\boldsymbol{\psi}}_M. \end{aligned} \tag{10} $$

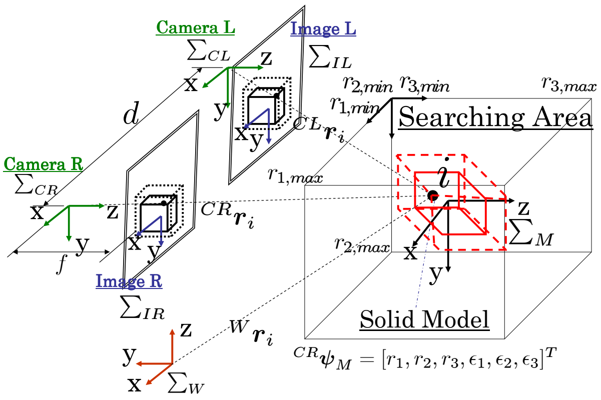


Fig.2 Coordinate systems

The matrix $\boldsymbol{J}_M$ in (10) describes how target pose change in Σ_{CR} with respect to changing manipulator pose in Σ_{CR}. The matrix $\boldsymbol{J}_N$ in (10) describes how target pose change in Σ_{CR} with respect to the pose changing of itself in real word.

In this paper, we do not deal with the prediction of the target's motion in the real world, we take account of the prediction of the target velocity in Σ_{CR} based on the joint velocity $\dot{\boldsymbol{q}}$ of the manipulator, so we can rewrite (10) as

$$ {}^{CR}\dot{\boldsymbol{\psi}}_M = \boldsymbol{J}_M(\boldsymbol{q}, {}^{CR}\boldsymbol{r}_M, {}^{CR}\boldsymbol{\epsilon}_M)\dot{\boldsymbol{q}}. \qquad (11)$$

Then the 3-D pose of the target at time $t + \Delta t$ can be predicted based on the motion of the end-effector motion at time t, presented by

$$ {}^{CR}\hat{\boldsymbol{\psi}}_M(t+\Delta t) = {}^{CR}\boldsymbol{\psi}_M(t) + {}^{CR}\dot{\boldsymbol{\psi}}_M \Delta t. \qquad (12)$$

(12) shows $\boldsymbol{J}_M$ is a function of $\boldsymbol{q}, {}^{CR}\boldsymbol{r}_M, {}^{CR}\boldsymbol{\epsilon}_M$. $\boldsymbol{q}$ can be considered known from the robot manipulator without errors, while ${}^{CR}\boldsymbol{r}_M(t), {}^{CR}\boldsymbol{\epsilon}_M(t)$ is the result of recognition at time t by using model-based matching in which errors exist probablely. Then the errors included in $\boldsymbol{J}_M$ from ${}^{CR}\boldsymbol{r}_M(t), {}^{CR}\boldsymbol{\epsilon}_M(t)$ will lead to incorrect prediction and cause the recognition errors at the next time $t + \Delta t$. It is seems as a difficulty in 3-D pose prediction since the errors may increase exponentially due to such a vicious circle. However, a proposed "1-Step GA" method will limit the increasing errors by correcting the recognition result based on the prediction at each time t, which is explained in detail in 3·3.

3. 3D Measurement Method

3·1 Kinematics of Stereo-Vision

We utilize perspective projection as projection transformation. Fig. 2 shows the coordinate system of our stereo vision system. The target object's coordinate system is represented by Σ_M and image coordinate systems of the left and right cameras are represented by Σ_{IL} and Σ_{IR}. 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,

$$ {}^{CR}\boldsymbol{r}_i = {}^{CR}\boldsymbol{T}_M \; {}^{M}\boldsymbol{r}_i. \qquad (13)$$

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,

$$ {}^{W}\boldsymbol{r}_i = {}^{W}\boldsymbol{T}_{CR} \; {}^{CR}\boldsymbol{r}_i. \qquad (14)$$

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,

$$ {}^{IR}\boldsymbol{r}_i = \boldsymbol{P} \; {}^{CR}\boldsymbol{r}_i. \qquad (15)$$

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,

$$ {}^{CL}\boldsymbol{r}_i = {}^{CL}\boldsymbol{T}_{CR} \; {}^{CR}\boldsymbol{r}_i. \qquad (16)$$

As we have obtained ${}^{IR}\boldsymbol{r}_i$, ${}^{IL}\boldsymbol{r}_i$ is described by the following (17) through projection matrix $\boldsymbol{P}$.

$$ {}^{IL}\boldsymbol{r}_i = \boldsymbol{P} \; {}^{CL}\boldsymbol{r}_i \qquad (17)$$

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 (15), (17) are rewritten as,

$$\left\{ \begin{array}{l} {}^{IR}\boldsymbol{r}_i = \boldsymbol{f}_R({}^{CR}\boldsymbol{\psi}_M, \; {}^{M}\boldsymbol{r}_i) \\ {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L({}^{CR}\boldsymbol{\psi}_M, \; {}^{M}\boldsymbol{r}_i). \end{array} \right. \qquad (18)$$

This relation connects the arbitrary points on the object and projected points on the left and right images corresponding to a 3-D pose ${}^{CR}\boldsymbol{\psi}_M$ of the object. 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" which will be explained in 3·3.

3·2 Model-based matching

The 3-D solid model named S for the target object of a rectangular block is shown in Fig. 3 (on the top). The set of coordinates inside of the block model 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}. Projecting S_{in} and S_{out} onto the 2-D coordinates of left camera Σ_{IL}, we have

$$ S_{L,in}({}^{CR}\boldsymbol{\psi}_M) = \sum_{k=1}^{m} S_{L,in,k} = \sum_{k=1}^{m} \{ {}^{IL}\boldsymbol{r}_i \in \Re^2 \mid {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L({}^{CR}\boldsymbol{\psi}_M, {}^{M}\boldsymbol{r}_i), {}^{M}\boldsymbol{r}_i \in S_{in,k} \in \Re^3 \} \qquad (19)$$

$$S_{L,out}({}^{CR}\boldsymbol{\psi}_M) = \left\{ {}^{IL}\boldsymbol{r}_i \in \Re^2 \,\middle|\, {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L({}^{CR}\boldsymbol{\psi}_M, {}^{M}\boldsymbol{r}_i), \right.$$
$$\left. {}^{M}\boldsymbol{r}_i \in S_{out} \in \Re^3 \right\} \quad (20)$$

where $m<n$ denotes the number of the visible surfaces. The projection for the right camera is in the same way. The left and right 2-D searching models, named S_L and S_R, are shown in Fig. 3(on the bottom).

We suppose there are many solid models in the searching area, each has its own pose ${}^{CR}\boldsymbol{\psi}_M$. To determine which solid model is most close to the real target, a fitness function is defined for evaluation. 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 (20) 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.

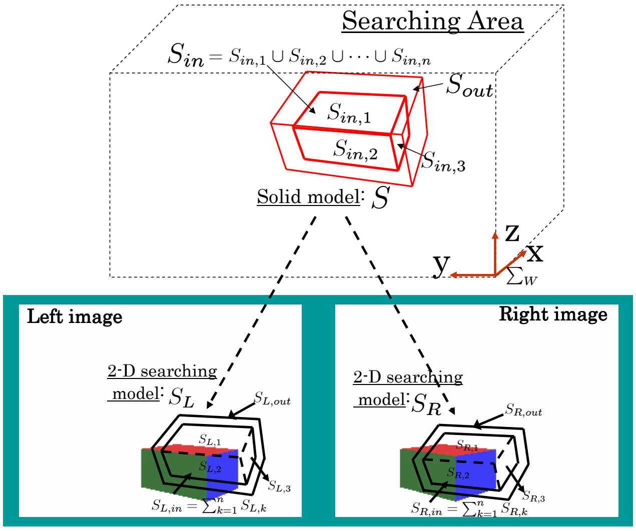
Here, we use color information to search for the target object in the images. Let therefore $b_k, (k = 1, 2, \cdots, n)$ denote the hue value of the color in $S_{in,k}$ surface of the target object. Let $h({}^{IL}\boldsymbol{r}_i)$ (or $h({}^{IR}\boldsymbol{r}_i)$) denote the hue value of the searching models 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,

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

where δ is the Kronecker delta function defined as

$$\boldsymbol{\delta}(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0. \end{cases} \qquad (21)$$

Here $\boldsymbol{\Lambda} = \sum_{k=1}^{m} n_k$, 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



Searching Area
$S_{in} = S_{in,1} \cup S_{in,2} \cup \cdots \cup S_{in,n}$
S_{out}
$S_{in,1}$
$S_{in,2}$
$S_{in,3}$
Solid model: S
z
y
x
Σ_W
Left image
2-D searching model: S_L
$S_{L,out}$
$S_{L,1}$
$S_{L,2}$
$S_{L,3}$
$S_{L,in} = \sum_{k=1}^{n} S_{L,k}$
Right image
2-D searching model: S_R
$S_{R,out}$
$S_{R,1}$
$S_{R,2}$
$S_{R,3}$
$S_{R,in} = \sum_{k=1}^{n} S_{R,k}$

Fig.3 Definition of a solid model and left/right searching models

$$F({}^{CR}\boldsymbol{\psi}_M) = (F_L({}^{CR}\boldsymbol{\psi}_M) + F_R({}^{CR}\boldsymbol{\psi}_M))/2. \qquad (22)$$

Equation (22) 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 position/orientation can be converted to searching ${}^{CR}\boldsymbol{\psi}_M$ that maximizes $F({}^{CR}\boldsymbol{\psi}_M)$. We solve this optimization problem by GA whose gene representing possible pose solution ${}^{CR}\boldsymbol{\psi}_{GA}$ is defined as,

$$\underbrace{\overbrace{01\cdots01}^{t_x}}_{12bit}\underbrace{\overbrace{00\cdots01}^{t_y}}_{12bit}\underbrace{\overbrace{11\cdots01}^{t_z}}_{12bit}\underbrace{\overbrace{01\cdots01}^{\epsilon_1}}_{12bit}\underbrace{\overbrace{01\cdots11}^{\epsilon_2}}_{12bit}\underbrace{\overbrace{01\cdots10}^{\epsilon_3}}_{12bit}.$$

The 72 bits of gene refers to the range of the searching area: $-150 \leq t_x \leq 150[mm]$, $0 \leq t_y \leq 300[mm]$, $650 \leq t_z \leq 950[mm]$, and $-0.5 \leq \epsilon_1, \epsilon_2, \epsilon_3 \leq 0.5$ which represents almost the same range of $-60 \leq roll, pitch, yaw \leq 60[deg]$.

3·3 On-line Evolutionary Recognition

Although GA has been applied to a number of robot control systems [9], it has not been yet applied to a robot manipulator control system to track a target in 3D space, since the general GA method costs much time until its convergence. So here, for real-time visual control purposes, we employ GA in a way that we denoted as “1-Step GA” evolution in which the GA evolutional iteration is applied one time to the newly input image. While using the elitist model of the GA, the position/orientation of a target can be detect in every new image by that of the searching model given

Table 1 Physical parameters of the PA-10

$Joint$	$Base$	$Link1$	$Link2$	$Link3$	$Link4$	$Link5$	$Link6$
$Length(m)$	0.200	0.115	0.307	0.143	0.225	0.245	0.080
$Center\ of\ mass\ (m)$	0.0750	-0.0518	0.0633	0.0536	0.0461	0.0803	-0.0186
$mass\ (Kg)$	3.04	9.78	8.41	3.51	4.31	3.45	1.70
$Inertia\ momentI_{xx}\ (Kgm^2)$	N/A	1.23×10^{-2}	6.86×10^{-2}	3.70×10^{-2}	2.79×10^{-2}	4.07×10^{-2}	1.09×10^{-2}
$Inertia\ momentI_{yy}\ (Kgm^2)$	N/A	6.36×10^{-2}	6.86×10^{-2}	2.62×10^{-2}	2.79×10^{-2}	5.83×10^{-3}	1.09×10^{-2}
$Inertia\ momentI_{xx}\ (Kgm^2)$	N/A	1.23×10^{-1}	1.19×10^{-2}	3.70×10^{-2}	6.48×10^{-3}	4.07×10^{-2}	6.97×10^{-4}

by the best individual in the population. This feature happens to be favorable for real-time visual recognition. We output the current best individual of the GA in every newly input image, and use it as real-time recognition result. Our previous research has confirmed the 2D recognition method enabled a eye-in-hand robot manipulator to catch a swimming fish by a net equipped at the hand [8]. Since the image inputting process is included in the GA iteration process seeking for the potential solution, so the evolving speed to the solution in the image should be faster than the speed of the target object in the successively input images, for the success of real-time recognition by "1-Step GA".

However, as the searching space extending to 3D, the time of each GA process will become longer since the parameters is increased to six. So the dynamics of recognition will become worse. The proposed MFF recognition method can help us conduct such a task since it can predict the motion of the target seeing from the cameras based on the motion of the robot. So when it got converged, GA group will move together with the moving of the target in the image, never loose it even under a high-speed moving of robot manipulator.

Using (12), the pose of the individuals ${}^{CR}\boldsymbol{\psi}_{GA}$ in the next generation can be predicted based on the current pose, presented by

$$ {}^{CR}\hat{\boldsymbol{\psi}}^{i+1}_{GA} = {}^{CR}\boldsymbol{\psi}^{i}_{GA} + {}^{CR}\dot{\boldsymbol{\psi}}_{M}{}^{i} \Delta t, \qquad (23)$$

where i represents the number of the generation. Since the effect on the recognition from the dynamics of manipulator can be compensated. Thus, recognition by hand-eye cameras will be independent of the dynamics of the manipulator, robust recognition can be obtained the same as using fixed cameras.

4. Controller

4·1 Desired-trajectory generation

The task in visual servoing is to use visual information to control the pose of the robot's end-effecter relative to a target object. In Fig. 4, suppose the motion of the target object ${}^{W}\boldsymbol{T}_{M}(t)$ (Homogeneous

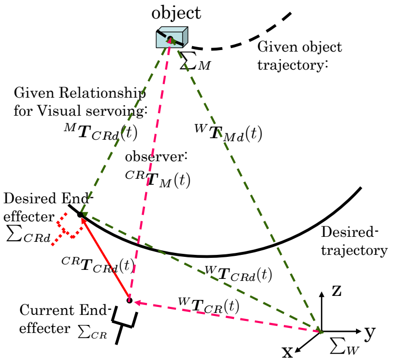

object
Given object trajectory:
Σ_M
Given Relationship for Visual servoing:
${}^{M}\boldsymbol{T}_{CRd}(t)$
${}^{W}\boldsymbol{T}_{Md}(t)$
observer:
${}^{CR}\boldsymbol{T}_{M}(t)$
Desired End-effecter
Σ_{CRd}
Desired-trajectory
${}^{CR}\boldsymbol{T}_{CRd}(t)$
${}^{W}\boldsymbol{T}_{CRd}(t)$
${}^{W}\boldsymbol{T}_{CR}(t)$
Current End-effecter Σ_{CR}
z
y
x
Σ_W

Fig.4 End-effector's motion trajectory

Transformation) is given, and the relative relationship of Σ_M and Σ_{CR} denoted by ${}^{CRd}\boldsymbol{T}_{M}(t)$ is also given, then a desired-trajectory of the robot's end-effecter is determined by

$$ {}^{W}\boldsymbol{T}_{CRd}(t) = {}^{W}\boldsymbol{T}_{M}(t) {}^{CRd}\boldsymbol{T}_{M}{}^{-1}(t). \qquad (24)$$

Denote the controlled end-effeter's coordinate as Σ_{CR}, and the relative relationship ${}^{CR}\boldsymbol{T}_{M}(t)$ can be observed by cameras, so the actual-trajectory of end-effecter is expressed by

$$ {}^{W}\boldsymbol{T}_{CR}(t) = {}^{W}\boldsymbol{T}_{M}(t) {}^{CR}\boldsymbol{T}_{M}{}^{-1}(t). \qquad (25)$$

By using Eq. (24), (25) the difference of Σ_{CRd} and Σ_{CR} denoted as ${}^{CR}\boldsymbol{T}_{CRd}$ can be deduced as

$$\begin{aligned} {}^{CR}\boldsymbol{T}_{CRd} &= {}^{W}\boldsymbol{T}^{-1}_{CR}(t) {}^{W}\boldsymbol{T}_{CRd}(t) && (26)\\ &= ({}^{W}\boldsymbol{T}_{M}(t) {}^{CR}\boldsymbol{T}^{-1}_{M}(t))^{-1}\ {}^{W}\boldsymbol{T}_{M}(t)\ {}^{CRd}\boldsymbol{T}^{-1}_{M}(t) \\ &= {}^{CR}\boldsymbol{T}_{M}(t)\ {}^{M}\boldsymbol{T}_{CRd}(t), && (27) \end{aligned}$$

Notice that Eq. (27) is a general deduction that satisfies arbitrary object motion ${}^{W}\boldsymbol{T}_{M}(t)$ and objective of visual servoing ${}^{CRd}\boldsymbol{T}_{M}(t)$.

Differentiating Eq. (27) with respect to time yields

$$ {}^{CR}\dot{\boldsymbol{T}}_{CRd} = {}^{CR}\dot{\boldsymbol{T}}_{M} {}^{M}\boldsymbol{T}_{CRd} + {}^{CR}\boldsymbol{T}_{M} {}^{M}\dot{\boldsymbol{T}}_{CRd}. \qquad (28)$$

Differentiating Eq. (28) with respect to time again

$$\begin{aligned} {}^{CR}\ddot{\boldsymbol{T}}_{CRd} = {}^{CR}\ddot{\boldsymbol{T}}_{M} {}^{M}\boldsymbol{T}_{CRd} + 2 {}^{CR}\dot{\boldsymbol{T}}_{M} {}^{M}\dot{\boldsymbol{T}}_{CRd} + \\ {}^{CR}\boldsymbol{T}_{M} {}^{M}\ddot{\boldsymbol{T}}_{CRd}. \qquad (29) \end{aligned}$$

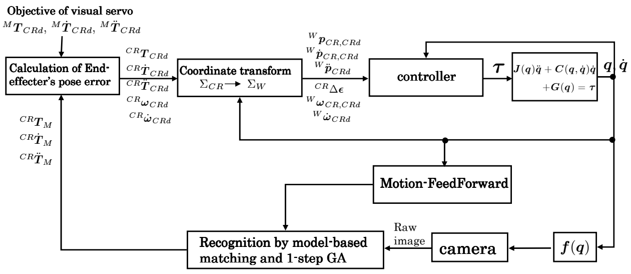

Objective of visual servo
${}^{M}\boldsymbol{T}_{CRd},\ {}^{M}\dot{\boldsymbol{T}}_{CRd},\ {}^{M}\ddot{\boldsymbol{T}}_{CRd}$
Calculation of End-effecter's pose error
${}^{CR}\boldsymbol{T}_{CRd}$
${}^{CR}\dot{\boldsymbol{T}}_{CRd}$
${}^{CR}\ddot{\boldsymbol{T}}_{CRd}$
${}^{CR}\boldsymbol{\omega}_{CRd}$
${}^{CR}\dot{\boldsymbol{\omega}}_{CRd}$
Coordinate transform
$\Sigma_{CR} \rightarrow \Sigma_{W}$
${}^{W}\boldsymbol{p}_{CR,CRd}$
${}^{W}\dot{\boldsymbol{p}}_{CR,CRd}$
${}^{W}\ddot{\boldsymbol{p}}_{CRd}$
${}^{CR}\Delta\boldsymbol{\epsilon}$
${}^{W}\boldsymbol{\omega}_{CR,CRd}$
${}^{W}\dot{\boldsymbol{\omega}}_{CRd}$
controller
$\boldsymbol{\tau}$
$\boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}$
$\boldsymbol{q}, \dot{\boldsymbol{q}}$
Motion-FeedForward
${}^{CR}\boldsymbol{T}_{M}$
${}^{CR}\dot{\boldsymbol{T}}_{M}$
${}^{CR}\ddot{\boldsymbol{T}}_{M}$
Recognition by model-based matching and 1-step GA
Raw image
camera
$\boldsymbol{f}(\boldsymbol{q})$

Fig.5 block diagram of the controller

Here, ${}^{M}\boldsymbol{T}_{CRd}$, ${}^{M}\dot{\boldsymbol{T}}_{CRd}$, ${}^{M}\ddot{\boldsymbol{T}}_{CRd}$ are given by the desired visual servoing objective. ${}^{CR}\boldsymbol{T}_{M}$, ${}^{CR}\dot{\boldsymbol{T}}_{M}$, ${}^{CR}\ddot{\boldsymbol{T}}_{M}$ can be assumed to be recognized by cameras. So ${}^{CR}\boldsymbol{T}_{CRd}$, ${}^{CR}\dot{\boldsymbol{T}}_{CRd}$, ${}^{CR}\ddot{\boldsymbol{T}}_{CRd}$ can be calculated.

4·2 Servoing controller

The block diagram of our proposed controller is shown in Fig. 5. The controller used in our visual servoing is proposed by B.Siciliano [10]. Here, we just show main equations of the controller to calculate $\boldsymbol{\tau}$, which is output to control the robot manipulator.

$$\boldsymbol{a}_p = {}^{W}\ddot{\boldsymbol{r}}_{CRd} + \boldsymbol{K}_{D_p}{}^{W}\dot{\boldsymbol{r}}_{CR,CRd} + \boldsymbol{K}_{P_r}{}^{W}\boldsymbol{r}_{CR,CRd}, \tag{30}$$

$$\boldsymbol{a}_o = {}^{W}\dot{\boldsymbol{\omega}}_{CRd} + \boldsymbol{K}_{D_o}{}^{W}\boldsymbol{\omega}_{CR,CRd} + \boldsymbol{K}_{P_o}{}^{W}\boldsymbol{R}_{CR}{}^{CR}\Delta\boldsymbol{\epsilon}, \tag{31}$$

$$\ddot{\boldsymbol{q}}_d = \boldsymbol{J}^{+}(\boldsymbol{q})(\begin{bmatrix} \boldsymbol{a}_p \\ \boldsymbol{a}_o \end{bmatrix} - \dot{\boldsymbol{J}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}) + (\boldsymbol{I} - \boldsymbol{J}^{+}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q}))$$

$$(\boldsymbol{E}_p(\boldsymbol{q}_0 - \boldsymbol{q}) + \boldsymbol{E}_d(\boldsymbol{0} - \dot{\boldsymbol{q}})), \tag{32}$$

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}). \tag{33}$$

Here, the vectors in Eq. (30), (31) is expressed in Σ_W, which can be obtained from the vectors in Σ_{CR} deduced in 3. by coordinate transformation as follows.

It has been proved in [10] that the system must be exponentially stable for any choice of positive definite $\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}$ and $\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}$, thus

$$\lim_{t\to\infty} {}^{W}\boldsymbol{r}_{CR,CRd} = \boldsymbol{0} \qquad \lim_{t\to\infty} {}^{W}\dot{\boldsymbol{r}}_{CR,CRd} = \boldsymbol{0} \tag{34}$$

$$\lim_{t\to\infty} {}^{CR}\Delta\boldsymbol{\epsilon} = \boldsymbol{0} \qquad \lim_{t\to\infty} {}^{W}\boldsymbol{\omega}_{CR,CRd} = \boldsymbol{0}. \tag{35}$$

Substituting Eq. (34), (35) to Eq. (26), we have

$$\lim_{t\to\infty} {}^{CR}\boldsymbol{T}_{CRd} = \boldsymbol{I} \qquad \lim_{t\to\infty} {}^{CR}\dot{\boldsymbol{T}}_{CRd} = \boldsymbol{0} \tag{36}$$

Substituting Eq. (36) to Eq. (27), we have

$$\lim_{t\to\infty} {}^{CR}\boldsymbol{T}_{M} = \lim_{t\to\infty} {}^{CRd}\boldsymbol{T}_{M} \tag{37}$$

Eq. (37) proves stable convergence of visual servoing.

5. Simulation of visual servoing

To verify the effectiveness of the proposed visual servoing system, we conduct the simulations of visual servoing to a rectangular solid block($100mm \times 150mm \times 200mm$) with symmetrical colored surfaces.

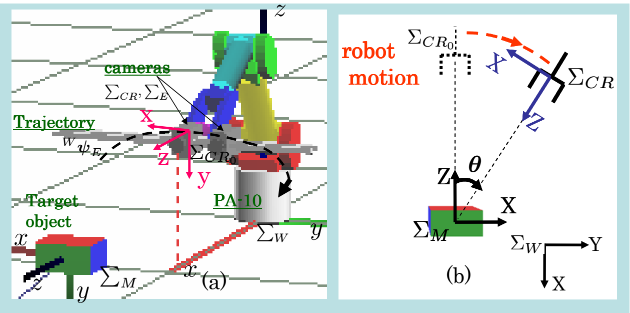


Fig.6 (a)Simulation experiment system created by OpenGL. (b) coordinate systems of simulation experiment

5·1 simulation condition

The simulation experiment is performed using software "Open GL". Here, a manipulator modeling the actual 7-link "PA-10", which is a 7 DOF robot arm manufactured by mitsubishi Heavy Industries. Two cameras are mounted on its end-effector, shown in Fig. 6. Physical parameters of the PA-10 are shown in Table 1. The initial pose of the end-effector (also the right camera) is defined as Σ_{CR_0}, and given by

$$\begin{aligned} {}^{W}\boldsymbol{\psi}_{CR_0} &= [{}^{W}\boldsymbol{r}_{CR_0}, {}^{W}\boldsymbol{\epsilon}_{CR_0}]^T \\ &= [0.918[m], 0, 0.455[m], -0.5, 0.5, -0.5]^T \end{aligned} \tag{38}$$

as shown in Fig. 6(a). We compare the visual servoing using "1-step GA " with that using "1-step GA + MFF" when the target object is set as

$$\begin{aligned} {}^{CR_0}\boldsymbol{\psi}_M &= [{}^{CR_0}\boldsymbol{r}_M, {}^{CR_0}\boldsymbol{\epsilon}_M]^T \\ &= [0[m], 0.1, 0.6[m], 0, 0.130, 0]^T, \end{aligned} \tag{39}$$

which means 15[deg] rotation of Σ_M around y axis of Σ_{CR_0}. Here, we suppose the target is static, and the step response ability of the visual servoing system is estimated. The objective of the visual servoing system is

$$\begin{aligned} {}^{CRd}\boldsymbol{\psi}_M &= [{}^{CRd}\boldsymbol{r}_M, {}^{CRd}\boldsymbol{\epsilon}_M]^T \\ &= [0[m], 0.1[m], 0.6[m], 0, 0, 0]^T, \end{aligned} \tag{40}$$

so the final pose of the end-effector should stay stablely as

$$\begin{aligned} {}^{CR}\boldsymbol{\psi}_M &= [{}^{CR}\boldsymbol{r}_M, {}^{CR}\boldsymbol{\epsilon}_M]^T \\ &= [0[m], 0.1[m], 0.6[m], 0, 0, 0]^T, \end{aligned} \tag{41}$$

shown in Fig. 6(b).

Notice that the pose of the target object ${}^{CR}\boldsymbol{\psi}_M$ is supposed to be unknown before visual servoing.

5·2 simulation results

The cost of one generation process of GA is 140[ms], here we assume it can be finished in 30[ms] in the simulation to perform real-time recognition (in the case

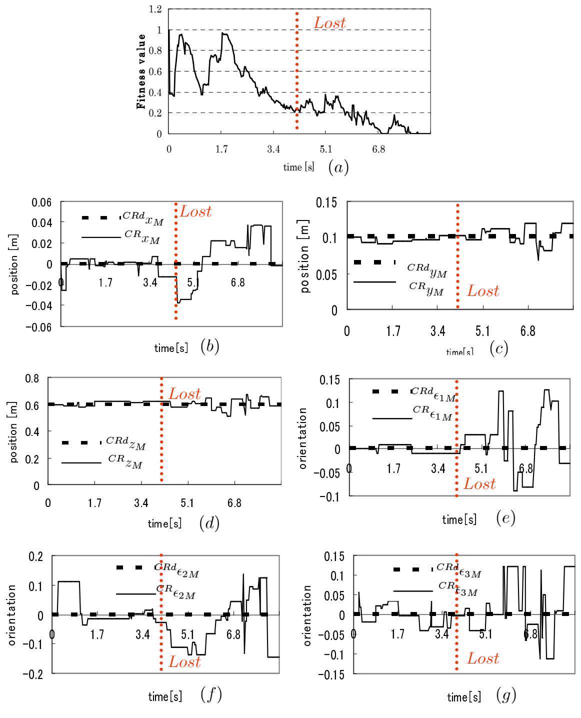


Fig.7 Visual servoing without MFF when $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [0.3, 0.6, 0.6, 0.6, 0.6, 0.6]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [0.5, 1, 1, 2, 2, 2]^T$.

the frame frequency is about 33fps). In the future practical experiment, fast recognition could be realized by using multi-computer to decrease the calculation time.

Table 2 GA Parameters

Population size	30 individuals
Selection rate	0.5
Crossover	Two-point
Mutation rate	0.10
Elitist model	*yes*

Firstly, we set the gains of controller in (30, 31) are set as $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [0.3, 0.6, 0.6, 0.6, 0.6, 0.6]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [0.5, 1, 1, 2, 2, 2]^T$.

Fig. 7 shows step response of visual servoing by using just “1-step GA”. When the robot starts to moving, the pose of target object in Σ_{CR} is changed due to the dynamics of the robot manipulator. The “1-Step GA” can not recognize precisely, wrong recognition result will lead to wrong control of the robot, which makes the recognition more difficult. As shown in Fig. 7(a), the fitness value of the recognition is decreasing, and after $t = 4[s]$, the fitness value is lower than 0.2, in such case, the target object is considered as to be lost, and the robot can not be normally controlled. The reason of target object lost is considered as GA’s convergence speed was not faster than the target speed relative to the camera.

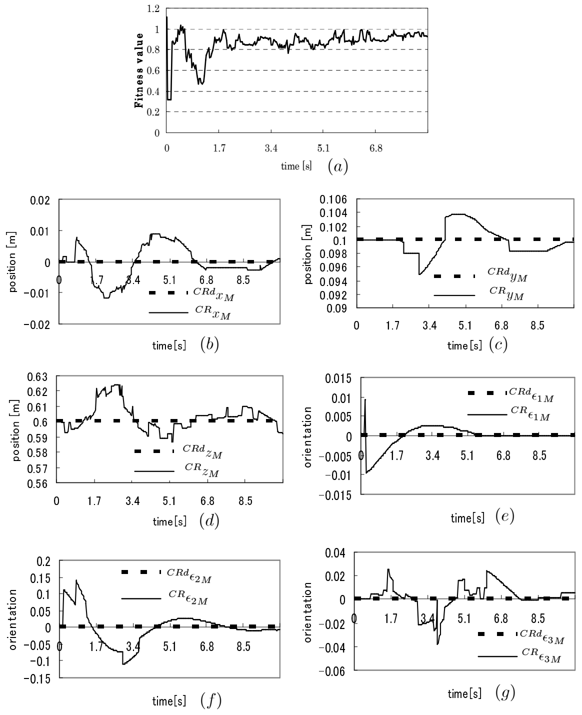


Fig.8 Visual servoing with MFF when $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [0.3, 0.6, 0.6, 0.6, 0.6, 0.6]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [0.5, 1, 1, 2, 2, 2]^T$.

On the other hand, “1-step GA + MFF” gives stable control of the robot manipulator, as shown in Fig. 8. Since the the target’s fictional motion coming from the moving cameras can be compensated by using MFF method, the recognition dynamics can be improved. Once “1-step GA” finds the closeness model of the target object, the model will keep overlapping the target object, never lose it. So in the case of using “1-step GA + MFF” recognition method, the visual servoing can be performed, and the end-effector’s motion will be converged after about 7s.

Secondly, we set the gains of controller in (30, 31) are set as $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [1.5, 3, 3, 3, 3, 3, 3]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [2.5, 5, 5, 10, 10, 10]^T$. In this case, the response of the robot will become more faster, so by using just “1-step GA”, the correct recognition can not be received. See Fig. 9(a), when the robot starts to moving, the fitness value decreased to 0.4 soon, and did not increase. After $t = 2[s]$, the target object is lost, and the robot can not be normally controlled. The results of visual servoing by using “1-step GA + MFF” are shown in Fig. 10. Even the response of the robot will become more faster, the models can still keep matching the target object well, and the end-effector’s motion will be converged after about 2s.

These simulations have confirmed that MFF method can compensate the fictional target motions in the camera view induced by the end-effctor’s mo-

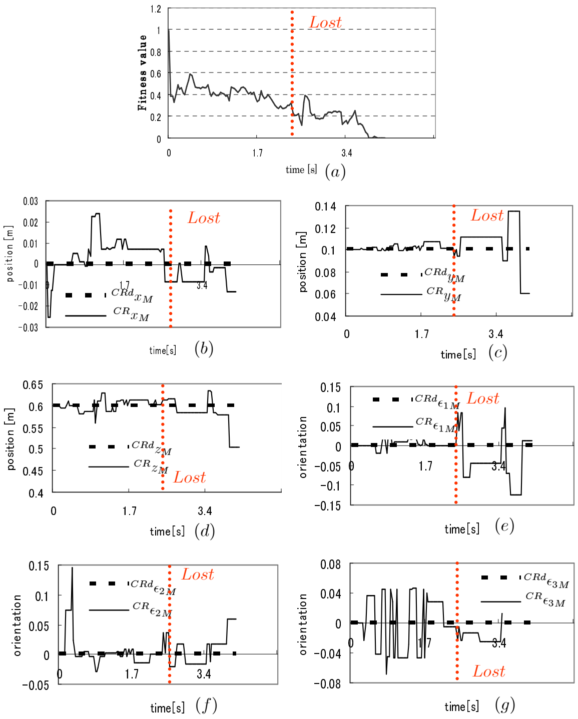


Fig.9 Visual servoing without MFF when $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [1.5, 3, 3, 3, 3, 3, 3]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [2.5, 5, 5, 10, 10, 10]^T$.

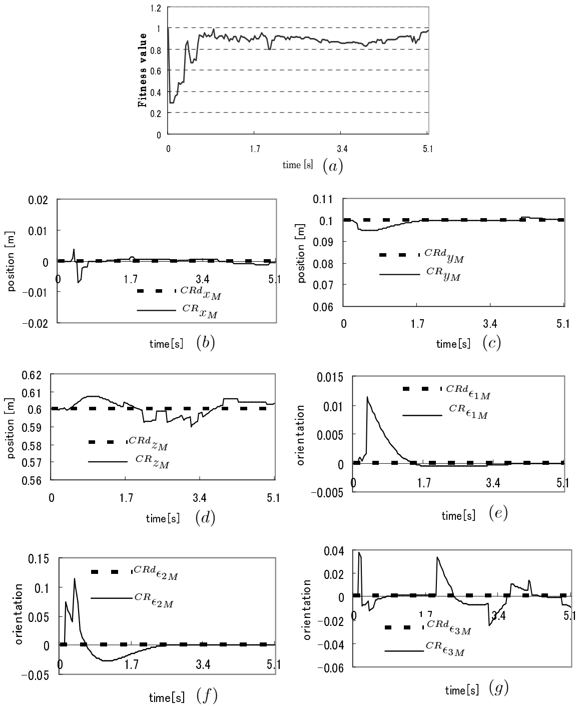


Fig.10 Visual servoing with MFF when $[\boldsymbol{K}_{D_p}, \boldsymbol{K}_{D_o}]^T = [1.5, 3, 3, 3, 3, 3, 3]^T$, $[\boldsymbol{K}_{P_p}, \boldsymbol{K}_{P_o}]^T = [2.5, 5, 5, 10, 10, 10]^T$.

tion, the recognition became robust to the dynamics of the manipulator. It has been proved that including "1-step GA + MFF" method to visaul servoing system gives stable control of the robot.

6. Conclusion

We have proposed a method of visual servoing to a 3D pose (6 degree of freedom) of a target object. The 3D pose measurement method 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 motion-feedforward compensation method is proposed to improve visual recognition dynamics, which become worse by disturbing hand-eye motion during visual servoing of the robot manipulator. Simulation results have been verified the effectiveness of the proposed MFF method to give stable visual servoing to a 3D pose.

[1] S.Hutchinson, G.Hager, and P.Corke, "A Tutorial on Visual Servo Control", IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pp. 651-670, 1996.

[2] P.Y.Oh, and P.K.Allen, "Visual Servoing by Partitioning Degrees of Freedom", IEEE Trans. on Robotics and Automation, vol. 17, no. 1, pp. 1-17, 2001.

[3] E.Malis, F.Chaumentte and S.Boudet, "2-1/2-D Visual Servoing", IEEE Trans. on Robotics and Automation, vol. 15, no. 2, pp. 238-250, 1999.

[4] P.K.Allen, A.Timchenko, B.Yoshimi, and P.Michelman, "Automated Tracking and Grasping of a Moving object with a Robotic Hand-Eye System", IEEE Trans. on Robotics and Automation, vol. 9, no. 2, pp. 152-165, 1993.

[5] 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

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

[7] 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

[8] M.Minami, H.Suzuki, J.Agbanhan, 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.

[9] T.Nagata, K.Konishi and H.Zha: "Cooperative manipulations based on Genetic Algorithms using contact information", Proceedings of the International Conference on Intelligent Robots and Systems, pp.400-5, 1995

[10] B.Siciliano and L.Villani: *Robot Force Control*, ISBN 0-7923-7733-8.