On-line Stable Evolutionary Recognition Based on Unit Quaternion Representation by Motion-Feedforward Compensation

Wei Song*, Mamoru Minami1, and Seiji Aoyagi2
1 University of Fukui, Graduate School of Engineering, Japan
2 Kansai University, Faculty of Engineering, Japan
Received 11 January 2007; revised 15 February 2007, accepted 6 May 2007

Abstract
This paper presents a pose measurement method of a 3D object. The proposed method utilizes an evolutionary search technique of the genetic algorithm (GA) and a fitness evaluation based on a stereo model matching 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 the simulation experiments.

Keywords
3D measurement, genetic algorithm (GA), unit quaternion, 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-Effecter of a manipulator toward a target object are referred to as visual servoing [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 3D pose measurement of a target.

There is a variety of approaches for 3D pose estimation, and they can be classified into three categories: feature-based, appearance-based, and model-based. Feature-based approaches use local features like points, line segments, edges, or regions. The main idea of this method is to select a set of feature points, which are matched against the incoming video to update the estimating pose. Feature-based techniques are naturally less sensitive to occlusions, as they are based on local correspondences. Several researches apply this method to head pose estimation based on tracking of small facial features like the corners of the eyes or mouth. Yang and Zhang [5] presented a head tracking algorithm using stereovision to overcome the occlusion problem. However, the tracker needs to know the initial head pose to start tracking that is determined by several corresponding landmark points in each image, which are selected manually.

In appearance-based approaches, the image is compared with various templates to determine which one most closely matches the image, resulting in wasting time to recognize. Some appearance-based methods include the work of Niyogi [6] and Masson [7]. In Ref. [7], the surface of the target 3D object is modeled by a set of small square patches, which are determined from several key views by a learning process. The model-based method is to use a model to search a target object in the image, and the model is composed based on how the target object can be seen in the input image [8], [9]. Our method is included in this category. The matching degree of the model to the target can be estimated by a fitness function, whose maximum value represents the best matching. The 3D pose to
give the best matching can be solved by GA. An advantage of our method is that we use a 3D solid model which possesses 6-DOF (both the position and orientation). 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 complicated and time consuming, e.g., Ref. [10].

There are three main contributions of our 3D object's pose recognition framework. One is that we use unit quaternion representation to express the target object's pose. In order to perform 3D pose measurement, both the target's position and orientation must be recognized simultaneously. And in visual servoing tasks, the six-DOF recognition results will be used to handle the end-effector position and orientation as feedback information, where the 3D measurement performance is connected with how to present the orientation. So a suitable representation of orientation should be sought for. Several parameterizations exist to describe the orientation angles, including a set of three Euler angles and angle/axis representations which describe the general orientation of a rigid body as a displacement of an angle around a unit axis. Whereas a drawback of the three Euler angles is the occurrence of representation singularities (for manipulator, the orientation Jacobian matrix is singular for some orientation). A general angle/axis representation is not unique since a rotation by an angle $-\theta$ about an axis $-\kappa$ can not be distinguished from a rotation by $\theta$ about $\kappa$. Moreover, a representational singularity happens when $\theta = 0$, the rotation axis can not be determined, resulting in that the singularity problem makes the GA confuse and the pose recognition can not be fulfilled. Thus, it can not be used to apply visual servoing tasks in which the desired target pose is $\theta = 0$. A unit quaternion is a different angle/axis representation. It can represent the orientation of a rigid body without singularities. Recently, unit quaternion has been successfully used for attitude control of rigid bodies [11] and control of robot manipulator [12], [13].

The other two contributions of this paper is about improving the measurement dynamics. When constructing visual servoing system of eye-in-hand robot, the dynamics of the robot is a problem standing against the stable motion control of robots. It is known in the control theory that the influence from changing the dynamics of the system to be controlled could be suppressed by the effect of feedback control loop. On the other hand, a changing of the dynamics of the sensing unit in the feedback loop causes direct influence to the output motion without suppression like changing the dynamics of the system. In this sense, improvement of the dynamics of the sensing unit is important. Generally, in visual servo system, the sensing unit is the part of recognition using cameras. Thus, the recognition dynamics can be defined as a phenomenon that the sensed variables (the position and orientation of the target object) can be detected with time delay because sensing mechanism generally be governed by differential equations in time domain. This time delay may cause this feedback system unstable. However, it looks like that the researches concerning the sensing dynamics for visual servoing has been ignored so far. To see clearly the measurement dynamics in 3D pose recognition, in this paper, we do not deal with the dynamics of the manipulator.

One of these two contributions is that a mathematical formulation is given to express the relationship between the motion of the target in the camera view and its two effect factors: the motion of the target in real world and the motion of the camera. It is important for the robot to distinguish what is the real motion of the target and what is a fictitious motion just coming from the hand-eye camera. Then we can predict the target's 3D pose along with the motion of the end-effector based on the current target's 3D pose to compensate the fictitious motion, which is defined as motion-feedforward compensation method. It is considered that the dynamics of recognition will be improved by using the motion-feedforward method to compensate the fictitious motion, giving stability to visual servo system. The combination of 3D model-based matching method and GA pave the way smoothly for improving the recognition dynamics by the feedforward recognition through predicting the 3D pose. On the other hand, the other methods which are based on the 2D image points, like feature-based, appearance-based methods, to do such a prediction may be complicated and time-consuming.

The other one is that we proposed a “1-Step GA” method for real-time recognition. The reasons to use GA for optimization, i.e., the recognition of the target are mainly two: the first is that GA is useful for optimization over multi-peak distribution; the second is its high ability to converge quickly into the optimum peak. The GA-based scene recognition method described here can be designated as “evolutionary recognition method,”
since for every step of the GA’s evolution, it
struggles to perform the recognition of a target in
the input image. 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 newly input image, which we called
“1-Step GA” [14]. In this way, the GA searching
process and the convergence to the target does not
consist in one image but the convergence is
achieved in the sequence of the continuously input
images to recognize it. This real-time recognition
nature of 1-Step GA is justified when GA’s
convergence speed be faster than that of moving
object in the dynamic images.

The remainder of this paper is organized as follows.
The following section describes motion-
feedforward compensation method. Next is 3D
measurement method, in which a solid model and a
fitness function is defined. Then effectiveness of
3D pose measurement method is evaluated through
the simulation experiments. Finally, the conclusion
of the work is presented.

2. MOTION-FEEDFORWARD
COMPENSATION

Most visual servo systems use an eye-in-hand
configuration, having the camera mounted on the
robot’s end-effector. In this case, the motion of
the target in the camera coordinate will be affected by
both the motion of the target in real world and the
motion of the camera. Here what we are interested in
is how to predict the target velocity based on the
motion of the hand-eye camera. This can be
considered the same as human’s action, i.e., we can
predict the target pose caused by the motion of
ourselves. To apply such intelligence into a
manipulator, a robust recognition has been
proposed that we named as a motion-feedforward
recognition method, in which the target velocity is
predicted based on the joint velocity of manipulator
to compensate the influence from the motion of the
camera itself.

Take the eye-in-hand robot as an instance, we
explain how to describe such a relationship between
a target and a moving camera in a mathematical
formulation. First, we establish relations among
relative velocities of three moving frames, world
double coordinate system \( \Sigma_w \), target coordinate system
\( \Sigma_m \) and camera coordinate system as \( \Sigma_{cr} \), shown
in Figure 1. Take \( \Sigma_w \) as the reference frame.
Denote the vector from \( O_w \) (the origin of \( \Sigma_w \)) to
\( O_{cr} \) expressed in \( \Sigma_w \) as \( w r_{cr} \), the vector from

\[ O_w \text{ to } O_m \text{ expressed in } \Sigma_w \text{ as } w r_{m}, \text{ and the vector from } \Sigma_{cr} \text{ to } \Sigma_m \text{ expressed in } \Sigma_{cr} \text{ as } cr r_{cr,m}. \]

We define robot's end-effector coordinate system as \( \Sigma_h \), which is considered same as \( \Sigma_{cr} \) since the camera is mounted on the robot's
end-effector. So the rotation matrix \( w r_{cr} \) is a
function of the joint vector \( \mathbf{q} \). Then the following
relation hold:

\[ \text{CR } r_{cr,m} = \text{CR } R_w (q)(w r_m - w r_{cr}(q)). \] (1)

Differentiating Eq. (1) with respect to time.

\[ \text{CR } r_{cr,m} = \text{CR } R_w (q)(w r_m - w r_{cr}(q)) + S(\text{CR } \omega_w) \] (2)

where \( S(\cdot) \) is the operator performing the cross
product between two \((3 \times 1)\) vectors.

Given \( \mathbf{w} = [\omega_x, \omega_y, \omega_z]^T \), \( S(\mathbf{w}) \) takes on the form

\[ S(\mathbf{w}) = \begin{bmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{bmatrix}. \] (3)

Similarly, the angular velocities of \( \Sigma_{cr} \) and \( \Sigma_m \)
with respect to \( \Sigma_w \) are \( w \omega_{cr} \) and \( w \omega_m \), and the
angular velocity of \( \Sigma_m \) with respect to \( \Sigma_{cr} \)
is \( \text{CR } \omega_{cr,m} \). Then it can be written as:

\[ \text{CR } \omega_{cr,m} = \text{CR } R_w (q)(w \omega_{m} - w \omega_{cr}(q)). \] (4)

In this paper, the target’s orientation is expressed by
a four-parameter representation, namely, the unit
quaternion. The target pose based on \( \Sigma_{cr} \) is defined as \( \text{CR } Q_m = \{ \text{CR } \eta_m, \text{CR } e_m \} \). Since \( \text{CR } \eta_m \) can
be determined by \( \text{CR } e_m \) as:

\[ \text{CR } \eta_m = \sqrt{1 - \text{CR } e_m^T \text{CR } e_m}, \] (5)
we use only three parameters $\varepsilon_{CR}$ to express the target’s orientation. So the 3D pose of the target can be expressed by a six-parameter representation:

$$CR \varphi_M = [CR \, r_{CR,M}, CR \, \varepsilon_M]_T,$$

(6)

where $CR \, r_{CR,M} = [t_x, t_y, t_z]^T$, $CR \, \varepsilon_M = [\varepsilon_1, \varepsilon_2, \varepsilon_3]^T$.

The target’s 3D pose velocity is defined as:

$$CR \tilde{\varphi}_M = [CR \, \dot{r}_{CR,M}, CR \, \dot{\varepsilon}_M]_T,$$

(7)

where the time derivation of target’s position $CR \, \dot{r}_{CR,M}$ is given by Eq. (2). The relation between $CR \, \dot{r}_{CR,M}$ and the angular velocity of the target $CR \, \dot{\varepsilon}_M$ can be written as

$$CR \dot{\varepsilon}_M = \frac{1}{2} \left( CR \eta W I - S(CR \varepsilon_M) CR \omega_{CR,M} \right)$$

(8)

where $CR \omega_{CR,M}$ is given by Eq. (4).

Moreover, the camera velocity (considered as the end-effector velocity) can be expressed using the Jacobian matrix $J(q) = [J^e (q), J^o (q)]^T$,

$$W \dot{r}_{CR} = J^e (q) \dot{q},
$$

(9)

$$W \omega_{CR} = J^o (q) \dot{q},
$$

(10)

$$S(CR \omega_W) = -CR R_W(q) S(W \omega_{CR}) W R_W(q) = -CR R_W(q) S(J_o(q) \dot{q}) W R_{CR}(q).$$

(11)

Substituting Eq. (9), (10), (11) to Eq. (2), (8), the target velocity $CR \dot{\varphi}_M$ can be described by:

$$CR \dot{\varphi}_M = CR \left[ \begin{array}{c} \dot{r}_{CR,M} \\ \dot{\varepsilon}_M \end{array} \right]$$

$$= \left[ \begin{array}{c} CR R_{W}(q) J^e (q) \dot{q} + CR \dot{r}_W(q) \\ S(W \omega_{CR}) \dot{r}_W(q) \end{array} \right]$$

$$+ \frac{1}{2} \left( CR \eta_M I - S(CR \varepsilon_M) CR \dot{r}_W(q) J^o(q) \right)$$

$$= \left[ \begin{array}{c} J^e (q) \dot{q} + J^o(q)^W \dot{\varphi}_M \\ 0 \end{array} \right].$$

(12)

The relationship $J_o(q)$ given by Eq. (12) describes how target pose change in $\Sigma_{CR}$ with respect to the pose changing of itself in real world. The relationship $J_e(q)$ given by Eq. (12) describes how target pose change in $\Sigma_{CR}$ with respect to changing manipulator pose that 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 $\Sigma_{CR}$ based on the joint velocity of manipulator $q$, so we can rewrite Eq. (12) as:

$$CR \dot{\varphi}_M = J_m(q) \dot{q}.$$

(13)

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

$$CR \dot{\varphi}_M (t + \Delta t) = CR \dot{\varphi}_M (t) + CR \phi_M \Delta t.$$

(14)

$CR \phi_M \Delta t$ is the changing extent from the current pose to the next. We consider that the recognition ability will be improved by using Eq. (14) 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.

However, from the above detailed representation of Eq. (12), it can be learned that to get $CR \dot{r}_{CR,M}$, not only the joint information of manipulator $q$ and $\dot{q}$, but also the current target’s position $CR \dot{r}_{CR,M}$ is required. Also, to get $CR \dot{\varepsilon}_M$ the current target’s orientation, $CR \eta_M, CR \varepsilon_M$ are needed. It means that the accuracy of the target’s 3D pose prediction depends on the detected accuracy of current 3D pose. In other words, the recognition performance relies on the initial situation.

3. 3D MEASUREMENT METHOD
Kinematics of Stereo-Vision

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

$$CR r_i = CR T_M^M r_i.$$

(15)
where \( M_r \) is predetermined fixed vectors. Using a homogeneous transformation matrix from \( \Sigma_w \) to \( \Sigma_{cr} \), i.e., \( W_{cr} \), then \( W_r \) is got as,

\[
W_r = W_{cr} \tilde{r}_1. \tag{16}
\]

The position vector of \( i \) point in right image coordinates, \( IR_r \), is described by using projection matrix \( P \) of camera as,

\[
IR_r = P CR i \tilde{r}_1. \tag{17}
\]

By the same way as above, using a homogeneous transformation matrix of fixed values defining the kinematical relation from \( \Sigma_{cl} \) to \( \Sigma_{cr} \), \( CL_{cr} \), \( IL_r \) is,

\[
CL_r = CL_{cr} CR i \tilde{r}_1. \tag{18}
\]

As we have obtained \( IR_r \), \( IL_r \) is described by the following Eq. (19) through projection matrix \( P \).

\[
IL_r = P CL i \tilde{r}_1. \tag{19}
\]

Then position vectors projected in the \( \Sigma_{ir} \) and \( \Sigma_{il} \) of arbitrary point \( i \) on target object can be described \( IR_r \) and \( IL_r \). Here, position and orientation of \( \Sigma_M \) based on \( \Sigma_{cr} \) has been defined as \( \phi = [t_x, t_y, t_z, e_x, e_y, e_z] \) (we use \( \phi \) to instead of \( \phi_M \) used in section 2), where \( e_x, e_y, e_z \) are quaternion parameters. Then Eq. (17), (19) are rewritten as,

\[
\begin{align*}
IR_r &= f_R(\phi, M_r) \\
IL_r &= f_L(\phi, M_r).
\end{align*} \tag{20}
\]

This relation connects the arbitrary points on the object and projected points on the left and right images with the variable \( \phi \), which is considered to be unknown in this paper. This measurement problem of \( \phi(t) \) in real time will be solved by consistent convergence of a matching model to the target object by a “1-Step GA” \[14\]. When evaluating each point above, the matching problem of corresponding point in left and right images mentioned in the introduction is arisen. Therefore, to avoid this problem, the 3D model-based matching that treats the image as a set, is chosen instead of point-based corresponding. The 3D model for the target object of a rectangular block is shown in Figure 3. The set of coordinates inside of the block is depicted as \( S_{in} \), which is composed of each surfaces \( S_{in,k}(k = 1,2,\ldots,n) \), 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 \( S \), which are projected onto \( \Sigma_{il} \) are expressed as,

\[
S_{in,l}(\phi) = \sum_{k=1}^{m} S_{in,k} \in \mathbb{R}^3 \quad \text{and} \quad S_{out,l}(\phi) = \sum_{k=1}^{m} S_{out,k} \in \mathbb{R}^3 \tag{21}
\]

where \( m \leq n \) denotes the number of the visible surfaces. The left searching model projected to \( \Sigma_{il} \) is shown in Figure 4(a). The area composed of \( S_{in,l} \) and \( S_{out,l} \) 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 Figure 4(b).

**Model Definition**

Here, we define evaluation function to estimate how much the moving solid model \( S \) defined by

![Figure 3. Solid model for a block.](image-url)
\( \phi \) 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 \( S_R \). Therefore, if the camera parameters and kinematical relations are completely accurate, and the solid searching model describes precisely the target object shape, then \( S_{L,in} \) and \( S_{R,in} \) will completely lie on the target reflected on the left and right images, provided that true value of \( \phi \) is given.

In order to search for the target object in the images, the surface-strips model shown in Figure 4 and its color information are used. It is easy to understand that the color can be limited only by hue value of the HSV color system as shown in Figure 5(b). Let therefore \( \delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \) be the Kronecker delta function defined as

\[
\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}
\]

where \( \delta \) is the Kronecker delta function defined as

\[
\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}
\]

Here \( H = \sum_{k=1}^{m} n_k \), and \( n_k \) represents the number of the searching points in \( S_{L,in} \) and \( S_{R,in} \). \( H \) is a scaling factor that normalizes \( F(\phi) \leq 1 \). In case of \( F(\phi) < 0 \), \( F(\phi) \) is given to zero. The first part of this function expresses how much each color area of \( S_{L,in} \) and \( S_{R,in} \) defined by \( \phi \) 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 subtraction of the internal surface summation and the contour-strips of the surface-strips model can make the estimation more sensible, especially in distance recognition between the object to the cameras which determine the size of the flat models. Eq. (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, the fitness function \( F(\phi) \) gives the maximum value. We confirm this as follows.

\[
F_L(\phi) = \frac{1}{H} \sum_{k=1}^{m} \left\{ \sum_{r_i \in S_{L,in},k(\phi)} \delta(h(L_r_i) - b_k) - \sum_{r_i \in S_{L,out}(\phi)} \delta(h(L_r_i) - b_k) \right\}
\]

\[
F_R(\phi) = \frac{1}{H} \sum_{k=1}^{m} \left\{ \sum_{r_i \in S_{R,in},k(\phi)} \delta(h(R_r_i) - b_k) - \sum_{r_i \in S_{R,out}(\phi)} \delta(h(R_r_i) - b_k) \right\}
\]

\[
F(\phi) = \frac{1}{2} (F_L(\phi) + F_R(\phi)) \tag{22}
\]

Figure 4. Searching models.

Figure 5. Color information.

In the left image \( S_L \) in; in; 3 \( S_L \) out; in; 1 \( S_L \) in; in; 2 \( S_R \) in; in; 3 \( S_R \) out; in; 1 \( S_R \) in; in; 2

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

Flat Model Flat Model

SR;in = SR;in;1 \[\ldots\] SR;in;n

SL;in = SL;in;1 \[\ldots\] SL;in;n

Figure 4. Searching models.

(a) Target object (b) Hue Circle

Figure 5. Color information.
The 3D pose of a target object is 
\((t_x, t_y, t_z) = (0,150,800)\;[mm]\), 
\((\phi, \varepsilon_2, \varepsilon_3) = (0,0,0)\), 
that we measured beforehand. Here, a 3D plot of fitness distribution calculated by Eq. (22) is shown in Figure 6, using a moving model with fixed orientation coinciding with the true values of the target object. Consider that the position \(t_x\) is fixed to several values as a parameter in Figure 6 (a)–(e). It can be seen that when the position of the model 
fits to the true position 
\((t_x, t_y, t_z) = (0,150,800)\;[mm]\), the fitness function has the maximum value as shown in Figure 6 (c).

In the same way, a 3D plot of fitness distribution by scanning the orientation space using a moving model with fixed position coinciding with the true values of the target object is shown in Figure 7. It shows that the fitness function has the maximum value when the orientation of the model fits to the true values 
\((\phi, \varepsilon_2, \varepsilon_3) = (0,0,0)\), as shown in Figure 7 (c).

Therefore the problem of recognition of target object and detection of its 3D pose can be converted to searching problem of \(\phi\) that maximizes \(F(\phi)\).

To recognize the target object in short time, we solve this optimization problem to search for \(\phi\) to maximize \(F(\phi)\) by GA whose gene denoted by 
\(CR\;\phi_{GA}\) is defined in Figure 8. The 72 bits of gene refers to the range of the searching area:

\[-150 \leq t_x \leq 150[mm], \quad 0 \leq t_y \leq 300[mm], \quad 650 \leq t_z \leq 950[mm], \quad -0.5 \leq \varepsilon_1, \varepsilon_2, \varepsilon_3 \leq 0.5\] 

which represents almost the same range of 
\(-60 \leq \text{roll}, \text{pitch}, \text{yaw} \leq 60[\text{deg}]\).

**On-line Evolutionary Recognition**

Although GA has been applied to a number of robot control systems [15], it has not been yet applied to a robot manipulator control system to track a target in 3D space with unpredictable movement in real time, 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. This means that the GA evolitional iteration is applied one time to the newly input image. While using the elitist model of the GA, the 3D pose of a target can be detected in every new image by that of the searching model given by the highest-fitness-function gene in the GA. In addition, this feature that the highest-fitness gene represents the current detected pose happens to be favorable for real-time visual recognition. Our previous research has confirmed the 2D recognition method enabled an eye-in-hand robot manipulator to catch a swimming fish by a net equipped at the hand Ref. [14]. The image inputting process is included in the
GA iteration process seeking for the potential solution, i.e., toward the target. For the success of real-time recognition by "1-Step GA," the evolving speed to the solution in the image should be faster than the speed of the target object in the successively input images.

However, when the target object moves quickly in the image, here we suppose the moving is just from the motion of the camera, the dynamics of recognition will become worse. It is considered that increasing the number of individuals of GA will be a possible way to solve this problem. But even more individuals exist in the searching area, only a part of them can converge to the vicinity of the target object when the convergence pressure of GA mechanism is only used, and the genes can not move together with the motion of the camera. Further, more individuals cost more time during each generation. Thus, time delay becomes more remarkable, resulting in the offset error between the position of the target and the distribution center of the individuals, which may cause loosing the target object in real-time recognition.

We will explain this by using Figure 9, which shows the recognition result of a ball under a given velocity of the hand-eye camera using different population size. Initially, the robot is not moving and GA converged to a static ball in the input image “i = 0”. When the ball starts moving in the image (like the images “i = k” and “i = k + 1”) at a velocity of “v”, time delay will exist using ”1-step GA” method. In Figure 9 (a), the delay time is expressed by $\Delta t_a$, and in Figure 9 (b), it is expressed by $\Delta t_b$. It is obvious that more individuals cost more time delay, that is $\Delta t_b > \Delta t_a$, so the center of the converged GA group got far from the center of the target in Figure 9 (b) than Figure 9 (a). Thus, it became more difficult to track to the moving ball with more individuals.

Using motion-feedforward recognition method, the motion of the target seeing from the cameras can be predicted by using the motion of the robot, so when it got converged, GA group will move together with the moving target in the image, never loose it even under a high-speed moving of hand-eye camera. As shown in Figure 9 (c), using motion-feedforward recognition method, only less individuals result in an effective recognition in real-time. As the searching space extending to 3D, the time of each GA process will become longer since the variables are increased to six. So the motion-feedforward recognition method is more important to decrease the time delay, and to improve the dynamics of recognition.

Using Eq. (14), the pose of the individuals $\phi$ in the next generation can be predicted from the current end-effector motion, presented by

$$\phi_{i+1}^{CR} = \phi_{i+1}^{GA}(t) + \phi_{M}^{CR} \Delta t.$$  \hspace{1cm} (24)

The recognition system of the proposed method is shown in Figure 10. We consider that the recognition ability will be improved by using Eq. (24) 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.

![Figure 9. Dynamics of recognition using different population size.](image)

![Figure 10. Motion-Feedforward recognition system.](image)
4. SIMULATION EXPERIMENT OF RECOGNITION

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

The simulation experiment is performed under software "OpenGL" which is an applications program interface for defining 2D and 3D objects. With OpenGL, an application can create the same effects in any operating system. Here, we create a manipulator which is the same as an actual 7-link manipulator namely “PA-10” robot of Mitsubishi Heavy Industries, LTD., shown in Figure 11. Two cameras are mounted on the robot end-effector. The frame frequency is about 5.5[fps], that is the newly images are input every 0.18[s], during each time “1-Step GA” is executed. The calculation of the motion-feedforward compensation for each generation costs less than 0.01[s] which is quite smaller than the cost of one generation of GA about 0.14[s].

Simulation experiments

Two kinds of motion is given to the robot end-effector while recognizing the target object's 3D pose. We compare the recognition results using the proposed motion-feedforward recognition method with that without using this method under these two robots' motions respectively. To evaluate the effectiveness of the proposed motion-feedforward recognition method, in this paper, the parameters of GA will be fixed to give the same search ability, which had been determined through our previous experiment that could satisfy fastest convergence.

(1) Recognition under translational motion (shown in Figure 12(a))

In this case, the shuttle motion in y axis of $\Sigma_w$ is given to the robot end-effector with period $T = 20[\text{s}]$. The initial pose of the end-effector is shown in Figure 11 (b) defined as $\varphi_0 = [x_0, y_0, z_0, \varphi_0]^T$, $Q_0 = [e_1, e_2, e_3, e_4, e_5, e_6]^T$. The desired motion track is given as

$$y_d = y_0 + 0.1 \sin \left( \frac{2\pi}{T} t \right),$$

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

Figure 13 shows the simulation result of recognition under position shuttle motion of end-effector without feedforward recognition method. Figure 13(a) shows the fitness value. Figure 13(b) shows the recognition result of position $x$, $y$, $z$ compared with the desired position (in $\Sigma_{CR}$), where the desired position is depicted in white lines. Figure 13(c) shows recognition result of orientation $e_1, e_2, e_3$ compared with the desired position (in $\Sigma_{CR}$). In the same way, the desired orientation is depicted in white lines. 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 Figure 13, Figure 14 shows the simulation results with motion-feedforward recognition method under the same motion of the robot end-effector. It shows that the simulation results in Figure 14 always overlap the real 3D pose which verifies that the motion-feedforward method works well.

(2) Recognition under rotational motion (shown in Figure 12 (b))

Here, the orientation changing of end-effector is defined as the motion on a circle with a fixed distance “d” to the target and with keeping the eye-line (z axis of $\Sigma_{CR}$) passes the center of the
The GA became converged at about \( \eta_1, \eta_2, \eta_3 \) compared with the desired position (in \( \sum_{CR} \)).

Discussions

In the end of section 2, we have mentioned that to predict the target's 3D pose, not only the joint information of manipulator, but also the current target's 3D pose in \( \sum_{CR} \) are required. The above simulation has confirmed that good tracking can be achieved with accurate prediction when the initial target's pose is known. However, in most cases, the robot is expected to find the target by itself. In our work, the matching solid model and GA is used to find the target. So the recognition accuracy depends on whether the defined solid model can give a good fitness function profile with a sharp peak. And the recognition speed relies on the convergence speed of GA.

Figure 17 is an example of the recognition result under translational motion with feedforward recognition method when the initial target's pose is unknown. The period of the end-effector's motion is \( T = 20[s] \). The GA became converged at about \( t = 7[s] \), cost 40 generations. The position errors \( (\mathbf{r}_d - \hat{\mathbf{r}}) \) are less than 1[mm], and orientation errors \( (\mathbf{\eta}_d - \hat{\mathbf{\eta}}) \) are less than 0.03 (about 3[deg]). In this
The recognition result with feedforward recognition method works well on keeping the model always matched the target. In the same way, Figure 18 is the recognition result with feedforward recognition method under rotational motion. The GA became converged at about $t = 19\text{s}$, cost 107 generations. The recognition error of position is less than $2\text{mm}$, and orientation error is less than $0.04$ (about $4\text{deg}$). In this case, the feedforward recognition method is also confirmed effective to keep the model always match the target. From Figure 17 and 18, we can conclude that the feedforward recognition possesses such a performance: good recognition results lead to good prediction, however, bad recognition results (when the GA is not converged) do not affect GA to get convergence.

In this paper, we supposed the target is static, and just discussed how to predict the target's pose with respect to camera using the joint information of the manipulator. Of course, the target's real motion in $\sum_B$ will also cause the target's pose changing in $\sum_G$, the relationship is expressed by $J_\theta(q)$, the other part of Eq. 12, which is not discussed here. In future work, such prediction will be completed, and the recognition will be robust to both the real motion of the target and the relative motion with respect to the camera's.

5. CONCLUSIONS

We have proposed a 3D pose measurement method which utilizes an evolutionary recognition technique of GA and a fitness evaluation based on a 3D matching solid model whose pose is expressed by unit quaternion. A proposed motion-feedforward compensation method has been confirmed that can improve the dynamics of recognition. Simulation results have 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 fictitious motion just comes from the camera.

ACKNOWLEDGEMENTS

This is a product of research which was financially supported by the Kansai University Grant-in-Aid for the Faculty Joint Research Program, Feasibility Study Program of Fukui prefecture, Key Research Program in University of Fukui and Incubation Laboratory Factory of University of Fukui.
Figure 17. Simulation under translational motion with feedforward recognition method when the initial target's pose is unknown. (a), (b), (c) is the same meaning as that in Figure 13.

Figure 18. Simulation under rotational motion with feedforward recognition method when the initial target's pose is unknown. (a), (b), (c) is the same meaning as that in Figure 13.

REFERENCES


AUTHOR INFORMATION

Wei Song received her B.E. degree in electronic communication engineering from Beijing Industry of Machinery University, Beijing, China, in 2003. She received her M.E. degree in Graduate School of Engineering from University of Fukui, Fukui, Japan, in 2006. Now she is a doctor student in Graduate School of Engineering of University of Fukui, Fukui, Japan. Her current research interests are dynamic-image recognition, visual servoing.

Mamoru Minami completed the M.E. program in aeronautical engineering at University of Osaka Prefecture in 1981, and the course in natural science in 1993 at the Graduate School of Kanazawa University, receiving a D.Eng. degree. He became an associate professor in 1994 in the Department of Mechanical Engineering, and a professor in 2002 in the Department of Human and Artificial Intelligence Systems, Faculty of Engineering, University of Fukui. He is engaged in research on the control of mobile manipulators, dynamic image recognition, visual servoing, and force control. He is a member of the Japan Society of Mechanical Engineers, the Robotics Society of Japan, SICE, and IEEE.

Seiji Aoyagi received his BE, ME, and PhD degrees in precision machinery engineering from the University of Tokyo, Tokyo, Japan, in 1986, 1988, and 1994, respectively. From 1988 to 1995, he was with the Mechanical System Engineering Department at Kanazawa University, Kanazawa, Japan as a research associate and an associate professor. He is currently a full professor in the Mechanical Engineering Department at Kansai University, Osaka, Japan. He was a visiting researcher at California Institute of Technology (Caltech) from 2002 to 2003. His current research interests are robotics, mechatronics, MEMS, with an emphasis on sensors, and actuators for micro robotics.