# An Intelligent Visual Servoing Method using Vanishing Point Features

Joon-Soo Lee and Il-Hong Suh


## Abstract

A visual servoing method is proposed for a robot with a camera in hand. Specifically, vanishing point features are suggested by employing a viewing model of perspective projection to calculate the relative rolling, pitching and yawing angles between the object and the camera. 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.


## I. 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[1].

In selecting image features for a visual servoing, image selection criteria including unique features, feature set robustness, cost of feature extraction, and feature set completeness have been proposed in[2]. Among them, uniqueness should be strongly considered, since it gives effects on the computational complexity of the visual servoing 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, $F_j$ ($i \neq j$), to be changed. To cope with such a difficulty, vanishing point features are suggested by employing a viewing model of perspective projection[3].

In this paper, 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. Specifically, for the latter case, FMFNN in[4, 5] is employed.

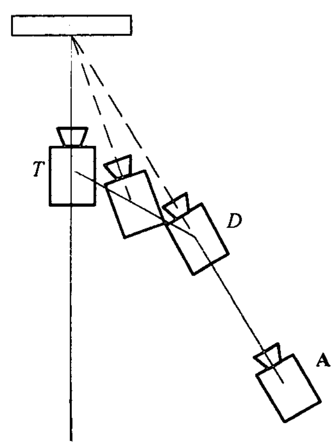



Fig. 1. Schematic diagram of camera motion by employing the proposed visual servoing method.

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.

To show the validity of our proposed algorithms, some experimental results are illustrated, where a four axis SCARA

Manuscript received November 10, 1997; accepted December 8, 1997.
J. S. Lee is with Intelligent System Control Research Center, KIST, Korea.
I. H. Suh is with Dept. of Electronics Eng., Hanyang University, Korea.

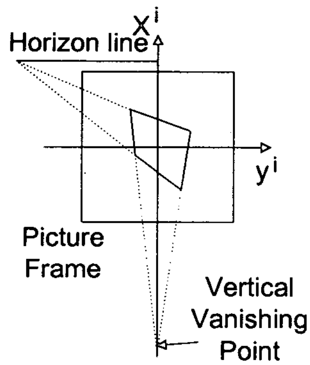



**Fig. 2.** Vanishing points.

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.

## II. Image Features for the Design of Visual Servoing

Consider a quadrangle (specifically, a rectangle for the case of the reference image) as shown in[1, 6], 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. Then, features, $F_i$, for $i$=1,2,...,6 are chosen as

$F_1$, $F_2$ : X and Y coordinates of the center of gravity of the quadrangle in the image plane,

$F_3$ : the area of the quadrangle in the image plane,

$F_4$, $F_5$, $F_6$ : rolling, pitching, and yawing angle of the quadrangle about camera axis.

$F_1$, $F_2$, $F_3$ can be found using chain code around the image of quadrangle. And $F_4$, $F_5$, $F_6$ can be obtained using vanishing points.

The images of all vertical lines pass through the unique vertical vanishing point. And horizontal vanishing point is the intersection of the image line with the horizon line of the picture. The horizon line of any picture is defined to be the intersection of the picture plane with a plane through the lens center parallel to the floor[3, 9]. Fig. 2. shows the vanishing points of the image of the quadrangle.

In order to find the coordinates of vanishing points, we need two coordinates systems as in Fig. 3 : an image coordinate system in which to locate image points, and a world coordinate system in which to locate everything else.

A vector $V$ in world coordinate can be transformed to the vector $V^i$ in the image coordinate by

$$V^i = PGRT\ V \tag{1}$$

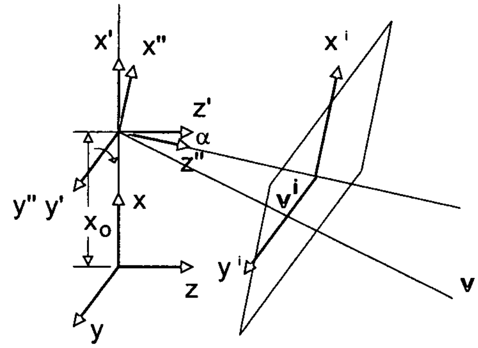



**Fig. 3.** Perspective transformation with object and camera frames.

where matrix $\boldsymbol{T}$ is the homogeneous transformation of the camera offset $x_0$, matrix $\boldsymbol{R}$ is the pans through a about $y'$ axis, matrix $\boldsymbol{G}$ is gimbal offset, and matrix $\boldsymbol{P}$ is the linear transformation that takes an object point to its image point. Specifically, $\boldsymbol{T}$, $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{P}$ are given as

$$T=\begin{bmatrix}1&0&0&-x_0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix},\quad R=\begin{bmatrix}c\alpha&0&-s\alpha&0\\0&1&0&0\\s\alpha&0&c\alpha&0\\0&0&0&1\end{bmatrix},\quad G=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&-f\\0&0&0&1\end{bmatrix},$$
$$P=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&1/f&1\end{bmatrix} \tag{2}$$

where $f$ is the focal length and $c\alpha$ and $s\alpha$ are $\cos\alpha$ and $\sin\alpha$, respectively. From (1) and (2), image coordinate of image point $(x^i, y^i)$ can be written as

$$x^i = f\frac{\cos\alpha(x-x_o)-\sin\alpha z}{\sin\alpha(x-x_o)+\cos\alpha z}, \tag{3}$$

and

$$y^i = f\frac{y}{\sin\alpha(x-x_o)+\cos\alpha z}. \tag{4}$$

By combining (3) with (4), $x^i$ can be written as

$$x^i = \left(-\frac{z}{\sin\alpha y}\right)y^i + \frac{f}{\tan\alpha}. \tag{5}$$

And we can obtain the coordinate of vertical vanishing point, $x^v$, by letting $y^i$ be equal to zero, and thus find pitching angle ( $F_5$) about $y'$ axis as follows :

$$x^v = \frac{f}{\tan\alpha}$$
$$\alpha = F_5 = atan(\frac{f}{x^i}). \tag{6}$$

To obtain horizontal vanishing point ( $x^h$, $y^h$), consider the image of an object line lying in the floor plane of the world coordinate system has the form $(0, y, my+b)^T$, where $m$ and $b$ are respectively the slope and y-intercept of the line. By substituting $(0, y, my+b)^T$ into (3) and (4), we obtain that

$$x^i = \frac{-m x_o y^i + f(x_o \cos\alpha + b \sin\alpha)}{\sin\alpha x_o - \cos\alpha b}. \tag{7}$$

Observe that $x^h$ can be simply obtained from the intersection of the image line with the horizon line of the picture as

$$x^h = -f \tan\alpha. \tag{8}$$

By replacing $x^i$ in (7) with $x^h$ in (8), we can find out the $y'$ -intercept of the image line (8) with the horizon line given as

$$y^h = \frac{f}{m \cos\alpha}. \tag{9}$$

Thus, the rolling angle $F_4$ can be found as

$$\beta = F_4 = atan\left(\frac{f}{y^h \cos(\alpha)}\right). \tag{10}$$

Finally, yawing angle $F_6$ is obtained using coordinate of vertical vanishing point, when vanishing point is not on the vertical axis of image plane. That is,

$$F_6 = atan2(y^v, x^v). \tag{11}$$

## III. Feature Trajectories and Learning of Line-of-sight Motion

Desired feature trajectories should be chosen in such a way that learning 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 $L$ to the target position $T$ in the camera frame. 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[7].

Fig. 5(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. 5., $(t_A, A)$ and $(t_D, D)$, respectively 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 technique, 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

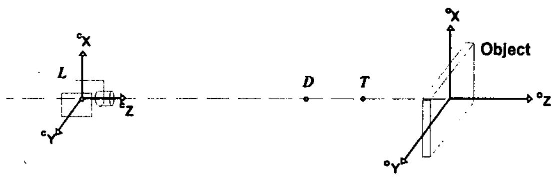
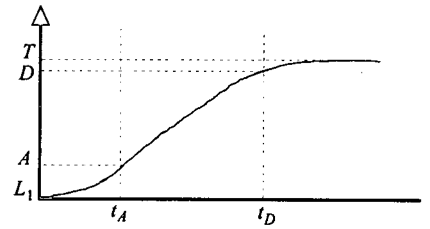
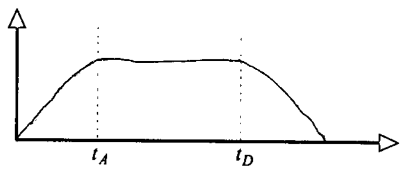



**Fig. 4.** The line of sight of camera which is aligned to the center of the object.




(a) position trajectory of the camera




(b) velocity profile of the camera

**Fig. 5.** A position trajectory and a velocity profile of a camera in hand.

the learning of visually guided motion are obtained only along the linear path from $D$ to $T$ at every visual sampling time.

Now, let $\delta X_3$ 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 $\delta x_3$. To approximately get $G(F_3, \delta F_3)$, a modified version of FMFNN in[4, 5] 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 $\delta F_{3i}$ is assumed, without loss of generality, to be given, and then is iteratively improved by FMFNN as in[4, 5].

## IV. 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, the rolling angle, $g_\beta$, and pitching angle, $g_\alpha$, for the gaze holding can be obtained by using the geometric relationship between camera and the object as follows:

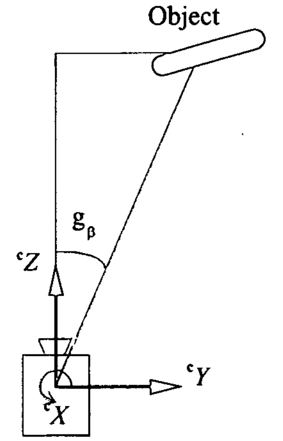



**Fig. 6.** The relative yawing angle of the camera with respect to the object.

$$F_2 = k_s \frac{f}{^cZ_o}\ ^cY_o, \tag{12}$$

$$g_\beta = \tan^{-1}\left(\frac{^cY_o}{^cZ_o}\right) = \tan^{-1}\left(\frac{F_2\ ^cZ_o/k_sf}{^cZ_o}\right)$$
$$= \tan^{-1}\left(\frac{F_2}{k_sf}\right), \tag{13}$$

and

$$g_\alpha = \tan^{-1}\left(\frac{F_1}{k_sf}\right), \tag{14}$$

where $f$ and $k_s$, respectively, denote the focal length of the camera and an image scaling factor and can be known a priori. And, $^cY_0$ and $^cZ_0$, respectively, are $^cY$ and $^cZ$ directional positions of the object with respect to the camera frame.

It is remarked that since the distance 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, actual size of the object can be estimated by $\widehat{F}_3$ given as

$$\widehat{F}_3 = F_3 \cos^{-1}(\alpha)\ \cos^{-1}(\beta) \tag{15}$$

where $\alpha$ and $\beta$ are calculated by (6), (10). Then, fuzzy rules for estimating the relative distance between the camera and the object can be given as

if $\widehat{F}_3$ is near $F_{3i}$
then $^oZ^*_c$ is near ( $^oZ_{Ti}$) (16)
for $i$=1,2,...,$n$,

where $^oZ^*_c$ and $^oZ_{Ti}$, respectively, are the estimated relative position of the camera and the pre-measured target position with respect to the object.

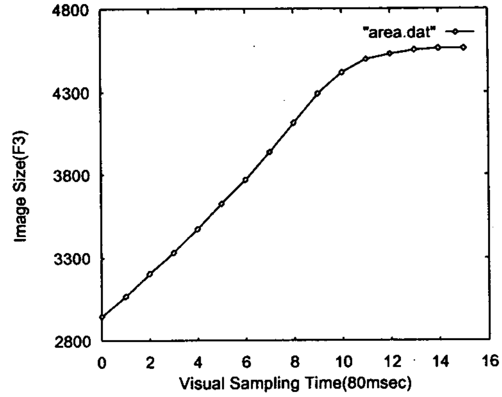



**Fig. 7.** Feature( $F_3$) trajectory.

Now, since T is given *a priori* with respect to the object frame and the current camera position is estimated by employing (18) and (19), the unit vector for the line from the current camera position to $T$, $\vec{u}_c$, can be obtained as

$$\vec{u}_c = \begin{vmatrix} u_{cx} \\ u_{cy} \\ u_{cz} \end{vmatrix} = \frac{1}{\sqrt{\tan^2\alpha + \tan^2\beta + 1}} \begin{vmatrix} tas\alpha \\ \tan\beta \\ 1 \end{vmatrix} \tag{17}$$

Then, the translational motion commands, $\delta^cX_1$, $\delta^cX_2$, and $\delta^cX_3$, of the camera along the linear path from $D$ to $T$ during a visual sampling time can be computed by using

$$\begin{bmatrix} \delta^cX_1 \\ \delta^cX_2 \\ \delta^cX_3 \end{bmatrix} = \delta X_3\ \vec{u}_c. \tag{18}$$

## V. 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[7], with a B/W CCD camera[8] 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. The object is given as a $4cm \times 4cm$ white square on the background. The target position $T$ is chosen as (25.3*cm*, 53.88*cm*, 6.9*cm*) in the robot frame. The maximum distances $\delta X_{max}$, $\delta Y_{max}$ and $\delta Z_{max}$ for the camera to move along $X$, $Y$ and $Z$ axis of the camera frame during one visual sampling time of 80*msec* are given as 5.5*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(0.994cm, 53.8cm, 6.4cm) is made to move to T with its maximum speed (5.5mm/80msec). Then, the deceleration position D is chosen from the position trajectory of the camera. To compute the feature

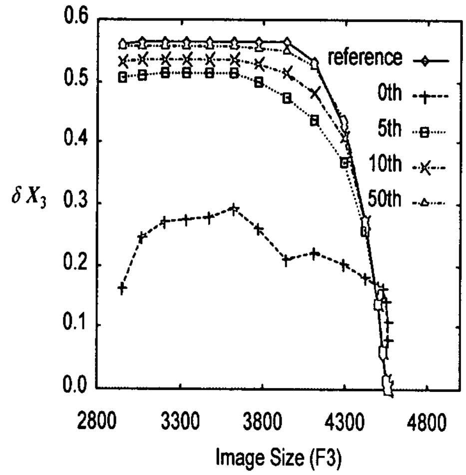



Fig. 8. The motion command, $\delta X_3$, versus image size.

trajectory, the camera is made to move from L to T, while F3 is computed, where $F_3$'s larger than $F_3^D$ are memorized for the feature trajectory as in Fig. 7.

Now, for the fine visual servoing, the weight of FMFNN as in [4, 5], $\lambda_i$, is adapted. Fig. 8 shows the motion command, $\delta X_3$, versus image size, while the FMFNN is trained.

Fig. 9 shows the servoing performance of the proposed methods, when the camera is visually controlled to move from A1 (-0.723, 49.663, 6.36cm), A2(-0.472, 44.303, 6.41), and A3 (-1.356, 40.38, 5.9) to the target location T.

## VI. Concluding Remarks

In this paper, vanishing point features and an improved visual servoing method were proposed. A viewing model of perspective projection was used to get the relative rolling, pitching and yawing angles between the object and the camera. The advantage of vanishing point feature are as follows; 1) insensitivity to the noise, 2) relative accuracy at the small angle, 3) independency of the size of the image and 4) no requirement of centering the image on CCD camera.

Control actions for the line-of-sight motion of the camera were approximately found by fuzzy-neural networks to follow such desired feature trajectories when the oriental 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.

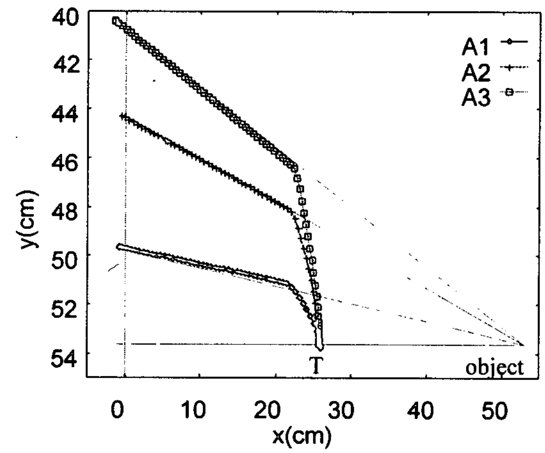



Fig. 9. The servoing performances of the proposed method.

## References

[1] S. Hutchinson, G. D. Hager, and Peter I. Corke, A Tutorial on Visual Servo Control, *IEEE Trans. on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670, Oct. 1996.

[2] John T. Feddema, C. S. George Lee, and Owen Robert Mitchell, Weighted Selection of Image Features for Resolved Rate Visual Feedback Control, *IEEE Trans. on Robotics and Automation*, Vol. 7, No. 1, pp. 31-47, Feb. 1991.

[3] Richard O. Duda, Peter E. Hart, *Pattern Classification and Scene Analysis*, A Wiley-Interscience Pub., 1973.

[4] 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.

[5] I. H. Suh and T. W. Kim, A Visual Servoing Algorithm using Fuzzy Logics and Fuzzy-Neural Networks, *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 3605-3612, 1996.

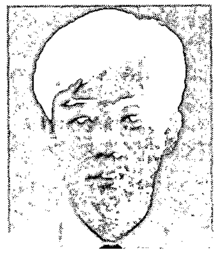
[6] K. Hashimoto, T. Ebine, and H. Kimura, Visual Servoing with Hand-Eye Manipulator--Optimal Control Approach, *IEEE Trans. Robotics and Automation*, Vol. 12, No. 5, pp. 766-774, Oct. 1996.

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

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

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

**Joon-Soo Lee** received the B.S. and M.S. degrees in electronics engineering from Hanyang University, in 1987 and 1989, respectively. He joined the Korea Institute of Science and Technology (KIST), Seoul, in 1989 where he is now research scientist. And he is currently working toward the Ph.D. degree with the Department of Electronics Engineering at Hanyang University. His research interests are in the area of machine vision, visual servoing, fuzzy logics.



**Il-Hong Suh** received the B.S. degree in electronics engineering from Seoul National University, in 1977 and the M.S. and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Seoul, in 1979 and 1982, respectively. From 1982 to 1985 he was a Senior Research Engineer at the Technical Center of Daewoo Heavy Industries, Ltd., Inchon, Korea. In 1987, he was a visiting Research Scientist at the Robotics Division, CRIM of the University of Michigan, Ann Arbor. From March, 1996 to March, 1998, he served as Vice Dean of Academic Affairs, Ansan campus, Hanyang University. Since 1985, he has been with the Department of Electronics Engineering, Hanyang University, where he is a Professor. His research interests include sensor based control of robot manipulators, coordination of multiple robot arms, fuzzy logics and neural networks.