DOI 10.1007/s10846-014-0065-2

Neural Network-Based Image Moments for Robotic Visual Servoing

Yi-Min Zhao · Wen-Fang Xie · Sining Liu · Tingting Wang

Received: 18 March 2013 / Accepted: 12 May 2014 / Published online: 10 June 2014

Abstract This paper applies two Neural Network (NN)-based image features Zhao et al. (2012) to solve the problem of decoupling the rotational velocities around *x* and *y* axes of camera frame in robotic visual servoing systems. Based on these two image features and the other four image features used in previous work Chaumette (IEEE Trans. Robot. 20(4):713–723 2004), the interaction matrix has a maximal decoupled structure and thus the singularity of interaction matrix is avoided in Image-Based Visual Servoing (IBVS). The analytical form of depth is given by using classical geometrical primitives and image moment invariants. The IBVS Proportional Derivative (PD) controller is then designed and the stability of the controller is proved by using Lyapunov method. The tracking performance is thus enhanced for a 6 degree-of-freedom (DOF) robotic system. Experimental results on the robotic system are provided to illustrate the effectiveness of the proposed method.

Keywords IBVS · Robotic systems · Image moment invariants · Interaction matrix · Neural network

1 Introduction

"Visual Servoing" has been widely used to make robotic systems increasingly dexterous and fast to work in its surrounding environment in the past two decades. Based on the error signal definition, visual servoing is classified into two major categories: Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS) [3, 4]. In IBVS, one of the open questions is to determine a set of image features in control scheme so that the decoupled behavior of the system is obtained [5]. Generally speaking, the image features used as input of control scheme are the coordinates of points, or the parameters describing the configuration of image segments such as straight lines and ellipses. In the recent years, the researchers in visual servoing adopted image moments, which have been commonly used for pattern-recognition in computer vision [6–8], for control scheme design. The main reason of using image moments for visual servoing is that they provide a generic representation of

Y.-M. Zhao · W.-F. Xie (✉) · S. Liu
Department of Mechanical and Industrial Engineering,
Concordia University Montreal, QC, Canada
e-mail: wfxie@encs.concordia.ca

Y.-M. Zhao
e-mail: ymnzhao@yahoo.com

S. Liu
e-mail: siningliu@live.ca

T. Wang
Department of Mechanical and Electrical Engineering,
Hohai University, Changzhou, 214122, China
e-mail: wtt624@163.com

any object, with simple or complex shape [2]. Furthermore, image moments can be computed easily from binary or segmented image or from a set of extracted points of interest, disregarding the object shape complexity. In addition, the judicious combination of image moments is invariant to some transformations such as 2D translation, 2D rotation and scale. This property is of great value in visual servoing [5]. The main question in choosing the image features for visual servoing is that how to combine those image features to obtain adequate form of interaction matrix for visual servoing systems. The basic strategy is just stacking [3]. However, the inappropriate combinations of image features may cause some potential convergence problems, such as local minimum and coupled image features that lead to inadequate robot trajectories [9].

Some researchers have tried to use different image features to solve this problem [2, 5, 10, 11]. In [2], six independent image features are adapted such that the corresponding interaction matrix has a maximal decoupled structure and thus the robustness and the numerical stability of robotic systems are improved remarkably [12, 13]. In [14, 15], laser points are used as image features to track the objects without enough detectable geometric features. In [10], the 2-D features and 3-D features are combined to decouple the rotation form the translation, and also to ensure the visibility of the object in the field of view (FOV) of camera. The global stability of the system is achieved by only using the measurement information from the current and desired images. In [16], the authors selected subsets of image features most relevant for determining robot pose variations along each of six degrees of freedom of camera by using a statistical measurement of variable interdependence.

The desired behaviors for visual serving systems can be obtained by determining the image moment invariants to translational velocity and rotational velocity around axes of the camera frame. In [2], the image moment invariants to 2-D translation, to 2-D rotation and to scale are exploited to decouple ω_x and ω_y (the rotational velocities around axes x and y of camera frame respectively) from other DOFs. In [5], the combinations of the image moment invariants are proposed and two of them are selected as image features to control ω_x and ω_y. It is noticed that the values of $L_{\omega x}$and $L_{\omega y}$ in the interaction matrix $J_{||} = \begin{bmatrix} 0 & 0 & 0 & L_{\omega x} & L_{\omega y} & 0 \end{bmatrix}$ [2, 5] are zero for centered symmetrical shape object, which causes singularity problem of interaction matrix. In fact, this problem occurs in almost all of Hu's invariants. Hence how to choose two new image features for the objects with both centered symmetrical and non-symmetrical shapes to decouple ω_x andω_y remains a major challenge in visual servoing. Furthermore, in the previous researches [2, 5, 17], the depth of target object is assumed to be constant [17] or is calculated from the planar object equation expressed in desired position [2, 5]. This assumption simplifies the mathematical development and such an approximation is generally accurate enough in practice because of the robustness of the visual servoing scheme to modeling errors [2]. However, for large displacement visual servoing, this assumption will cause remarkable error of interaction matrix and singularity problem of visual servoing. Therefore, an accurate estimation of depth online is an important issue for large displacement IBVS. It is known that NN has very strong ability of approaching generalization and has been widely applied in function approximation and data compression, prediction, nonlinearities compensation, etc. [16, 18–21].

In this paper, two neural network-based image features [1] are adopted to decouple ω_x and ω_y in a 6 DOF robotic visual servoing system. In addition, the accurate estimation of depth online for planar object is achieved. Therefore, the interaction matrices related to the chosen image features can be accurately determined based on the proposed scheme. Hence the visual servoing performance is enhanced significantly. In our previous research [1], the simulation has been carried out to validate the algorithm on tracking the rectangle and whale shape objects. In this paper, we have conducted experiment to validate the proposed algorithm on tracking the star and whale shape objects in a 6-DOF robotic system. In addition, the stability of the proposed controller is proved by using Lyapunov method.

This paper is organized as follows. In Section 2, six image features based on image moments are introduced. In Section 3, the NN is designed to map the relationship between the image moment invariants and the rotational angles around axes x and y of camera frame with respect to the desired position. The depth estimation of planar object is given in Section 4. The IBVS controller is designed and stability analysis is given in Section 5. The experimental results are given in Section 6 to validate the proposed method.

Finally, the conclusion and future work are presented in Section 7.

2 IBVS Using Image Moments

In this section, the development of IBVS using image moment features is introduced for 6-DOF robot systems. The eye-in-hand robotic system configuration is composed of a 6-DOF robot and a camera mounted on the robot end-effector as shown in Fig. 1, where H denotes the transformation matrix between two reference frames. From the images, the image moments are computed so that image features are extracted. The proposed visual servoing control block diagram is shown in Fig. 2, where $s_d$and s present the desired and current image features respectively and u is the control signal sent to robot controller. In our method, the off-line trained neural network can estimate the rotational angles around x and y axes of camera frame at the desired position. Based on the image features and estimated rotational angles, the depth can be estimated on- line. The errors of image feature are sent to image-based controller. The generated control signal drives the robot end-effector to approach the desired pose.

To accomplish IBVS for such a robotic system, six image features are chosen for control scheme design. The derivation of interaction matrix based on the image features is given as follows.

2.1 Interaction Matrix of Image Moments

To understand the proposed algorithm, some background knowledge on the image moments [2] is presented in the following. For a dense object in the image, its two-dimensional moments$m_{ij}$and central moments μ_{ij} of order $i + j$ are defined by

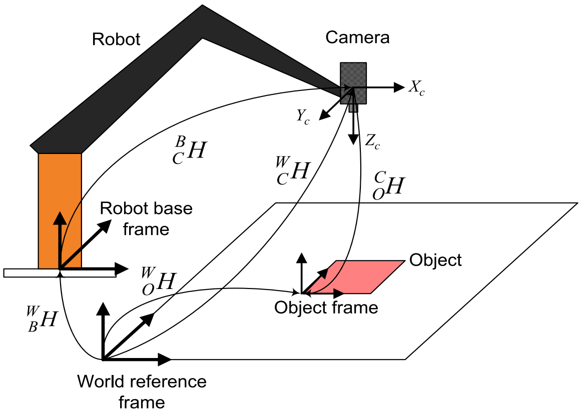


Fig. 1 Robotic eye-in-hand system configuration

$$m_{ij} = \iint\limits_{R(t)} x^i y^j dxdy \tag{1}$$

$$\mu_{ij} = \iint\limits_{R(t)} (x - x_g)^i (y - y_g)^j dxdy \tag{2}$$

where $R(t)$ represents the part of object projection in the image plane and (x_g, y_g) are the centroid coordinates in image plane $(x_g = m_{10}/m_{00}, y_g = m_{01}/m_{00})$. In the case of planar object, the equation of object surface in camera frame $F : X - Y - Z$ at instant t is given by

$$Z = k_1 X + k_2 Y + Z_0$$

where $k_1,\ k_2$are coefficients and Z_0 is the coordinate of the intersection point of the object plane and camera optical axis. In visual servoing [2], one has

$$\frac{1}{Z} = Ax + By + C \tag{3}$$

where $A = -\frac{k_1}{Z_0}$, $B = -\frac{k_2}{Z_0}$, $C = -\frac{1}{Z_0}$, and x, y are the coordinates of point in image plane corresponding to 3D point(X, Y, Z) in camera frame. From Eq. 1, it is known that only$R(t)$relates to the time$t$ in m_{ij}. By taking derivative of Eq. 1 with time and using the definition of "Contour integration" [2], the relationship between the velocities of $\mathbf{x} = (x, y)$in image plane and the velocity screw of camera is obtained

$$\dot{m}_{ij} = J_{mij}\dot{\mathbf{r}} \tag{4}$$

where the velocity screw of the camera is $\dot{\mathbf{r}} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$, the interaction matrix is

$$J_{mij} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{\omega x} & m_{\omega y} & m_{\omega z} \end{bmatrix} \tag{5}$$

where

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j} + Cm_{i-1,j+1}) - Am_{ij} \\ m_{vy} = -j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i+j+3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{\omega x} = (i + j + 3)m_{i,j+1} + jm_{i,j-1} \\ m_{\omega y} = -(i + j + 3)m_{i+1,j} - im_{i-1,j} \\ m_{\omega z} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

Similarly for the central moments μ_{ij}, one obtains

$$\dot{\mu}_{ij} = J_{\mu ij}\dot{\mathbf{r}} \tag{6}$$

where the interaction matrix is

$$J_{\mu ij} = \begin{bmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{\omega x} & \mu_{\omega y} & \mu_{\omega z} \end{bmatrix} \tag{7}$$

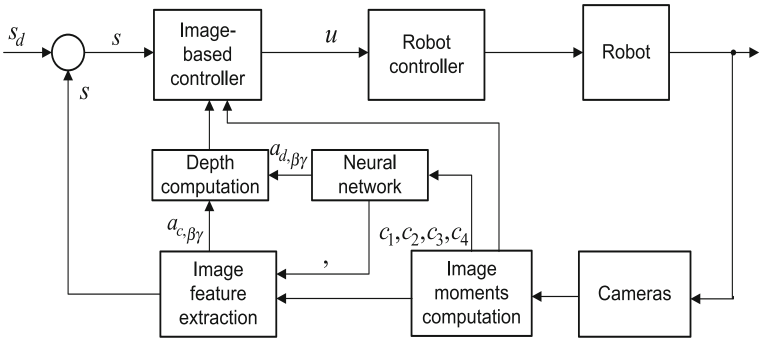


Fig. 2 Block diagram of control system

and

$$\begin{cases} \mu_{vx} = -(i+1)A\mu_{ij} - iB\mu_{i-1,j+1} \\ \mu_{vy} = -jA\mu_{i+1,j-1} - (j+1)B\mu_{ij} \\ \mu_{vz} = -A\mu_{\omega y} + B\mu_{\omega x} + (i+j+2)C\mu_{ij} \\ \mu_{\omega x} = (i+j+3)\mu_{i,j+1} + ix_g\mu_{i-1,j+1} \\ \quad + (i+2j+3)y_g\mu_{i,j} - 4i\left(\frac{m_{11}}{m_{00}} - x_g y_g\right)\mu_{i-1,j} \\ \quad - 4j\left(\frac{m_{02}}{m_{00}} - y_g^2\right)\mu_{i,j-1} \\ \mu_{\omega y} = -(i+j+3)\mu_{i+1,j} - iy_g\mu_{i+1,j-1} \\ \quad + (2i+j+3)x_g\mu_{i,j} + 4i\left(\frac{m_{20}}{m_{00}} - x_g^2\right)\mu_{i-1,j} \\ \quad + 4j\left(\frac{m_{11}}{m_{00}} - x_g y_g\right)\mu_{i,j-1} \\ \mu_{\omega z} = i\mu_{i-1,j+1} - j\mu_{i+1,j-1} \end{cases}$$

In conventional IBVS, A, B and C are treated as constants during visual servoing and this assumption is applicable in the vicinity of the desired position. Whereas, A, B and C will change due to the camera motion. The calculation of A, B and C will be discussed in Section 4.

2.2 Choice of Image Features

Although the normalized features proposed in [5] allow obtaining the better decoupling results on three translational motions, the improvement on the overall decoupling performance is not as significant as those related to the rotational motions. Three image features related to the translational motions and one feature related to orientation angle are selected as the same as those in [2] due to their less computational demand. Two Neural Network (NN)-based image features are expected to provide the decoupled rotational velocities around x and y axes of camera frame and to remedy the performance of using those non-normalized features. The four image features are selected as: $a = m_{00}$ the area of the image, x_g, y_g the coordinates of centroid, and α the orientation angle. As shown in Fig. 3, the orientation angle α is defined [7] as $\alpha = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}}\right)$. The details on the interaction matrix related to four image features are referred to [2].

The other two image features are determined such that the interaction matrix has maximal decoupled structure and minimal nonlinearities [5]. In fact, the interaction matrices would have such form as

$$J_{mx} = \begin{bmatrix} 0 & 0 & 0 & c_x & 0 & 0 \end{bmatrix} \tag{8}$$

$$J_{mx} = \begin{bmatrix} 0 & 0 & 0 & 0 & c_y & 0 \end{bmatrix} \tag{9}$$

where c_x and c_y are constant. To address this challenge, it is assumed that there exist two image moment invariants (to 2D translation, to 2D rotation and to scale), which are referred to as virtual image moments and denoted as m_x and m_y respectively. It is also assumed that$m_x$and $m_y$have such form as follows

$$m_x = f_x(\beta) = c_x\beta \tag{10}$$

$$m_y = f_y(\gamma) = c_y\gamma \tag{11}$$

where $\beta, \gamma$are the rotational angles around x and y axes of the desired camera frame respectively. If β and

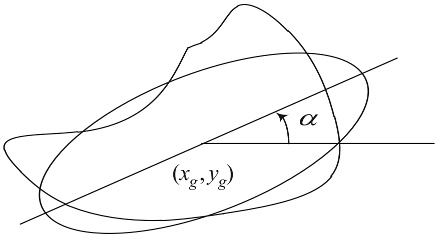


Fig. 3 Orientation of an object defined as the orientation of the ellipse

γ are available, m_x and m_y can be easily calculated during visual servoing. Hence the problem is how to estimate β and γ based on the image, which will be given in Section 3.

2.3 Interaction Matrices Related to Six Image Features

The rotational matrix from camera current frame to its desired frame is expressed as

$$R_{xyz} = R_z(\alpha)R_y(\gamma)R_x(\beta) = \begin{bmatrix} c\gamma c\alpha & (s\beta s\gamma c\alpha - c\beta s\alpha) & (c\beta s\gamma c\alpha + s\beta s\alpha) \\ c\gamma s\alpha & (s\beta s\gamma s\alpha + c\beta c\alpha) & (c\beta s\gamma s\alpha - s\beta c\alpha) \\ -s\gamma & s\beta c\gamma & c\beta c\gamma \end{bmatrix}$$

where β, γ, $\alpha$are Euler angles around x, y, z of the camera desired frame. The rotation order is $x \rightarrow y \rightarrow z$, which coincides with the neural network training procedure in Section 3. It is known that

$$\begin{bmatrix} d\beta/dt \\ d\gamma/dt \\ d\alpha/dt \end{bmatrix} = \begin{bmatrix} c\gamma c\alpha & (s\beta s\gamma c\alpha - c\beta s\alpha) & (c\beta s\gamma c\alpha + s\beta s\alpha) \\ c\gamma s\alpha & (s\beta s\gamma s\alpha + c\beta c\alpha) & (c\beta s\gamma s\alpha - s\beta c\alpha) \\ -s\gamma & s\beta c\gamma & c\beta c\gamma \end{bmatrix} \times \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

where $\begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$is an angular velocity vector expressed in camera current frame.

When m_x and m_y are chosen as image features, the interaction matrix can be computed by taking the time derivatives of $m_x$and m_y:

$$\frac{dm_x}{dt} = c_x \frac{d\beta}{dt} = \omega_x c\gamma c\alpha + \omega_y (s\beta s\gamma c\alpha - c\beta s\alpha) + \omega_z (c\beta s\gamma c\alpha + s\beta s\alpha) \quad (12)$$

$$\frac{dm_y}{dt} = c_y \frac{d\gamma}{dt} = \omega_x c\gamma s\alpha + \omega_y (s\beta s\gamma s\alpha + c\beta c\alpha) + \omega_z (c\beta s\gamma s\alpha - s\beta c\alpha) \quad (13)$$

where $c\alpha$is shorthand for $\cos\alpha$, $s\alpha$ for $\sin\alpha$, and so on; α is rotational angle around z axis of camera desired frame. ω_x, ω_y, ω_z are angular velocities around x, y, z axes of camera current frame.

The interaction matrices related to the two image features m_x and m_y are

$$J_{mx} = [0\ 0\ 0\ c_x c\gamma c\alpha\ c_x(c\beta s\gamma s\alpha - s\beta c\alpha)\ c_x(c\beta s\gamma c\alpha + s\beta s\alpha)] \quad (14)$$

$$J_{my} = \left[0\ 0\ 0\ c_y c\gamma s\alpha\ c_y(s\beta s\gamma s\alpha + c\beta c\alpha)\ c_y(c\beta s\gamma s\alpha - s\beta c\alpha)\right] \quad (15)$$

It is clear that in the vicinity of camera desired position ($\alpha = \beta = \gamma = 0$), the interaction matrices can be approximated as follows:

$$J_{mx} = \begin{bmatrix} 0 & 0 & 0 & c_x & 0 & 0 \end{bmatrix} \quad (16)$$

$$J_{mx} = \begin{bmatrix} 0 & 0 & 0 & 0 & c_y & 0 \end{bmatrix} \quad (17)$$

Hence the interaction matrices related to the two image features have the same forms as Eqs. 8 and 9 in the vicinity of camera desired position. From [5], the interaction matrices related to image features a, x_g, y_g and α are as follows

$$J_{xg} = \begin{bmatrix} -1/Z_g & 0 & x_{gvz} & x_{g\omega x} & x_{g\omega y} & y_g \end{bmatrix}$$

$$J_{xg} = \begin{bmatrix} 0 & -1/Z_g & y_{gvz} & y_{g\omega x} & y_{g\omega y} & -x_g \end{bmatrix}$$

$$J_{xg} = \begin{bmatrix} -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 \end{bmatrix}$$

$$J_\alpha = \begin{bmatrix} \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix}$$

The definitions of the variables x_{gvz}, $x_{g\omega x}$, $x_{g\omega y}$, y_{gvz}, $y_{g\omega x}$, $y_{g\omega y}$, α_{vx}, α_{vy}, α_{vz}, $\alpha_{\omega x}$, $\alpha_{\omega y}$, $\alpha_{\omega z}$ are referred to [5]. By stacking the interaction matrices related to$a$, x_g, y_g, m_x, m_y and α, one obtains the overall interaction matrix of six image features at camera current position as follows

$$J_{image} = \begin{bmatrix} -1/Z_g & 0 & x_{gvz} & x_{g\omega x} & x_{g\omega y} & y_g \\ 0 & -1/Z_g & y_{gvz} & y_{g\omega x} & y_{g\omega y} & -x_g \\ -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 \\ 0 & 0 & 0 & c_x c\gamma c\alpha & c_x(s\beta s\gamma c\alpha - c\beta s\alpha) & c_x(c\beta s\gamma c\alpha + s\beta s\alpha) \\ 0 & 0 & 0 & c_y c\gamma s\alpha & c_y(s\beta s\gamma s\alpha + c\beta c\alpha) & c_y(c\beta s\gamma s\alpha - s\beta c\alpha) \\ \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix} \quad (18)$$

The overall interaction matrix of six image features at camera desired position ($\beta = \gamma = 0$) is

$$J_{image} = \begin{bmatrix} -1/Z_g & 0 & x_{gvz} & x_{g\omega x} & x_{g\omega y} & y_g \\ 0 & -1/Z_g & y_{gvz} & y_{g\omega x} & y_{g\omega y} & -x_g \\ -aA & -aB & a(3/Z_g - C) & 3ay_g & -3ax_g & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & c_y & 0 \\ \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix} \tag{19}$$

3 Estimation of Rotational Angles Around x and y Axe of Camera Frame

3.1 Neural Network Estimation

It is noticed that the following image moments are the invariants to 2D translation, to 2D rotation [5]

$$I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_9, I_{13}I_{14}, I_{15} \tag{20}$$

The analytical forms of I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_9, $I_{13}I_{14}$, I_{15} are given in the Appendix of [5]. On the other hand, several combinations of the image moment invariants in Eq. 19 are presented in order to normalize the image moment invariant for the same shape object, which are given as follows

$$c_1 = \frac{I_1}{I_2}, c_2 = \frac{I_3}{I_4}, c_3 = \frac{I_6}{I_6}, c_4 = \frac{I_7}{I_6}, \\ c_6 = \frac{I_9}{I_6}, c_9 = \frac{I_{13}}{I_{15}}, c_{10} = \frac{I_{14}}{I_{15}}, \tag{21}$$

where c_1, c_2, c_3, c_4, c_6, c_9, c_{10} are invariant to 2D translation, to 2D rotation, and to scale, i.e., $c_1, c_2, c_3, c_4, c_6, c_9, c_{10}$ are only the functions of $\beta,\ \gamma$ when planar object is parallel to the image plane of camera at the desired position. However, we found out thatc_1, c_2, c_3 and c_4 has nonlinear relationship with $\beta,\ \gamma$. In this paper, NN is used to map the nonlinear relationship between c_1, c_2, c_3, c_4 and β, γ, i.e.,

$\beta = f_{NNX}(c_1, c_2, c_3, c_4)$ and $\gamma = f_{NNY}(c_1, c_2, c_3, c_4)$

If β, γ are available, the virtual image moments can be calculated by Eqs. 10 and 11. To estimate β, γ from c_1, c_2, c_3, c_4, NN needs to be trained in advance. As shown in Fig. 4, the object plane is parallel to the image plane of camera at the desired position, i.e., $\beta = \ \gamma = 0$. To acquire the data sets to train NN easily, the origin of camera frame is fixed and β, γ the rotational angles around axes x and y of the camera desired frame are changing. The images of the planar object are taken at each specific $\beta, \gamma$and then image moment invariants c_1, c_2, c_3, c_4 as well as the image area $a_{d,\beta\gamma}$ are calculated over the certain range of β, γ. After being trained by using the acquired data sets, the NN is used to map the relationship between image moment invariants c_1, c_2, c_3, c_4 and β, γ. In Fig. 4, $F_0 : \mathrm{X}_0 - Y_0 - Z_0$ and$F_i : \mathrm{X_c} - Y_c - Z_c$ are the camera desired frame (i.e., $\beta = \gamma = 0$) and the camera current frame (i.e., $\beta \neq 0$ or $\gamma \neq 0$) respectively. Z_d is the depth at the desired position.

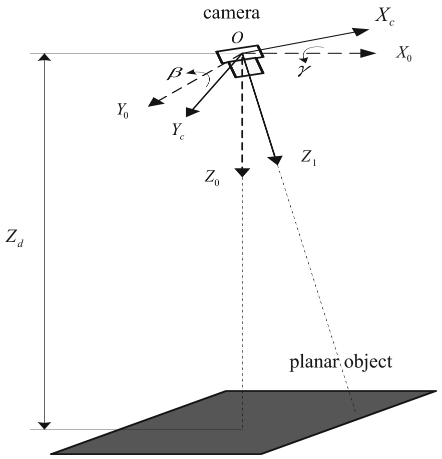


Fig. 4 Acquisition of data sets for training NN

As shown in Fig. 5, the NN of a multi-layer perceptron (MLP) [24] is used to map the nonlinear relationship of c_1, c_2, c_3, c_4 and β, γ. The MLP structure is composed of one input layer, one hidden layer with m hidden neurons, and one output layer. The inputs of NN are the image moment invariant c_1, c_2, c_3, c_4 and the outputs of NN are $\hat{\beta},\ \hat{\gamma}$ which are the estimated values of rotational angles around axes x and y of camera desired frame respectively.

The nonlinear functions mapped by NN are denoted as follows

$$\begin{aligned} \hat{\beta} &= f_2\left(\sum_{i=1}^{m} w^o_{i,1} f_1\left(\sum_{l=1}^{m}\sum_{j=1}^{4} w^h_{j,l} c_j\right)\right) \\ &= \sum_{i=1}^{m} w^o_{i,1} f_1\left(\sum_{l=1}^{m}\sum_{j=1}^{4} w^h_{j,l} c_j\right) \\ \hat{\gamma} &= f_2\left(\sum_{i=1}^{m} w^o_{i,2} f_1\left(\sum_{l=1}^{m}\sum_{j=1}^{4} w^h_{j,l} c_j\right)\right) \\ &= \sum_{i=1}^{m} w^o_{i,2} f_1\left(\sum_{l=1}^{m}\sum_{j=1}^{4} w^h_{j,l} c_j\right) \end{aligned} \tag{22}$$

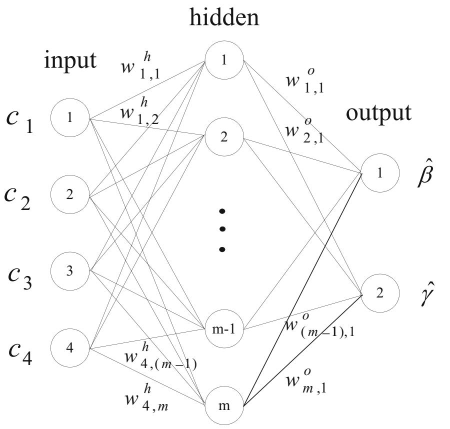


Fig. 5 NN for estimating β and γ

where $w^o_{i,1}$, $w^o_{i,2}$and$w^h_{j,l}$ are the synopsis weights of NN, $f_1(\bullet)$ is the activation function for hidden layer nodes, $f_2(\bullet)$ is the activation function for the output layer.

In this paper, the NN is trained by using Matlab2008a Neural Networks Toolbox. It is noted that if the NN is trained for a certain shape planar object, it can be reused to estimate the rotational angles for the similar shape objects in visual servoing.

3.2 NN Generalization and Verification

One of the major advantages of NN is its ability of generalization. The generalization of NN stands for the ability to have the outputs of the network when the inputs are not in the training set. In this paper, once the NN has been trained, it can be used as an estimator of rotational angles around axes x and y of camera desired frame within the training area. The experiments had been carried out to validate NN in estimating rotational angles around axes x and y of camera desired frame by using image moment invariants.

4 Depth Estimation

The depth estimation is a crucial step to obtain the accurate interaction matrix. In this section, the detailed derivations of depth estimation and the equation of planar object in the current camera frame are given. The camera frames for depth estimation are defined in Fig. 6. In this figure, $F_o : X_o - Y_o - Z_o$ is the camera desired frame and $F_c^{'} : X_c^{'} - Y_c^{'} - Z_c^{'}$ is the camera current frame. $F_o^{'} : X_o^{'} - Y_o^{'} - Z_o^{'}$ is a subsidiary frame which is parallel to frameF_o. We define $\beta = \gamma = 0$ at camera desired frame, where the planar object is both centered and parallel to the image plane of camera. In addition, two subsidiary camera frames are defined in convenience when describing the motion of camera mounted on robot end-effector. $X_c - Y_c - Z_c$ is the camera current frame which is parallel to $X_c^{'} - Y_c^{'} - Z_c^{'}$ and the origin of $X_c - Y_c - Z_c$ coincides with that of $X_0 - Y_0 - Z_0$. Similarly, $X_o^{'} - Y_o^{'} - Z_o^{'}$ the camera desired frame which is parallel to $X_0 - Y_0 - Z_0$ and their origins coincide. In case of binary image, $a_{d,00}$ indicate the image areas at the desired position. Hence $a_{c,00}$ indicates the image area for the parallel case with camera desired frame and $a_{c,\beta\gamma}$ for the non-parallel case respectively. It is well known that $a_{c,00}$ is invariant to all cameras' motions except the translation along the optical axis of camera, i.e., $a_{c,00}$ only varies with translation along optical axis of camera frame). If A_r is the area of the object, we have the following equation when the object and image planes are parallel:

$$A_r = \frac{a_{d,00}}{Z_d^2} = \frac{a_{c,00}}{Z_i^2} \quad \text{i.e.,} Z_i = Z_d\sqrt{\frac{a_{c,00}}{a_{d,00}}}$$

Assuming $a_{c,\beta\gamma}$can be used as the approximate value of $a_{c,00}$, we obtain:

$$Z_i = Z_d\sqrt{\frac{a_{c,\beta\gamma}}{a_{d,00}}}$$

Since $a_{c,\beta\gamma} = m^{'}_{00,\beta\gamma}$ can be directly measured from the current image, such that the current depth can be estimated as:

$$Z_i = Z_d\sqrt{\frac{a_{c,\beta\gamma}}{a_{d,00}}} \tag{23}$$

The equation of planar object is expressed in the current camera frame $F_c^{'}$:

$$Z_c^{'} = A^{'}X_c^{'} + B^{'}Y_c^{'} + Z_i^{'} \tag{24}$$

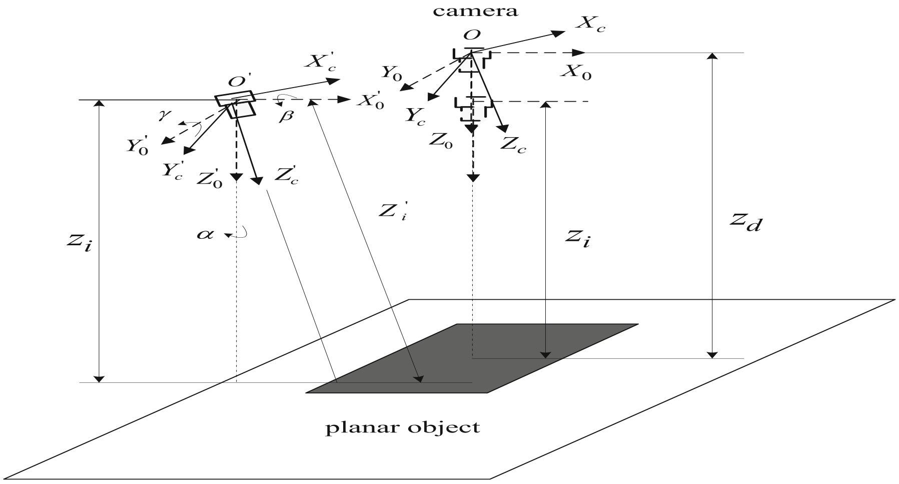
On the other hand, $Z_c^{'}$ can be expressed as a continuous function of its image coordinates x and y [2]

$$\frac{1}{Z_c^{'}} = Ax + By + C \tag{25}$$

From Eqs. 23 and 24, one obtains

$$A = -\frac{A^{'}}{Z_i^{'}}, \quad B = -\frac{B^{'}}{Z_i^{'}}, \quad C = \frac{1}{Z_i^{'}} \tag{26}$$

Fig. 6 Depth estimation



It is noticed that the equation of the planar object expressed in frame $F_0^{'}$ is given by $Z_0^{'} = Z_i$. Meanwhile, the rotation matrix from $F_c^{'}$ to $F_0^{'}$ is denoted as

$$R_{xyz} = R_z(\alpha)R_y(\gamma)R_x(\beta) = \begin{bmatrix} c\gamma c\alpha & (s\beta s\gamma c\alpha - c\beta s\alpha) & (c\beta s\gamma c\alpha + s\beta s\alpha) \\ c\gamma s\alpha & (s\beta s\gamma s\alpha + c\beta c\alpha) & (c\beta s\gamma s\alpha - s\beta c\alpha) \\ -s\gamma & s\beta c\gamma & c\beta c\gamma \end{bmatrix}$$

where $c\alpha$is shorthand for$\cos\alpha$, $s\alpha$for$\sin\alpha$, and so on. Suppose that S is a point on planar object and $S\left(X_0^{'}, Y_0^{'}, Z_0^{'}\right)$ and $S(X_c, Y_c, Z_c)$ are the coordinates expressed in frame$F_0^{'}$ and current camera frame $F_{\mathrm{c}}^{'}$ respectively. One has

$$\begin{bmatrix} X_0^{'} \\ Y_0^{'} \\ Z_0^{'} \end{bmatrix} = \begin{bmatrix} c\gamma c\alpha & (s\beta s\gamma c\alpha - c\beta s\alpha) & (c\beta s\gamma c\alpha + s\beta s\alpha) \\ c\gamma s\alpha & (s\beta s\gamma s\alpha + c\beta c\alpha) & (c\beta s\gamma s\alpha - s\beta c\alpha) \\ -s\gamma & s\beta c\gamma & c\beta c\gamma \end{bmatrix} \begin{bmatrix} X_c^{'} \\ Y_c^{'} \\ Z_c^{'} \end{bmatrix}$$

Because of the fact that$Z_0^{'} = Z_i$, one obtains

$$Z_0^{'} = Z_i = -s\gamma X_c^{'} + s\beta c\gamma Y_c^{'} + c\beta c\gamma Z_c^{'}$$

$$Z_c^{'} = \frac{s\gamma}{c\beta c\gamma} X_c^{'} - \frac{s\beta}{c\beta} Y_c^{'} + \frac{Z_i}{c\beta c\gamma} \tag{27}$$

From Eqs. 23 and 27, one has

$$A^{'} = \frac{s\gamma}{c\beta c\gamma}, B^{'} = -\frac{s\beta}{c\beta}, Z_i^{'} = \frac{Z_i}{c\beta c\gamma} \tag{28}$$

From Eqs. 25 and 27, one obtains

$$A = -\frac{s\gamma}{Z_i}, \quad B = \frac{\mathrm{s}\beta c\gamma}{Z_i}, \quad C = \frac{c\beta c\gamma}{Z_i}$$

A, B and $C$are used to calculate the interaction matrix.

5 IBVS Controller and Stability Analysis

5.1 IBVS Controller

As mentioned in Section 2, the image features are chosen as follows

$$s = \begin{bmatrix} x_g & y_g & a & m_x & m_y & \alpha \end{bmatrix}^T$$

Based on chosen image features, a proportional control law is given by

$$u = -K\hat{J}_{image}^{-1}\Delta s \tag{29}$$

where u is the camera screw velocity sent to robot controller, K is the proportional gain or error convergence rate (K>0), $\Delta s$is the error of image features and $\hat{J}_{image}$ is the estimated interaction matrix. The desired image features are denoted as

$$s_d = \begin{bmatrix} x_{gd} & y_{gd} & a_d & m_{xd} & m_{yd} & \alpha_d \end{bmatrix}^T$$

Thus the errors of image features are $\Delta s = s - s_d$.

5.2 Stability Analysis

Define V $= \frac{1}{2}\Delta s^T \Delta s$ as a Lyapunov function candidate, where $\Delta s$is the error vector of image features. The time derivative of the Lyapunov function is

$$\dot{V} = \Delta s^T \dot{\Delta s} = \Delta s^{\mathrm{T}}(-K J_{image}\hat{J}_{image}^{-1})\Delta s = -K\Delta s^T(J_{image}\hat{J}_{image}^{-1})\Delta s \tag{30}$$

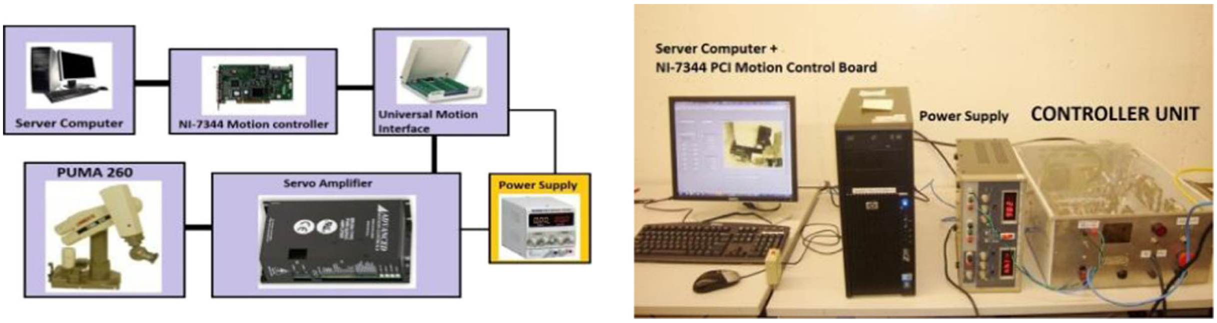


Fig. 7 PUMA 260 hardware components and connections of controller unit

where J_{image} is interaction matrix; $\hat{J}_{image}$ is estimated interaction matrix based on the proposed algorithm. Since the interaction matrix related to the chosen image features is accurate, it is reasonable to assume $J_{image} \cong \hat{J}_{image}$. Hence we have $J_{image}\hat{J}_{image}^{-1} \cong I$, which is an identity matrix (a positive definite matrix). Thus, we have $\dot{V} \leq 0$. By using Lyapunov stability theorem, we can infer that the system is stable and the local convergence is guaranteed when we have accurate enough.

5.3 Summary of the Proposed Algorithm

- **Acquisition of data sets for training NN** In the training area, the images of the object are taken at each integer β, γ and then the image moment invariantsc_1, c_2, c_3, c_4 and the image area $a_{d,\beta\gamma}$ are computed. At this stage, six desired image features s_d at position ($\beta = \gamma = 0$) are also obtained.
- ***NN*training** At this stage, based on the data sets in the first stage, the NN with input of image moment invariantsc_1, c_2, c_3, c_4 is trained as shown in Fig. 2.
- **System integration** The trained NN can be integrated into the robotic control system and IBVS control signal is calculated by Eq. 23. It is noticed that $\hat{\beta}$, $\hat{\gamma}$ (the output of NN in Fig. 2) are used to compute two virtual image moments m_x, m_y as well as to estimate the parameters A, B and C in Section 4. This implies that the equation of planar object can be obtained on-line and thus the overall interaction matrix is updated during visual servoing. As a result, the control performance of visual servoing is improved.

6 Experimental Results

The proposed algorithm has been validated on the robotic visual servoing system consisting of a 6 DOF robot and a camera installed on the end effector. The robot is PUMA 260 [22], which has been retrofitted with motion controller unit shown in Fig. 7. And the camera is JAI CM-030 GE [23]. The image processing, image moment calculation and IBVS controller are implemented in Labview environment in the server computer. In the following experiments, the sampling time is set as 50 ms, and the convergence rate K is tuned as 0.6 by trial and error method. The desired position and two initial positions of camera are shown in Fig. 8. Two objects are tested: a simple centered

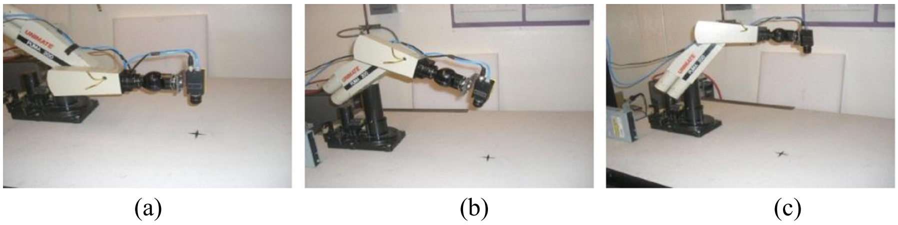

(a) (b) (c)

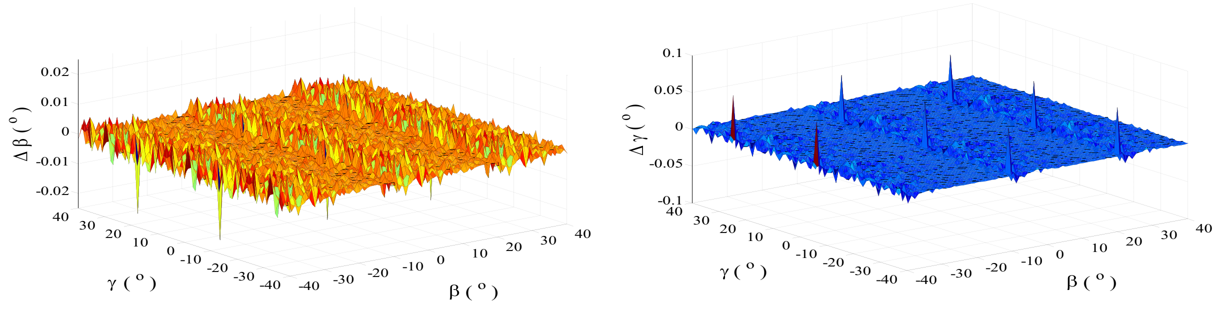
Fig. 8 Desired position and two initial position: **a** Desired position of camera **b** Initial position 1 of camera **c** Initial position 2 of camera

Table 1 Data set used for training NN

Angles(°)		Image moment invariant			
β	γ	c_1	c_2	c_3	c_4
−40	−40	−2.51	8.64	0.86	378.76
−40	−20	−2.44	9.89	0.76	42.12
−40	0	−1.94	6.83	1.79	18.34
−40	20	−204	7.58	1.35	110.89
−40	40	−2.21	9.64	0.96	218.23
−30	−40	−1.78	2.31	1.28	16.55
−30	−20	−2.03	6.33	5.28	33.54
−30	0	−1.83	5.67	3.31	56.79
−30	20	−1.93	4.31	1.17	127.66
−30	40	−38.07	42.36	26.86	308.41
−20	−40	−1.67	1.86	1.09	95.81
−20	−20	−1.84	5.07	1.27	65.21
−20	0	−2.19	9.25	4.16	10.76
−20	20	−2.04	7.41	1.56	42.42
−20	40	−1.56	4.32	1.15	13.01
−10	−40	−1.67	2.38	1.19	15.31
−10	−20	−1.43	2.35	1.17	15.55
−10	0	−1.78	2.31	1.28	16.55
−10	20	−2.03	6.33	5.28	13.54
−10	40	−1.72	2.04	1.11	14.57
0	−40	−1.471	1.14	1.26	40.11
0	−20	−1.60	1.59	1.41	41.28
0	0	−1.98	11.51	19.93	70.01
0	20	−1.76	1.42	1.36	43.28
0	40	−1.52	1.03	1.14	43.01
10	−40	−1.66	2.37	1.15	15.11
10	−20	−1.44	2.45	1.16	15.57
10	0	−1.76	2.37	1.25	16.65
10	20	−2.08	6.43	5.56	13.63
10	40	−1.75	2.12	1.14	14.65
20	−40	−1.65	1.89	1.12	95.86
20	−20	−1.86	5.69	1.21	65.19
20	0	−2.25	9.31	4.31	10.76
20	20	−2.12	7.37	1.78	42.48
20	40	−1.58	4.38	1.17	13.06
30	−40	−1.76	2.40	1.31	16.57
30	−20	−2.11	6.36	5.29	33.58
30	0	−1.88	5.69	3.35	56.78
30	20	−1.99	4.35	1.23	127.58
30	40	−38.11	42.39	26.87	308.67
40	−40	−2.58	8.69	0.87	378.74
40	−20	−2.46	9.87	0.79	42.18
40	0	−1.93	6.82	1.75	18.38
40	20	−210	7.59	1.38	110.81
40	40	−2.18	9.67	0.93	218.56

Table 2 Neural network estimation results

Desired angles (°)		Image moment invariant				Estimated NN (°)		Errors (°)	
β	γ	c_1	c_2	c_3	c_4	$\hat{\beta}$	$\hat{\gamma}$	$\Delta\beta$	$\Delta\gamma$
40.5	−40.5	−2.31	9.54	0.97	406.71	−40.52	−40.51	0.02	0.01
−40.5	−20.5	−2.21	9.34	0.92	203.86	−40.62	−20.52	0.12	0.02
−40.5	0.5	−1.94	6.83	1.79	18.34	−40.49	0.62	−0.01	−0.12
−40.5	20.5	−2.24	9.04	0.94	401.65	−40.53	20.49	0.03	0.01
−40.5	40.5	−2.27	9.09	0.96	408.83	−40.70	40.49	0.20	0.01
−30.5	−40.5	−16.23	3.36	12.11	16.57	−30.49	−40.47	−0.01	−0.04
−30.5	−20.5	−2.24	4.78	5.64	30.54	−30.52	−20.42	0.02	−0.08
−30.5	0.5	−2.03	6.33	5.28	33.54	−30.42	−0.52	−0.08	0.02
−30.5	20.5	−1.93	4.31	1.17	27.66	−30.49	20.51	−0.01	−0.01
−30.5	40.5	−32.07	28.36	26.86	308.41	−30.44	40.50	−0.06	0
−20.5	−40.5	−1.77	1.86	1.09	90.81	−20.56	−40.34	0.06	−0.16
−20.5	−20.5	−1.84	5.07	1.27	65.21	−20.44	−20.52	−0.06	0.02
−20.5	0.5	−2.19	9.25	4.16	10.76	−20.43	0.50	−0.07	0
−20.5	20.5	−2.04	7.41	1.56	42.42	−20.41	20.51	−0.09	−0.01
−20.5	40.5	−1.56	4.32	1.15	93.01	−20.50	40.51	0	−0.01
−10.5	−40.5	−1.65	2.35	146	17.42	−10.49	−40.51	−0.01	0.01
−10.5	−20.5	−15.9	2.49	1.41	15.89	−10.53	−20.52	0.03	0.02
−10.5	0.5	−1.78	2.31	1.25	16.55	−10.56	−0.47	0.06	−0.03
−10.5	20.5	−2.03	2.33	328	33.54	−10.54	20.57	0.04	0.07
−10.5	40.5	−1.78	2.31	1.28	16.55	−10.49	40.51	−0.01	−0.01
0.5	−40.5	−1.46	1.11	1.34	40.18	0.34	−40.49	0.16	−0.01
0.5	−20.5	−1.76	1.42	1.36	41.28	0.46	−20.47	0.04	−0.03
0.5	0.5	−1.98	11.51	19.93	47.03	0.51	0.52	−0.01	−0.02
0.5	20.5	−1.66	1.47	1.28	43.23	0.49	20.51	0.01	−0.01
0.5	40.5	−1.51	1.07	1.11	43.09	0.51	40.51	−0.01	−0.01



(a) (b)

Fig. 9 Neural network estimation error **a** for β **b** for γ

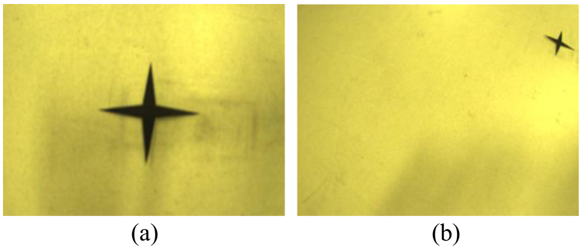


Fig. 10 Desired image (**a**) and initial image (**b**) of Task 1

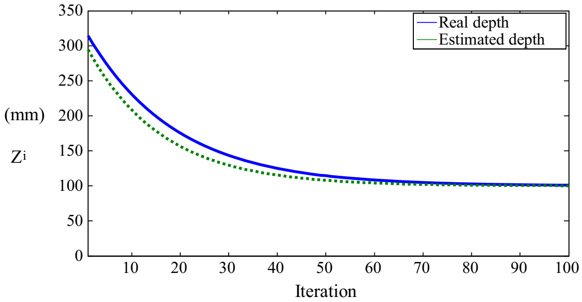


Fig. 11 Depth estimation result of Task 1

symmetrical planar object (with star shape) and a complex nonsymmetrical planar object (with whale shape). The design of NN mapping the relationship between moment invariantsc_1, c_2, c_3, c_4 and β, $\gamma$is a key step. The number of hidden neurons in the hidden layer is set as 40. The activation function $f_1(\bullet)$used for hidden layer nodes is hyperbolic tangent sigmoid, and the activation function $f_2(\bullet)$used for the output layer is linear function.

The acquisition system configuration of data sets for training NN is shown in Fig. 4, which consists of a camera JAI CM-030 GE and a planar object. At the desired position, the object plane is parallel to the image plane (i.e., $\beta = \gamma = 0$), and the depth of object centroid $Z_d = 100$ (mm). The ranges of rotational angles $\beta$and γaround the camera desired frame are ±40°respectively. The training area is divided into small grids by (1°×1°) in β and γdirections. The images of the two shapes of planar objects are taken at each integer βandγwithin the training area. Then the data sets for training NN are computed. Table 1 shows the portion of data sets for training NN, and Table 2 shows the estimation results of the rotational angles around axes$x$ and y of camera desired frame by the trained NN. In Table 2, $\Delta\beta = \beta - \hat{\beta}$ and $\Delta\gamma = \gamma - \hat{\gamma}$ represent the estimation errors of β and γ respectively. Figs. 9–10 demonstrate the estimation errors of β andγ in the training area respectively. From Figs. 9a and b, it is clear that the maximal absolute errors of estimation are$|\Delta\beta_{\max}| = 0.2°$ $|\Delta\gamma_{\max}| = 0.16°$. In order to validate the proposed scheme, the tasks with two different initial camera positions are carried out. In the following experiments, we choose$c_x = c_y = 1$.

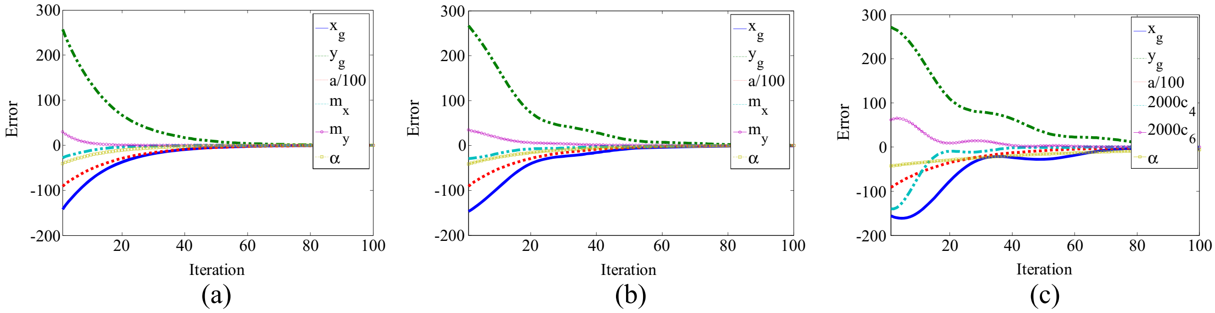
The camera desired position and two initial positions in the world reference frame are given as

$$T_d = \begin{bmatrix} 1 & 0 & 0 & 0.45 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{i1} = \begin{bmatrix} 0.6860 & -0.2359 & 0.0027 & 0.2481 \\ -0.1276 & -0.7430 & 0.2926 & -0.3491 \\ 0.1024 & 0.3878 & -0.6562 & 0.6183 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{i2} = \begin{bmatrix} 0.7754 & 0.2311 & 0.2234 & 0.3354 \\ 0.1056 & -0.6712 & 0.310 & 0.2241 \\ 0.2571 & -0.2051 & -0.7607 & 0.3145 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 3 Initial values and desired values of image features for star shape object in Tasks 1 and 2

Image features	Desired value	Initial value for Task 1	Errors of Task 1	Initial value for Task 2	Errors of Task 2
x_g (pixel)	0	−152	−152	156	156
y_g (pixel)	0	275	275	−164	−164
a($pixel^2$)	10138	495	−9643	235	−9903
m_x(deg)	0	−30	−30	38	38
m_y(deg)	0	36	36	−36	−36
α(deg)	0	−43.6	−43.6	−70.4	−70.7
c_4[5]	−0.0656	0.0034	0.069	0.0018	0.0674
c_6[5]	0.0236	0.0526	0.029	−0.0443	−0.0679



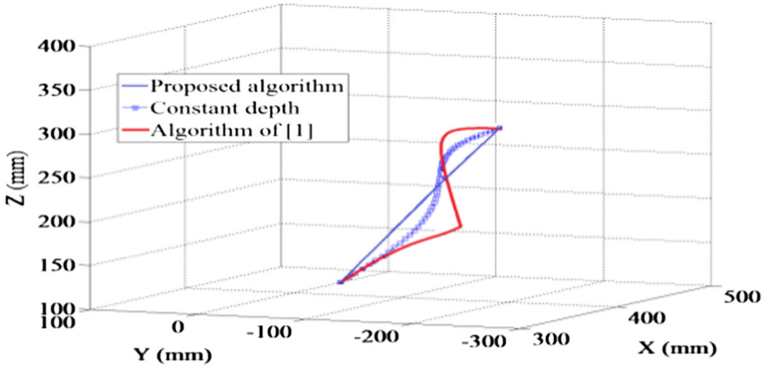
(a)

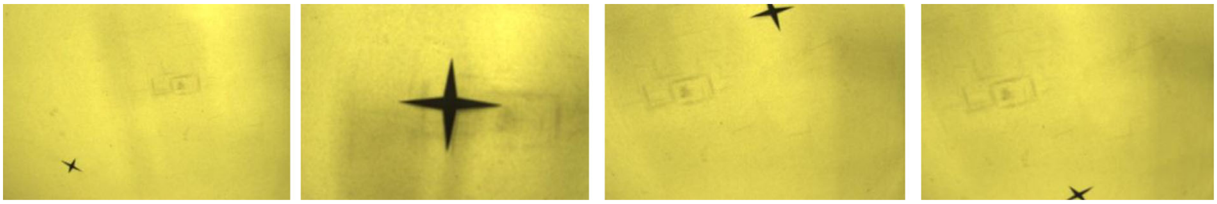
(b)

(c)

Fig. 12 Experimental results of image feature errors in Task 1: **a** with the proposed algorithm **b** with the constant depth **c** with the algorithm in [5]

Fig. 13 Camera trajectories in 3D robot base frame





(a)

(b)

(c)

(d)

Fig. 14 Initial image (**a**) and desired image (**b**) of Task 2, **c** and **d** the images when object gets out of the field of view of camera

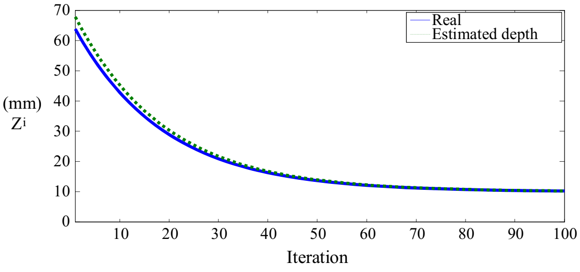


Fig. 15 Depth estimation result of Task 2

As expected, at the desired position,$A = B = 0$, $C = -\frac{1}{Z_g}$, the interaction matrix is rewritten as

$$J_{image} = \begin{bmatrix} -1/Z_g & 0 & x_{gvz} & x_{g\omega x} & x_{g\omega y} & y_g \\ 0 & -1/Z_g & y_{gvz} & y_{g\omega x} & y_{g\omega y} & -x_g \\ 0 & 0 & 2a/Z_g & 3ay_g & -3ax_g & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

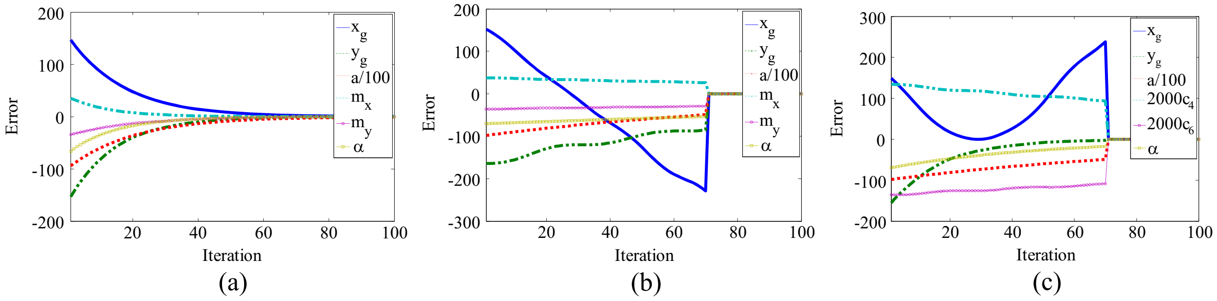


Fig. 16 Experimental results of image feature errors in Task 2: **a** with the proposed algorithm **b** with the constant depth **c** with the algorithm in [5]

Fig. 17 Camera trajectories in 3D robot base frame

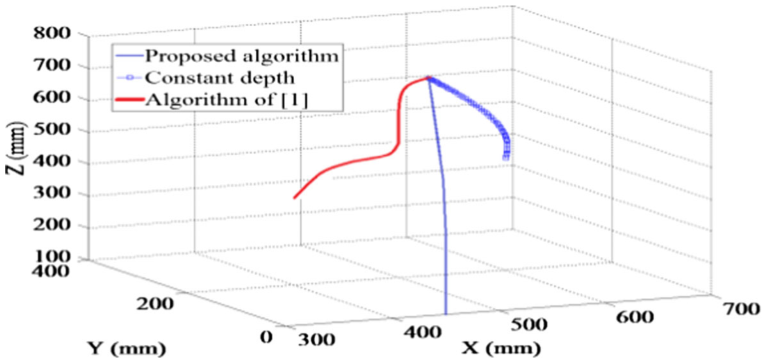


Table 4 The initial and the desired values of image features for whale shape object in Tasks 3 and 4

Image features	Desired value	Initial value for Task 3	Errors of Task 3	Initial value for Task 4	Errors of Task 4
x_g (pixel)	0	−147	−147	154	154
y_g (pixel)	0	267	267	−164	-164
a($pixel^2$)	16655	1148	−15507	558	−16097
m_x(deg)	0	−30	−30	38	38
m_y(deg)	0	36	36	−36	−36
α(deg)	0	45.7	45.7	−49	−49
c_9[5]	0.0514	−0.1346	−0.186	0.0225	−0.0289
c_{10}[5]	0.1059	0.0374	−0.0685	0.0584	−0.0475

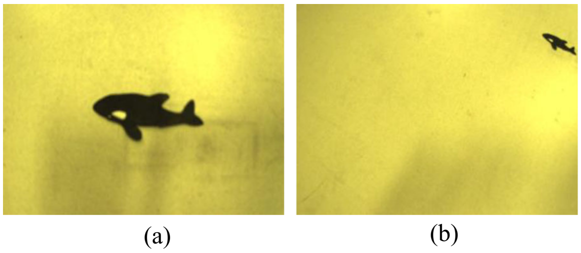


Fig. 18 Desired image (**a**) and initial image (**b**) of Task 3

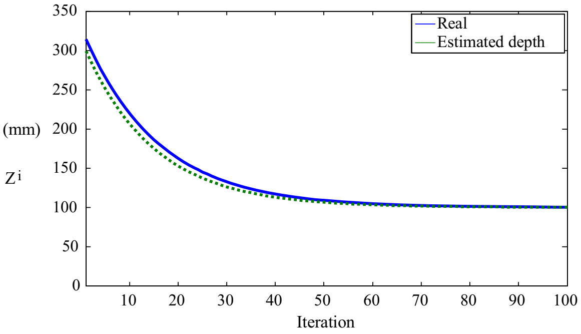


Fig. 19 Depth estimation result of Task 3

6.1 Star Shape Object

Two tasks with different camera initial positions for the star shape object have been carried out. The initial values and the desired values of image features are shown in Table 3. The experimental results are shown in Figs. 10–17. The interaction matrix at the desired position is given by

$$\gamma = \begin{bmatrix} -0.131 & 0 & 0 & 0 & -1.41 & 0 \\ 0 & -0.615 & 0 & 0.22 & 0 & 0 \\ 0 & 0 & 0.098 & 0.0033 & 0.051 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

It is noticed that its condition number is 1.612, which is very satisfactory. The condition number is a measure of stability or sensitivity of a matrix (or the linear system it represents) to numerical operations [25]. Normally, the best condition number is 1. Therefore if the condition number is too big, we may not be able to trust the results of inversion of an ill-conditioned interaction matrix which may lead to bad performance or the failure of visual servoing.

In Task 1, Fig. 10a and b show the desired and initial images respectively. Fig. 11 shows the depth estimation result. Fig. 12a shows the image feature errors by using the proposed algorithm. It is shown that the proposed algorithm obtains pure exponential decrease of image feature errors. The good convergence performance attributes to the fact that the accurate interaction matrix is used in the control scheme. Fig. 12b shows the experimental results by using the proposed algorithm with constant depth. Fig. 12c shows the experimental results by using the algorithm of [5]. The camera trajectories in 3D robot base frame are shown in Fig. 13. As expected, the camera trajectory in 3D robot base frame by the proposed algorithm is a straight line. Whereas, the proposed algorithm with constant depth leads to a curved camera trajectory, and the algorithm of [5] leads a complex curved camera trajectory in 3D robot base frame.

The large displacement from camera initial to desired position is set in Task 2. Fig. 14a and b show the desired and initial images respectively. Figs. 15–17 show the experimental results of the proposed algorithm. It is shown in Fig. 16a that the proposed algorithm obtains exponential decrease of image feature errors due to the fact that the accurate interaction matrix is used in the control scheme. Fig. 16b shows the experimental results achieved by using the proposed algorithm with constant depth. It is demonstrated that the trajectories of the vertices of the star object in image plane get out of the FOV of camera and thus leads to the failure of visual servoing. Fig. 16c shows the experimental results achieved by using the algorithm of [5], which also leads to the failure of visual servoing in the case of large displacement. Fig. 14c and d show the image that the star object gets out of the FOV of camera. The camera trajectories in 3D robot base frame are shown in Fig. 17. The proposed algorithm with

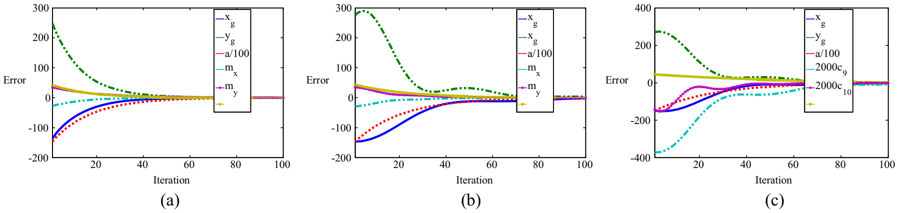


Fig. 20 Experimental results of image feature errors in Task 3: **a** with the proposed algorithm **b** with the constant depth **c** with the algorithm in [5]

Fig. 21 Camera trajectories in 3D robot base frame

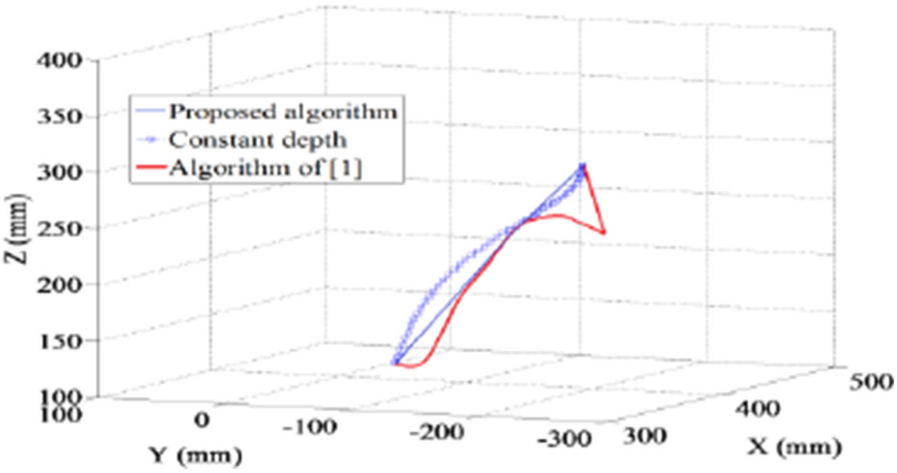
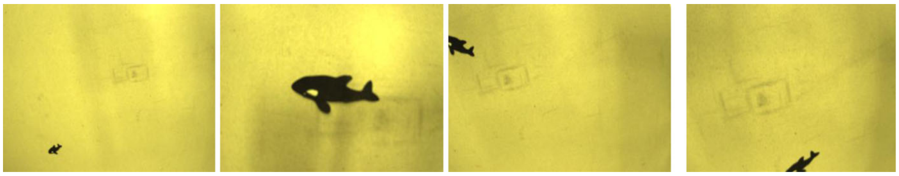


Fig. 22 Initial image (**a**) and desired image (**b**) of Task 4, **c** and **d** the images when the object gets out of the view of camera



(a) (b) (c) (d)

Fig. 23 Depth estimation result of Task 4

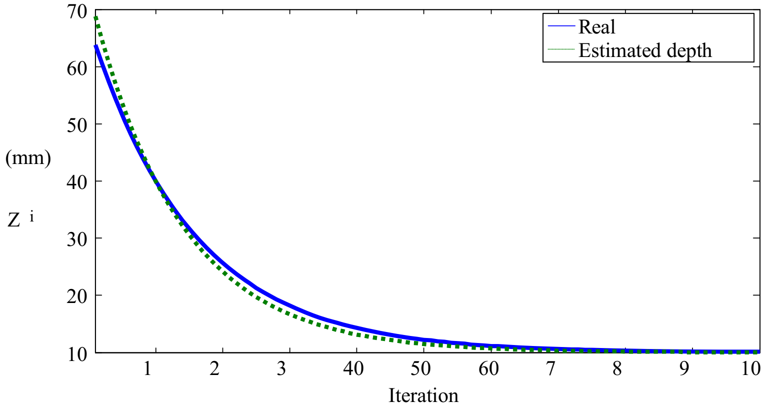
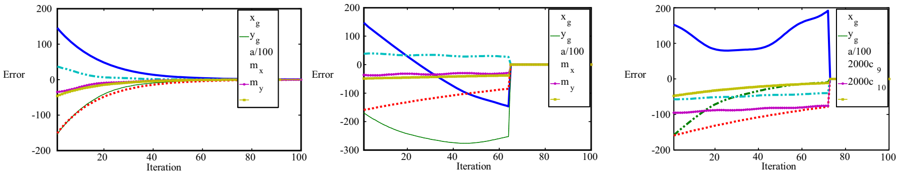


Fig. 24 Experimental results of image feature errors in Task 3: **a** with the proposed algorithm **b** with the constant depth **c** with the algorithm in [5]



(a) (b) (c)

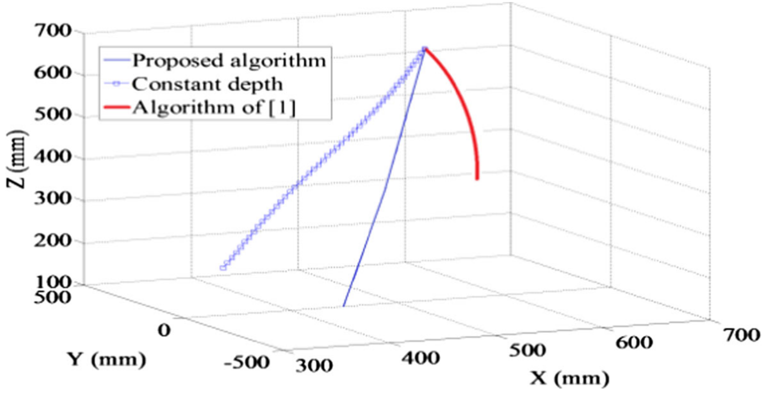


Fig. 25 Camera trajectories in 3D robot base frame

constant depth and the algorithm of [5] fail to drive the camera to approach the desired position in 3D robot base frame.

6.2 Whale Shape Object

Two tasks with different camera initial positions for the whale shape object have been carried out. The initial values and the desired values of image features are shown in Table 4. The experimental results are shown in Figs. 18–25. The interaction matrix at the desired position is given by

$$\gamma = \begin{bmatrix} -0.56 & 0 & 0 & 0 & -0.55 & 0 \\ 0 & -0.38 & 0 & 0.65 & 0 & 0 \\ 0 & 0 & 0.75 & -0.22 & -1.15 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

It is noticed that its condition number is 1.98, which is satisfactory.

In Task 3, Figs. 18a and b show the desired and initial images respectively. Fig. 19 shows the depth estimation result. Fig. 20a shows that the proposed algorithm obtains exponential decrease of image feature errors since the accurate interaction matrix is used in the control scheme. Fig. 20b shows the experimental results of the proposed algorithm with constant depth and of the algorithm of [5]. The camera trajectories in 3D robot base frame are shown in Fig. 21. The camera trajectory in 3D robot base frame by the proposed algorithm is a straight line. The proposed algorithm with constant depth leads to curved camera trajectory, and the algorithm of [5] leads to complex curved camera in 3D robot base frame.

In Task 4, Fig. 22a and b show the initial and desired images respectively. Fig. 23 shows the depth estimation result. Fig. 24a shows that the proposed algorithm obtains exponential decrease of image feature errors since accurate interaction matrix is used in the control scheme. Fig. 24b and c shows the experimental results of the proposed algorithm with constant depth and of the algorithm in [5] respectively. The camera trajectories in 3D robot base frame are shown in Fig. 25. As expected, the camera trajectory of the proposed algorithm in 3D robot base frame is a straight line. Both the proposed algorithm with constant depth and the algorithm of [5] lead to the failure of visual servoing.

The experimental results show that the proposed algorithm can achieve exponential decrease of image feature errors for different shape planar objects and from different camera initial positions. The errors of image features approach to zero exponentially during visual servoing and the camera reaches the desired position from different initial positions by using the proposed algorithm. The experimental results also demonstrate that the proposed algorithm outperforms the proposed algorithm by setting depth constant, which implies that accurate depth estimation online plays key role in visual servoing. Since the proposed scheme can provide the accurate interaction matrix with decoupled form, the proposed scheme achieves better performance in visual servoing compared with the other two algorithms in this paper.

7 Conclusion and Future Work

This paper presents a new scheme to solve the problem of decoupling the rotational velocities around x and y axes of camera frame in robotic visual servoing systems. Based on the proposed scheme, the depth of target object is estimated online, which results in the accurate interaction matrix. Meanwhile, the proposed scheme provides the interaction matrix with particular decoupled form. As result, the singularity is avoided and the local stability of visual servoing system is improved remarkably. The robustness with respect to calibration errors, the global stability analysis, and the visibility constraints on controller design of visual servoing systems will be studied in the future work.

Acknowledgments This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC), Canada and Support of Research Thesis of Faculty of Engineering and Computer Science, Concordia University.

References

1. Zhao, Y.M., Xie, W.F., Wang, T.T.: Neural network-based image moments for visual servoing of planar objects. In: Proceedings of ASME/IEEE Int. Conference on Advanced Intelligent Mechatronics, pp. 268–274 (2012)
2. Chaumette, F.: Image moments: A general and useful set of features for visual servoing. IEEE Trans. Robot. **20**(4), 713–723 (2004)
3. Chaumette, F., Hutchinson, S.: Visual servo control, Part I: Basic approaches. IEEE Robot. Autom. Mag. **13**(4), 82–90 (2006)
4. Chaumette, F., Hutchinson, S.: Visual servo control, Part II: Advanced approaches. IEEE Robot. Autom. Mag. **14**(1), 109–118 (2007)
5. Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Trans. Robot. **21**(6), 1116–1127 (2005)
6. Hu, M.K.: Visual Pattern Recognition by Moment Invariants. IRE Trans. Info. Theory, IT-8, 179–187 (1962)
7. Mukundan, P., Ramakrishnan, K.R.: Moment Functions in Image Analysis: Theory and Application. World Scientific, Singapore (1998)
8. Prokop, R.J., Reeves, A.P.: A survey of moments-based techniques for unoccluded object representation. Graph. Model. Image Process. **54**(5), 438–460 (1992)
9. Chaumette, F.: Potential problems of stability and convergence in image-based and position-based visual servoing. The Confluence of Vision and Control, vol. 237, pp. 6678. Springer-Verlag, New York (1998)
10. Malis, E., Chaumette, F., Boudet, S.: 2-1/2 D visual servoing. IEEE Trans. Robot. Autom. **5**(2), 238–250 (1999)
11. Corke, P.I., Hutchinson, S.: A new partitioned approach to image based visual servo control. IEEE Trans. Robot. Autom. **17**(4), 507–515 (2001)
12. Feddema, J.T., Lee, C.S.G., Mitchell, O.R.: Automatic selection of image features for visual servoing of a robot manipulator. In: Proceedings of IEEE Int. Conference on Robotics and Automation, vol. 2, pp. 832–837 (1989)
13. Nelson, B.J., Khosla, P.K.: The resolvability ellipsoid for visual servoing. In: Proceedings of IEEE Int. Conference on Computer Vision and Pattern Recognition, pp. 829–832 (1994)
14. Krupa, A., Gangloff, J.: Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser pointers. In: Proceedings of IEEE Int. Conference on Robotics and Automation, pp. 3769–3774 (2002)
15. Xie, W.F., Li, Z., Tu, X.W., Perron, C.: Switching control of image based visual servoing with laser pointer in robotic manufacturing systems. IEEE Trans. Ind. Electron. **56**(2), 520–529 (2009)
16. Wells, G., Torras, C.: Assessing Image Features for Vision-Based Robot Positioning. J. Intell. Robot. Syst. **30**, 95–118 (2001)
17. Liu, S.N., Xie, W.F., Su, C.Y.: Image-based visual servoing using improved image moments. In: Proceedings of IEEE Int. Conference on Information and Automation, pp. 577–582 (2009)
18. Yalcin, B., Ohnishi, K.: Infinite-model neural networks for motion control. IEEE Trans. Ind. Electron. **56**(8), 2933–2994 (2009)
19. Gadoue, S.M., Giaouris, D., Finch, J.W.: Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers. IEEE Trans. Ind. Electron. **8**, 3029–3039 (2009)
20. Xia, C.L., Gao, C., Shi, T.N.: A Neural-network-identifier and Fuzzy-controller-based algorithm foe dynamic decoupling control of Permanent-magnet spherical motor. IEEE Trans. Ind. Electron. **57**(8), 2868–2877 (2010)
21. Cotton, N.J., Wilamowski, B.M.: Compensation of nonlinearities using neural networks implemented on inexpensive microcontrollers. IEEE Trans. Ind. Electron. **58**(3), 733–740 (2011)
22. PUMA 260 Reference Manual, UNIMATE INC (1984)
23. CM-030 GE/CB-030 GE Users Manual, Document Version 2.0 (2011)
24. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Englewood Cliffs (1998)
25. Cheney, E.W., Kincaid, D.R.: Numerical Mathematics and Computing, Cengage Learning (2007)