Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

A Visual Servoing Algorithm using Fuzzy Logics and Fuzzy-Neural Networks

Il Hong Suh

Intelligent Control and Robotics Lab., Dept. of Electronics Eng., Hanyang Univ., 396 Daehak-Dong, Ansan-Si, Kyungki-Do 425-791, KOREA

and

Tae Won Kim

Mechatronics Team, Precision Instruments R&D Center, Samsung Aerospace Industries, Ltd., 145-3, Sangdaewon 1-Dong, Jungwon-Gu, Sungnam-City, Kyungki-Do 462-121, KOREA

ABSTRACT

A visual servoing algorithm is proposed for a robot with a camera in hand, where fuzzy logics and fuzzy-neural networks are employed to represent and/or learn camera motion commands to track a moving object in terms of image features and their variations. Specifically, novel image features are suggested by employing a viewing model of perspective projection to estimate relative pitching and yawing angles between the object and the camera. And, owing to the uniqueness of the proposed image features, at most two input variables are shown to be sufficient for the design of fuzzy logics and/or fuzzy-neural networks. To compensate dynamic characteristics of the robot, desired feature trajectories for the learning of visually guided line-of-sight robot motion are obtained by measuring features by the camera in hand not in the entire workspace, but on a single linear path along which the robot moves under the control of a commercially provided function of linear motion. And then, control actions of the camera are approximately found by fuzzy-neural networks to follow such desired feature trajectories.

To show the validity of proposed algorithm, some experimental results are illustrated, where a four axis SCARA robot with a B/W CCD camera is used.

1. Introduction

Recently, visual servoing has been considered as one of powerful tools for intelligent robotic applications. Especially, feature Jacobian has been mainly used for visual servoing. The feature Jacobian can be classified as a pose-based one, if its elements are represented by a relative translation and orientation between the camera and the object. Otherwise, it is classified as a feature-based feature Jacobian. Since the pose based feature Jacobian is a function of the relative pose, it requires the depth information from the camera to the object, which is often difficult to obtain[16, 19, 20]. And, since a correctional motion of the robot end effector with a camera in hand is given by the inverse of the feature Jacobian, computational complexity and the singularity of the feature Jacobian should be considered for real time control of robot[7, 11, 12]. In contrast with the pose-based feature Jacobian, a feature-based feature Jacobian is represented by a function of features. In [12], a feature Jacobian was partially described by features. But, it may also suffer from real-time calculation of the inverse of the feature Jacobian at every visual sampling time. And all of six dimensional motions of robot end effector have not been related with only features, since it is usually difficult to analytically obtain all such relations without couplings. Several approaches using neural networks were proposed to learn pose-based or feature-based feature Jacobian[14, 10, 15]. A good related literature survey can be found in [5, 9].

On the other hand, 6 DOF motions of a robot were directly described in terms of six features and their variations, and has been approximated by a fuzzy-neural network, *fuzzy membership function-based neural network* (FMFNN)[13, 18] to avoid the use of the inverse of Jacobian. However, since the robot dynamics was not considered, the robot had to move slowly in practical applications. And, orientational motions of the camera was not completely considered.

In this paper, a different type of visual servoing approach from the ones in [1, 3, 17] is proposed, where two control policies are applied according to the size of the object as in Fig.1 : If the size of the object is measured by a feature to be smaller than a pre-specified size, then the camera is controlled to move to the object while keeping gaze holding along the line of sight. Otherwise, the camera is controlled to move along the linear path from the current position to a target position in front of the object, while trying to have a given desired relative orientation between the camera and the object. it is remarked that our proposed visual servoing approach seems to be similar to visual servoing schemes of human or animals. Specifically, for the latter case, FMFNN in [13, 18] is employed, where number of fuzzy rules is shown to be reduced by employing novel features based on a viewing model of perspective projection. It is also remarked that proposed novel features satisfy the image recognition criteria for visual servoing in [8] and can be easily implemented for

This work was supported in part by ERC-ACI of SNU by KOSEF.

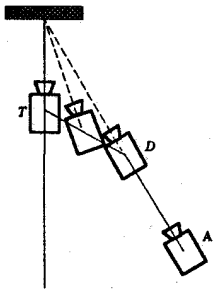


Fig. 1. Schematic diagram of camera motion by employing the proposed visual servoing method

practical applications.

And, to incorporate dynamic characteristics of the robot, desired feature trajectories for the learning of visually guided line-of-sight robot motion are here obtained by measuring features by the camera in hand not in the entire workspace, but on a single linear path along which the robot moves under the control of a commercially provided function of linear motion. And then, control actions of the camera are approximately found by FMFNN to follow such desired feature trajectories. On the other hand, since gaze motions are designed for the alignment of the line-of-sight of the camera with the center of the object regardless of the orientation of the camera with respect to the object[2, 4], it is necessary for the orientation of the camera to be adjusted to become a given desired orientation with respect to the object for correct grasping of the object. However, in our approach, such an orientational motion control of the robot end-effector is applied only near the target position. When an orientational motion control as well as a gaze control are simultaneously performed, a line of sight of the camera does not usually coincide with the center of the object, and thus the camera will not move along the desired linear path. Thus, corrective motions on the plane perpendicular to the approaching direction of the robot end effector should be considered. Such corrective motion control algorithms are proposed by fuzzily estimating current position of the camera with respect to the object, where a perspective-model based features are used.

To show the validity of our proposed algorithms, some experimental results are illustrated, where a four axis SCARA robot with a CCD camera is utilized. It is remarked that the camera is mounted on the rolling axis (S-axis) of the robot in such a way that the line of sight of the camera lies in X-Y plane.

2. Image Features for the Design of Visual Servoing Algorithms

In selecting image features for a visual feedback control, image selection criteria including unique features, feature set robustness, computational inexpensive features, and feature set completeness have been proposed in [8]. Among them, uniqueness should be strongly considered, since it gives effects on the computational complexity of the visual feedback control algorithms. However, it is usually difficult to find an unique feature because camera motion along the i-th axis of the camera frame usually cause not only the i-th feature, F_i, but also other features, $F_j (j \neq i)$, to be changed. To cope with such a difficulty, a viewing model of perspective projection in [6] is here employed.

Specifically, consider a quadrangle (specially, a rectangle for the case of the reference image) in the image as shown in Fig.2. As shown in [11], [13], [3], and [5], features obtained from four points or a quadrangle in the image plane could be generally applicable to real tasks. Hereinafter, an object will be considered as a quadrangle. Let (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) be the corner points of the rectangle in the image plane as shown in Fig.2. Then, features, F_i, for $i = 1, 2, \ldots, 6$, are chosen as

$$\begin{aligned} F_1 &= (x_1 + x_2 + x_3 + x_4)/4, \\ F_2 &= (y_1 + y_2 + y_3 + y_4)/4, \\ F_3 &= (x_2 + x_4 - x_1 - x_3) \bullet (y_2 + y_4 - y_1 - y_3), \\ F_4 &= (m_1 x_1 - m_2 x_3 + y_3 - y_1)/(m_1 - m_2) - F_1, \\ F_5 &= (m_3 [m_4 (x_1 - x_2) + y_2 - y_1]) / (m_3 - m_4) + y_1 - F_2, \end{aligned} \tag{1}$$

and

$$F_6 = \tan^{-1}([y_1 + y_2 - y_3 - y_4] / [x_1 + x_2 - x_3 - x_4]).$$

In (1), m_1, m_2, m_3, and m_4 are given as $(y_2-y_1)/(x_2-x_1)$, $(y_4-y_3)/(x_4-x_3)$, $(x_1-x_3)/(y_1-y_3)$, and $(x_2-x_4)/(y_2-y_4)$, respectively.

It is remarked that F_1 and F_2, respectively, are X and Y coordinates of the center of gravity of the quadrangle in the image plane. F_3 is the area of the quadrangle in the image plane. Let $l_{ij}(x,y) = 0$ be the line connecting (x_i, y_i) with (x_j, y_j). F_4 implies the difference between F_1 and X coordinate of the point of intersection of two lines $l_{12}(x,y) = 0$ and $l_{34}(x,y) = 0$ as shown in Fig.2. And, F_5 is the difference between F_2 and Y coordinate of the point of intersection of two lines $l_{13}(x,y) = 0$ and $l_{24}(x,y) = 0$. F_6 is vector

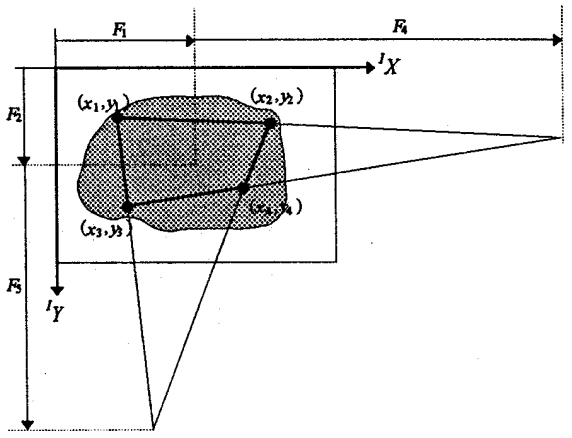


Fig. 2. A quadrangle extracted from simple 4 dot pattern in the image plane

of the image plane. It is also remarked that F_1, F_2 the degree of rotation of the quadrangle about the normal and F_3, respectively, are used for translational motions of the camera along the ${}^cX$, ${}^cY$ and ${}^cZ$ axis. And F_4, F_5 and F_6, respectively, are used to determine magnitudes of a pitching, a yawing, and a rolling motion of the camera about ${}^cX$, ${}^cY$ and ${}^cZ$ axis[18].

Now, consider a viewing model of perspective projection of a regular hexahedron as shown in Fig.3 to know how relative pitching and yawing angles between the object and the camera affect image features. Let a quadrangle with corner points C_1, C_2, C_3 and C_4 be denoted as $Q(C_1, C_2, C_3, C_4)$. And let $\mathcal{R}(R_1, R_2, R_3 R_4)$ in Fig.2 be the image of $Q(C_1, C_2, C_3, C_4)$ on the object to be visually tracked, and R_{vp} be the viewpoint of the camera which is transformed from the camera frame to the image plane. Also let the plane including three *vanishing points*, V_1, V_2 and V_3, be represented as $\wp(V_1, V_2, V_3) = 0$. Here, note that $\wp(V_1, V_2, V_3) = 0$ is orthogonal to the line of sight of the camera[6][13]. Assume that the viewpoint of the camera coincides with the center of $Q(R_1, R_2, R_3 R_4)$. Then, the angle α between $\wp(V_1, V_2, V_3)$ and the line connecting V_2 with R_1, $\overline{V_2R_1}$, which is called as a *vanishing line*, becomes a pitching angle of the camera with respect to the object. And the angle β between $\wp(V_1, V_2, V_3)$ and $\overline{V_3R_1}$ becomes a relative yawing angle. It is here remarked that the angles α and β, respectively, can be simply represented by the difference of the X coordinates of R_{vp} and V_2 and the difference of the Y coordinates of R_{vp} and V_3[6]. It is also remarked that the distance between R_{vp} and V_2 or V_3 is not varied even if the viewing scale is changed as shown in Fig.4. Thus, to obtain relative pitching and yawing angles, α and β, R_{vp}, V_2 and V_3 should be found. Here, transformed R_{vp} in the image plane can be easily known, since the origin of the camera frame is always mapped onto a fixed point in the image plane. And, V_2 and V_3 can be computed by using the intersection of $\overline{R_1R_2}$ and $\overline{R_3R_4}$, and $\overline{R_1R_4}$ and $\overline{R_2R_3}$, respectively. However, V_2 and V_3 are not linearly proportional to the angle α and β, respectively. Thus, fuzzy rules are here designed to represent the relations between the angle α and V_2, and the angle β and V_3. It is remarked that V_2 and V_3 in Fig.3 are the same as F_4 and F_5, respectively. For this, the line of sight of the camera is made to be aligned to the center of the object, and the camera is in front of the object as shown in Fig.4. To get relations between F_4 and the relative yawing angle of the camera with respect to the object, the object is rotated in a clockwise or counter-clockwise direction about X axis of the object frame, ${}^oX_1$, from $-\beta_{max}$ to β_{max} by an increment of β_{step}, while computing F_4 at every yawing angle. Then, fuzzy rules can be given to represent the relations between β and F_4 as follows;

$$\text{If } F_4 \text{ is NEAR } \mathfrak{I}_4^i, \text{ then } \beta \text{ is NEAR } \Omega_y^i, \quad \text{for } i = 1, 2, \ldots, n, \tag{2}$$

where n is the number of fuzzy rules given as $(2\beta_{max} / \beta_{step} + 1)$, NEAR $\mathfrak{I}_4^i$ is the i-th linguistic value of F_4, and NEAR Ω_y^i is the i-th linguistic value of β.

In case of the relative pitching angle, the difference of the Y coordinate between R_{vp} and F_5 is employed. And, since F_6 is linearly proportional to the relative rolling angle, the scale factor is only required to get a real relative rolling angle.

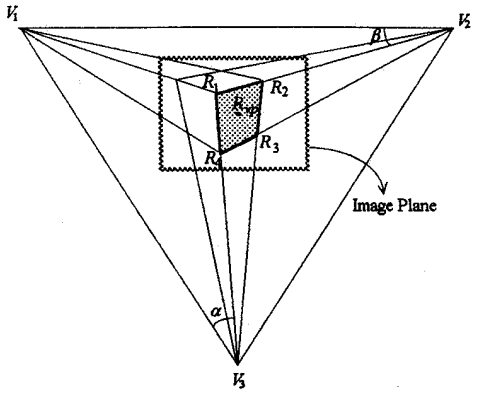


Fig.3. A viewing model of perspective projection of a regular hexadron.

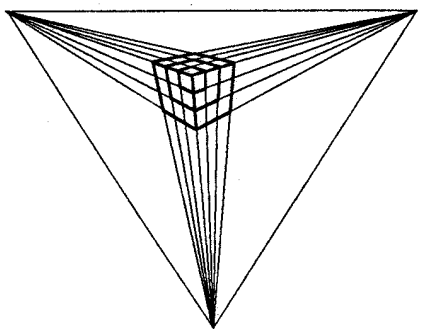


Fig. 4. A schematic diagram of a viewing model of perspective projection to show that all vanishing lines are met into vanishing points even if the viewing scale is changed.

3. Desired Feature Trajectories and Learning of Line-of-sight Motion

Desired feature trajectories should be chosen in such a way that learnings of both the line-of-sight motion of a robot end-effector and correct positioning without oscillations at the target position are guaranteed. For this, consider the case, without loss of generality, that a robot end-effector is made to move along a linear path from a position L_1 to the target position T in the camera frame as shown in Fig.5. Here, such a linear motion is achieved by means of the function of *linear move* which is basically provided in most of commercial industrial robot controllers[18]. Fig.6(a) and (b), respectively, show a typical position trajectory and the velocity profile for such a linear motion of the robot end-effector when the dynamics of the robot is assumed as $6.5/(s+6.5)$. In Fig.6, (t_A, A) and (t_D, D), respectively,

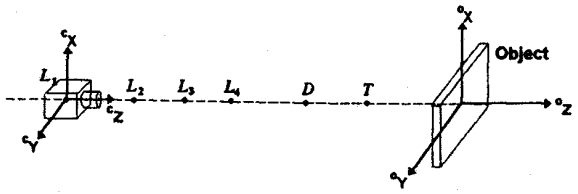
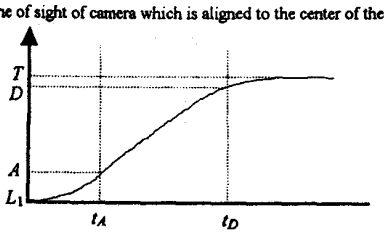
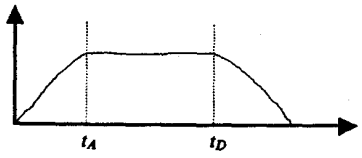


Fig.5. The line of sight of camera which is aligned to the center of the object.



(a) position trajectory of the camera



(b) velocity profile of the camera

Fig.6. A position trajectory and a velocity profile of a camera in hand when the robot moves from L1 to T.

imply the time and the position at which acceleration becomes zero, and deceleration begins to reduce the velocity. To let the end-effector of a robot learn how to follow such a position trajectory or a velocity profile under our feature-based feature Jacobian control algorithm, we will let the robot move with its maximum velocity unless magnitude of feature F_3 is smaller than F_3^D, magnitude of F_3 measured at D. And, reference features to be necessary for the learning of visually guided motion are obtained only along the linear path from D to T at every visual sampling time. We will now call them as a desired feature trajectory, $F_3^R(t)$.

On the other hand, the robot end-effector can begin to move from an arbitrary position L_i, for $i = 2, 3, ..., n$, on the linear path as shown in Fig.5. Thus, corresponding velocity profiles become different from the case that the robot begins to move from L_1. Thus, feature trajectories corresponding to several velocity profiles are obtained along the same linear path from D to T by letting the end effector of the robot begin to move from several different positions. Therefore, robot end effector velocity as well as features are required to identify which feature trajectory should be chosen as follow. Here, variation of feature F_3, δF_3, is to be used instead of robot velocity for such an identification.

Now, let δX_3 be the camera motion along ${}^cZ$ axis during one visual sampling time, and let $G(F_3,\delta F_3)$ be the relationship between $(F_3, \delta F_3)$ and δX_3. To approximately get $G(F_3,\delta F_3)$, a modified version of FMFNN in [13, 18] is used. Specifically, $G(F_3,\delta F_3)$ is approximated by fuzzily combining m functions, $G_i(F_{3i}, \delta F_{3i})$, for $i = 1, 2, ..., m$. For this, an approximated $G_i(F_{3i},\delta F_{3i})$ is initially found by fuzzy rules, where δF_{3i} is assumed, without loss of generality, to be given, and then is iteratively improved by FMFNN. To design such initial m fuzzy rules for the coarse tracking, it is noted that precise control actions need to be applied near the set point to prevent an overshoot of the feature trajectory. Thus, fuzzy values to represent the feature trajectory should be given to be dense near the set point. For the case that δF_3 is similar to δF_i, an example of fuzzy rules can be given as

$$\begin{aligned} &\text{If } F_3 \text{ is } \texttt{Large}, \quad \text{then } \delta^c X_3 \text{ is } \texttt{Small}, \text{ or} \\ &\text{If } F_3 \text{ is } \texttt{Medium}, \text{ then } \delta^c X_3 \text{ is } \texttt{Medium}, \text{ or} \\ &\text{If } F_3 \text{ is } \texttt{Small}, \quad \text{then } \delta^c X_3 \text{ is } \texttt{Large}, \end{aligned} \tag{3}$$

Here, F_3 is used for implicitly representing the distance between the robot and the object. Now, for the fine visual tracking, initial fuzzy rules for coarse tracking are represented as a form of a neural network, FMFNN[13, 18], given by

$$\delta^c X_3 = \sum_{i=1}^{q} \lambda_i \Phi_i(F_3). \tag{4}$$

The basis function of FMFNN, $\Phi_i(\bullet)$, is a triangular membership function of the input fuzzy variable of the i-th fuzzy rule. The weight of FMFNN, λ_i, is the singleton membership function of output fuzzy variable. Now, λ_i is iteratively refined by using reference feature trajectory, $F_3^R(t)$. For this, an error function for an FMFNN is given as

$$E(t) = \frac{1}{2}\left(F_3^R(t) - F_3(t)\right)^2. \tag{5}$$

Thus, as in [13, 18], the learning rule for adapting weight of modified FMFNN is given as follows;

$$\lambda_i(t+1) = \lambda_i(t) + k\left[F_3^R(t) - F_3(t)\right]\mathrm{sgn}\left(\frac{F_3(t) - F_3(t-1)}{\lambda_i(t) - \lambda_i(t-1)}\right), \tag{6}$$

where k is a slope of linear output node function of modified FMFNN.

It is remarked that feature trajectories which are not learned *a priori* can be encountered. Such cases can be handled by using previously learned line-of-sight motions. For this, a simple linear interpolation method is here used, since visual servoing dynamics for line of sight motions are not expected to be much different. That is, δX_3, the motion command for ${}^cZ$ axis of the camera is determined as

$$\delta X_3 = \Psi(F_3)\big|_{\delta F_3} = \frac{\left|\delta F_3^i - \delta F_3\right| G_{i+1}(F_3,\delta F_3) + \left|\delta F_3^{i+1} - \delta F_3\right| G_i(F_3,\delta F_3)}{\left|\delta F_3^i + \delta F_3^{i+1}\right|} \tag{7}$$

when δF_3 is measured as a value between δF_3^i and δF_3^{i+1}.

4. Control of Gaze and Orientational Motions

To endow our visual controller with a gaze holding capability for the case that the line-of-sight of the camera does not coincide with the center of an object, it is necessary to determine how much the camera should be rotated about perpendicular axes with respect to ${}^cZ$ direction along which the camera approaches to the object. For this, without loss of generality, consider a simple 2D configuration of a camera and an object as shown in Fig.7.

Then, by using the geometric relationship between the camera and the object, F_1 in (1) can be represented as

$$F_1 = k_s \frac{f}{{}^cZ_o} {}^cY_o \,, \tag{8}$$

where f and k_s, respectively, denote the focal length of the camera and an image scaling factor and can be known *a priori*. In (8), ${}^cY_o$ and ${}^cZ_o$, respectively, are ${}^cY$ directional and ${}^cZ$ directional positions of the object with respect to the camera frame. Thus, by using (8), the angle g_β for the gaze holding can be obtained using only feature F_1 as

$$g_\beta = \tan^{-1}\left(\frac{{}^cY_o}{{}^cZ_o}\right) = \tan^{-1}\left(\frac{F_1 {}^cZ_o / k_s f}{{}^cZ_o}\right) = \tan^{-1}\left(\frac{F_1}{k_s f}\right). \tag{9}$$

It is here remarked that since dynamics for rotational motions of the camera are relatively faster than those for translational motions of the camera, orientational motions of the camera can be controlled without considering the robot dynamics. To be specific, let $g_{\beta m}$ be the angle of the camera to be maximally rotated during one visual sampling time, which can be determined by a mechanical and electrical specification of the robot to be used. Then, if g_β in (9) is smaller than $g_{\beta m}$, the camera is made to be rotated by the angle of g_β in one visual sampling time. Otherwise, the camera is made to be rotated by the angle of $g_{\beta m}$ until current gaze angle g_β becomes smaller than $g_{\beta m}$.

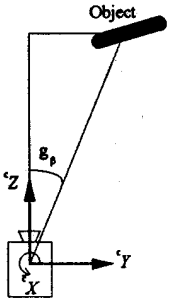


Fig.7. 2D configuration for representing the relative yawing angle of the camera with respect to the object.

It is also remarked that pitching angle of the camera for gaze holding in 3D configuration, g_α, can be determined by

$$g_\alpha = \tan^{-1}\left(\frac{F_2}{k_s f}\right). \tag{10}$$

Recall that in our visual servoing problem, a desired path to be visually followed is not given as an actual line-of-sight path from a position D to the center of the object, but given as the linear path from D to the target position T as shown in Fig.1. Thus, for the visually guided linear motion from D to T, $D({}^oX_D, {}^oY_D, {}^oZ_D)$ is necessary to be estimated. To avoid use of a rather complex and/or expensive distance measuring techniques in [16], a fuzzy-based position-estimation technique is now proposed. Specifically, fuzzy rules are designed by using empirically obtained relational data between F_3 and the actual distance, where relative pitching and yawing angles are fixed to be null. For this, let d_{DT} be the distance between D and T, and let F_3^D and F_3^T be the features measured at D and T, respectively. Also let δF_3^{TD} be defined as $(F_3^T - F_3^D) / n$, where n is an integer implying the number of fuzzy rules to be designed. Then, choose d_{DTi} on a linear path from D to T in such a way that

$$F_{3i} = F_3 \text{ at } d_{DTi} = F_3^D + i \cdot \delta F_3^{TD}, \text{ for i} = 1, 2, \ldots, n. \tag{11}$$

It is remarked that since the position measuring rules to be proposed are designed under the assumption that the relative orientation between the camera and the object is near zero, some errors may be generated if the relative orientation is not zero. That is, F_3 can be decreased due to nonzero yawing and pitching angles. To reduce such errors, consider the 3D configuration of the camera and the object, where if an object whose the size is L is declined to α and β with respect to ${}^cY$ and ${}^cY$, respectively, the length of the object is scaled down as $L \cos\alpha \cos\beta$. Since α and β can be estimated by the fuzzy rules in (2), actual size of the object can be estimated by $\hat{F}_3$ given as

$$\hat{F}_3 = F_3 \cos^{-1}(\alpha) \cos^{-1}(\beta). \tag{12}$$

Then, fuzzy rules for estimating the relative position between the camera and the object can be given as

$$\begin{aligned} &\text{If } \hat{F}_3 \text{ is } \textbf{near } F_{3i}, \\ &\text{then } {}^oZ_c^* \text{ is } \textbf{near } ({}^oZ_T + d_{DTi}), \\ &\qquad \text{for } i = 1, 2, \ldots, n, \end{aligned} \tag{13}$$

where ${}^oZ_c^*$ and ${}^oZ_T$, respectively, are the estimated relative position of the camera and the pre-measured target position with respect to the object. Here, linguistic values **near** F_{3i} and **near** $({}^oZ_T + d_{DTi})$, respectively, are given as triangular and singleton

membership functions.

Now, since T is given *a priori* with respect to the object frame and the current camera position is estimated by employing Eqs.(12) and (13), the unit vector for the line from the current camera position to T, $\vec{u}_c$, can be obtained as

$$\vec{u}_c = \begin{bmatrix} u_{cx} \\ u_{cy} \\ u_{cz} \end{bmatrix} = \frac{1}{\sqrt{\tan^2\alpha + \tan^2\beta + 1}} \begin{bmatrix} \tan\alpha \\ \tan\beta \\ 1 \end{bmatrix}. \tag{14}$$

Then, the translational motion commands, $\delta^c X_1$, $\delta^c X_2$, and $\delta^c X_3$, of the camera along the linear path from D to T during a visual sampling time can be computed by using $\Psi(\hat{F}_3)$ in (7) as

$$\begin{bmatrix} \delta^c X_1 \\ \delta^c X_2 \\ \delta^c X_3 \end{bmatrix} = \Psi(\hat{F}_3)\vec{u}_c . \tag{15}$$

It is remarked that controls of $\delta^c X_1$, $\delta^c X_2$ and $\delta^c X_3$ for camera motions along a linear path from D to T might cause the camera to lose gaze holding. Thus, it is necessary that the orientation of the robot should be readjusted for the gaze holding. For this, a pitching angle, E_α, and a yawing angle, E_β, during the camera motion along a linear path from D to T can be obtained by using ${}^oZ_c^*$, $\delta^c X_1$, $\delta^c X_3$, α and β as follows;

$$E_\alpha = \tan^{-1}\left(\frac{{}^oZ_c^*\cos\alpha - \delta^c X_1}{{}^oZ_c^* - \delta^c X_3}\right), \tag{16}$$

and

$$E_\beta = \tan^{-1}\left(\frac{{}^oZ_c^*\cos\beta - \delta^c X_1}{{}^oZ_c^* - \delta^c X_3}\right). \tag{17}$$

It is here noted that E_α and E_β in Eqs.(16) and (17) are valid for the case that the camera is modeled as pin-hole lens and output values of fuzzy rules in Eqs.(2) and (14) are exactly the same as actual values. Thus, in addition to E_α and E_β in Eqs.(16) and (17), an additional orientational motion may be required to reduce orientational errors. To comply with such a requirement, pitching and yawing control commands, δX_4 and δX_5 are given by incorporating gazing angles g_α and g_β, respectively, in (9) and (10) as follows;

$$\delta X_4 = E_\alpha + k_p\, g_\alpha, \tag{18}$$

and

$$\delta X_5 = E_\beta + k_p\, g_\beta, \tag{19}$$

where k_p is given as a positive constant less than unity.

Now, rolling control command, δX_6, of the camera in hand is simply given to be proportional according to the error between the desired and the actual rolling degree, where actual rolling degree is estimated

5. Experimental Results

To show the validity of the proposed visual servoing method, some experimental results are illustrated. For this, a four axis SCARA robot, SPR-600[25], with a B/W CCD camera[23] is utilized, where the camera is mounted on the rolling axis (S-axis) of the robot in such a way that line of sight lines in the X-Y plane of the robot frame. Specifically, the CCD camera is composed of a signal module, IK-M41MK, and a lens module, IK-M30M, of which a focal length is 7.5*mm*. The control system employs an MC68030/68882-based commercial CPU board (Force CPU 30[21]) and a commercialized real-time multitasking O.S. VxWorks[24]. An additional FORCE CPU 30[21] and a B/W frame grabber, DT-1451[22], are embedded into the control system for image acquisition and feature extraction. On the other hand, the object is given as a 4*cm* × 4*cm* white square on the background. The target position T is chosen as (25.3*cm*, 55.8*cm*, 10.8*cm*) in the robot frame. The maximum distances δX_{max}, δY_{max}, and δZ_{max} for the camera to move along X, Y and Z axis of the camera frame during one visual sampling time of 160*msec* are given as 11*mm*. Here, the sampling time for the control of the robot is chosen as 40*msec*.

To compute the position trajectory, the camera at L_1(13.0*cm*, 55.8*cm*, 10.8*cm*) is made to move to T with its maximum speed. Then, the deceleration position D is chosen from the position trajectory of the camera. And, F_3^D is computed at D. To compute the feature trajectory, the camera is also mode to move from L_1 to T, while F_3 is computed, where F_3's larger than F_3^D are memorized for the feature trajectory. For the learning of line-of-sight motion, 11 initial fuzzy rules in (2) are used, and membership functions of their input variables are shown in Fig.8. Now, for the fine visual servoing, the weight of FMFNN, λ_i, in (4) is adapted by employing (6). Fig.9 shows the trajectory of F_3 while the FMFNN is trained. It is observed from Fig.9 that the steady state error converges to near zero within 30 trials. To compute other feature trajectories for various robot speeds passing through the position D, the camera in hand at L_2(15.0*cm*, 55.8*cm*, 10.8*cm*), L_3(17*cm*, 55.8*cm*, 10.8*cm*) and L_4(19*cm*, 55.8*cm*, 10.8*cm*) is moved to target position T. After learning such feature trajectories, the linear interpolation method in (7) is employed to handle feature trajectories which are not learned *a priori*.

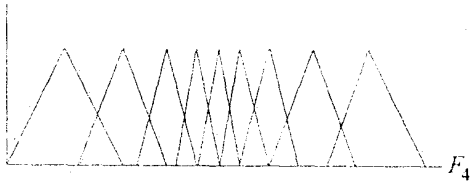


Fig.8. Membership functions for input

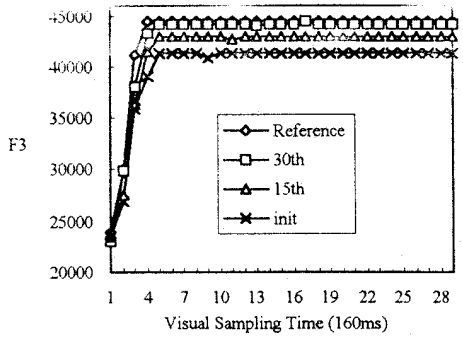


Fig.9. Feature trajectories with respect to the number of learning trials, when the FMFNN is trained in such a way that the camera moves from D to T.

Fig.10 shows the servoing performances of the proposed algorithm, when the camera is visually controlled to move from A_1(13.0*cm*, 58.5*cm*, 10.8*cm*), A_2(13.1*cm*, 61.2*cm*, 10.8*cm*), A_3(13.0*cm*, 63.96*cm*, 10.8*cm*) and A_4(13.0*cm*, 66.87*cm*, 10.8*cm*) to the target location T. Specifically, at the starting positions A_i, i = 1, 2, 3, and 4, the camera is turned by the gaze angle, g_β, in (9) for the initial gaze holding. Then, the camera is controlled to move from A_i to D_i along the line of sight with its maximum speed. It is recalled that visual servoing controller determines whether it reduces speed commands by comparing the current feature F_3 and F_3^D. When the current F_3 is larger than F_3^D, translational motions of the camera to the target position T are controlled by incorporating (2), (7), and (12) thru (15). And, the orientational motions are controlled by (18) and (19). It is observed from Fig.10 that actual paths of the camera from A_i to D_i, i = 1, 2, 3, and 4, are almost linear. This implies that the line-of-sight motion control with gaze holding works successfully in the whole workspace as expected. It is also observed from Fig.10 that the actual paths from D_i, i = 1, 2, 3, and 4, to T shows tolerable average error of 2*mm*, which might be caused by untrained camera motions along ${}^cX$ and ${}^cY$ axes, but shows little positioning errors at T. Thus, the proposed visual servoing algorithm for the control phase form D_i to T seems to be valid on

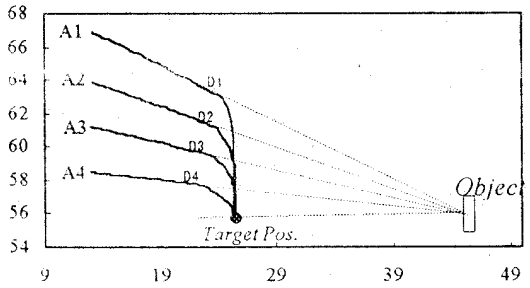


Fig. 10. The servoing performances of the proposed algorithm, when the camera is visually controlled to move from A_1, A_2, A_3 and A_4 to the target location T

any paths in the workspace, even though FMFNN is trained only on a single path.

On the other hand, we investigate the capability of the proposed visual servoing method to track the moving object along the line of sight. For this, we consider the case that the object moves along a linear path from (43.0*cm*, 55.8*cm*, 43.0*cm*) to (44.7*cm*, 55.2*cm*, 41.2*cm*) with a velocity of 3*mm/sec*, and the camera trace the object from (39*cm*, 80.0*cm*, 39.0*cm*). The desired and the actual trajectories of the camera are shown in Fig.11, where steady state error of 0.5*mm* is observed for the Y axis of the camera frame. It is remarked that the steady state error is mainly caused by computational time delay of one visual sampling time. It is also remarked that the camera motions in 6 DOF can be easily extended by using similar approaches used in this experiment.

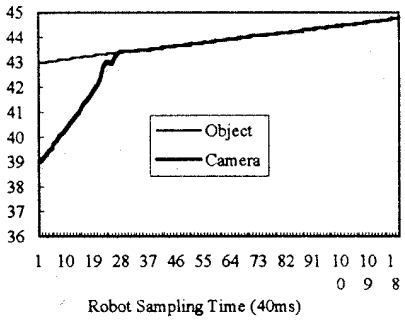


Fig.11. Tracking performance of the proposed algorithm when the object moves along a linear path from (43.0*cm*, 55.8*cm*, 43.0*cm*) to (44.7*cm*, 55.2*cm*, 41.2*cm*) with a velocity of 3*mm/sec*, and the camera trace the object from (39*cm*, 80.0*cm*, 39.0*cm*)

6. Concluding Remarks

In this paper, novel features and an improved visual servoing algorithm were proposed. A viewing model of perspective projection was used to get image features F_4 and F_5 with relatively high noise-immunity and size invariant characteristics. And, owing to the uniqueness of the proposed image features, at most two input variables were employed for the design of fuzzy logics and/or fuzzy-neural networks. To compensate dynamic characteristics of the robot, desired feature trajectories for the learning of visually guided line-of-sight robot motion were obtained by measuring features by the camera in hand not in the entire workspace, but on a single linear path along which the robot moves under the control of a commercially provided function of linear motion. And then, control actions for the *line-of-sight motion* of the camera are approximately found by fuzzy-neural networks to follow desired feature trajectories when the orientational motions of the camera are controlled by gaze holding. From the experimental results, it was shown that the proposed visual servoing method worked successfully on any

paths in the whole workspace.

The advantages of our proposed visual servoing can be now summarized as follows; 1) The robot dynamics can be effectively considered by using the feature trajectories. 2) A novel perspective model based image feature is robust against image noises and/or computation errors. 3) Relatively little geometric information is required on the camera, the object and the environment. 4) The amount of learning is small since line-of-sight motions of a robot-mounted camera are required to be learned not in the entire work space, but on a single linear path. 5) Line-of-sight motions of the robot can be guaranteed in the whole workspace by using both the line of sight motion on a single linear path and the gaze holding. 6) Comparing with the conventional feature Jacobian, relatively fast computation can be achieved since no computations of the inverse of the feature Jacobian are required.

References

[1] P.K.Allen, B.Yoshimi, and A.Timcenko, "Real-time visual servoing," In Proc. of *IEEE Int. Conf. Robotics and Automation*, pp.851-856, 1991.

[2] C.Brown, "Gaze Controls with Interactions and Delays," *IEEE Trans. on System, Man, and Cybernetics*, Vol.20, No.1, pp.518-527, March/April 1990.

[3] F.Chaumette, P.Rives, and B.Espiau, "Positioning of a Robot with respect to an Object, Tracking it and Estimating its Velocity by Visual Servoing," In Proc. of *IEEE Int. Conf. on Robotics and Automation*, pp.2248-2253, April 1991(Sacramento, California, USA).

[4] D.J.Coombs and C.M.Brown, "Cooperative Gaze Holding in Binocular Vision," *IEEE Control Systems*, Vol.11, No.4, pp.24-33, June 1991.

[5] P.I.Corke, "Visual Control of Robot Manipulator - A Review," In *Visual Servoing: Real-Time Control of Robot Manipulators Based on Visual Sensory Feedback*, K.Hashimoto, Ed., World Scientific Publishing Co., 1993.

[6] J. Doblin, *Perspective - A New System for Designer*, 1956

[7] J.T.Feddema and O.R.Mitchell, "Vision-guided servoing with feature-based trajectory generation," *IEEE Trans. Robotics and Automation*, Vol.5, No.5, pp.691-700, Oct. 1989.

[8] J.T.Feddema, C.S.G.Lee, and O.R.Mitchell, "Weighted selection of image features for resolved rate visual feedback control," *IEEE Trans. Robotics and Automation*, Vol.7, No.1, pp.31-47, Feb. 1991.

[9] G.D.Hager and S.Hutchinson, "Visual Servoing : Achievements, Issues, and Applications," Workshop Notes of *IEEE Int. Conf. on Robotics and Automation* on Visual Servoing , May 1994 (San Diego, California, USA)

[10]. H.Hashimoto, T.Kubota, W-C.Lo, and F.Harashima, "A control scheme of visual servo control for robotic manipulators using artificial neural network," In Proc. of *IEEE Int. Conf. Control and Applications*, pp.TA-3-6, 1989 (Jerusalem, Israel).

[11] K.Hashimoto, T.Kimoto, T.Ebine, and H.Kimura, "Manipulator control with image-based visual servo," In Proc. of *IEEE Int. Conf. Robotics and Automation*, pp.2767-2272, 1991.

[12] W.Jang and Z.Bien, "Feature-based visual servoing of an eye-in-hand robot with improved tracking performance," In Proc. of *IEEE Int. Conf. Robotics and Automation*, pp.2254-2260, 1991.

[13] T.W.Kim, "A Study on Fuzzy-Neural Network-based Visual Servoing of a Robot Manipulator," *Ph.D. thesis*, Hanyang Univ., Seoul, Korea, 1995 (in Korean)

[14] M.Kuperstein, "Generalized neural model for adaptive sensory-motor control of single postures," In Proc. of *IEEE Int. Conf. Robotics and Automation*, pp.140-143, 1988.

[15] W.T.Miller, "Sensor-based control of robotic manipulators using a general learning algorithm," *IEEE Trans. Robotics and Automation*, Vol.3, No.2, pp.157-165, April 1987.

[16] R.Nevatia, "Depth measurement by motion stereo," *Computer Graphics and Image Processing*, Vol.5, pp.203-214, 1976.

[17] N.P.Papanikolopoulos and P.K.Khosla, "Shared and traded telerobotic visual control," In Proc. of *IEEE Int. Conf. Robotics and Automation*, pp.878-885, 1992.

[18] I.H.Suh and T.W.Kim, "Fuzzy Membership Function Based Neural Networks with Applications to the Visual Servoing of Robot Manipulators," *IEEE Trans. Fuzzy Systems*, Vol.2, No.3, pp.203-220, Aug. 1994.

[19] D.Vernon and M.Tistarelli, "Using camera motion to estimate range for robotic parts manipulation," *IEEE Trans. Robotics and Automation*, Vol.6, No.5, pp.509-521, Oct. 1990.

[20] J.S.-C.Yuan, "A general photogrammetric method for determining object position and orientation," *IEEE Trans. Robotics and Automation*, Vol.5, No.2, pp.129-142, April 1989.

[21] *CPU-30 User's Manual*, FORCE Computers, Inc., 1991

[22] *User Manual for DT-1451*, Data Translation, Inc., 1988

[23] *User Manual for IK-M41MK*, Toshiba, 1993

[24] *VxWorks Programmer's Guide*, Wind River System, Inc., 1990

[25] *Operating Manual for SPR-600 SCARA Robot*, Samsung Aerospace Ind., 1990