Proceedings of the 2008 IEEE/ASME
International Conference on Advanced Intelligent Mechatronics
July 2 - 5, 2008, Xi'an, China

On-line Motion-Feedforward Pose Recognition Invariant for Dynamic Hand-eye motion

Wei Song and Mamoru Minami

***Abstract*— This paper presents a pose measurement method of a 3D object detected by hand-eye cameras. We propose a motion-feedforward (MFF) method to improve visual recognition dynamics, which become worse by disturbing hand-eye motion during visual servoing of the robot manipulator. The MFF method is used to compensate the fictional target motions in the camera view induced by the end-effctor's motion, so the pose recognition can be assumed invariant for dynamic hand-eye motion. The effectiveness of the proposed method is confirmed by simulations of object's 3D pose recognition being affected by dynamical oscillations of hand-eye cameras.**

I. Introduction

Tasks in which visual information are used to direct a manipulator toward a target object are referred to visual servoing in [1],[2]. 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.

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. 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 [3] 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 [4] 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 [5] 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.

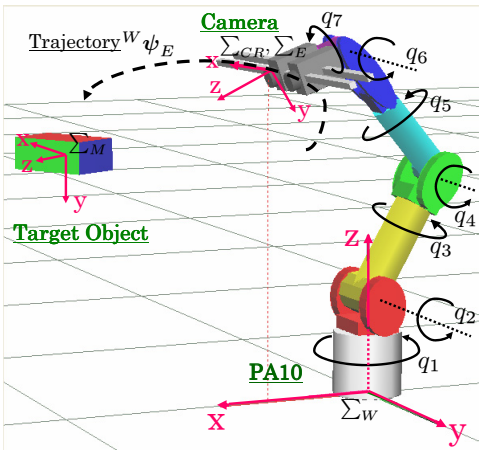

Fig. 1. Visual servo system of PA-10

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" [6]. 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 cause multi-solution for a given orientation that is difficult

Wei Song is with Graduate school of Engineering, University of Fukui, Fukui, 910–8507, Japan songwei@rc.his.fukui-u.ac.jp
Mamoru Minami is with Faculty of Engineering, University of Fukui, Fukui, 910–8507, Japan minami@rc.his.fukui-u.ac.jp

978-1-4244-2495-5/08/$25.00 © 2008 IEEE.

for GA to converge.

Our final objective of this research is to perform visual servoing to a moving target in 3D space by using a hand-eye robot. We consider it is important for the robot to distinguish the target motion coming from the hand-eye cameras and the target motion of itself in real world. And we describe such a relationship by a mathematical function (Eq. (10)), which can distinguish these two motions. If both of the two motions can be predicted, then the target motion in camera will be considered as known, and the recognition will become robust. As a fundamental research step, in this paper, we will confirm the effectiveness of MFF method to compensate the dynamic motion of the robot end-effector, so we use a static target object. Prediction of a moving target in real world will not be discussed here, but will be considered in future research.

II. 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 expressed by unit quaternion. Please refer to [9] for a detailed explanation of 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 (A.8) 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:

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

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

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

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

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

(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 III-C.

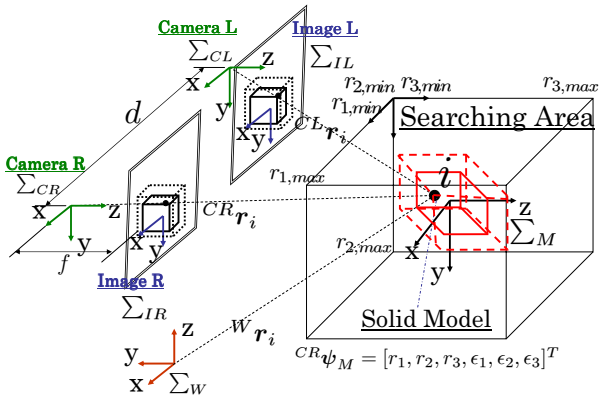

Fig. 2. Coordinate systems

III. 3D Measurement Method

In this paper, we take a rectangular solid block whose size is known as an example of the target object. However, other different kinds of shape targets can also be measured by model-based matching strategy if their size is given, for example, in [6] a model of fish is used to recognize fish in real time, and in [7] a model of human face is used for face detection.

A. 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,

$$ {}^{CR}\boldsymbol{r}_i = {}^{CR}\boldsymbol{T}_M \, {}^{M}\boldsymbol{r}_i. \tag{13}$$

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,

$$ {}^{W}\boldsymbol{r}_i = {}^{W}\boldsymbol{T}_{CR} \, {}^{CR}\boldsymbol{r}_i. \tag{14}$$

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,

$$ {}^{IR}\boldsymbol{r}_i = \boldsymbol{P} \, {}^{CR}\boldsymbol{r}_i. \tag{15}$$

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,

$$ {}^{CL}\boldsymbol{r}_i = {}^{CL}\boldsymbol{T}_{CR} \, {}^{CR}\boldsymbol{r}_i. \tag{16}$$

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

$$ {}^{IL}\boldsymbol{r}_i = \boldsymbol{P} \, {}^{CL}\boldsymbol{r}_i \tag{17}$$

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,

$$ \begin{cases} {}^{IR}\boldsymbol{r}_i = \boldsymbol{f}_R({}^{CR}\boldsymbol{\psi}_M, \ {}^{M}\boldsymbol{r}_i) \\ {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L({}^{CR}\boldsymbol{\psi}_M, \ {}^{M}\boldsymbol{r}_i). \end{cases} \tag{18}$$

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 section III B.

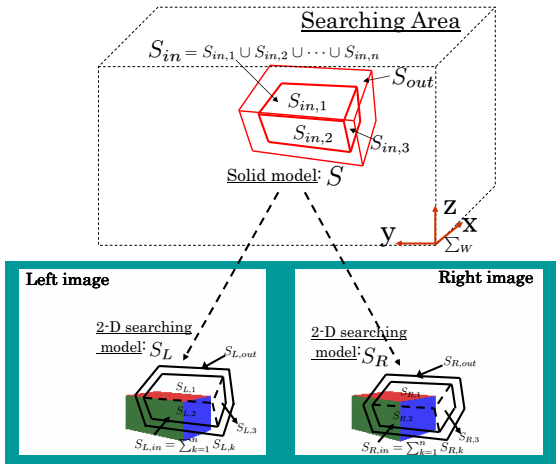

Fig. 3. Definition of a solid model and left/right searching models

B. 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

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

$$ S_{L,out}({}^{CR}\boldsymbol{\psi}_M) = \{ {}^{IL}\boldsymbol{r}_i \in \Re^2 \mid {}^{IL}\boldsymbol{r}_i = \boldsymbol{f}_L({}^{CR}\boldsymbol{\psi}_M, {}^{M}\boldsymbol{r}_i), {}^{M}\boldsymbol{r}_i \in S_{out} \in \Re^3 \} \tag{20}$$

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

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

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

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

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, which will be explained in the next section. The genes of GA representing possible pose solution ${}^{CR}\boldsymbol{\psi}_{GA}$ is defined as,

$$\underbrace{01\cdots01}_{12bit}\underbrace{00\cdots01}_{12bit}\underbrace{11\cdots01}_{12bit}\underbrace{01\cdots01}_{12bit}\underbrace{01\cdots11}_{12bit}\underbrace{01\cdots10}_{12bit}.$$
(t_x, t_y, t_z, ϵ_1, ϵ_2, ϵ_3)

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 $-60 \leq roll, pitch, yaw \leq 60[deg]$.

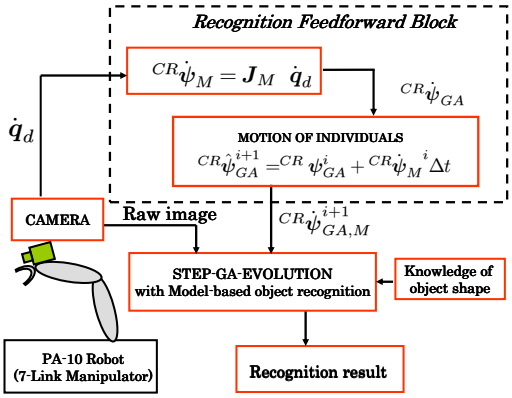

Fig. 4. Feedforward recognition system

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

$${}^{CR}\hat{\boldsymbol{\psi}}_{GA}^{i+1} = {}^{CR}\boldsymbol{\psi}_{GA}^{i} + {}^{CR}\dot{\boldsymbol{\psi}}_M{}^{i}\Delta t, \tag{23}$$

where i represents the number of the generation.

The recognition system of the proposed method is shown in Fig. 4. 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.

C. On-line Evolutionary Recognition

Here, we use $\boldsymbol{\psi}$ instead of ${}^{CR}\boldsymbol{\psi}_M$ for a clear explanation. In fact we cannot always guarantee the highest peek $\boldsymbol{\psi}^{max}(t)$

$$\boldsymbol{\psi}^{max}(t) = \big\{\boldsymbol{\psi}(t) \;\big|\; \max_{\boldsymbol{\psi}\in L} F(\boldsymbol{\psi}(t))\big\}, \tag{24}$$

in the filtered image coinsides with the actual pose of the target object $\boldsymbol{\psi}^{Obj}(t)$ in raw images, however we can set such an environment and elaborated function $F(\boldsymbol{\psi}(t))$ related to matching model so that $\boldsymbol{\psi}^{max}(t)$ should correspond to $\boldsymbol{\psi}^{Fish}(t)$). Then the problem of recognition of a target object is converted to an optimization problem of $F(\boldsymbol{\psi}(t))$.

$F(\boldsymbol{\psi}(t))$ is used as a fitness function in Genetic Algorithm (GA) to find $\boldsymbol{\psi}^{max}(t)$ through i-th gene $(i = 1, 2, \cdots, p)$ of j-th generation, $\boldsymbol{\psi}_i^j(j\Delta t)$,

$$\boldsymbol{\psi}_j^{max}(t) = \big\{\boldsymbol{\psi}_{i,j}(t) \;\big|\; \max_{i} F(\boldsymbol{\psi}_{i,j}(t))\big\}. \tag{25}$$

Assuming the time " t " be fixed to the current time the problem to find $\boldsymbol{\psi}^{max}(t)$ in the current input image through GA process is described as

$$\boldsymbol{\psi}_j^{max}(t) \overset{\text{evolve}}{\longrightarrow} \boldsymbol{\psi}^{max}(t), \quad (j \to \infty), \tag{26}$$

where the above convergence to target pose $\boldsymbol{\psi}^{max}(t)$ can be thought generally to be guranteed with appropriate parameter setting.

To recognize a target in a dynamic image input by video rate, 33 [fps], the recognition system must have real-time nature, that is, the searching model must converge to the

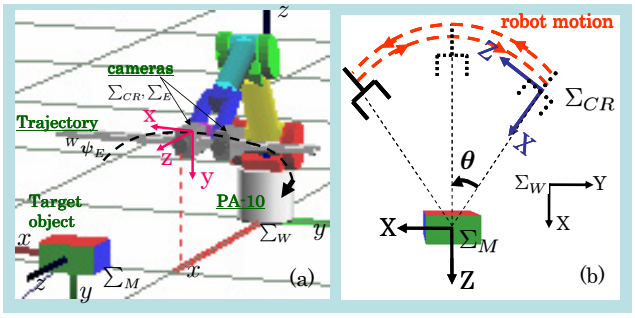

Fig. 5. (a)Simulation experiment system created by OpenGL. (b) coordinate systems of simulation experiment

fish in the successively input raw images. To give the GA process the real-time nature, the evolving process

$$\boldsymbol{\psi}_{i,j}(t) \overset{\text{evolve}}{\longrightarrow} \boldsymbol{\psi}_{i,j}(t+\Delta t) \tag{27}$$

is executed only one time within the time interval of Δt. We named it as "1-step GA". Should the converging speed of the model to the target in the dynamic images be faster than the moving speed of the target, then the pose indicated by the highest gene represents the target's actual pose at the time. The on-line measurement of target's pose by 1-step GA optimization can be written as

$$|\boldsymbol{\psi}^{max}(t) - \boldsymbol{\psi}_j^{max}(t)| < \epsilon, \tag{28}$$

where ϵ represents tolerable extent as a observing error. We have confirmed that the above time-variant optimization problem to solve $\boldsymbol{\psi}_{max}(t)$ maximizing $F(\boldsymbol{\psi}(t))$ could be solved by 1-step GA through several experiments [6].

IV. Recognition Simulation

To verify the effectiveness of the proposed MFF recognition method, we conduct the simulation experiments to recognize a rectangular solid block($100mm \times 150mm \times 200mm$) with symmetrical colored surfaces, shown in Fig. 5(a). We compare the recognition using "1-step GA " with that using "1-step GA + MFF" under a given trajectory of the end-effector. To see clearly the result of proposed method, here, we keep the target object static, so the motion in the camera view is just from the motion of the end-effctor.

A. simulation condition

The simulation experiment is performed using software "Open GL". Here, a manipulator modeling the actual 7-link "PA-10" is used, and two cameras are mounted on its end-effector, shown in Fig. 5.

A trajectory of end-effector is given as 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 initial hand pose is defined as v and the homogeneous transformation matrix from Σ_{E_0} to Σ_W is

$${}^W\boldsymbol{T}_{E_0} = \begin{bmatrix} 0 & 0 & 1 & 918 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 455 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{29}$$

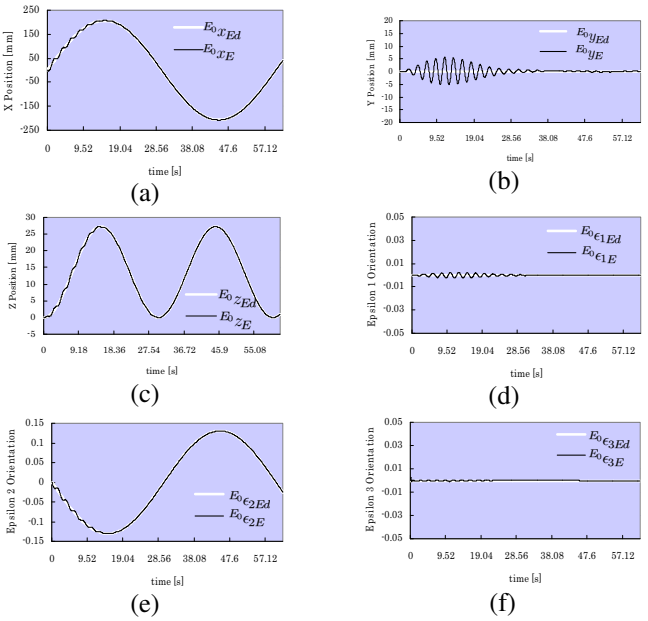

Fig. 6. Desired hand trajectory ${}^{E_0}\boldsymbol{\psi}_{Ed}(t)$ and the actual trajectory ${}^{E_0}\boldsymbol{\psi}_E(t)$ with dynamics, $\omega = 0.105[rad/s]$.

The desired hand trajectory expressed in Σ_{E_0} is

$$\begin{cases} {}^{E_0}x_{Ed}(t) = d * sin\theta_d(t) \\ {}^{E_0}y_{Ed}(t) = 0 \\ {}^{E_0}z_{Ed}(t) = d - d * cos\theta_d(t) \\ {}^{E_0}\epsilon_{1Ed}(t) = 0 \\ {}^{E_0}\epsilon_{2Ed}(t) = sin\frac{\theta_d(t)}{2} \\ {}^{E_0}\epsilon_{3Ed}(t) = 0 \end{cases} \tag{30}$$

where $d = 800[mm]$, $\theta_d(t) = 15sin(\omega t)[deg]$, ω represents the frequency of end-effector's motion. By using (30), (29), the desired hand trajectory can be expressed in Σ_W as

$${}^W\boldsymbol{T}_{Ed}(t) = {}^W\boldsymbol{T}_{E_0}\ {}^{E_0}\boldsymbol{T}_{Ed}(t). \tag{31}$$

The desired hand pose and the actual hand pose in Σ_{E_0} obtained by PD control are shown in Fig.6 when ($\omega = 0.105[rad/s]$). It can be found that there is oscillation in the actual hand trajectory at the first 30 seconds due to the dynamics of the manipulator, which would make the recognition of the object more difficult because the hand-eye camera is moving together with the manipulator. The effectiveness of MFF method that is proposed to solve this kind of problem will be evaluated in the following simulation, in which the target object is assumed to be static in Σ_W, set as ${}^{CR}\boldsymbol{\psi}_M(0) = [0, 155[mm], 800[mm], 0, 0, 0]^T$.

B. simulation result

We compare the methods of "1-step GA " and "1-step GA + MFF" by changing the moving speed of end-effector, that is, changing ω as $\omega = 0.105[rad/s], 0.157[rad/s], 0.314[rad/s]$ (corresponding to the periods $T = 60[s], 40[s], 20[s]$).

Fig. 7 shows comparison of the methods "1-step GA " and "1-step GA + MFF" under $\omega = 0.105[rad/s]$. The true values of the 3-D pose of the target object in Σ_{CR} are represented by ${}^{CR}\boldsymbol{\psi}_M$. The recognition results using

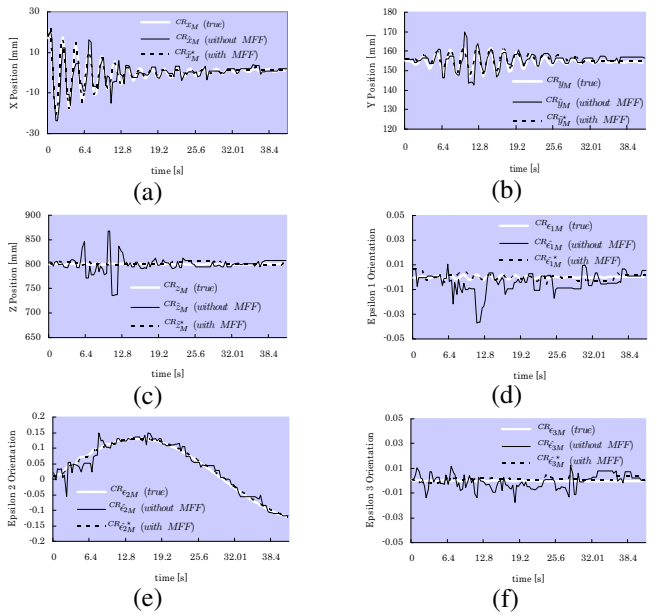

Fig. 7. Comparison of the recognition by "1-step GA " ${}^{CR}\hat{\boldsymbol{\psi}}_M$ and by " MFF + 1-step GA" ${}^{CR}\hat{\boldsymbol{\psi}}^{\star}{}_M$ under $\omega = 0.105[rad/s]$.

only "1-step GA " without MFF method are represented by ${}^{CR}\hat{\boldsymbol{\psi}}_M$, and the recognition results using both "1-step GA " and MFF are represented by ${}^{CR}\hat{\boldsymbol{\psi}}^{\star}{}_M$. Due to the dynamics of the manipulator, the target object can not move smoothly in the images. Fig. 7(a), (b) show the obvious unstable motion during the first 30 seconds. Such oscillations surely bring difficulty to object recognition and the oscillations are increasing when the speed of the end-effctor is faster (see later simulations).

Compare the recognition results using "1-step GA " and "1-step GA + MFF" in Fig. 7, the "1-Step GA" can not recognize precisely, expecially in the oscillation time. On the other hand, "1-step GA + MFF" gives more correct result even in the oscillation time. The error between ${}^{CR}\boldsymbol{\psi}_M$ and ${}^{CR}\hat{\boldsymbol{\psi}}^{\star}{}_M$ using "1-step GA + MFF" is smaller than that of using "1-Step GA", as shown in the first rank of Table 8. It has been confirmed that MFF method can compensate the fictional target motions in the camera view induced by the end-effctor's motion, the recognition became robust to the dynamics of the manipulator. In other words, recognition by hand-eye cameras can be independent of the manipulator, and robust recognition can be obtained the same as using fixed cameras by MFF.

The figures of simulations for comparisons of the methods "1-step GA " and "1-step GA + MFF" under $\omega = 0.157[rad/s]$ and $\omega = 0.314[rad/s]$ are not shown here. But the evaluated results are shown in Fig. 8. We can see that using only "1-step GA ", the mean value of the fitness function, defined as $\bar{F}$, from 0s to 40s gets lower, and the root-mean-square value of the error of position/orientation, defined as $\Delta^{CR}\tilde{\boldsymbol{\psi}}_M$, gets bigger (to about 33[mm],12[deg]) along with ω changing from 0.105[rad/s] to 0.314[rad/s]. By "1-step GA + MFF", the end-effector's motion has been compensated completely, even the end-effector moves faster and faster, both $\bar{F}$ and $\Delta^{CR}\tilde{\boldsymbol{\psi}}_M$ are not changed much (about

$\bar{F}, \Delta^{CR}\tilde{\boldsymbol{\psi}}_M$ \ ω [rad/s]		$\omega = 0.105$	$\omega = 0.157$	$\omega = 0.314$
1-step GA	$\bar{F}$	**0.8026**	**0.7371**	**0.6280**
	$\Delta^{CR}\tilde{x}_M$ [mm]	3.1742	3.4841	7.6398
	$\Delta^{CR}\tilde{y}_M$ [mm]	3.5439	4.4271	21.7151
	$\Delta^{CR}\tilde{z}_M$ [mm]	14.9828	17.1283	33.8106
	$\Delta^{CR}\tilde{\epsilon}_{1M}$	0.0188	0.0251	0.0494
	$\Delta^{CR}\tilde{\epsilon}_{2M}$	0.0203	0.0323	0.0656
	$\Delta^{CR}\tilde{\epsilon}_{3M}$	0.0053	0.0072	0.0091
MFF + 1-step GA	$\bar{F}$	**0.9806**	**0.9764**	**0.9733**
	$\Delta^{CR}\tilde{x}_M$ [mm]	0.6235	1.49	1.5601
	$\Delta^{CR}\tilde{y}_M$ [mm]	1.16	1.712	1.7493
	$\Delta^{CR}\tilde{z}_M$ [mm]	2.8536	5.5322	5.281
	$\Delta^{CR}\tilde{\epsilon}_{1M}$	0.0075	0.008	0.0086
	$\Delta^{CR}\tilde{\epsilon}_{2M}$	0.0091	0.0092	0.0098
	$\Delta^{CR}\tilde{\epsilon}_{3M}$	0.0022	0.002	0.0029

Fig. 8. Conclusion of simulations

5[mm],2[deg]). It is confirmed that by using MFF recognition method, the recognition in a dynamic hand-eye system is invariant, just like using a fixed camera system.

V. 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 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 recognize the pose of a target object being affected by dynamical oscillations of hand-eye cameras. Some experimental validations of the proposed method will be shown in the conference presentation, including the real-time visual servoing to a moving 3D moving target.

References

[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] 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.

[3] 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

[4] 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).

[5] 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

[6] 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.

[7] W. Song, M. Minami, Y. Mae and S. Aoyagi, "On-line Evolutionary Head Pose Measurement by Feedforward Stereo Model Matching", IEEE Int. Conf. on Robotics and Automation (ICRA), pp.4394-4400, 2007.

[8] 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

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