Partially Decoupled Image-Based Visual Servoing Using Different Sensitive Features

De Xu, *Senior Member, IEEE*, Jinyan Lu, Peng Wang, *Member, IEEE*, Zhengtao Zhang, and Zize Liang

***Abstract*—A new image-based visual servoing method based on sensitive features is presented to separately realize the position control and orientation control. Line features are used for the orientation control because of their sensitivities to rotational motions. Point features and area size features are employed to realize the position control since area size features are very sensitive to the objects' depths. The translations resulting from rotational motions are introduced into the position control as the compensation in order to eliminate the influence of the camera's motions on the point features. The depths for all active features are estimated via interaction matrices, features variations, and the executed camera motions. The proposed method can keep the tracked objects in the camera's field of view in the visual servoing process. In addition, the determination methods of the interaction matrices for point, line, and area size features are proposed. Comparing to the traditional method, the proposed determination method of the interaction matrix for line is independent from the parameters of the plane containing the line. Experimental results verify the effectiveness of the proposed methods.**

***Index Terms*—Depth estimation, interaction matrix, sensitive features, visual control, visual servoing.**

I. Introduction

VISUAL servoing has been attracting the attention of researchers from the robotics and automation society for decades [1], [2], [26]–[32]. As early as 1996, Hutchinson *et al.* [1] surveyed the visual servoing approaches in their tutorial. The visual servoing approaches are classified as two categories: 1) image-based visual servoing (IBVS) and 2) position-based visual servoing, according to the features belonging to image space or Cartesian space. The IBVS methods have the advantages of high accuracy over the position-based visual servoing methods, but the position-based methods have merits that the control laws are designed in Cartesian space and the robot's trajectory is more reasonable and steady. The image-based methods require online computation of the image Jacobian matrix or the interaction matrix which depends on the distance from the camera to the target particularly in a monocular vision system. The large errors in image Jacobian matrix will result in servoing failure. Malis *et al.* [3] proposed a 2.5-D visual servoing method to combine the characteristics of the image-based and position-based visual servoing methods. The position control is with the IBVS method to achieve high accuracy. The orientation control is realized via the position-based visual servoing method in order to avoid servoing failure in the procedure of orientation adjustment. It ensures the convergence of control system in the whole task space [4]. Many researchers have put in a great deal of effort to develop visual servoing approaches or apply them in practice [5]–[8]. For example, Alkhalil and Doignon [5] proposed a stereo visual servoing method based on the state vectors built from the two-view affine geometry and the visual measurements. Liu *et al.* [2] presented an adaptive visual servoing method based on the depth-independent interaction matrix using common image features, in which the depths are estimated online for the features. Mills *et al.* [6] developed an IBVS approach for a fixed wing unmanned aerial vehicle (UAV) to track the power lines, which are taken as parallels and their pose is estimated via the line interaction matrix. Generally speaking, IBVS methods are paid more attention to because of their high accuracy, needless of 3-D reconstruction and robustness with respect to camera and robot calibration errors [9], [10]. The problem is how to avoid servoing failure becasuse of inadequate camera/robot motions resulting from using variable depths in the interaction matrix in the procedure of orientation adjustment [4]. One of the solutions is to decouple the orientation and translation controls [3], [11], [12]. Malis *et al.* [3] decomposed the Euclidean homography matrix as the sum of a rotation matrix and a translation vector to decouple the rotation and translation. The Euclidean homography matrix is estimated with point features. Furthermore, Benhimane and Malis [24] developed an IBVS method relying on the homography matrix but not requiring any decomposition, and not requiring any estimation of the depth. Tahri *et al.* [11] proposed a decoupled IBVS method to control the three translational degree-of-freedoms (DOFs) via selecting features invariant to rotations. It is a big problem to know the plane equations of triangles in the camera's frame. Andreff *et al.* [12] proposed a visual servoing method based on lines, which partially decouples the orientation and translation controls. The line's orientation is estimated in 3-D Cartesian space in order to realize orientation control. The lines junction is used as feature for translation control. The control law is independent from the depth and asymptotically

Manuscript received August 4, 2016; accepted December 12, 2016. Date of publication January 9, 2017; date of current version July 17, 2017. This work was supported by the National Natural Science Foundation of China under Grant 61227804, Grant 61421004, and Grant 61379097. This paper was recommended by Associate Editor C. J. F. Silvestre.

D. Xu, J. Lu, P. Wang, and Z. Zhang are with the Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: de.xu@ia.ac.cn).

Z. Liang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2016.2641951

stable in the case of the orthogonal trihedron formed by lines. In general cases, a laser pointer is employed to present depth information if the lines cannot form an orthogonal trihedron.

The features used in visual servoing include points, lines, distances, angles, and areas. The interaction matrix for points was deduced in [1] and [13]. It is very common to employ points as features in visual servoing [14]–[17]. For example, the point features and the interaction matrix were used to realize the visual servoing in the joint space for a manipulator with eye-in-hand and eye-to-hand vision systems in [14] and [15], respectively. The edge points and their normal vectors are tracked and matched with ones in CAD model, then the visual tracking is realized via the optimization of matching errors [16]. Point features are sensitive to 6 DOFs, but their sensitivities are different. The sensitivity for the DOF along the optical axis of the camera is lowest. In addition, the variations of point features are coupled with the translational and rotational motions. Line features are very sensitive to rotational motions. Espiau *et al.* [13] also deduced the interaction matrices for straight lines, circles, and spheres. Comport *et al.* [18] deduced the interaction matrix for the distance from a point to a line based on the interaction matrices for points and straight lines, which is widely used for the target tracking in augmented reality. In the determination of the interaction matrix for a straight line, the parameters of the plane containing the line but not containing the optical center are needed. This requirement prevents the line interaction matrix from visual servoing for robots. Some works use line features for visual servoing in other modes. For example, the moment of parallel lines and its interaction matrix were used for visual servoing in power lines checking with a rotary wing UAV [19]. Coutard *et al.* [20] used the vanishing point and angles of parallels as features to estimate the aircraft's pose relative to the aircraft carrier's deck. The interaction matrix for the angles of parallels is presented. In [21], line features were employed to realize the orientation alignment and point features were used for the position alignment. Multiple kinds of features are helpful to improve the performance of visual servoing system. For example, Liu *et al.* [2] employed points, lines, and area features in their work. But they did not distinguish the roles of different features.

The motivation of this paper is to develop a new visual servoing method based on sensitive features. It is a good idea to use different sensitive features to adjust the orientation and position separately. Point features, line features, and area size features are sensitive to different DOFs. They can be appointed different roles in visual servoing system. Line features are very sensitive to rotation, they are used for orientation control. Area size features are very sensitive to objects' depths, they are employed to realize position control combining with point features. The determination of interaction matrices for line and area size features is investigated. A new visual servoing system is designed, whose orientation control is realized using line features and position control is realized using point and area size features. The translation compensation resulting from rotational motions is considered in the position control in order to help keeping the tracked objects in the camera's field of view.

The rest of this paper is organized as follows. The interaction matrices for points, lines, and area sizes are given in Section II. In Section III, the visual servoing system based on sensitive features is proposed. The control law and depth estimation method are presented. In Section IV, experiments and results are provided. Finally, this paper is concluded in Section V.

II. Interaction Matrices for Features

A. Interaction Matrix for Point Feature

The intrinsic parameters of a camera can be described using the pinhole model after the lens distortion is corrected. In context, the plane with the normalized focal length 1 is called the normalized imaging plane. For a point (x_c, y_c, z_c) in the camera's frame, its imaging point on the normalized imaging plane is denoted as $(x_{1c}, y_{1c}, 1)$. Generally, (x_c, y_c, z_c) is unknown for monocular vision. But x_{1c} and y_{1c} can be calculated from the image coordinates and the camera's intrinsic parameters.

The variation of a point feature on the normalized imaging plane is given in (1) when the camera moves [13]

$$\begin{aligned}\begin{bmatrix}\dot{x}_{1c}\\ \dot{y}_{1c}\end{bmatrix} &= \begin{bmatrix}-\frac{1}{z_c} & 0 & \frac{x_{1c}}{z_c} & x_{1c}y_{1c} & -\left(1+x_{1c}^2\right) & y_{1c}\\ 0 & -\frac{1}{z_c} & \frac{y_{1c}}{z_c} & 1+y_{1c}^2 & -x_{1c}y_{1c} & -x_{1c}\end{bmatrix}\\ &\quad\times\begin{bmatrix}v_{ca}\\ \omega_{ca}\end{bmatrix}\\ &= L_x\begin{bmatrix}v_{ca}\\ \omega_{ca}\end{bmatrix}\end{aligned} \tag{1}$$

where $v_{ca} = [v_{cax}, v_{cay}, v_{caz}]^{\mathrm{T}}$ and $\omega_{ca} = [\omega_{cax}, \omega_{cay}, \omega_{caz}]^{\mathrm{T}}$ are the translational and rotational velocities of the camera, L_x is the interaction matrix for a point. The lower index "$1c$" represents the camera's normalized imaging plane.

The camera's intrinsic parameters are required in the calculation of x_{1c} and y_{1c} from the point's image coordinates. In other words, the camera should be calibrated in advance. In addition, the point's depth z_c in the camera's frame must be estimated in order to compute the interaction matrix L_x.

B. Interaction Matrix for Line Feature

The line equation on the normalized imaging plane can be expressed in polar coordinates as

$$x_{1c}\cos\theta + y_{1c}\sin\theta = \rho \tag{2}$$

where θ is the angle between its orthogonal line and x_{1c}-axis of the imaging frame, ρ is the distance from the origin of the imaging frame to the line on the normalized imaging plane.

The imaging frame is established on the normalized imaging plane, as shown in Fig. 1. Its origin O_{1c} is at the intersection of the optical line on this plane. Its x_{1c}- and y_{1c}-axis are parallel to the x_c- and y_c-axis of the camera frame, respectively.

The projection of the line not parallel to the optical axis on the normalized imaging plane is denoted as l_{c1}. The orthogonal line $O_{1c}P_{1c_0}$ is from the origin O_{1c} to the line l_{c1}. The length of

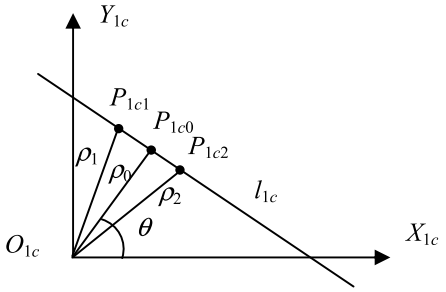


Fig. 1. Projection of a line on the image plane.

the line $O_{1c}P_{1c0}$ is the parameter ρ_0 of the line l_{c1} . The angle betweent the line $O_{1c}P_{1c0}$ and the x_{1c}-axis is the parameter θ of the line l_{c1}. P_{1c0} is the projection of the point P_0. In other words, P_0 is the point on the line corresponding to the point P_{1c0}. It can be found from Fig. 1 that the variation of the line l_{c1} can be obtained from the variations of point P_{1c0} and its neighborhood points.

The coordinates of points P_0 and P_{1c_0} are denoted as (x_{c0}, y_{c0}, z_{c0}) and $(x_{1c_0}, y_{1c_0}, 1)$ in the camera frame, respectively. The relation between polar parameters (ρ_0, θ_0) and Cartesian coordinate (x_{1c_0}, y_{1c_0}) is

$$\rho_0 = \sqrt{x_{1c0}^2 + y_{1c0}^2}, \theta_0 = \arctan\frac{y_{1c0}}{x_{1c0}}. \tag{3}$$

With the time derivative of (3), and after $\rho_0 \cos\theta$ takes the place of x_{1c_0} and $\rho_0 \sin\theta$ takes the place of y_{1c_0}, we have (4) [25], shown at the bottom of the page.

Two points P_{1c_1} and P_{1c_2} are symmetrically selected nearby the point P_{1c_0}. Their parameters satisfy

$$\begin{cases} \theta_1 = \theta + \Delta\theta \\ \theta_2 = \theta - \Delta\theta \\ \rho_1 = \rho_2 \end{cases} \tag{5}$$

where $\Delta\theta$ is the angle increment, it is positive and near zero.

It can be seen from Fig. 1 that the variations of ρ_0, ρ_1, and ρ_2 are very near if the angle increment $\Delta\theta$ is near to zero when the camera moves. Therefore, the variation of ρ_0 can be considered as approximately same as that of ρ. Submitting the polar coordinate of the points P_{1c_1} and P_{1c_2} to (2), we have

$$\begin{cases} \rho_1 \cos\theta_1 \cos\theta + \rho_1 \sin\theta_1 \sin\theta = \rho_0 \\ \rho_2 \cos\theta_2 \cos\theta + \rho_2 \sin\theta_2 \sin\theta = \rho_0. \end{cases} \tag{6}$$

Then the parameter θ expressed with the polar coordinate of the points P_{1c_1} and P_{1c_2} is obtained from (6)

$$\theta = \arctan\frac{\rho_1 \cos\theta_1 - \rho_2 \cos\theta_2}{\rho_2 \sin\theta_2 - \rho_1 \sin\theta_1}. \tag{7}$$

Submitting (5) into the time derivative of (7) and simplification, we have

$$\dot\theta = \frac{1}{2\rho_1}(\dot\rho_2 - \dot\rho_1)\cot\Delta\theta + \frac{1}{2}\left(\dot\theta_1 + \dot\theta_2\right). \tag{8}$$

Submitting the variations of polar coordinate of P_{1c_1} and P_{1c_2} as given in (4) into (8), we have (9). The interaction matrix for line feature is obtained as given in (11) with the combination of (4) and (9)

$$\dot\theta = \begin{bmatrix} L_{\theta vx} & L_{\theta vy} & \frac{1}{2z_{c2}} - \frac{1}{2z_{c1}} & -\rho_1\cos\theta & -\rho_1\sin\theta & -1 \end{bmatrix} \times \begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} \tag{9}$$

where z_{c_1} and z_{c_2} are the z coordinate corresponding to the points P_{1c_1} and P_{1c_2} (see (10) and (11), shown at the bottom of the page).

It can be concluded that $L_{\theta vx} = 0$ and $L_{\theta vy} = 0$ if $z_{c_1} = z_{c_2}$. In this case, (9) is rewritten as

$$\dot\theta = \begin{bmatrix} 0 & 0 & 0 & -\rho_1\cos\theta & -\rho_1\sin\theta & -1 \end{bmatrix} \begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix}. \tag{12}$$

It can be found from (12) that only rotation has contribution to the angle variation of a line on image in the case $z_{c_1} = z_{c_2}$. In addition, z_{c_1} and z_{c_2} are approximately equal if the line is about orthogonal to the camera's optical axis. In this case, rotation has main contribution to the line's angle variation on image.

C. Interaction Matrix for the Distance From Point to Line

The distance from the ith feature point $(x_{cpi}, y_{cpi}, z_{cpi})$ to the jth feature line on the normalized imaging plane is described as

$$d_{ij} = \rho_j - x_{1ci}\cos\theta_j - y_{1ci}\sin\theta_j \tag{13}$$

where (x_{1ci}, y_{1ci}) are the coordinates of the ith feature point $(x_{cpi}, y_{cpi}, z_{cpi})$ on the normalized imaging plane, ρ_j and θ_j are

$$\begin{bmatrix} \dot\rho_0 \\ \dot\theta_0 \end{bmatrix} = \begin{bmatrix} -\frac{\cos\theta_0}{z_{c0}} & -\frac{\sin\theta_0}{z_{c0}} & \frac{\rho_0}{z_{c0}} & \left(1+\rho_0^2\right)\sin\theta_0 & -\left(1+\rho_0^2\right)\cos\theta_0 & 0 \\ \frac{\sin\theta_0}{\rho_0 z_{c0}} & -\frac{\cos\theta_0}{\rho_0 z_{c0}} & 0 & \frac{\cos\theta_0}{\rho_0} & \frac{\sin\theta_0}{\rho_0} & -1 \end{bmatrix} \begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} \tag{4}$$

$$\begin{cases} L_{\theta vx} = \frac{1}{2\rho_1}\left(\frac{\cos\theta_1}{z_{c1}} - \frac{\cos\theta_2}{z_{c2}}\right)\cot\Delta\theta + \frac{1}{2\rho_1}\left(\frac{\sin\theta_2}{z_{c2}} + \frac{\sin\theta_1}{z_{c1}}\right) \\ L_{\theta vy} = \frac{1}{2\rho_1}\left(\frac{\sin\theta_1}{z_{c1}} - \frac{\sin\theta_2}{z_{c2}}\right)\cot\Delta\theta - \frac{1}{2\rho_1}\left(\frac{\cos\theta_1}{z_{c1}} + \frac{\cos\theta_2}{z_{c2}}\right) \end{cases} \tag{10}$$

$$\begin{bmatrix} \dot\rho \\ \dot\theta \end{bmatrix} = \begin{bmatrix} -\frac{\cos\theta}{z_{c0}} & -\frac{\sin\theta}{z_{c0}} & \frac{\rho}{z_{c0}} & \left(1+\rho^2\right)\sin\theta & -\left(1+\rho^2\right)\cos\theta & 0 \\ L_{\theta vx} & L_{\theta vy} & \frac{1}{2z_{c2}} - \frac{1}{2z_{c1}} & -\rho_1\cos\theta & -\rho_1\sin\theta & -1 \end{bmatrix} \begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} \tag{11}$$

the polar parameters of the jth feature line, d_{ij} is the distance from the ith feature point to the jth feature line.

The following formula is gotten from the time derivative of (13):

$$\dot{d}_{ij} = \dot{\rho}_j - \dot{x}_{1ci}\cos\theta_j - \dot{y}_{1ci}\sin\theta_j + x_{1ci}\sin\theta_j\dot{\theta}_j - y_{1ci}\cos\theta_j\dot{\theta}_j. \tag{14}$$

Substituting (1) and (11) into (14), then

$$\begin{aligned}\dot{d}_{ij} &= \begin{bmatrix} L_{dijvx} & L_{dijvy} & L_{dijvz} & L_{dij\omega x} & L_{dij\omega y} & 0 \end{bmatrix}\begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} \\ &= L_{dij}\begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix}\end{aligned} \tag{15}$$

where

$$\begin{cases} L_{dijvx} = \left(-1/z_{c0j} + 1/z_{cpi}\right)\cos\theta_j \\ L_{dijvy} = \left(-1/z_{c0j} + 1/z_{cpi}\right)\sin\theta_j \\ L_{dijvz} = \rho_j/z_{c0j} - \left(x_{1ci}\cos\theta_j - y_{1ci}\sin\theta_j\right)/z_{cpi} \\ L_{dij\omega x} = \left(1+\rho_j^2\right)\sin\theta_j - x_{1ci}y_{1ci}\cos\theta_j - \left(1+y_{1ci}^2\right)\sin\theta_j \\ \qquad - \rho_j\cos\theta_j\left(x_{1ci}\sin\theta_j - y_{1ci}\cos\theta_j\right) \\ L_{dij\omega y} = -\left(1+\rho_j^2\right)\cos\theta_j + \left(1+x_{1ci}^2\right)\cos\theta_j \\ \qquad + x_{1ci}y_{1ci}\sin\theta_j - \rho_j\sin\theta_j\left(x_{1ci}\sin\theta - y_{1ci}\cos\theta_j\right) \end{cases} \tag{16}$$

where $z_{c_{0j}}$ is the z coordinate of the point P_{0j} on the jth line, z_{cpi} is the z coordinate of the ith feature point in the camera frame.

It can be concluded from (15) that the camera's rotation around z_c-axis does not change the distance from the ith feature point to the jth feature line. In addition, if $z_{c_{0j}} = z_{cpi}$, then $L_{dijvx} = 0$ and $L_{dijvy} = 0$. In this case, the camera's translations along x_c- and y_c-axis do not change the distance from the ith feature point to the jth feature line.

D. Interaction Matrix for Area Size

Suppose the lengths of the two edges of a rectangle are d_{11} and d_{12} on the normalized imaging plane. The variation of the rectangle's area size is defined as

$$\dot{S} = d_{11}\dot{d}_{12} + d_{12}\dot{d}_{11} \tag{17}$$

where S is the rectangle's area size on the normalized imaging plane.

Of course, the edge length can be taken as the distance from a point to a line. Substituting (15) into (17), we have

$$\dot{S} = \left(L_{d11}d_{12} + L_{d12}d_{11}\right)\begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} = L_s\begin{bmatrix} v_{ca} \\ \omega_{ca} \end{bmatrix} \tag{18}$$

where $L_{d_{11}}$ and $L_{d_{12}}$ are the interaction matrices of the distances listed in (15), L_s is the interaction matrix for area size.

Comparing (15) and (17), we have the conclusion that the variation of the area size is much larger than the line's variation if $d_1 > 1$ and $d_2 > 1$ in the condition of the same movements of the camera.

III. Image-Based Visual Servoing With Separate Design for Orientation and Position Control

A. Control Law Design

It can be found from (9) and (10) that the influences of the camera's translations on the angles of the feature lines are very little if the lines are approximately perpendicular to the camera's optical axis. The variations of the angles of the feature lines are mainly resulted from the camera's rotations. It is very natural to select the angles' variations of the feature lines as the sensitive features to represent the rotational motions of the camera. The camera's rotation can be controlled using the errors between the desired and current angles of at least three lines. From (12), we have

$$\begin{aligned}\begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_m \end{bmatrix} &= \begin{bmatrix} -\rho_1\cos\theta_1 & -\rho_1\sin\theta_1 & -1 \\ \vdots & \vdots & \vdots \\ -\rho_m\cos\theta_m & -\rho_m\sin\theta_m & -1 \end{bmatrix}\begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} \\ &= L_{lw}\begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix}\end{aligned} \tag{19}$$

where L_{lw} is the interaction matrix for the angles' variations of the feature lines.

Formula (19) is rewritten as

$$\begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} = L_{lw}^{+}\begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_m \end{bmatrix} \tag{20}$$

where L_{lw}^{+} is the pseudo-inverse matrix of L_{lw}.

The relationship of the error variation and the error is given in (21) in order to let the error reduce with exponential speed

$$\dot{e} = -\lambda e. \tag{21}$$

Combining (20) and (21), the orientation control law is designed as

$$\begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} = -\lambda_w L_{lw}^{+}\begin{bmatrix} \Delta\theta_1 \\ \vdots \\ \Delta\theta_m \end{bmatrix} \tag{22}$$

where λ_w is the rotation adjustment factor.

It is sure that point's imaging coordinates will vary when the camera moves with any of the 6 DOFs. Especially, the camera's rotation has apparent influence on points' imaging coordinates if the point is far from the camera. Therefore, the compensation of rotational motions is introduced into the position control in order to eliminate the influence of the camera's rotation on point's imaging coordinates. Generally, the camera's translation in limited specified distance will not lose the tracked objects. So the tracked objects can be kept in the camera's field of view after the compensation. The variations of point features on the normalized imaging plane are deduced

$$\begin{bmatrix} {}^1\dot{S}_w \\ \vdots \\ {}^k\dot{S}_w \end{bmatrix} = \begin{bmatrix} {}^1L_{d1wx}{}^1d_2 + {}^1L_{d2wx}{}^1d_1 & {}^1L_{d1wy}{}^1d_2 + {}^1L_{d2wy}{}^1d_1 \\ \vdots & \vdots \\ {}^kL_{d1wx}{}^kd_2 + {}^kL_{d2wx}{}^kd_1 & {}^kL_{d1wy}{}^kd_2 + {}^kL_{d2wy}{}^kd_1 \end{bmatrix} \begin{bmatrix} \omega_{cax} \\ \omega_{cay} \end{bmatrix} = L_{sw} \begin{bmatrix} \omega_{cax} \\ \omega_{cay} \end{bmatrix} \tag{26}$$

$$\begin{bmatrix} v_{capx} & v_{capy} & v_{capz} \end{bmatrix}^{\mathrm{T}} = -\lambda_v L_{xsv}^{+} \begin{bmatrix} \Delta x_{1c1} & \Delta y_{1c1} & \cdots & \Delta x_{1cn} & \Delta y_{1cn} & \Delta S_1 - {}^1\dot{S}_w T_c & \cdots & \Delta S_k - {}^k\dot{S}_w T_c \end{bmatrix}^{\mathrm{T}} \tag{27}$$

$$L_{xsv} = \begin{bmatrix} -\frac{1}{z_{cp1}} & 0 & \frac{x_{1c1}}{z_{cp1}} \\ 0 & -\frac{1}{z_{cp1}} & \frac{y_{1c1}}{z_{cp1}} \\ \vdots & \vdots & \vdots \\ -\frac{1}{z_{cpn}} & 0 & \frac{x_{1cn}}{z_{cpn}} \\ 0 & -\frac{1}{z_{cpn}} & \frac{y_{1cn}}{z_{cpn}} \\ {}^1L_{d1vx}{}^1d_2 + {}^1L_{d2vx}{}^1d_1 & {}^1L_{d1vy}{}^1d_2 + {}^1L_{d2vy}{}^1d_1 & {}^1L_{d1vz}{}^1d_2 + {}^1L_{d2vz}{}^1d_1 \\ \vdots & \vdots & \vdots \\ {}^kL_{d1vx}{}^kd_2 + {}^kL_{d2vx}{}^kd_1 & {}^kL_{d1vy}{}^kd_2 + {}^kL_{d2vy}{}^kd_1 & {}^kL_{d1vz}{}^kd_2 + {}^kL_{d2vz}{}^kd_1 \end{bmatrix} \tag{28}$$

$$\begin{bmatrix} v_{cax} & v_{cay} & v_{caz} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} v_{capx} & v_{capy} & v_{capz} \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} v_{cawx} & v_{cawy} & v_{cawz} \end{bmatrix}^{\mathrm{T}}. \tag{29}$$

from (1)

$$\begin{bmatrix} \dot{x}_{1c1w} \\ \dot{y}_{1c1w} \\ \vdots \\ \dot{x}_{1cnw} \\ \dot{y}_{1cnw} \end{bmatrix} = \begin{bmatrix} x_{1c1}y_{1c1} & -\left(1+x_{1c1}^2\right) & y_{1c1} \\ 1+y_{1c1}^2 & -x_{1c1}y_{1c1} & -x_{1c1} \\ \vdots & \vdots & \vdots \\ x_{1cn}y_{1cn} & -\left(1+x_{1cn}^2\right) & y_{1cn} \\ 1+y_{1cn}^2 & -x_{1cn}y_{1cn} & -x_{1cn} \end{bmatrix} \begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} = L_{xw} \begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} \tag{23}$$

where L_{xw} is the interaction matrix between point features and the camera's rotation, $(\dot{x}_{1ciw}, \dot{y}_{1ciw})$ are the variations of the ith point feature on the normalized imaging plane resulting from the camera's rotation.

The compensation of rotational motions is computed from the variations of point features on the normalized imaging plane and their interaction matrix

$$\begin{bmatrix} v_{cawx} \\ v_{cawy} \\ v_{cawz} \end{bmatrix} = -L_{xv}^{+} \begin{bmatrix} \dot{x}_{1c1w} \\ \dot{y}_{1c1w} \\ \vdots \\ \dot{x}_{1cnw} \\ \dot{y}_{1cnw} \end{bmatrix} = -L_{xv}^{+} L_{xw} \begin{bmatrix} \omega_{cax} \\ \omega_{cay} \\ \omega_{caz} \end{bmatrix} \tag{24}$$

where L_{xv} is the interaction matrix between point features and the camera's translation

$$L_{xv} = \begin{bmatrix} -1/z_{cp1} & 0 & x_{1c1}/z_{cp1} \\ 0 & -1/z_{cp1} & y_{1c1}/z_{cp1} \\ \vdots & \vdots & \vdots \\ -1/z_{cpn} & 0 & x_{1cn}/z_{cpn} \\ 0 & -1/z_{cpn} & y_{1cn}/z_{cpn} \end{bmatrix} \tag{25}$$

where z_{cpi} is z_c coordinate of the ith point feature.

The conditions $x_{1c} << 1$ and $y_{1c} << 1$ are satisfied since the limitation of the camera's view angle. Therefore, the variations of point features on the normalized imaging plane are more sensitive to the translations along x_c- and y_c-axis than the translations along z_c-axis. It is necessary to introduce areas' sizes as features to indicate the translations along z_c-axis.

The variations of areas' sizes resulting from the camera's rotation are given in (26), shown at the top of the page. Combining (18) and (26), the position control law is designed as given in (27), shown at the top of the page, where L_{xsv} is the interaction matrix (see (28), shown at the top of the page) of point and area size features with respective to the camera's rotation, T_c is the control cycle time, λ_v is the translation adjustment factor. The translational velocity of the camera is obtained via merging the results computed from (24) and (27), and it is given in (29).

B. Depth Estimation

The depth of feature point can be estimated with its variation on the normalized imaging plane and the camera's velocity after the camera moves. From (1), we have

$$z_{cpi} = \frac{1}{2} \frac{x_{1ci}v_{caz} - v_{cax}}{\dot{x}_{1ci} - x_{1ci}y_{1ci}\omega_{cax} + \left(1+x_{1ci}^2\right)\omega_{cay} - y_{1ci}\omega_{caz}} + \frac{1}{2} \frac{y_{1ci}v_{caz} - v_{cay}}{\dot{y}_{1ci} - \left(1+y_{1ci}^2\right)\omega_{cax} + x_{1ci}y_{1ci}\omega_{cay} + x_{1ci}\omega_{caz}}. \tag{30}$$

The depth of point $P_{1c_{0j}}$ on feature line can be estimated with the line's parameters and variations on the normalized imaging plane and the camera's velocity after the camera moves. From the first equation in (11), it is deduced,

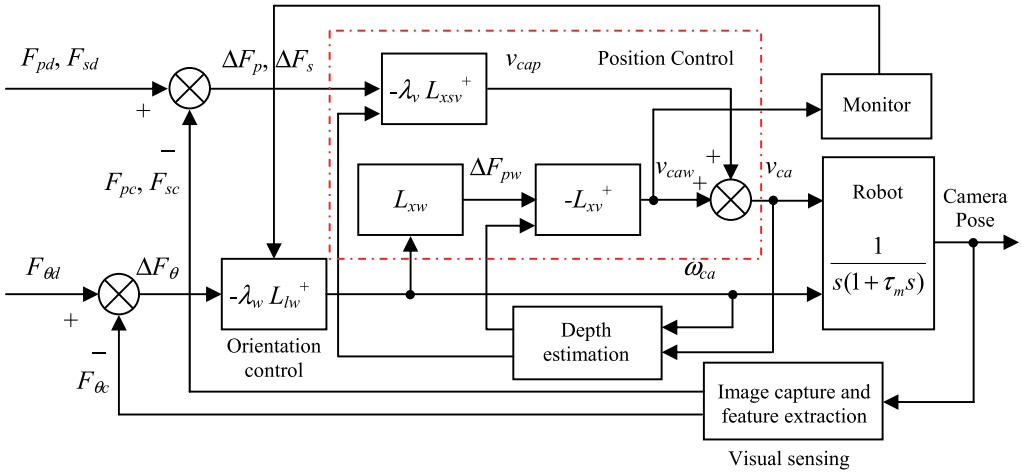


Fig. 2. Control system block diagram.

as given in

$$z_{c0j} = \frac{\rho_j v_{caz} - \cos\theta_j v_{cax} - \sin\theta_j v_{cay}}{\dot{\rho}_j - \left(1+\rho_j^2\right)\sin\theta_j \omega_{cax} + \left(1+\rho_j^2\right)\cos\theta_j \omega_{cay}}. \quad (31)$$

C. Control System Design

The orientation and position control blocks are separately designed in the proposed IBVS system [3], [21], [22]. The proposed visual servoing system consists of six blocks such as orientation control, position control, visual sensing, depth estimation, monitor, and robot. Its block diagram is given in Fig. 2. The orientation control is realized with line features. The angular velocity ω_{ca} is computed according to the desired lines' features $F_{\theta_d} = (\rho_{1_d}, \theta_{1_d}, \ldots, \rho_{\mathrm{md}}, \theta_{\mathrm{md}})$ and current ones $F_{\theta_c} = (\rho_{1c}, \theta_{1c}, \ldots, \rho_{\mathrm{mc}}, \theta_{\mathrm{mc}})$ and the interaction matrix for line features. It is used to control the robot's rotation to adjust the camera's orientation. The control law is as given in (22). The position control is realized with points and area size features. The compensated translational velocity v_{caw} resulting from the camera's rotation is computed according to the angular velocity ω_{ca} with (24). The translational velocity v_{cap} is calculated in (27) with the interaction matrix for points and areas sizes according to the errors between the desired point features $F_{pd} = (x_{1c_{1d}}, y_{1c_{1d}}, \ldots, x_{1\mathrm{cnd}}, y_{1\mathrm{cnd}})$ and current ones $F_{pc} = (x_{1c_{1c}}, y_{1c_{1c}}, \ldots, x_{1\mathrm{cnc}}, y_{1\mathrm{cnc}})$ and the errors between the desired area size features $F_{sd} = ({}^1S_d, \ldots, {}^kS_d)$ and current ones $F_{sc} = ({}^1S_c, \ldots, {}^kS_c)$. The sum v_{ca} of v_{caw} and v_{cap} is considered as the translational velocity to be sent to the robot to adjust the camera's position. The depth estimation block is used to estimate the depths of the feature points using (30) and the points $P_{1c_{01}}$ to $P_{1c_{0m}}$ on the feature lines using (31). The visual sensing block is used to capture images and extract the features, including points, lines, distances, and area sizes. The monitor checks whether the translational velocity is adequate or not. If the compensated translational velocity v_{caw} was too large, then the factor λ_w would be reduced to change it to the acceptable value. The robot is the plant of the control system, which executes the motion commands from the orientation and position control blocks.

D. Stability Analysis

Here, the stability of orientation control is first considered. Let $e_w = [\Delta\theta_1, \ldots, \Delta\theta_m]^{\mathrm{T}}$. Then

$$\dot{e}_w = L_{lw}\omega_{ca} = -\lambda_w L_{lw}\hat{L}_{lw}^+ e_w \quad (32)$$

where $\hat{L}_{lw}^+$ is the estimated pseudo-inverse matrix of L_{lw}.

The candidate Lyapunov function is defined by the squared error norm

$$\mathcal{L} = \frac{1}{2}\|e_w\|^2. \quad (33)$$

The derivative of Lyapunov function is

$$\dot{\mathcal{L}} = e_w^{\mathrm{T}}\dot{e}_w = -\lambda_w e_w^{\mathrm{T}} L_{lw}\hat{L}_{lw}^+ e_w. \quad (34)$$

The global asymptotic stability of the orientation control system is obtained when the following sufficient condition is satisfied:

$$L_{lw}\hat{L}_{lw}^+ > 0. \quad (35)$$

It can be found from Fig. 2, (22), and (35) that the orientation control loop is stable if the pseudo-inverse matrix $\hat{L}_{lw}^+$ is accurately estimated and its rank is 3, and the factor λ_w is adequate. The more accurate the estimated pseudo-inverse matrix $\hat{L}_{lw}^+$ is, the more approximate to identity matrix the $L_{lw}\hat{L}_{lw}^+$ is. The condition (35) is satisfied if the estimated pseudo-inverse matrix $\hat{L}_{lw}^+$ is not too coarse. Hence, the global asymptotic stability of the orientation control system can be obtained if the lines' parameters ρ_j, θ_j, $j = 1, \ldots, m$ are not with coarse errors.

The position control is simplified to a control system similar to the orientation control after the compensation of the camera's rotational motions. It is decoupled from rotation if the compensation is accurate. However, it is inevitable that there exists error in the compensation. The compensation reduces the impact of rotation on translation control to such low a level that the error in compensation can be taken as disturbance. Similarly, the position control loop is stable if $L_{xsv}\hat{L}_{xsv}^+ > 0$. In orther words, the position control loop is stable if the pseudo-inverse matrix $\hat{L}_{xsv}^+$ is accurately estimated and its rank is 3, and the factor λ_v is adequate. Therefore, it is necessary to

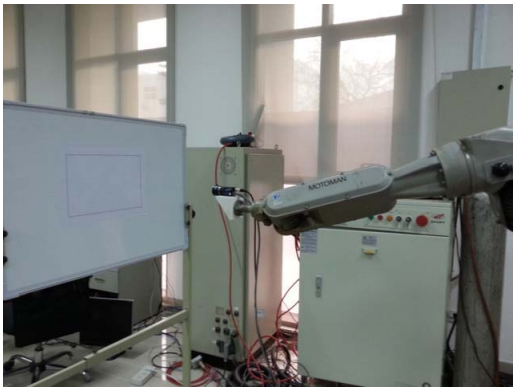


Fig. 3. Experimental system.

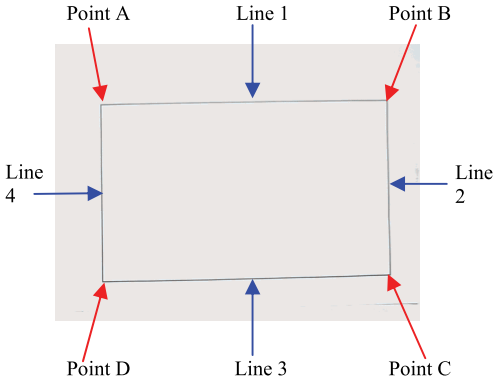


Fig. 4. Desired image and features of the rectangle object.

select adequate features to indicate the motions in order to ensure the stability of the control system.

IV. Experiment and Results

A. Experiment System and Calibration

The experiment system consisted of a manipulator, a camera, a computer, and a whiteboard, as shown in Fig. 3. The manipulator was Motoman UP6, a 6-DOF robot produced by Yaskawa Company. The camera was fixed on the end-effector of the manipulator, which formed an eye-in-hand vision system. The camera was MER-200-14GX produced by Daheng Company, its image size was 1626 × 1238 in pixel. A rectangle was attached on the whiteboard as the object to be tracked. The computer was used to deal with the image processing and visual servoing algorithms.

The camera's intrinsic parameters and the hand–eye relation were calibrated before experiments. The intrinsic parameters model was pinhole model. The 4-order Brown distortion model was adopted for the lens distortion. The hand–eye relation was denoted as the transformation matrix M_{ex} from the end-effector coordinates to the camera coordinates.

B. Approaching Experiment With the Proposed Method

In experiments, the angle error's threshold was 0.5°. The point feature error's threshold was 0.02 mm on the normalized imaging plane. The area size error's threshold was 0.01 mm^2. The orientation adjustment of one step for each direction was limited to 5°. The translation adjustment of one step for each direction was limited to 200 mm. The rotation adjustment factor including the sampling cycle time, $\lambda_w T_c$, was set to 0.6. The translation adjustment factor including the sampling cycle time, $\lambda_v T_c$, was set to 0.5. The adjustments finished once the errors were smaller than the thresholds above. The end-effector's pose and the features in each step were recorded as experimental data.

In experiments, the velocity was replaced with the increment divided by the sampling cycle time since the manipulator only accepted pose commands. It was to be noticed that the camera's motion should be converted to the end-effector's motion with (36) in order to control the manipulator

$$T_e = M_{ex} T_{ca} M_{ex}^{-1} \tag{36}$$

TABLE I
Desired Line Features

Line	Line 1	Line 2	Line 3	Line 4
$\rho(mm)$, $\theta(°)$	0.1898, -90.94	0.3646, -1.1153	0.2269, 88.7526	0.3170, 179.453

TABLE II
Desired Point Features

Point	Point A	Point B	Point C	Point D
x_{1c}, y_{1c} (mm)	-0.3188, -0.1846	0.3608, -0.1958	0.3689, 0.2189	-0.3148, 0.2338

where T_e is the matrix formed with the end-effector's motion, T_{ca} is the matrix formed with the camera's motion.

In the experiment, the end-effector was first moved to the rectangle object to capture the desired image as shown in Fig. 4. The four lines and four corners were selected as the image features. The edges were detected with Canny operator. Then the feature lines were extracted using probabilistic Hough transform, the corners were the lines' intersections calculated from the lines. The points' coordinates on the normalized imaging plane were computed with their image coordinates and the intrinsic parameters of the camera. Then the desired features of the lines and corners were obtained, as listed in Tables I and II. The desired area size of the rectangle on the normalized imaging plane was 0.2013 mm^2.

The end-effector was moved away after the desired image of the object was captured. The end-effector's pose had a large offset from the desired one. As described in Section III, the depths of feature points and lines needed to be estimated in order to control the manipulator's motion with the control law. The camera was actively moved a specified offset (−20 mm, −10 mm, 0 mm, 0°, 0°, 0°) as the first step. Then the depths of feature points and lines were estimated with (30) and (31).

Once the depths of feature points and lines were estimated, the visual servoing could be conducted. In the following steps, the current features, including points, lines, and area were extracted and their errors to desired ones were computed. The orientation adjustment of the camera was computed with (22). Then the translation compensation was computed with (24). The area size variation caused by rotation was calculated with (26). The translation adjustment of the camera to remove

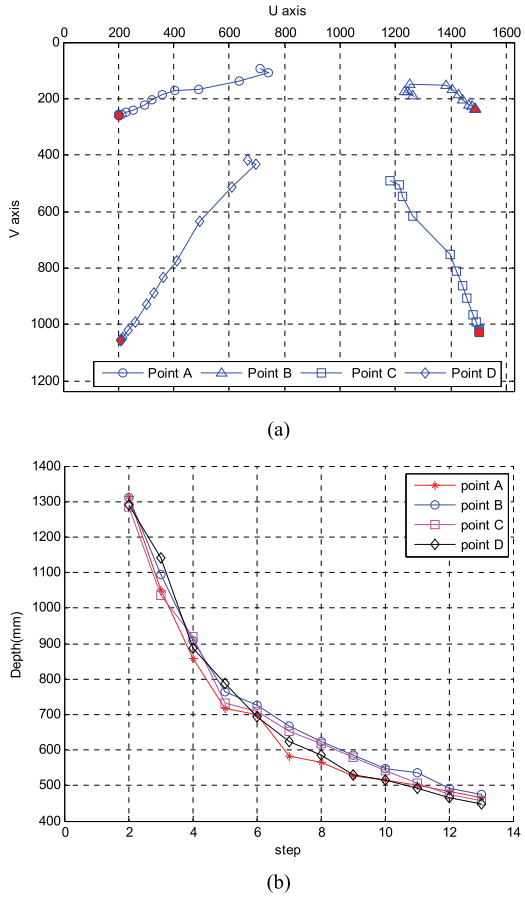


Fig. 5. Feature points trajectories and estimated depths with the proposed method. (a) Trajectories. (b) Estimated depths.

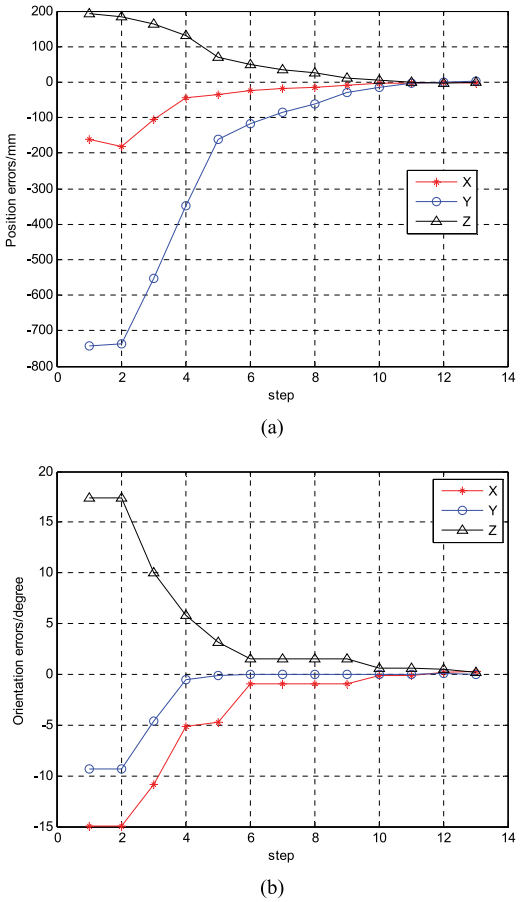


Fig. 6. Position and orientation errors of the end-effector with the proposed method. (a) Position errors. (b) Orientation errors.

the position errors was computed with (27). Then the merged translation adjustment of the camera was obtained with (29).

The experiments were well conducted with the proposed control laws as described above. The end-effector and the camera were well moved to the desired position with desired orientation in experiments. The phenomenon of losing the object did not occur in all experiments. The results in one experiment were shown in Figs. 5–7. Fig. 5(a) shows the feature points' image coordinates. The solid symbols indicated the desired positions. Fig. 5(b) gave the estimated depths of the four points. Fig. 6 gave the position and orientation errors of the end-effector. Fig. 7 was the end-effector's trajectory. The approaching process finished after 13 steps. The errors of the end-effector after the approaching were (−2.53 mm, 2.63 mm, −1.67 mm, 0.26°, −0.04°, 0.20°).

It can be found from Fig. 5(b) that the estimated depths were steady. It can be found from Figs. 6 and 7 that the errors reduced quickly and steadily, the trajectories were reasonable.

C. Comparison Experiments

The 2.5-D visual servoing method in [3] and the traditional IBVS method in [1] were selected to conduct comparison experiments. In comparison experiments, the four corner points were used as the features. The conditions including the initial pose of the manipulator, the point feature error's

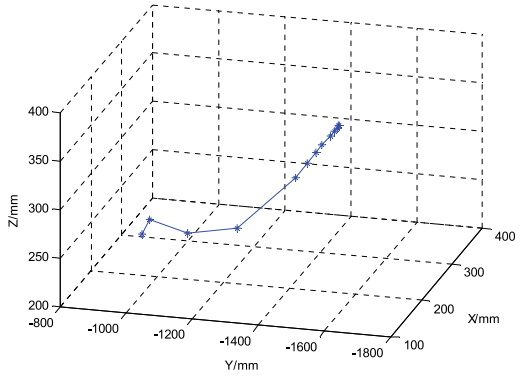


Fig. 7. End-effector trajectory with the proposed method.

threshold, the adjustment limition for each step were as same as ones in above experiment.

In the experiment with the 2.5-D visual servoing method, the Euclidean homogragh matrix was estimeted and decomposed. The desired distance from the camera to the target plane was manully measured. It was 480 mm. Point C was selected as the refernce feature point in translation control. The extended image coordinates were computed according to the image coordinates of the point C and the Euclidean homogragh matrix. The error's threshold for the third element of extended image coordinates was 0.01. The orientation

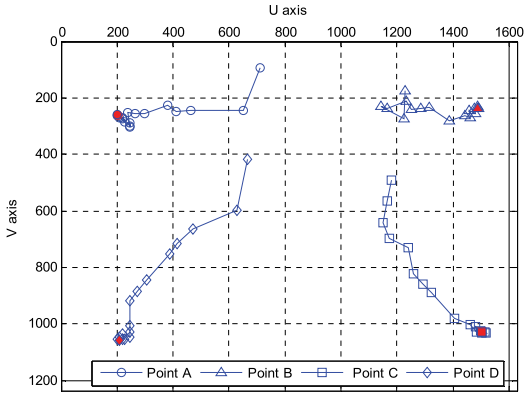
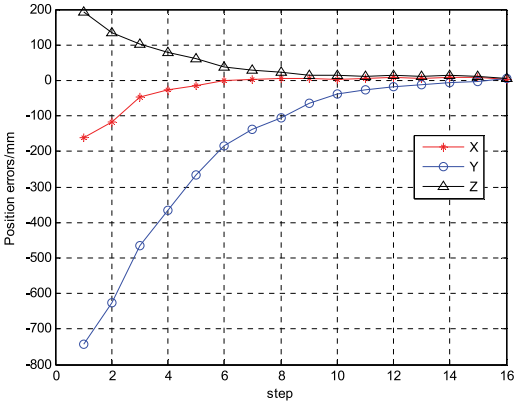
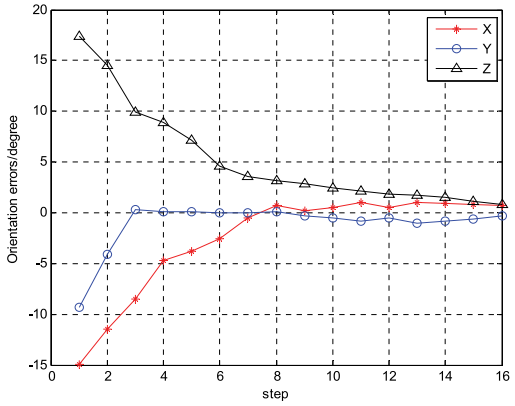


Fig. 8. Feature points trajectories with the 2.5-D visual servoing method.



(a)



(b)

Fig. 9. Position and orientation errors of the end-effector with the 2.5-D visual servoing method. (a) Position errors. (b) Orientation errors.

error's threshold was set to 0.5°. The orientation adjustment factor, including the sampling cycle time, $\lambda_w T_c$, was set to 0.6. The translation adjustment factor, including the sampling cycle time, $\lambda_v T_c$, was set to 0.5. The experimental results were shown in Figs. 8–10. Fig. 8 shows the feature points' image coordinates. The solid symbols indicated the desired positions. Fig. 9 gave the position and orientation errors of the end-effector. Fig. 10 was the end-effector's trajectory. It used 16 steps to finish the approaching. The errors of the end-effector after the approaching were (3.81 mm, 5.41 mm, 5.35 mm, 0.76°, −0.30°, 0.77°). They were slightly larger

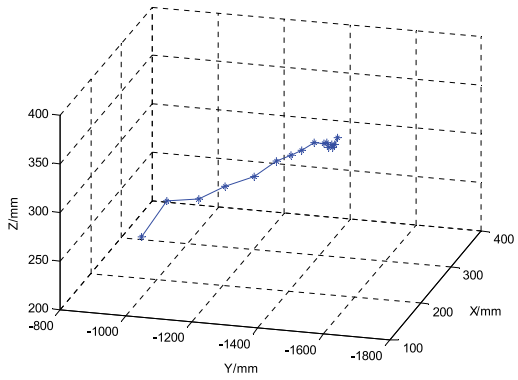


Fig. 10. End-effector trajectory with the 2.5-D visual servoing method.

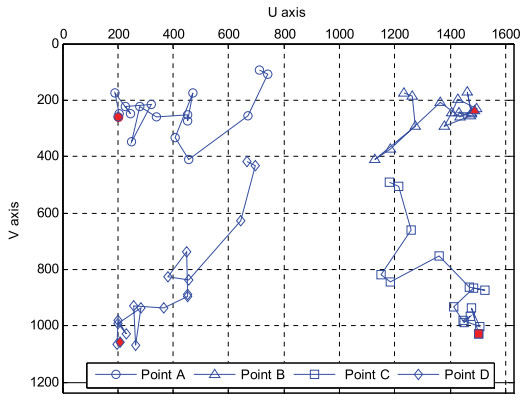


Fig. 11. Feature points trajectories with the traditional IBVS method.

than the errors with the proposed method. It can be found from Fig. 9 that the errors reduced slowly while the errors were small.

In the experiment with the traditional IBVS method, the four corner points were used as the image features too. One active step as same as one in the experiment with the proposed method was conducted to estimate the feature points' depths. The orientation and translation adjustment factors were set to the same values as above experiments. The experimental results were shown in Figs. 11–13. Fig. 11 shows the feature points' image coordinates. The solid symbols indicated the desired positions. Fig. 12 shows the position and orientation errors of the end-effector. Fig. 13 gave the end-effector's trajectory. Sixteen steps were used to finish the approaching. The errors of the end-effector after the approaching were (−14.95 mm, 8.49 mm, 11.44 mm, 1.20°, −1.05°, −1.64°). They were much larger than the errors with the proposed method. It can be found from Fig. 11 that the image features varied in a large range, which may result in visual servoing failure once the image features were out of the image. It can also be found from Fig. 12 that the orientation errors fluctuated, especially the orientation errors around y-axis.

The 2.5-D visual servoing method needs to estimate and decompose the Euclidean homography matrix online in order to decouple the orientation and translation controls. The proposed method does not need to estimate and decompose the Euclidean homography matrix. It compensates the translations resulting from rotational motions to decouple the orientation

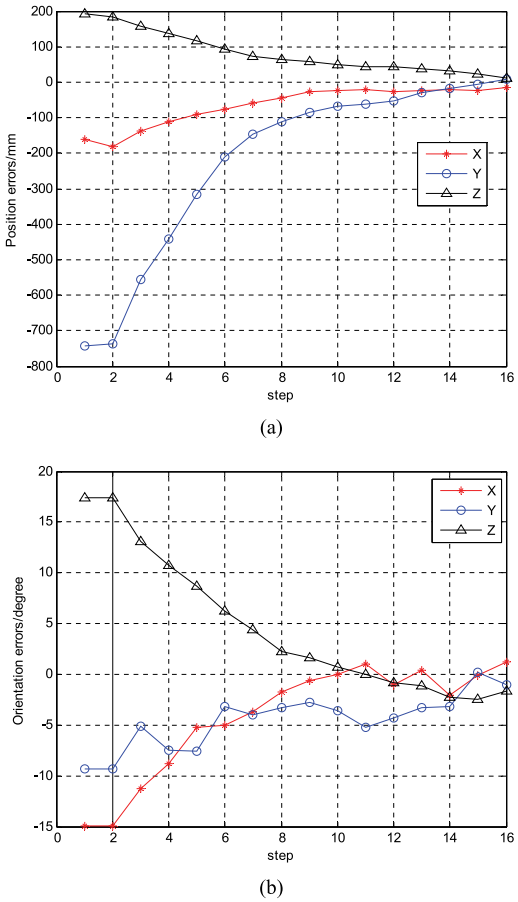


Fig. 12. Position and orientation errors of the end-effector with the traditional image-based servoing method. (a) Position errors. (b) Orientation errors.

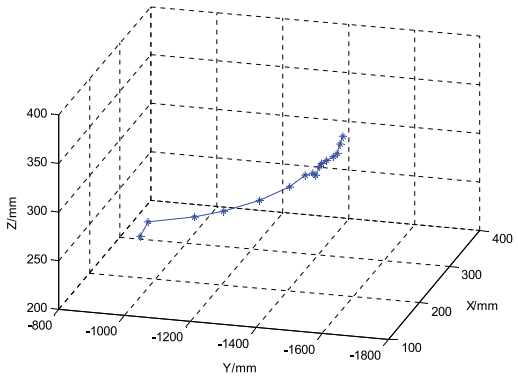


Fig. 13. End-effector trajectory with the traditional IBVS method.

and translation controls. It has better performance than the 2.5-D visual servoing method and its computation is much simple.

As pointed out in [23], a quadrangle pattern is sufficient to ensure that the perspective-*n*-point problem has unique solution. Therefore, three lines at least in which two lines have different directions and four feature points at least that form a quadrangle pattern are necessary to avoid the local minimum in the visual servoing procedure with the proposed method.

V. Conclusion

The main contribution of this paper is the new IBVS method with separately designed position and orientation controllers based on different sensitive features. The orientation control is realized using line features based on the fact that the directions of lines are very sensitive to rotational motions. The translations resulting from rotational motions are introduced into the position control as the compensation in order to eliminate the influence of the camera's rotation on point features. The position controller is designed based on point features and area size features with the considerations that point features are sensitive to the motions perpendicular to the camera's optical axis and area size features are very sensitive to the motions along the camera's optical axis. The advantages of the proposed mothod are as follows. The orientation control is independent of the position control, which has the merit that the position controller and orientation controller can be saperately designed. The translation compensation of the rotations is very helpful to keep the tracked object in the camera's field of view. It avoids estimating and decomposing the Euclidean homography matrix and has better performance than the 2.5-D visual servoing method. It is easy to select adequate features to ensure the stability of the control system.

The interaction matrices for point, line, and area size features are investigated. The proposed determination method of the interaction matrix for line is independent from the parameters of the plane containing the line, which is more easy to get the interaction matrix comparing to the traditional method. The depths for all active features, including points and lines are estimated via interaction matrices, features' variations and the executed camera motions. It is very helpful to apply line features to control the camera's orientation in visual servoing. Experimental results verify the effectiveness of the proposed methods.

In future, our emphasis will be put on the application of the proposed methods in practice.

References

[1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 651–670, Oct. 1996.

[2] Y.-H. Liu, H. Wang, W. Chen, and D. Zhou, "Adaptive visual servoing using common image features with unknown geometric parameters," *Automatica*, vol. 49, no. 8, pp. 2453–2460, 2013.

[3] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[4] D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," Comput. Vis. Act. Percept. Lab., Stockholm, Sweden, Tech. Rep. ISRN KTH/NA/P-02/01-Sen, CVAP259, 2002.

[5] F. Alkhalil and C. Doignon, "Stereo visual servoing with decoupling control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 1671–1676.

[6] S. Mills, N. Aouf, and L. Mejias, "Image based visual servo control for fixed wing UAVs tracking linear infrastructure in wind," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 5769–5774.

[7] R. Pieters, Z. Ye, P. Jonker, and H. Nijmeijer, "Direct motion planning for vision-based control," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1282–1288, Oct. 2014.

[8] D. Kruse, J. T. Wen, and R. J. Radke, "A sensor-based dual-arm telerobotic system," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 4–18, Jan. 2015.

[9] Z. Liu, F. Wang, and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled

dynamic," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 10, pp. 1301–1312, Oct. 2015.
[10] J. Gao, A. A. Proctor, Y. Shi, and C. Bradley, "Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2323–2334, Oct. 2016.
[11] O. Tahri, Y. Mezouar, F. Chaumette, and P. Corke, "Decoupled image-based visual servoing for cameras obeying the unified projection model," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 684–697, Aug. 2010.
[12] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *Int. J. Robot. Res.*, vol. 21, no. 8, pp. 679–699, 2002.
[13] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.
[14] Y. Liu and H. Wang, "An adaptive controller for image-based visual servoing of robot manipulators," in *Proc. World Congr. Intell. Control Autom.*, Jinan, China, 2010, pp. 988–993.
[15] H. Wang, Y.-H. Liu, and W. Chen, "Visual tracking of robots in uncalibrated environments," *Mechatronics*, vol. 22, no. 4, pp. 390–397, 2012.
[16] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 932–946, Jul. 2002.
[17] Y.-H. Liu, H. Wang, C. Wang, and K. K. Lam, "Uncalibrated visual servoing of robots using a depth-independent interaction matrix," *IEEE Trans. Robot.*, vol. 22, no. 4, pp. 804–817, Aug. 2006.
[18] A. I. Comport, E. Marchand, M. Pressigout, and F. Chaumette, "Real-time markerless tracking for augmented reality: The virtual visual servoing framework," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 4, pp. 615–628, Jul./Aug. 2006.
[19] H. Xie, A. Lynch, and M. Jagersand, "IBVS of a rotary wing UAV using line features," in *Proc. Can. Conf. Elect. Comput. Eng.*, Toronto, ON, Canada, 2014, pp. 1–6.
[20] L. Coutard, F. Chaumette, and J.-M. Pflimlin, "Automatic landing on aircraft carrier by visual servoing," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, 2011, pp. 2843–2848.
[21] S. Liu, D. Xu, D. Zhang, and Z. Zhang, "High precision automatic assembly based on microscopic vision and force information," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 1, pp. 382–393, Jan. 2016.
[22] D. Xu, M. Tan, Y. Liu, and J. Wang, "Control strategy for a low cost manipulator to transport and align IC mask-plates," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 5, pp. 1018–1027, Sep. 2009.
[23] D. Xu, Y. F. Li, and M. Tan, "A general recursive linear method and unique solution pattern design for the perspective-*n*-point problem," *Image Vis. Comput.*, vol. 26, no. 6, pp. 740–750, 2008.
[24] S. Benhimane and E. Malis, "Homography-based 2D visual tracking and servoing," *Int. J. Robot. Res.*, vol. 26, no. 7, pp. 661–676, 2007.
[25] F. Chaumette and S. Hutchinson, "Visual servo control, part II: Advanced approaches," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.
[26] P. Jiang and R. Unbehauen, "Robot visual servoing with iterative learning control," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 2, pp. 281–287, Mar. 2002.
[27] K. C. C. Peng, W. Singhose, and P. Bhaumik, "Using machine vision and hand-motion control to improve crane operator performance," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 6, pp. 1496–1503, Nov. 2012.
[28] J. Oyekan, D. Gu, and H. Hu, "Visual imaging of invisible hazardous substances using bacterial inspiration," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1105–1115, Sep. 2013.
[29] V. Bruni and D. Vitulano, "An improvement of kernel-based object tracking based on human perception," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 11, pp. 1474–1485, Nov. 2014.
[30] X. Zhang *et al.*, "Human pose estimation and tracking via parsing a tree structure based human model," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 5, pp. 580–592, May 2014.
[31] Z. Liu, F. Wang, and Y. Zhang, "Adaptive visual tracking control for manipulator with actuator fuzzy dead-zone constraint and unmodeled dynamic," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 10, pp. 1301–1312, Oct. 2015.
[32] Z. Cao *et al.*, "A fast orientation estimation approach of natural images," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1589–1597, Nov. 2016.

De Xu (M'05–SM'09) received the B.Sc. and M.Sc. degrees from the Shandong University of Technology, Jinan, China, in 1985 and 1990, respectively, and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2001, all in control science and engineering.

He has been with the Institute of Automation, Chinese Academy of Sciences, Beijing, China since 2001, where he is a Professor with the Research Center of Precision Sensing and Control. His research interests include robotics and automation, such as visual measurement, visual control, intelligent control, welding seam tracking, visual positioning, microscopic vision, and micro-assembly.

Jinyan Lu received the B.Sc. degree in computer science and technology from Xidian University, Xi'an, China, in 2008, the M.Sc. degree in computer science and technology from Beihang University, Beijing, China, in 2011, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, in 2016.

She is currently an Assistant Professor with IACAS. Her research interests include cover pose detection, visual measurement, and robot control.

Peng Wang (M'11) received the B.Sc. degree in electrical engineering and automation from Harbin Engineering University, Harbin, China, in 2004, the M.Sc. degree in automation science and engineering from the Harbin Institute of Technology, Harbin, in 2007, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, China, in 2010.

He is an Associate Professor with the Research Center of Precision Sensing and Control, IACAS. His research interests include visual tracking, visual detection, visual attention models, and visual perception models.

Zhengtao Zhang received the B.Sc. degree from the China University of Petroleum, Dongying, China, in 2004, the M.Sc. degree from the Beijing Institute of Technology, Beijing, China, in 2007, and the Ph.D. degree from Institute of Automation, Chinese Academy of Sciences (IACAS), Beijing, in 2010, all in control science and engineering.

He is an Associate Professor with the Research Center of Precision Sensing and Control, IACAS. His research interests include visual measurement, micro-assembly, and automation.

Zize Liang received the B.Sc. and M.Sc. degrees in automatic control engineering from Tsinghua University, Beijing, China, in 1986 and 1989, respectively.

Since 2000, he has been with the Institute of Automation, Chinese Academy of Sciences, Beijing, where he is currently a Professor with the State Key Laboratory of Management and Control for Complex Systems. His research interests include robotics and servo control.