Paper

Visual Feedback Using Virtual Object Points In Stereo Vision Based Manipulator

Student member Djoko Purwanto (Keio University)

Member Toshiyuki Murakami (Keio University)

Member Kouhei Ohnishi (Keio University)

This paper presents a visual servoing method using the stereo camera to control the manipulator pose (position and orientation) with respect to an object pose (position and orientation). To represent the object pose with respect to cameras mounted at the end of manipulator, we use the virtual object points. Based on the translation and the rotation operation applied to a predefined template points, the virtual object points are generated. The translation operation is determined from the single feature point that represents the estimation of the object position. The rotation operation is determined from the virtual angle calculated from the difference of the object area and the reference model area. The control signal to move the manipulator is derived from the error signal between the virtual object points and its desired values. Experiments conducted on a 3-links serial planar manipulator demonstrated the performance of the proposed method.

***Keywords**: vision-based control, stereo vision, multi degree of freedom manipulator*

1. Introduction

The visual perception and the control strategy are the important aspects in visual servoing. The feature extraction is widely used in visual perception to represent the important information. Based on the feature, the visual servoing algorithm is constructed. Bernardino et al. (1999) defines the image feature including the binocular disparity and the target average position, and uses this feature in the binocular tracking system (1). In manipulator control strategy, Ohnishi at al. (1996) develops the disturbance-observer-based robust motion control that performs the estimation and suppression of the disturbance existing in the manipulator, and to make the motion controller to be an acceleration controller (4). In our previous work (2), using the merits of the disturbance-observer-based control strategy, we proposed the position-velocity-based trajectory control to move the manipulator smoothly in linear path between two designated pose (position and orientation). We also applied the proposed control strategy to construct a simple visual tracking system using single camera.

This paper proposes the virtual object points extracted from the visual perception to provide the visual feedback signals in the tracking system. The virtual object points are the feature points in camera frame, and represent the object pose with respect to the camera pose. The control approach used to construct the visual servoing is the extension of the position-velocity-based control method. Based on the translation and the rotation operation applied to a predefined template points, the positions of the virtual object points are determined. The translation operation is determined using a single feature point that represents the estimation of the object position. The virtual angle based on the difference of the object area and the reference model area, determines the rotation operation. The error signals between the position of virtual object points and its desired values are used to generate the signals to control the manipulator movement.

In this paper, we consider the visual tracking system using the 3-links serial planar manipulator with the stereo camera mounted at the end of manipulator. The manipulator moves only in horizontal plane (*x*-*z* plane). The object can be chosen freely as long as the left and right side of the object is symmetric in shape. The object movement is same as the manipulator movement that is in horizontal plane (*x*-*z* plane). The change in object orientation is only the rotation about *y* axis. To obtain the faster visual perception time, we use the simple background. The flexibility to perform the visual tracking with respect to various object shapes is the main advantage of our purpose method.

Our proposed method can be extended for the control system of the camera-equipped automatic wheel chair. The moving ability to satisfy the desired posture with respect to the owner is the basic ability of the wheel chair. In this case, the simplified algorithm is required to generate the desired motion reference and our proposed approach is practicable in the well-known environment. Improving the visual perception processing to obtain the better perception in more complex and dynamic environments is the important future work.

2. Imaging Geometry

2.1. Translation, Scaling, and Rotation Translation, scaling, and rotation are useful transformation that widely used in robotic and vision field. For convenience, in this paper we use the concept of homogeneous coordinates and homogeneous transformation matrix.

Homogeneous coordinates for the position vector of the object point in three-dimensional space $[x_p \;\; y_p \;\; z_p]^T$, is represented by the 4×1 vector $[kx_p \;\; ky_p \;\; kz_p \;\; k]^T$, where k is an arbitrary nonzero constant. Converting a point represented as a 4×1 vector from a homogeneous coordinate representation back to the three-dimensional coordinate representation can be accomplished by dividing the first three homogeneous coordinates by the fourth.

Homogeneous transformation matrix is 4×4 matrix that is defined for purpose of a homogeneous position vector mapping from one coordinate system into another. The basic transformations include translation, scaling, and rotation.

The homogeneous transformation matrix for pure translation of a moving frame with respect to a fixed frame, is expressed as

$$\mathbf{G}(x_0, y_0, z_0) = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)$$

$[x_0 \;\; y_0 \;\; z_0]^T$ is the position of the origin of the moving frame relative to the fixed frame. Figure 1 describes the translation operation.

Using the homogeneous transformation matrix, a point in moving frame coordinate system ${}^{B}\mathbf{p}_o$, can be expressed easily in fixed frame coordinate system ${}^{A}\mathbf{p}_o$, that is

$$ {}^{A}\mathbf{p}_{o(h)} = \mathbf{G}(x_0, y_0, z_0)\,{}^{B}\mathbf{p}_{o(h)} \qquad (2) $$

Here, the *(h)* notation expresses the homogeneous representation.

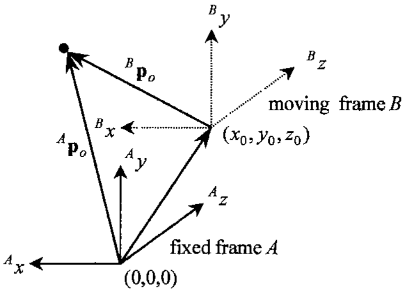


Fig. 1 Translation operation

The scaling transformation matrix which s_x, s_y, and s_z, denote the scaling factors in *x*, *y*, and *z* axis, respectively, is given by equation (3). The scaling transformation applies the individual scale factors to a coordinate system. The change of measurement units for example, needs the scaling operation.

$$ \mathbf{S}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3) $$

The homogeneous transformation matrix for rotation operation about *y* direction $\mathbf{R}_y(\beta)$ and *z* direction $\mathbf{R}_z(\gamma)$, are given by equations (4) and (5). Figure 2 illustrates the rotation operation.

$$ \mathbf{R}_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\beta & 0 & \cos\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4) $$

$$ \mathbf{R}_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5) $$

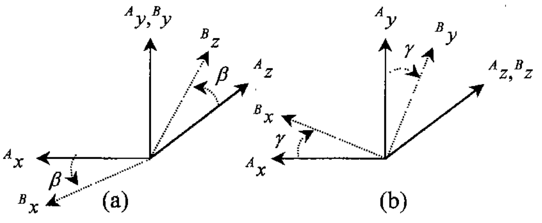


Fig. 2 Rotation operation about *y* axis (a) and *z* axis (b)

2.2. Object-Image Point Transformation Figure 3 shows the triangle relationship in the object-image point transformation. The point in image plane ${}^{i}O$ is the point of projection and λ is the focal length of the lens.

The perspective transformation matrix for projection operation $\mathbf{P}(\lambda)$ is given by equation (6). Using the perspective transformation matrix $\mathbf{P}(\lambda)$, the projection of an object point in camera frame ${}^{c}O$ onto image plane ${}^{i}O$ can be determined easily as expressed in equation (7). Note that equation (7) is written using homogeneous coordinate notation.

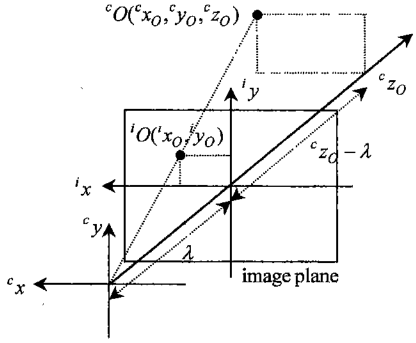


Fig. 3 Object-image point transformation

$$ \mathbf{P}(\lambda) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/\lambda & 0 \end{bmatrix} \qquad (6) $$

$$ {}^{i}\mathbf{O}_{(h)} = \mathbf{P}(\lambda)\,{}^{c}\mathbf{O}_{(h)} \qquad (7) $$

The mapping of three-dimensional object onto the image plane is a many to one transformation. The three-dimensional object point cannot be recovered from its image unless something more about the point is known or if the image taken from two cameras that are a fixed distance apart.

2.3. Computer Screen Representation The image preprocessing usually conducted in computer screen plane and use the standard pixel unit. The transformation matrix is required to convert a value from screen plane to image plane and vice versa. Figure 4 shows the image plane and screen plane representation. The transformation from image plane to screen plane can be determined by sequences operation of the rotation about *z* axis, scaling to pixel unit, and the translation with respect to the half size of screen plane. The transformation matrix from image plane to screen plane ${}^{i}\mathbf{T}_s$ and its inverse transformation ${}^{s}\mathbf{T}_i$, are expressed in equations (8) and (9), respectively.

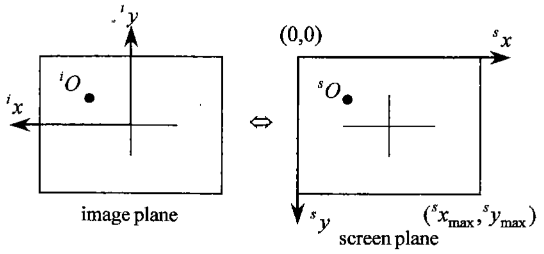


Fig. 4 The image plane and the computer screen plane

$$ {}^{i}\mathbf{T}_s = \mathbf{G}\left({}^{s}x_{\max}/2,\ {}^{s}y_{\max}/2, 0\right)\mathbf{S}(1/ps, 1/ps, 1)\mathbf{R}_z(\pi) \qquad (8) $$

$$ {}^{s}\mathbf{T}_i = ({}^{i}\mathbf{T}_s)^{-1} \qquad (9) $$

Here, *ps* is pixel scale factor in m/pixel unit, and it is assumed that in both *x* and *y* direction have same value. The scaling factor in *z* direction is an arbitrary nonzero value. The unit for rotation about *z* axis is radian.

2.4. Manipulator and Stereo camera Figure 5 shows the system configuration used to construct the vision-based control system. Here, the stereo camera are mounted at the end of manipulator. The manipulator works in x-z plane with the parameters including the workspace position (x_m, z_m) and orientation α_m. The vector representation of the manipulator workspace $\mathbf{h}$ is expressed in equation (10). An object point can be expressed either in world frame O or in camera frame ${}^{c}O$. The transformation matrix to change the coordinate representation from world frame to camera frame ${}^{w}\mathbf{T}_c$ and its inverse tranformation matrix ${}^{c}\mathbf{T}_w$, are given by equations (11) and (12), respectively.

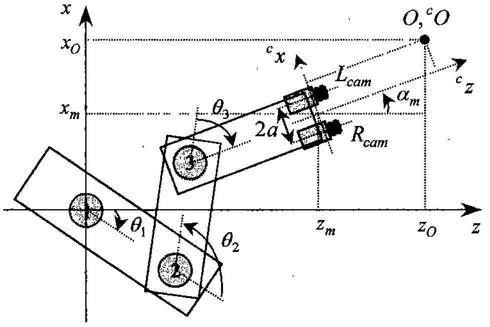


Fig. 5 Manipulator and stereo camera

$$\mathbf{h} = [x_m \quad z_m \quad \alpha_m]^T \tag{10}$$

$$^{w}\mathbf{T}_c = \mathbf{R}_y(-\alpha_m)\mathbf{G}(-x_m, 0, -z_m) \tag{11}$$

$$^{c}\mathbf{T}_w = \left({}^{w}\mathbf{T}_c\right)^{-1} \tag{12}$$

In the stereo camera perception, if the left camera and right camera have similar focal length λ, the projection of an object point in camera frame ${}^{c}O$ onto left image plane ${}^{i}O_L$ and right image plane ${}^{i}O_R$ are as follows.

$$\textit{left camera}: \quad {}^{i}\mathbf{O}_{L(h)} = \mathbf{P}(\lambda)\mathbf{G}(-a,0,0)\,{}^{c}\mathbf{O}_{(h)} \tag{13}$$

$$\textit{right camera}: \quad {}^{i}\mathbf{O}_{R(h)} = \mathbf{P}(\lambda)\mathbf{G}(a,0,0)\,{}^{c}\mathbf{O}_{(h)} \tag{14}$$

On the contrary, if the projection points in image plane from left camera ${}^{i}O_L({}^{i}x_L, {}^{i}y_L)$ and right camera ${}^{i}O_R({}^{i}x_R, {}^{i}y_R)$ are available, by solving simultaneous equations (13) and (14), the three-dimensional object point position vector in camera frame ${}^{c}\mathbf{O}$ can be calculated using equation (15).

$$^{c}\mathbf{O} = \begin{bmatrix} {}^{c}x_O \\ {}^{c}y_O \\ {}^{c}z_O \end{bmatrix} = \frac{2}{{}^{i}x_R - {}^{i}x_L} \begin{bmatrix} 0.5a({}^{i}x_R + {}^{i}x_L) \\ a\,{}^{i}y_R \\ \lambda a \end{bmatrix} \tag{15}$$

Here, ${}^{i}y_L = {}^{i}y_R$

3. Visual Servoing

3.1. Image preprocessing Figure 6 shows the image preprocessing to obtain the visual information that includes the object position in screen plane (${}^{s}x_O$,${}^{s}y_O$) and the area comparison between the object and the reference model in screen plane, ${}^{s}D_O$. In the normalized cropping region N × N, the normalized object area and the reference model area are determined to obtain the area information that does not depend on the distance between the object and the camera. Using this approach, the area comparison result contains only the basic information about the object orientation with respect to the reference model, or in other word, expresses the object orientation with respect to the camera. The image preprocessing procedure applied to both of images captured by the stereo camera.

The image preprocessing starts with image thresholding operation on original image to obtain the binary image. The binary image separates clearly the object and its background. The second step is to determine the cropping region for the object using the *bounding box* concept. The bounding box concept is an operation to find a rectangle with horizontal and vertical sides enclose the region of the object, and touch the topmost, bottommost, leftmost, and rightmost position of the object. The horizontal and vertical side of the bounding box are compared and rearranged to obtain the square cropping region. The third step consists of two operations, the object position calculation and the resizing of the cropping region with its content. The object position (${}^{s}x_O$,${}^{s}y_O$) is taken from the center position of the square cropping region. The resizing operation converts the square cropping region to the N × N normalization region. Finally, the normalized object area and the reference model area are compared in the object area comparator to generate the object-model area difference ${}^{s}D_O$.

The object position vector in left screen plane ${}^{s}\mathbf{O}_L$ and right screen plane ${}^{s}\mathbf{O}_R$ can be transformed to left image plane ${}^{i}\mathbf{O}_L$ and right image plane ${}^{i}\mathbf{O}_R$ by using the screen plane to image plane transformation matrix ${}^{s}\mathbf{T}_i$ described in equation (9).

$$\textit{left plane}: \quad {}^{i}\mathbf{O}_{L(h)} = {}^{s}\mathbf{T}_i \; {}^{s}\mathbf{O}_{L(h)} \tag{16}$$

$$\textit{right plane}: \quad {}^{i}\mathbf{O}_{R(h)} = {}^{s}\mathbf{T}_i \; {}^{s}\mathbf{O}_{R(h)} \tag{17}$$

If the position vector of the object in left image plane ${}^{i}\mathbf{O}_L$ and right image plane ${}^{i}\mathbf{O}_R$ are known, the three-dimensional object position vector in camera frame ${}^{c}\mathbf{O}$ can be determined using equation (15). In short notation is as equation (18).

$$^{c}\mathbf{O} = F_1\left({}^{i}\mathbf{O}_L, {}^{i}\mathbf{O}_R\right) \tag{18}$$

The object position can be described in world frame coordinate. Using camera frame to world frame transformation matrix ${}^{c}\mathbf{T}_w$ described in equation (12), the homogeneous coordinate representation of the object position in world frame $\mathbf{O}_{(h)}$ is as equation (19).

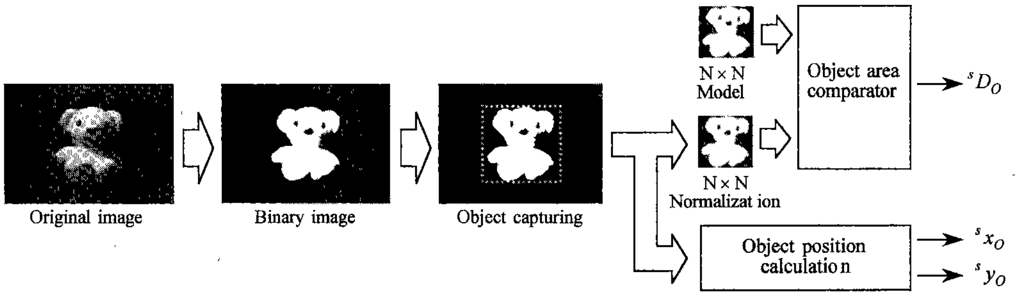


Fig. 6 Image preprocessing

$$\mathbf{O}_{(h)} = {}^c\mathbf{T}_w\ {}^c\mathbf{O}_{(h)} \tag{19}$$

Suppose that Mdl_{LArea} and Mdl_{RArea} denote the left side and the right side of the reference model area, Obj_{LArea} and Obj_{RArea} denote the left side and the right side of the object area calculated in normalized cropping area. The difference of area between the reference model and the object calculated in the left side and the right side of the object area comparator, ${}^sD_{OL}$ and ${}^sD_{OR}$, are formulated as equations (20) and (21).

$${}^sD_{OL} = Mdl_{LArea} - Obj_{LArea} \tag{20}$$

$${}^sD_{OR} = Mdl_{RArea} - Obj_{RArea} \tag{21}$$

3.2. Virtual Object Points The virtual object points is the feature points in certain configuration that is defined for purpose of the object pose (position and orientation) representation in camera frame coordinate. Based on predetermined template points, the estimation of the object position, and the change object area with respect to the reference model, the virtual points are determined. The error values generated from the difference between the actual values and its desired values of the virtual object points, are used to realize the vision-based control system for the manipulator.

The predetermined template points configuration depend on the movement properties of the object, especially the change in the object orientation. If the orientation of the object rotates in one axis direction, at least the template points in straight-line configuration are needed. Furthermore, if the orientation of the object rotates in two or three axis direction, at least the template points in the triangle configuration are needed. The number of points are chosen with respect to the number of the control signals applied to the manipulator. The size of the template configuration can be design freely as long as the virtual points can be viewed properly in screen frame when the object takes the nearest or farthest position with respect to the camera. The template points are placed at the position of the origin of the camera frame coordinate.

In this paper, we discuss the situation where the object orientation rotates only in one axis direction, that is the rotation about y axis in camera frame. The straight-line template configuration is suitable for this purpose. The template posture is arranged so that it can describe the rotation direction of the object. For the object tracking using the planar manipulator, we choose the three points, ${}^cA$, ${}^cB$, and ${}^cC$, as described in figure 7. The length of template points is $2l$.

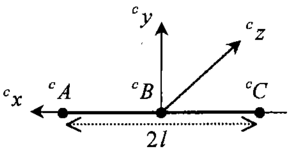


Fig. 7 The template points

The next step is to define an angle ${}^c\phi_O$ based on the area information taken from the left and the right object area comparator in screen plane, ${}^sD_{OL}$ and ${}^sD_{OR}$.

$${}^c\phi_O = \frac{\left|{}^sD_{OL}\right| + \left|{}^sD_{OR}\right|}{2} \frac{{}^c\phi_{\max}}{NORM_{Area}} \tag{22}$$

$NORM_{Area}$ is the area used for image normalization procedure that is N × N pixels, and ${}^c\phi_{\max}$ is the maximum value setting for the angle.

The virtual angle is determined according the angle ${}^c\phi_O$ and the area comparison taken by stereo camera. Equation (23) shows the methods to calculate the virtual angle in camera frame ${}^c\phi_{OV}$.

$${}^c\phi_{OV} = \begin{cases} -{}^c\phi_O & {}^sD_{OL} < {}^sD_{OR} \\ {}^c\phi_O & {}^sD_{OL} \geq {}^sD_{OR} \end{cases} \tag{23}$$

The virtual object points in camera frame determined by translating and rotating the predetermined template points in figure 7. The translation process based on the estimated object position in camera frame ${}^c\mathbf{O} = [{}^cx_O\ {}^cy_O\ {}^cz_O]^T$ as calculated using equation (18), and rotation process based on the virtual angle ${}^c\phi_{OV}$ as in equation (23). The augmented result of the virtual object points positions in homogeneous representation ${}^c\mathbf{p}_{OV(h)}$ is as equation (24).

$$\begin{aligned} {}^c\mathbf{p}_{OV(h)} &= [{}^c\mathbf{A}_{OV(h)}\quad {}^c\mathbf{B}_{OV(h)}\quad {}^c\mathbf{C}_{OV(h)}] = \\ &\mathbf{G}\left({}^cx_O, {}^cy_O, {}^cz_O\right)\mathbf{R}_y\left({}^c\phi_{OV}\right)[{}^c\mathbf{A}_{(h)}\quad {}^c\mathbf{B}_{(h)}\quad {}^c\mathbf{C}_{(h)}] \end{aligned} \tag{24}$$

The desired virtual points positions in camera frame are calculated in similar way as equation (24). Suppose that the desired situation is to obtain the visual perception similar as model in certain distance in z axis direction. It means that the virtual angle should set to zero and the translation applied only in z direction. The augmented matrix of the desired virtual points positions in homogeneous representation ${}^c\mathbf{p}_{DV(h)}$ is formulated in equation (25) below.

$$\begin{aligned} {}^c\mathbf{p}_{DV(h)} &= [{}^c\mathbf{A}_{DV(h)}\quad {}^c\mathbf{B}_{DV(h)}\quad {}^c\mathbf{C}_{DV(h)}] = \\ &\mathbf{G}\left(0,0,{}^cz_D\right)\mathbf{R}_y(0)[{}^c\mathbf{A}_{(h)}\quad {}^c\mathbf{B}_{(h)}\quad {}^c\mathbf{C}_{(h)}] = \\ &\mathbf{G}\left(0,0,{}^cz_D\right)[{}^c\mathbf{A}_{(h)}\quad {}^c\mathbf{B}_{(h)}\quad {}^c\mathbf{C}_{(h)}] \end{aligned} \tag{25}$$

In general term, ${}^cz_D$ is the desired camera-object relative distance.

The error function ${}^c\mathbf{e}_{V(h)}$ to realize the control mechanism for the manipulator is simply calculated from equation (24) and (25) as follows.

$${}^c\mathbf{e}_{V(h)} = {}^c\mathbf{p}_{DV(h)} - {}^c\mathbf{p}_{OV(h)} \tag{26}$$

3.3. Control Framework To build the general control framework, the augmented matrix in equations (24), (25), and (26) are rearranged so that they are represented in Cartesian coordinate as follows.

$${}^c\mathbf{p} = \begin{bmatrix} {}^c\mathbf{A}_{OV} \\ {}^c\mathbf{B}_{OV} \\ {}^c\mathbf{C}_{OV} \end{bmatrix} \tag{27}$$

$${}^c\mathbf{p}_d = \begin{bmatrix} {}^c\mathbf{A}_{DV} \\ {}^c\mathbf{B}_{DV} \\ {}^c\mathbf{C}_{DV} \end{bmatrix} \tag{28}$$

$${}^c\mathbf{e} = {}^c\mathbf{p}_d - {}^c\mathbf{p} \tag{29}$$

The general relation of the virtual object points in camera frame ${}^c\mathbf{p}$, the manipulator pose in world frame $\mathbf{h}$, and the virtual object points in world frame $\mathbf{p}$, is expressed as follows.

$${}^c\mathbf{p} = F_2(\mathbf{h}, \mathbf{p}) \tag{30}$$

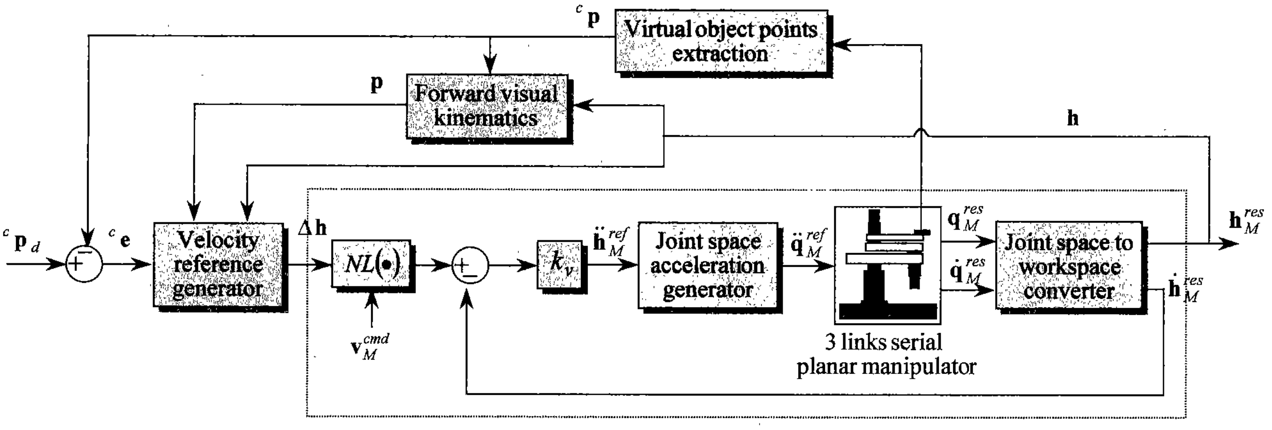


Fig. 8 The overall structure of the visual tracking system

In this visual servoing, the control strategy is determined from the virtual object points, through the linearization of (30) about the operating point.

$$\Delta^c\mathbf{p} = \frac{\partial F_2(\mathbf{h},\mathbf{p})}{\partial \mathbf{h}}\Delta\mathbf{h} + \frac{\partial F_2(\mathbf{h},\mathbf{p})}{\partial \mathbf{p}}\Delta\mathbf{p} \tag{31}$$

or $\quad \Delta^c\mathbf{p} = \mathbf{J}_1\,\Delta\mathbf{h} + \mathbf{J}_2\,\Delta\mathbf{p}$ (32)

When the object configuration parameters and their variation are known (or estimated), a suitable kinematic control solution to compensate for the virtual points motion expressed by (32) is

$$\Delta\mathbf{h} = \mathbf{J}_1^{+}\cdot\Delta^c\mathbf{p} - \mathbf{J}_1^{+}\,\mathbf{J}_2\cdot\Delta\mathbf{p} \tag{33}$$

The position difference in camera frame $\Delta^c\mathbf{p}$ can be replaced by error function in equation (29), and equation (33) becomes

$$\Delta\mathbf{h} = \mathbf{J}_1^{+}\cdot{}^c\mathbf{e} - \mathbf{J}_1^{+}\,\mathbf{J}_2\cdot\Delta\mathbf{p} \tag{34}$$

In case of static (unmoving) object, equation (34) expressed as follows.

$$\Delta\mathbf{h} = \mathbf{J}_1^{+}\cdot{}^c\mathbf{e} \tag{35}$$

Calculate the $\mathbf{J}_1$ and $\mathbf{J}_2$, requires that the current configuration parameters, **h** and **p** are known. The manipulator pose **h** is obtained from the manipulator internal encoders, and the virtual object point position **p** is estimated from *forward visual kinematics* process based on equation (19) or in simple notation is expressed as follows.

$$\mathbf{p} = F_3\left(\mathbf{h}, {}^c\mathbf{p}\right) \tag{36}$$

3.4. Control Structure

Figure 8 shows the overall structure of the visual servo system applied to 3-links serial planar manipulator. Based on the general control framework discussed in previous section, the position feedback path generates the velocity reference $\Delta\mathbf{h}$. The control system type in this structure is the position-velocity-based control. This control system performs internal interpolation mechanism to handle the different sampling rates used for motion control purpose and for the visual perception purpose. The nonlinear function $NL(\bullet)$ along with the velocity command determines the interpolation mechanism. The pure position-based control obtained by setting the velocity command at relative large value, or by replacing the nonlinear block with a position gain block. The position-velocity-based control system discussed in detail in reference (2).

4. Experimental Results

4.1. Experimental System

Figure 9 shows the experimental environment including the 3-links serial planar manipulator with stereo camera mounted at the end of manipulator, and a bear doll used as an object in visual servo system. Figure 10 shows the software and hardware description and the signal flow diagram for purpose of experiment.

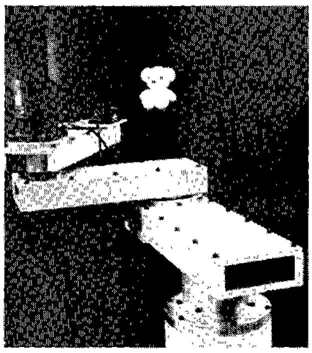


Fig. 9 Experimental environment

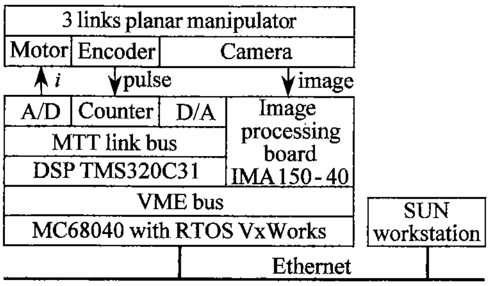


Fig.10 Signal flow diagram

4.2. Object Area

In this section, we investigated the properties of object area with respect to model area in a fixed distance and in a various orientation. Note that the area calculation does not depend on the camera-object distance because the area is determined in normalized cropping area. Figure 11 illustrates the situation in experiment. The plus and minus signs indicates the angle orientation of the object with respect to the camera. The reference model is taken at the orientation of 0 degree, and plotted as the binary image as shown at right side of figure 11. Table 1 describes the parameters used in this experiment, and figure 12 expresses the experimental results.

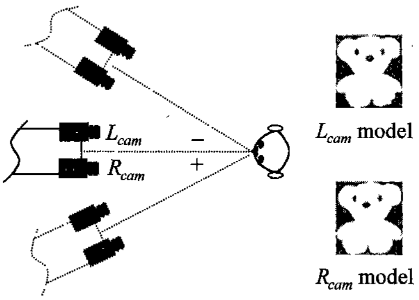


Fig. 11 Measurement of object area

Table 1 Parameters setting for area calculation

Parameter	Notation	Value
image normalization	N × N	32 × 32 pixels
object distance	${}^{c}z_O$	50 cm

The experimental result in figure 12 illustrates the difference of area between the reference model and the current object taken from both of the left and right cameras. The difference of area tends to increase if the orientation angle far away from the reference angle (0 degree) where the reference model is taken. The model-object area difference taken from the left camera and the right camera have values close to zero pixels at an orientation where the object appearance is similar to the model.

When the stereo camera takes place at left side, the model-object area difference taken from the left camera tends to large than from the right camera. On the other hand, when the stereo camera takes place at right side, the model-object area difference taken from the right camera tends to large than from the left camera.

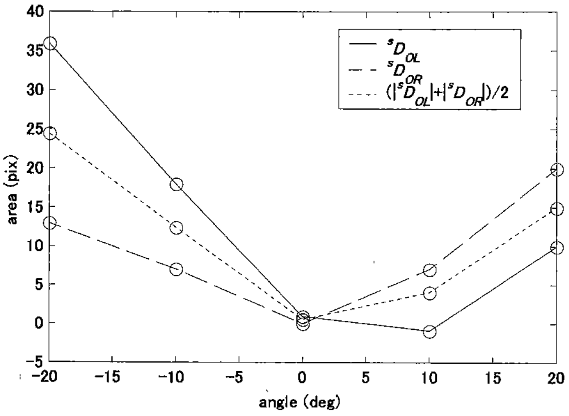


Fig. 12 Model-object area difference

4.3. Visual Servoing The experimental results from image plane, camera frame, and world frame point of view, are discussed in this section. The reference model is taken at the pose (position and orientation) where the cameras take place in front of the object [see figure 11]. The tracking process is to control the manipulator/camera pose to move to a certain pose so that the current object is similar to the model and the camera-object distance satisfies the desired distance ${}^{c}z_D$.

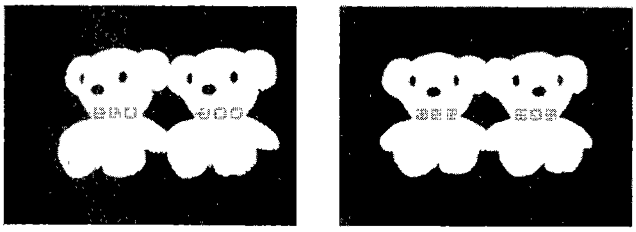
The parameters used in the visual servoing experiment are shown in table 2. As shown in table 2, the desired position in z direction or desired camera-object distance ${}^{c}z_D$ is 54 cm. Equation (25) expresses that the desired position in x direction ${}^{c}x_D$ and in y direction ${}^{c}y_D$ are both 0 cm. In this experiment, because of planar manipulator structure that moves only in x-z direction, we omit the desired position value in y direction ${}^{c}y_D$ in the control system structure.

Table 2 Parameters for visual tracking

Parameter	Notation	Value
Image normalization	N × N	32 × 32 pixels
Length of links	L_1, L_2, L_3	25,26,31 cm
Motion control sampling time	t_{s1}	1 ms
Visual perception samp. time	t_{s2}	600 ms
Template points length	$2\,l$	3 cm
Desired camera-object distance	${}^{c}z_D$	54 cm
Dead zone threshold in camera frame	th_{DZ}	0.3 cm

In the visual servoing experiment, the initial posture of the object with respect to the cameras is shown on the left side of figure 13. In the initial posture, the object position in z direction in camera frame or the actual camera-object distance ${}^{c}z_O$ is 50.1 cm, and the position in x direction in camera frame ${}^{c}x_O$ is −1.8 cm, as shown in initial position in table 3. The object orientation in the initial posture is around −7 degrees as shown in virtual angle performance at t=0 in figure 14. In figure 13, the current virtual points positions are shown as three small boxes, and the desired virtual points positions are shown as three small plus signs.

The final posture as shown on the right side of figure 13, the final object position in camera frame as shown in table 3, and the virtual angle value at t>35 seconds in figure 14, indicate the effectiveness of the visual servo system. The all plus signs that coincide the all boxes at the final posture as shown in figure 13, illustrate the situation where the object posture satisfies the reference model posture at the desired distance. In the table 3, the final position is close to the desired position for both of x and z direction, and it shows that the object positions in camera frame satisfy the desired setting. From figure 14, as the virtual angle is nearly constant at around −1 degree at t>35 seconds, it describes the situation where the area of the object close to area of the reference model, and it means that the object orientation satisfies the desired orientation. The dead zone processing applied in camera frame produce the offset error in this performance.



Initial posture Final posture

Fig. 13 Object visualization from left and right cameras

Table 3 Object position in camera frame
(Desired position, ${}^{c}x_D$ = 0 cm, ${}^{c}z_D$ = 54 cm)

Object position	Initial position	Final position
${}^{c}x_O$	−1.8 cm	0.0 cm
${}^{c}z_O$	50.1 cm	53.8 cm

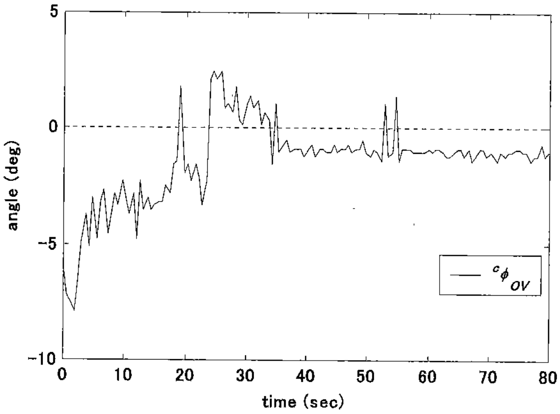
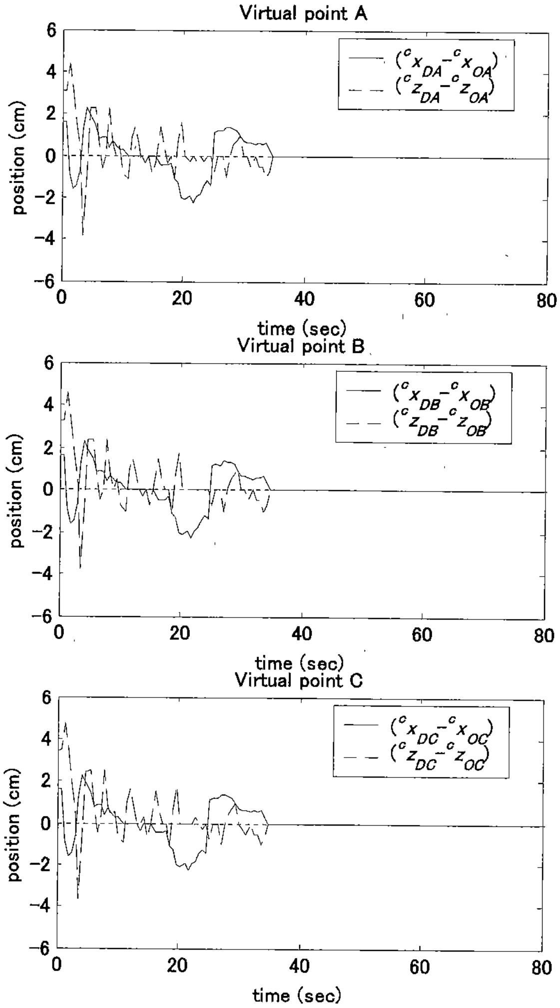


Fig. 14 Virtual angle performance


Virtual point A

Virtual point B

Virtual point C

Fig. 15 Error performance for virtual points in camera frame

Figure 15 shows the error performance of virtual points in camera frame. Here, we only show the information in x and z direction because we use the serial planar manipulator. The error of the virtual object points positions decrease the zero values at around t>35 seconds. It implies that the desired camera-object pose can be achieved properly. The perfect outputs are obtained by the application of the dead zone processing with 0.3 cm of the threshold setting in camera frame.

Finally, figure 16 shows the manipulator movement from initial pose (plotted as thin line) to final pose (plotted as thick line). It completes the clear picture about the proposed visual feedback control system.

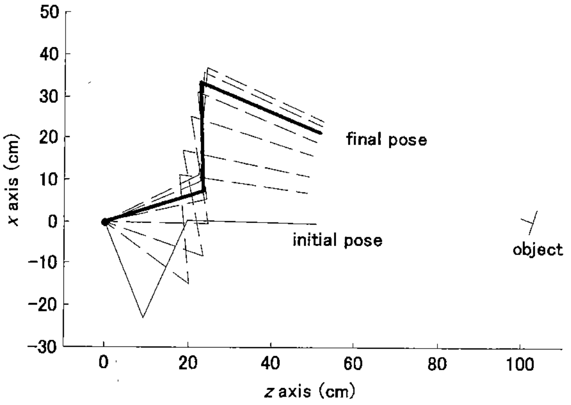


Fig. 16 Manipulator movement

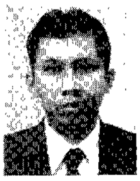
5. Conclusion

We presented the control framework based on the virtual object points to construct the visual tracking system in both of theoretical and experimental approach. Creating the virtual object points as representation of object pose (position and orientation) based on the template points, the model-object area difference, and the object position estimation, is the general tool that can be applied to the various object shapes. Several experimental results verified the effectiveness of the proposed method.

(Manuscript received Aug. 6, 2001, revised Nov. 9, 2001)

References

(1) Alexandre Bernardino, Jose Santos-Victor, "Binocular tracking: integrating perception and control", *IEEE Transaction on Robotics and Automation*, Vol. 15, No. 6, pp. 1080-1094 (1999)
(2) D. Purwanto, T. Murakami, and K. Ohnishi, "Position-velocity-based trajectory control for manipulator and its application to visual tracking system", A Publication of Industry Applications Society, The Transaction of The Institute of Electrical Engineers of Japan, Vol. 121-D, No. 8, pp. 894-901 (2001)
(3) Gregory D. Hager, "A modular system for robust positioning using feedback from stereo vision", *IEEE Transaction on Robotics and Automation*, Vol. 13, No. 4, pp. 582-595 (1997)
(4) K. Ohnishi, M. Shibata, and T. Murakami, "Motion control for advance mechatronics", *IEEE/ASME Transaction on Mechatronics*, Vol.1 No.1, pp. 56-67 (1996)
(5) Seth Hutchinson, Gregory D. Hager, and Peter I. Corke, "A tutorial on visual servo control", *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 5, pp. 651-670 (1996)
(6) William J. Wilson, Carol C. Williams Hull, and Graham S. Bell, "Relative end-effector control using cartesian position based visual servoing", *IEEE Transaction on Robotics and Automation*, Vol. 12, No. 5, pp. 684-696 (1996)



Djoko Purwanto (Member) received B.E. degree in electrical engineering from Institute of Technology "Sepuluh Nopember" Surabaya, Indonesia, in 1989. The M.E. degree in electrical engineering received from Keio University in 1999. He is currently working towards Ph.D. degree at Keio University. His research interests include image processing, motion control, and robotic vision.

Toshiyuki Murakami (Member) received B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University in 1988, 1990, and 1993, respectively. Since 1993, he has been with Keio University. His research interests include robotics, micromechatronics, and motion control.

Kouhei Ohnishi (Member) received B.E., M.E., and Ph.D. degrees, all in electrical engineering from the University of Tokyo in 1975, 1977, and 1980, respectively. Since 1980, he has been with Keio University. His research interests include mechatronics, motion control and robotics. Dr. Ohnishi is an IEEE Fellow for contributions to the development of disturbance observer and its applications to motion control.