Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
January-February 2018: 1–12
© The Author(s) 2018
DOI: 10.1177/1729881417753859
journals.sagepub.com/home/arx
SAGE

Dynamic performance improvement of direct image-based visual servoing in contour following

Che-Liang Li, Ming-Yang Cheng and Wei-Che Chang

Abstract

Image-based visual servoing (IBVS) has increasingly gained popularity and has been adopted in applications such as industrial robots, quadrotors, and unmanned aerial vehicles. When exploiting IBVS, the image feature velocity command obtained from the visual loop controller is converted to the velocity command of the workspace through the interaction matrix so as to converge image feature error. However, issues such as the noise/disturbance arising from image processing and the smoothness of image feature command are often overlooked in the design of the visual loop controller, especially in a contour following task. In particular, noise in the image feature will contaminate the image feedback signal so that the visual loop performance can be substantially affected. To cope with the aforementioned problem, this article employs the sliding mode controller to suppress the adverse effects caused by image feature noise. Moreover, by exploiting the idea of motion planning, a parametric curve interpolator is developed to generate smooth image feature commands. In addition, a depth observer is also designed to provide the depth information essential in the implementation of the interaction matrix. In order to assess the feasibility of the proposed approach, a two-degrees-of-freedom planar robot that employs an IBVS structure and an eye-to-hand camera configuration is used to conduct a contour following task. Contour following results verify the effectiveness of the proposed approach.

Keywords

Image-based visual servoing, parametric curve interpolator, depth observer, sliding mode controller

Date received: 21 August 2017; accepted: 10 December 2017

Topic: Special Issue – Robot Path Planning Design and Implementation in Manufacturing, Healthcare and Service Systems
Topic Editor: Andrey V Savkin
Associate Editor: Hamed Fazlollahtabar

Introduction

Recently, image-based visual servoing (IBVS) has increasingly gained popularity over the past two decades and has been adopted in applications such as industrial manufacturing,[1–3] quadrotors, unmanned aerial vehicles,[4,5] industrial robots, and mobile robots.[6,7] Several challenging problems related to IBVS remain unsolved.[8] Many existing studies related to IBVS focus on the transformation problems between different spaces such as the singularity problem and decoupling problem for image Jacobian,[9] while some investigate stability and convergence issues in IBVS.[10] Moreover, if the direct IBVS structure is adopted,[1] dynamics issues in visual servoing such as path/trajectory planning, visual loop controller design, and depth estimation need to be further explored.[11]

Vision-based path-planning problems in visual servoing applications are often encountered in point-to-point motions and contour following motions.[12] In particular, one

Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Corresponding author:
Ming-Yang Cheng, Department of Electrical Engineering National Cheng Kung University, Tainan 701, Taiwan, R.O.C.
Email: mycheng@mail.ncku.edu.tw

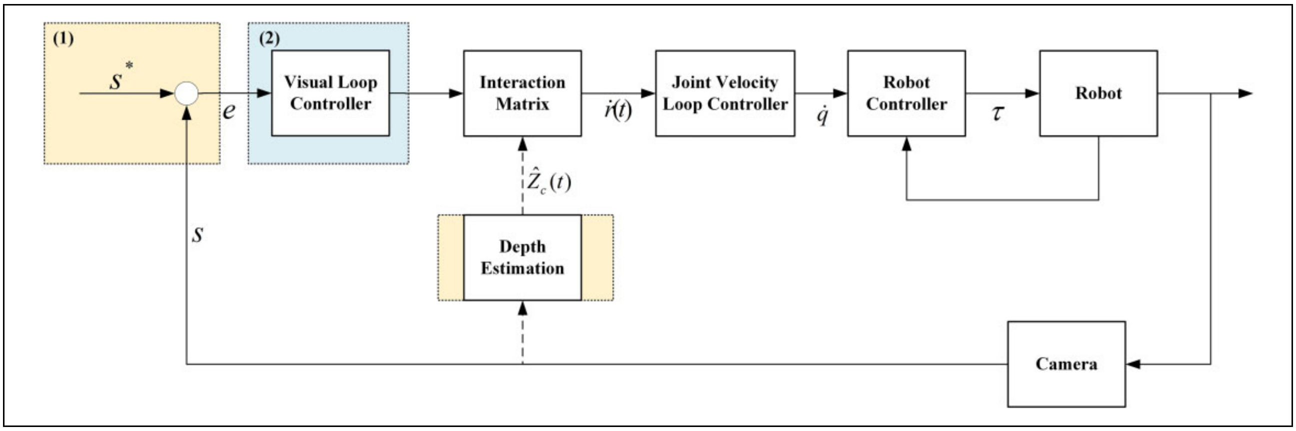


Figure 1. Control block diagram for the direct IBVS scheme employed in this article. IBVS: image-based visual servoing.

can divide vision-based path planning into two categories—offline path planning and online path planning. For offline path planning, Cherubini et al.[6] developed an IBVS scheme for nonholonomic robots to follow a continuous path drawn on the ground. In Chang et al.,[13] the path command for an unknown planar shape contour following task is generated by a Pythagorean-hodograph (PH)-spline interpolator. In Chang,[14] a binocular camera is adopted in a three-dimensional (3D) contour following application and the correspondence feature mapping problem in binocular systems is also addressed. On the other hand, for online path planning, Mezouar and Chaumette[15,16] proposed a new approach that performs path planning in an image space for IBVS. In particular, specific constraints are made to develop an interpolator ensuring that certain goals, such as ensuring the object remains in the field of view (FOV) of the camera and avoiding the joint limits of the robot, can be achieved. Later, Schramm and Morel[17] proposed a camera-parameter-free approach that can provide analytical solutions to the path-planning problem of the feature points in the image plane. Kyrki et al.[18] proposed a shortest-path approach for visual servoing to guarantee both the shortest Cartesian trajectory and object visibility. Under the assumption that the camera pose has been estimated and taking into account issues such as workspace, joint limits, and visibility margin, Chesi and Hung[19] proposed an optimal path-planning approach in the image space for an IBVS controller. Kazemi et al.[20] further proposed a global path-planning approach that accounts for constraints such as visibility of the target, visual occlusion avoidance, collision avoidance, and joint limits, in which the generated camera path is projected into the image space to be used in an IBVS scheme. Sharma and Hutchinson[21] proposed the concept of motion perceptibility and applied it to the problems of active camera trajectory planning and simultaneous camera/robot trajectory planning.

In IBVS, as mentioned previously, several challenging issues such as control design of visual loop and depth estimation deserve further investigation. In DeLuca et al.,[22] a nonlinear observer is exploited to estimate the depth value that is essential in the calculation of the interaction matrix in an IBVS scheme. In Keshmiri et al.,[23] unlike most previous approaches generating velocity commands for robot controllers, in order to achieve smoother feature tracking results, an augmented IBVS controller is proposed to generate acceleration commands for robot controllers instead. Becerra et al.[24] proposed a sliding mode controller that uses desired epipole trajectories for an IBVS scheme to achieve robust global stabilization without the need of performing accurate camera calibration. In order to avoid system failures when feature points are not in the FOV of the camera, Gans and Hutchinson[25] proposed a hybrid visual servo approach that switches between position-based visual servoing and IBVS. Jiang et al.[26] proposed an indirect Iterative Learning Control scheme for vision-guided robots to perform the task of trajectory imitation, in which the unknown interaction matrix essential to implementing IBVS is approximated by a series of neural networks so as to avoid possible singularities in uncalibrated (or poorly calibrated) camera applications.

Due to the fact that the issues of system dynamics and nonlinearities are crucial in the design of a direct IBVS scheme, this article focuses on enhancing the performance of the direct IBVS scheme so as to improve the entire visual servoing system. The contributions of this article are twofold (as shown in Figure 1):

1. By exploiting the concept of acceleration/deceleration motion planning commonly adopted in motion control applications, smooth image feature commands to be used in a direct IBVS scheme can be generated by a parametric curve interpolator. In addition, a depth observer[22] is also designed to provide the estimated depth value to the direct IBVS scheme in real time so that the actual image trajectory can be properly controlled to converge to the desired feature trajectory in the image space.[27]

2. Since the sliding mode control (SMC) is a nonlinear control method that can increase the robustness and disturbance suppression ability of the control system, in this article, a sliding mode visual loop controller is designed to cope with the possible contour following accuracy problems caused by modeling error/image feature noise.

In order to verify the effectiveness of the proposed approach, an experimental platform consisting of a two-degrees-of-freedom (2-DOF) planar manipulator equipped with an eye-to-hand camera is used to perform a vision-based contour following motion with respect to an object without a known geometric model.

The remainder of the article is organized as follows. The "Problem statement" section lays out the dynamics issues encountered in designing IBVS. The issues concerning image feature command generation, target image detection, and depth observer are addressed in the "Image-based path planning with a depth observer" section. Design of the direct IBVS controller is included in the "Direct IBVS controller design" section. Experimental setup and results are provided in the "Experimental setups and results" section. Finally, the conclusions are summarized in the "Conclusions" section.

Problem statement

Suppose in an image contour following task, the desired image feature command is obtained through a parametric curve interpolator. Without the loss of generality, the desired image feature command can be expressed as $f^*(r(t))$. In general, an IBVS control scheme is designed to converge the tracking error $e(r(t))$ between the vector of the desired image feature $f^*(r(t))$ and the vector of the image feature $f(r(t))$. Taking the derivative of $e(r(t))$ with respect to time t will yield

$$\begin{aligned}\dot{e}(r(t)) &= \frac{de(r(t))}{dt} = \frac{d(f(r(t)) - f^*(r(t)))}{dt}\\ &= \frac{\partial f(r(t))}{\partial r(t)} \cdot \frac{dr(t)}{dt} - \frac{\partial f^*(r(t))}{\partial r(t)} \cdot \frac{dr(t)}{dt}\\ &= J_I(r(t))\dot{r}(t) - J_I^*(r(t))\dot{r}(t) = J_e(r(t))\dot{r}(t) \qquad (1)\end{aligned}$$

where $f(r(t)) = [f_1(r(t)), f_2(r(t)), \ldots, f_k(r(t))]^T$, $f^*(r(t)) = [f_1{}^*(r(t)), f_2{}^*(r(t)), \ldots, f_k{}^*(r(t))]^T$ and $r(t) = [r_1(t), r_2(t), \ldots, r_m(t)]^T$

In addition,

$$J_I = \begin{bmatrix} \dfrac{\partial f_1(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial f_1(r(t))}{\partial r_m(t)} \\ \vdots & & \vdots \\ \dfrac{\partial f_k(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial f_k(r(t))}{\partial r_m(t)} \end{bmatrix} \in R^{k\times m,} \qquad (2)$$

$$J_I^* = \begin{bmatrix} \dfrac{\partial f_1^*(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial f_1^*(r(t))}{\partial r_m(t)} \\ \vdots & & \vdots \\ \dfrac{\partial f_k^*(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial f_k^*(r(t))}{\partial r_m(t)} \end{bmatrix} \in R^{k\times m} \qquad (3)$$

$$\begin{aligned} J_e &= \begin{bmatrix} \dfrac{\partial[f_1(r(t)) - f_1^*(r(t))]}{\partial r_1(t)} & \cdots & \dfrac{\partial[f_1(r(t)) - f_1^*(r(t))]}{\partial r_m(t)} \\ \vdots & & \vdots \\ \dfrac{\partial[f_k(r(t)) - f_k^*(r(t))]}{\partial r_1(t)} & \cdots & \dfrac{\partial[f_k(r(t)) - f_k^*(r(t))]}{\partial r_m(t)} \end{bmatrix} \\ &= \begin{bmatrix} \dfrac{\partial e_1(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial e_1(r(t))}{\partial r_m(t)} \\ \vdots & & \vdots \\ \dfrac{\partial e_k(r(t))}{\partial r_1(t)} & \cdots & \dfrac{\partial e_k(r(t))}{\partial r_m(t)} \end{bmatrix} \in R^{k\times m} \end{aligned} \qquad (4)$$

where $r(t)$ are the coordinates of end effector in some parameterization of task space T; $\dot{r}(t)$ is the end-effector velocity (velocity screw); $f(r(t))$ is the vector of image feature parameter; $\dot{f}(r(t))$ is the rate of change of vector of image feature parameter; $f^*(r(t))$ is the vector of desired image feature parameter; $\dot{f}^*(r(t))$ is the rate of change of vector of desired image feature parameter; J_I is the interaction matrix that relates $\dot{f}(r(t))$ and velocity screw $\dot{r}(t)$, J_I^* is the interaction matrix that relates $\dot{f}^*(r(t))$ and velocity screw $\dot{r}(t)$; and J_e is the interaction matrix that relates $\dot{e}(r(t))$ and velocity screw $\dot{r}(t)$.

Note that all three interaction matrices are functions of the depth values of the feature points in the camera frame. In general, the depth values of the feature points in the camera frame are not known and need to be estimated. The accuracy of the estimated depth value affects the accuracy of the interaction matrices as well as the performance of IBVS.[9] Suppose that the visual loop controller is of P type. That is, the image feature error $e(r(t))$ is controlled to exponentially converge to zero as described by

$$\dot{e}(r(t)) + \lambda e(r(t)) = 0 \qquad (5)$$

where λ is the gain constant of the P-type visual loop controller.

Substituting equation (1) into equation (5) will yield

$$J_e\dot{r}(t) + \lambda e(r(t)) = 0 \qquad (6)$$

Suppose that the matrix $J_e^T J_e$ is invertible. The velocity screw of the robot can be calculated using

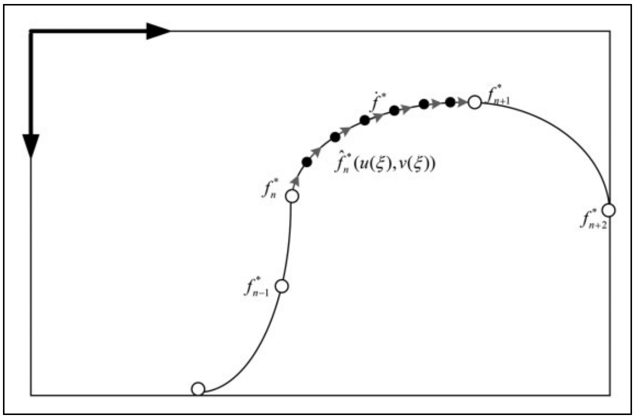


Figure 2. With the depth information provided by a depth observer, the actual image contour can be controlled to converge to the desired contour.

$$\dot{r}(t) = -\lambda(J_e^T J)^{-1} e(r(t)) \tag{7}$$

The velocity screw is then converted into the joint velocity command $\dot{q}(t)$ through the robot Jacobian J_R using

$$\dot{q}(t) = J_R^{-1}\dot{r}(t) = -\lambda J_R^{-1}(J_e^T J)^{-1} e(r(t)) \tag{8}$$

Moreover, this article focuses on the control design problem of a direct IBVS scheme. With the joint velocity command obtained from equation (8), one can design a joint velocity loop controller for generating the torque command to the servomotors installed at each joint of the robot, as shown in Figure 1.

Image-based path planning with a depth observer

In this article, motion planning is performed on a desired contour in the image plane. Moreover, with the depth information provided by a depth observer and a properly designed visual loop controller, the actual image contour can be controlled to converge to the desired image contour, as shown in Figure 2.

In Figure 2, f_n^* represents the desired image feature points that the desired image contour needs to pass through. Suppose that the desired image contour consists of l contour segments, in which the nth contour segment $\hat{f}^*_n(u(\xi),v(\xi))$ connects the nth image feature point f_n^* and the (n+1)th image feature point f^*_{n+1}. In this article, each desired image contour segment $\hat{f}^*_n(u(\xi),v(\xi))$ is represented by a parametric curve, for which ξ is the parameter and $0 \leq \xi \leq 1$. With the parameter value ξ given, one can obtain interpolation points between f_n^* and f^*_{n+1}. Different kinds of parametric curves such as Non-Uniform Rational B-Splines (NURBS),[28–30] B-spline,[31] Bezier curve,[31] cubic Hermite spline,[32] and PH spline[33,34] can be used to represent the image contour segment. In this article, a PH spline is constructed to pass through all the desired image feature points. In IBVS applications, the desired image feature command is a function of time. However, the PH spline $\hat{f}^*_n(u(\xi),v(\xi))$ adopted in this article is a parametric curve of parameter ξ. In order to generate desired image feature commands that can be used in IBVS applications, a relationship between the parameter ξ and time t must be developed. By definition, the feed rate $V(t)$ along the desired image contour is the time derivative of the PH spline $\hat{f}^*_n(u(\xi),v(\xi))$ described by

$$\begin{aligned}\|V(t)\| &= \left\|\frac{d\hat{f}^*_n(u(\xi),u(\xi))}{dt}\right\| \\ &= \left\|\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial u(\xi)}\cdot\frac{\partial u(\xi)}{\partial\xi}\cdot\frac{d\xi}{dt}+\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial v(\xi)}\cdot\frac{\partial v(\xi)}{\partial\xi}\cdot\frac{d\xi}{dt}\right\| \\ &= \left\|\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial u(\xi)}\cdot\frac{\partial u(\xi)}{\partial\xi}+\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial v(\xi)}\cdot\frac{\partial v(\xi)}{\partial\xi}\right\|\cdot\frac{d\xi}{dt}\end{aligned} \tag{9}$$

Denote

$$\sqrt{u'^2(\xi)+v'^2(\xi)} = \left\|\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial u(\xi)}\cdot\frac{\partial u(\xi)}{\partial\xi}+\frac{\partial\hat{f}^*_n(u(\xi),u(\xi))}{\partial v(\xi)}\cdot\frac{\partial v(\xi)}{\partial\xi}\right\| \tag{10}$$

Equation (9) can be rewritten as

$$\frac{d\xi(t)}{dt} = \frac{V(t)}{\sqrt{u'^2(\xi)+v'^2(\xi)}} \tag{11}$$

Equation (11) is an ordinary differential equation for $\xi(t)$ and can be solved by a numerical method such as the Taylor first-order method, the Taylor second order method, and the Runge–Kutta method.[35] After the parameter value $\xi(t)$ at time t is known, one can calculate the desired image feature point $\hat{f}^*_n(u(\xi(t)),v(\xi(t)))$ corresponding to this parameter value. Note that in equation (11), the feed rate $V(t)$ (i.e. image feature velocity) along the desired image contour can be obtained from trapezoidal or S-curve acceleration/deceleration motion planning. However, as shown in Figure 1, the output of the visual loop controller cannot be directly sent to the servomotors installed at each joint of the robot. Instead, the output of the visual loop controller needs to be converted into the joint velocity command for the servomotor by using the interaction matrix and robot Jacobian as described by equation (8). The problem is that the depth value of the feature point is essential in calculating the interaction matrix. An inaccurate depth value may lead to inappropriate joint velocity command so that the robot may not be able to move its effector properly to converge the tracking error of image feature to zero. In order to cope with the aforementioned difficulty, this article employs the depth observer proposed in DeLuca et al.[22] to estimate the depth values $\hat{Z}_c(t)$ of feature points in real time. The dynamic equation for the depth observer is described as

$$\begin{aligned}\dot{\tilde{x}}(t) &= \begin{bmatrix} -\mu\tilde{\mathrm{x}}_3(t) & 0 & \tilde{\mathrm{x}}_1(t)\tilde{\mathrm{x}}_3(t) & \dfrac{\tilde{\mathrm{x}}_1(t)\tilde{\mathrm{x}}_2(t)}{\mu} & -\left(\mu+\dfrac{\tilde{\mathrm{x}}_1^2(t)}{\mu}\right) & \tilde{\mathrm{x}}_2(t) \\ 0 & -\mu\tilde{\mathrm{x}}_3(t) & \tilde{\mathrm{x}}_2(t)\tilde{\mathrm{x}}_3(t) & \mu+\dfrac{\tilde{\mathrm{x}}_2^2(t)}{\mu} & -\dfrac{\tilde{\mathrm{x}}_1(t)\tilde{\mathrm{x}}_2(t)}{\mu} & -\tilde{\mathrm{x}}_1(t) \\ 0 & 0 & \tilde{\mathrm{x}}_3^2(t) & \dfrac{\tilde{\mathrm{x}}_2(t)\tilde{\mathrm{x}}_3(t)}{\mu} & -\dfrac{\tilde{\mathrm{x}}_1(t)\tilde{\mathrm{x}}_3(t)}{\mu} & 0 \end{bmatrix}\dot{r}(t) \\ y(t) &= \begin{bmatrix}\tilde{x}_1(t)\\ \tilde{x}_2(t)\end{bmatrix}\end{aligned} \tag{12}$$

where μ is the calibrated focal length in pixels and the state vector

$$\tilde{x}(t) = [\tilde{x}_1(t) \quad \tilde{x}_2(t) \quad \tilde{x}_3\,(t)]^T = \begin{bmatrix} u(t) & v(t) & \dfrac{1}{\hat{Z}_c(t)}\end{bmatrix}^T$$

The updated equation for the estimation of the unknown state is described by equation (13), where $\alpha(\widehat{x}(t), y(t))$ stands for the original system and $\beta(\widehat{x}(t), y(t), \dot{r}(t))$ is used for recovering the originally unobservable states[22]

$$\dot{\widehat{x}}(t) = \alpha(\widehat{x}(t), y(t))\dot{r}(t) + \beta(\widehat{x}(t), y(t), \dot{r}(t)) \tag{13}$$

$$\alpha\left(\widehat{x}(t), y(t)\right) = \begin{bmatrix} -\mu\widehat{\mathrm{x}}_3(t) & 0 & y_1(t)\widehat{\mathrm{x}}_3(t) & \dfrac{y_1(t)y_2(t)}{\mu} & -\left(\mu+\dfrac{y_1^2(t)}{\mu}\right) & y_2(t) \\ 0 & -\mu\widehat{\mathrm{x}}_3(t) & y_2(t)\widehat{\mathrm{x}}_3(t) & \mu+\dfrac{y_2^2(t)}{\mu} & -\dfrac{y_1(t)y_2(t)}{\mu} & -y_1(t) \\ 0 & 0 & \widehat{\mathrm{x}}_3^2(t) & \dfrac{y_2(t)\widehat{\mathrm{x}}_3(t)}{\mu} & -\dfrac{y_1(t)\widehat{\mathrm{x}}_3(t)}{\mu} & 0 \end{bmatrix} \tag{14}$$

$$\begin{aligned}\beta\left(\widehat{x}(t), y(t),\ \dot{\mathrm{r}}(t)\right) \;=\; \mathrm{Ke}_{\mathrm{o}}(t) &= \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ k_3(-\mu\dot{r}_1(t)+y_1\dot{r}_3(t)) & k_3(-\mu\dot{r}_2(t)+y_2(t)\dot{r}_3(t)) & 0\end{bmatrix}\begin{bmatrix}e_{o1}(t)\\ e_{o2}(t)\\ e_{o3}(t)\end{bmatrix} \\ &= \begin{bmatrix} k_1 e_{o1}(t) \\ k_2 e_{o2}(t) \\ k_3((-\mu\dot{r}_1(t)+y_1(t)\dot{r}_3(t))e_{o1}(t)+(-\mu\dot{r}_2(t)+y_2(t)\dot{r}_3(t))e_{o2}(t))\end{bmatrix}\end{aligned} \tag{15}$$

where $\widehat{x}(t) \in \Re^3$ is the estimate of the unknown state $\tilde{x}(t)$, and $e_o(t) = \tilde{x}(t) - \hat{x}(t) = [\,e_{o1}(t) \quad e_{o2}(t) \quad e_{o3}\,(t)]^T$ is the error vector. In addition, k_1, k_2, $k_3>0$. In practice, the image measurement ($u(t)$, $v(t)$) contains noise and may affect the estimation accuracy of the depth observer. In order to cope with this difficulty, in this article the image measurement ($u(t)$, $v(t)$) used in equations (14) and (15) is replaced by the image feature command ($u(\xi(t))$, $v(\xi(t))$) obtained from the parametric curve interpolator, that is

$$\tilde{x}(t) = \begin{bmatrix}\tilde{x}_1(t) & \tilde{x}_2(t) & \dfrac{1}{\tilde{x}_3(t)}\end{bmatrix}^T = \begin{bmatrix} u(\xi(t)) & v(\xi(t)) & \dfrac{1}{\hat{Z}_c(t)}\end{bmatrix}^T \tag{16}$$

Figure 3 illustrates the block diagram for the proposed depth estimation approach based on the parametric curve.

Direct IBVS controller design

SMC is one of the most commonly used nonlinear control schemes that can increase the robustness and the disturbance suppression ability of the control system. SMC belongs to the category of variable structure control[36] and has several attractive features such as simple controller design, good dynamic response, being robust, and having excellent disturbance suppression ability. In SMC, a sliding surface should be chosen in advance. Based on the system state, the sliding model control law generates switching force and equivalent force. In particular, the switching force makes the dynamics "slide" to the predefined sliding surface, while the equivalent force keeps the states of the system on the sliding surface after these states intersect the sliding surface. Under proper design, the SMC can guarantee asymptotic stability. In this article, an SMC is adopted in the design of the visual loop controller to reduce the image feature error e between image feature measurement $f(t)$ and desired image feature $\hat{f}_n^*(u(\xi(t)), v(\xi(t)))$. In this article, the sliding surface described by equation (17) is adopted

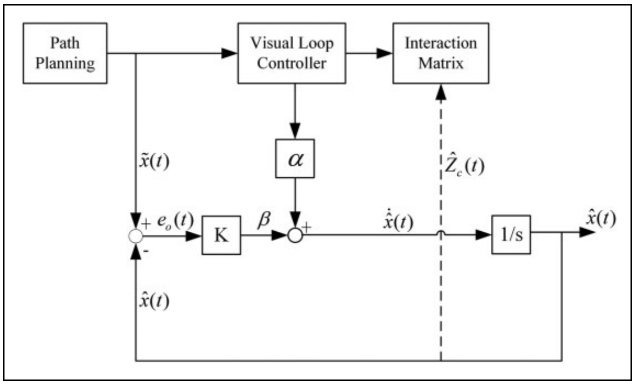


Figure 3. Block diagram for the proposed depth estimation approach based on the parametric curve.

$$S = e + k\int edt \ , \quad k > 0 \tag{17}$$

Suppose that in equation (1), the desired image feature $\hat{f}_n^*(u(\xi(t)), v(\xi(t)))$ is not a function of $r(t)$. Then equation (1) can be rewritten as $\dot{e} = J_I\dot{r} - \dot{\hat{f}}_n^*(u(\xi(t)), v(\xi(t)))$.

Differentiating equation (17) will result in

$$\dot{S} = \dot{e} + ke = v_c - \dot{\hat{f}}_n^*(u(\xi(t)), v(\xi(t))) + ke \tag{18}$$

where $v_c = J_I\dot{r}$.

In this article, the control output v_c of the SMC is described by

$$v_c = v_s + v_e \tag{19}$$

where v_s is the switching control and v_e is the equivalent control.

In particular, when the system state is on a sliding surface, that is, $S = 0$, the equivalent control can be obtained by letting $\dot{S} = 0$, which is

$$v_e = \dot{\hat{f}}_n^*(u(\xi(t)), v(\xi(t))) - ke \tag{20}$$

In contrast, when the system state is not on a sliding surface, that is, $S \neq 0$, the switching control is activated to drive the system toward the sliding surface.

To derive the switching control law, consider the following Lyapunov function candidate

$$V(t) = \frac{1}{2}S^2 \tag{21}$$

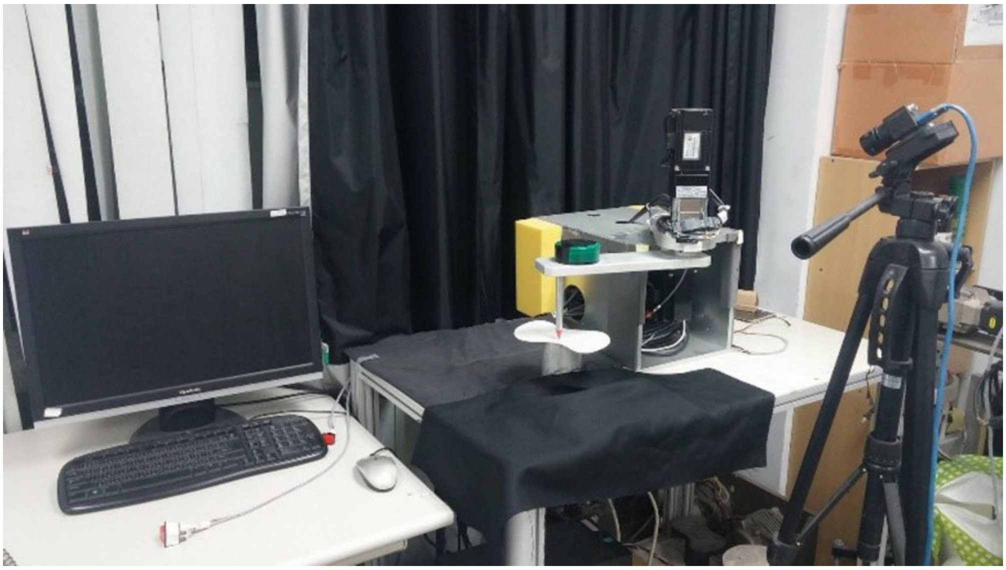


Figure 4. Experimental platform consisting of a 2-DOF planar robot equipped with an eye-to-hand camera, a personal computer, and an IMP-2 motion control card by Industrial Technology Research Institute (ITRI). DOF: degree of freedom.

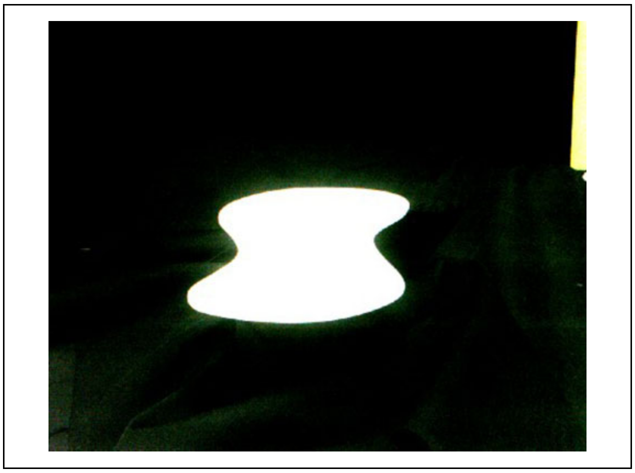


Figure 5. Extraction of the exterior contour of an object on the image plane.

Clearly, $V(t)$ is positive definite. The derivative of $V(t)$ versus t along the system trajectory is

$$\dot{V}(t) = \frac{d}{dt}\left(\frac{1}{2}S^2\right) = S\dot{S} \tag{22}$$

Substituting equation (18) into equation (22) will yield

$$\dot{V}(t) = S(v_c - \dot{\hat{f}}_n^{*}(u(\xi(t)), v(\xi(t))) + ke) \tag{23}$$

In addition, substituting equations (19) and (20) into equation (23) will result in

$$\dot{V}(t) = S(v_s + v_e - \dot{\hat{f}}_n^{*}(u(\xi(t)), v(\xi(t))) + ke) = Sv_s \tag{24}$$

By choosing the switching control law $v_s = -K_s\,\mathrm{sgn}(S)$, equation (24) becomes

$$\dot{V}(t) = Sv_s = -K_s S\,\mathrm{sgn}(S) < 0 \tag{25}$$

where $K_s \in \Re^{2\times 2}$ is a positive definite constant gain matrix.

According to the Lyapunov stability theorem, the system under SMC is stable.[37,38]

However, the switching control v_s contains a signum function that often results in the so-called chattering phenomenon. In order to cope with this problem, a hyperbolic tangent function tanh(●) is adopted to replace the signum function so as to smoothen the switching control. That is

$$\begin{aligned} v_s &= -K_s \tanh(K_b S) \\ &= -K_s \tanh(K_b(e + k\int edt)) \end{aligned} \tag{26}$$

where $K_b \in \Re^{2\times 2}$ is a positive definite constant gain matrix.

Substituting equations (20) and (26) into equation (19) will result in

$$v_c = -K_s \tanh(K_b(e + k\int edt)) + \dot{\hat{f}}_n^{*}(u(\xi(t)), v(\xi(t))) - ke \tag{27}$$

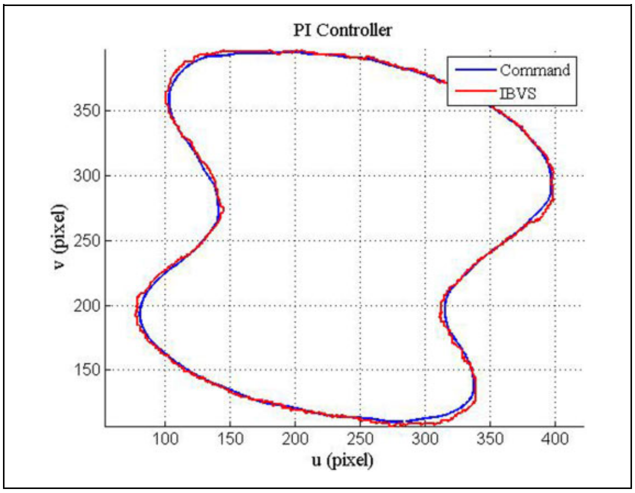


Figure 6. Contour following results of the Proportional-Integral-type (PI-type) visual loop controller.

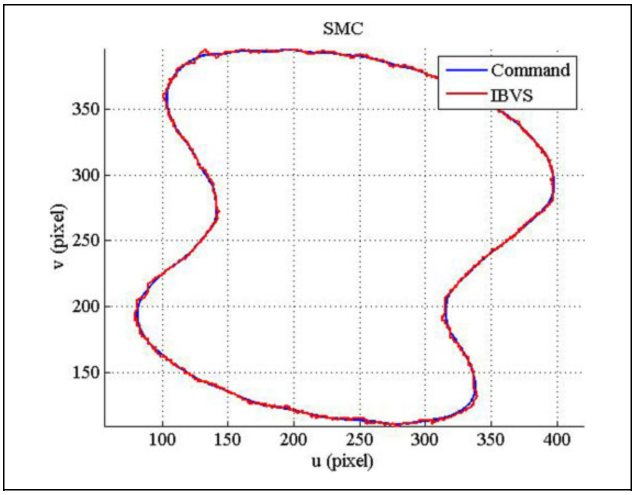


Figure 7. Contour following results of the sliding mode visual loop controller.

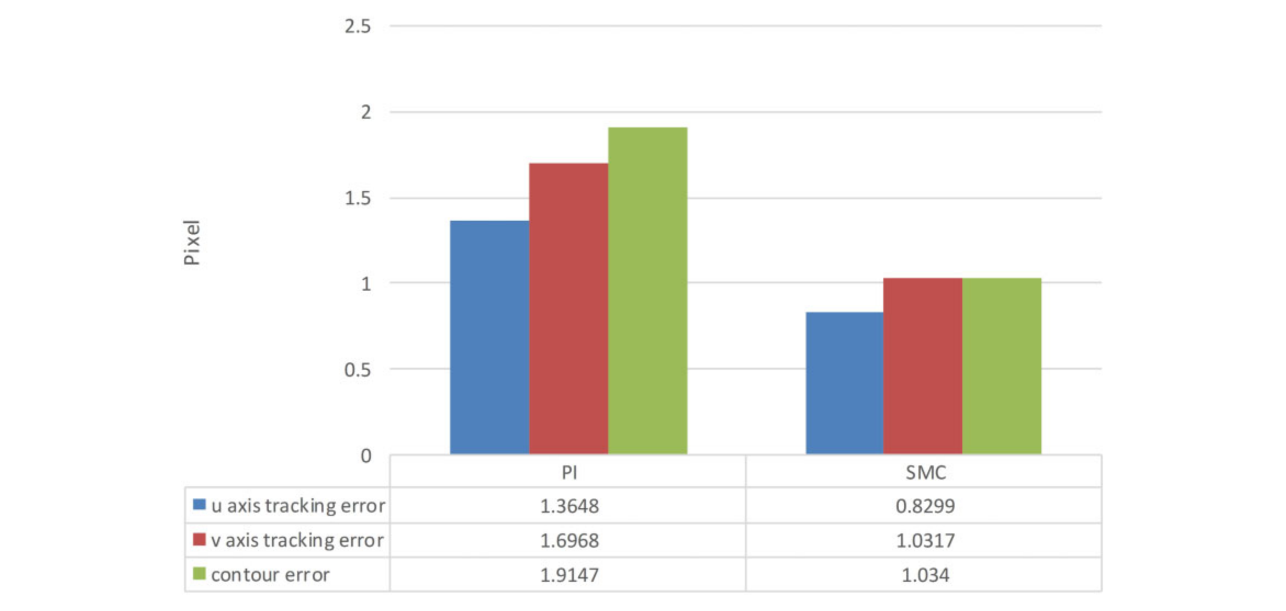
In addition, since $v_c = J_I\dot{r}$, the velocity screw of the end effector can be obtained using

$$\dot{r} = J_1^{+} v_c \tag{28}$$

where J_1^{+} is the pseudo inverse of J_I. Subsequently, the joint velocity command $\dot{q}$ can be obtained using equation (8) with $\dot{r}$ computed from equation (28). Based on the joint velocity command $\dot{q}$, a PI-type velocity loop controller is employed in this article to provide the torque command to the servomotor installed at each joint of the robot manipulator.

Experimental setups and results

The experimental platform consisting of a 2-DOF planar robot equipped with an eye-to-hand camera, a personal computer, and an IMP-2 motion control card by ITRI,



	PI	SMC
🟦 u axis tracking error	1.3648	0.8299
🟥 v axis tracking error	1.6968	1.0317
🟩 contour error	1.9147	1.034

Figure 8. Performance comparison between two different visual loop controllers.

Taiwan, ROC, as shown in Figure 4, is used to assess the performance of the proposed approach. The two joints of the planar robot are actuated by two AC servomotors, ECMA-C10804E7 and ECMA-C10602AS (Delta electronics, Taiwan). Throughout the experiments, the motor drives of the AC servomotors are set to torque mode. In addition, each servomotor is coupled with a 1:50 planetary gear reducer. The resolution of the encoder is 10,000 pulses/rev. In addition, the eye-to-hand camera is an FLEA3 FL3-U3-13E4C-C camera of resolution 640 pixel × 512 pixel and frame rate of 60 Hz (Point Grey, Canada). Its SV-1614 V lens with a focus length of 16 mm is manufactured by VS Technology (Japan).

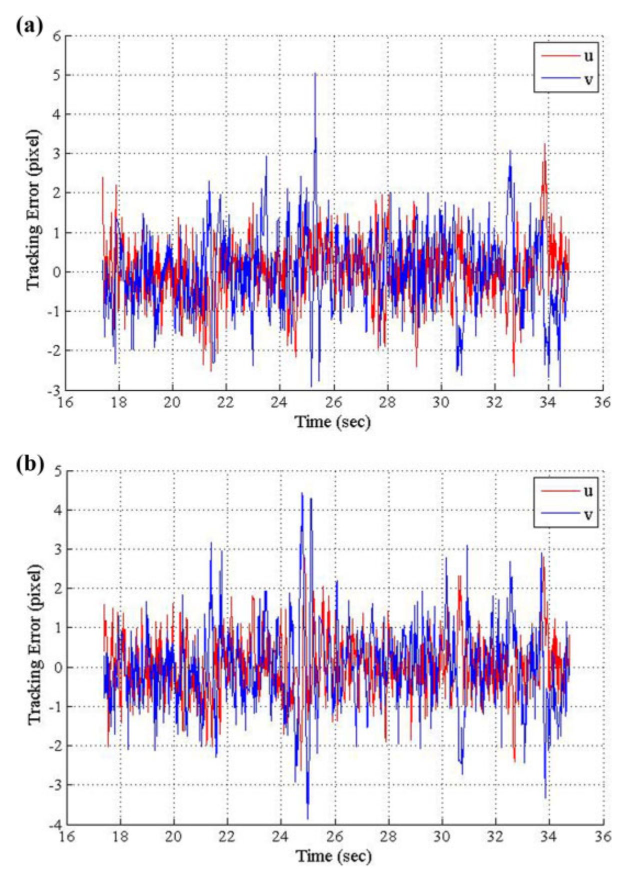


Figure 9. Tracking error using (a) a fixed depth value and (b) a depth observer.

Throughout the experiments, the geometric model of the object is not known in advance. The exterior contour of the object on the image plane is extracted, as shown in Figure 5. However, the extracted contour consists of pixels (i.e. grid points) in the image plane. Very likely, the pixels on the extracted contour are not dense enough to be directly used as the desired image feature commands in IBVS. In order to cope with this problem, as discussed in the "Image-based path planning with a depth observer" section, in this article a PH spline is constructed to pass through all the pixels on the extracted exterior contour of the object in the image plane. Moreover, with the desired feed rate provided, a parametric curve interpolator is developed to generate smooth desired image feature commands based on the constructed PH spline. Several contour following experiments are conducted to compare the performance of two visual loop controllers and to assess the effectiveness of the depth observer.

Performance comparison of visual loop controllers

In the contour following experiment, the tip of the planar robot is controlled to move along the contour of the object on the image plane. As mentioned previously, the geometric model of the object is not known in advance. By using the methods introduced in the "Image-based path planning with a depth observer" section, the generation of the desired image feature command and the detection of the vision feedback signal can be accomplished. In each experiment, two different visual loop controllers in IBVS—a PI controller and a sliding mode controller—are tested. In the experiments, a fixed depth value is used in the calculation of the interaction matrix. For each visual loop control scheme, ten contour following experiments are conducted. Note that the sampling rate for the visual loop is 50 Hz, and the desired feed rate for the parametric curve interpolator is 60 pixel/s. The cycle time of one contour following experiment is 17.36 s, indicating that there are a total of 868 image feature command points used in the contour following experiment. In order to observe the steady state response, two cycles of contour following motions are executed and only the results of the second cycle will be analyzed.

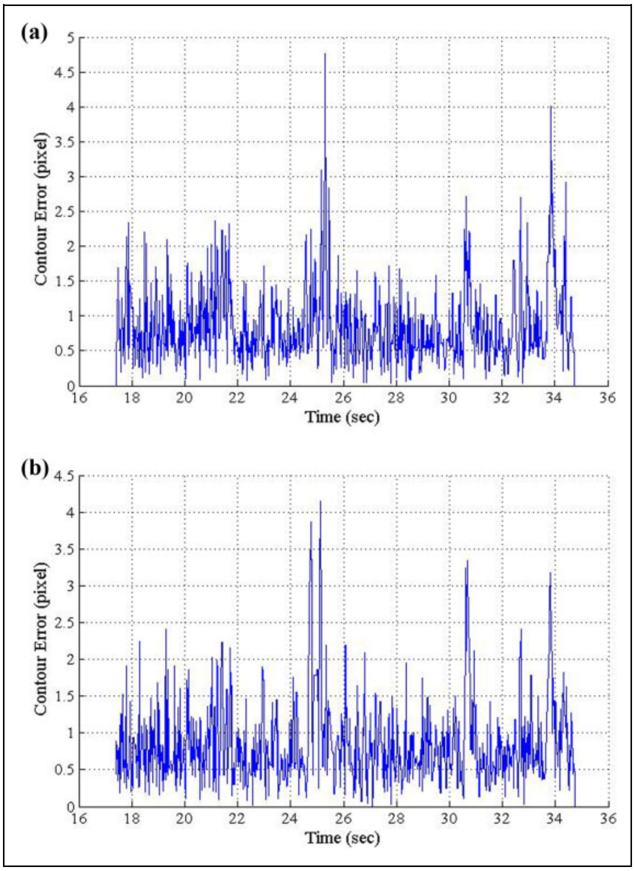
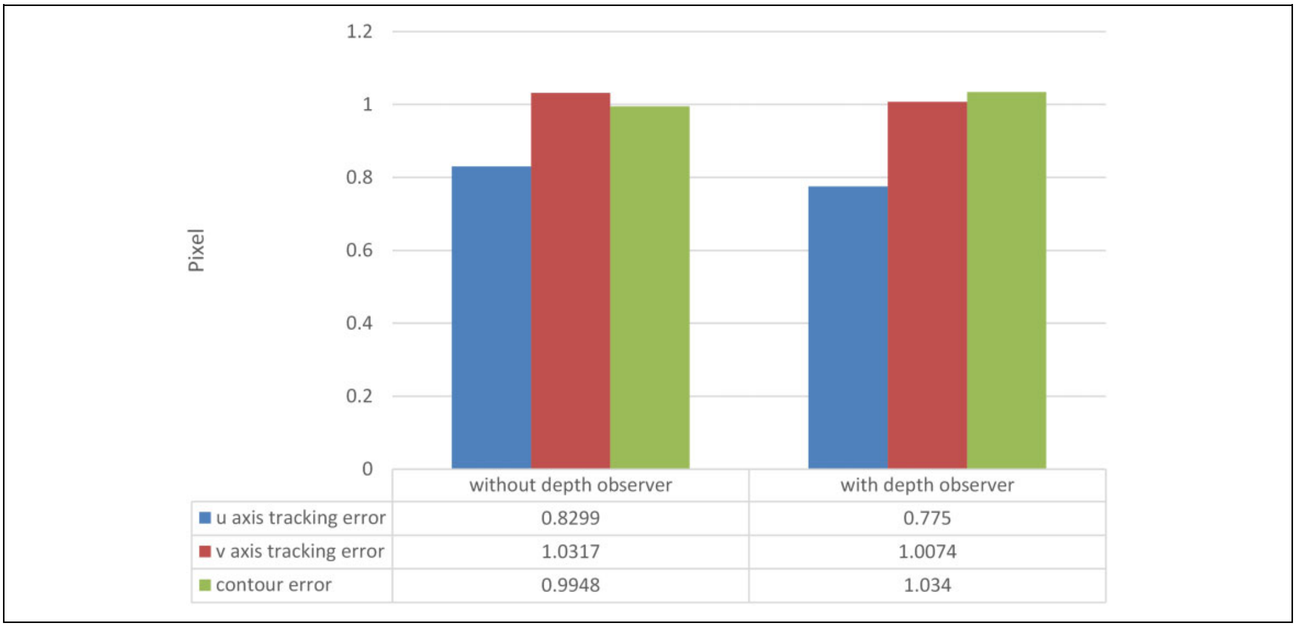


Figure 10. Contour error using (a) a fixed depth value and (b) a depth observer.

Figure 6 shows the contour following results of the PI-type visual loop controller, while Figure 7 shows the contour following results of the sliding mode visual loop controller. Clearly, the sliding mode visual loop controller yields significantly smaller contour error around the portion of the contour with large curvature. Figure 8 shows the average tracking error and contour errors of ten contour following experiments for two tested visual loop control schemes. It is clearly evident that the sliding mode visual loop controller outperforms the PI-type visual loop controller in all categories.

Performance assessment of depth observer

In IBVS, the interaction matrix plays a crucial role in converting the output of the visual loop controller into the velocity screw of the end effector. However, the depth



	without depth observer	with depth observer
🟦 u axis tracking error	0.8299	0.775
🟥 v axis tracking error	1.0317	1.0074
🟩 contour error	0.9948	1.034

Figure 11. Average tracking error and contour errors of 10 contour following experiments.

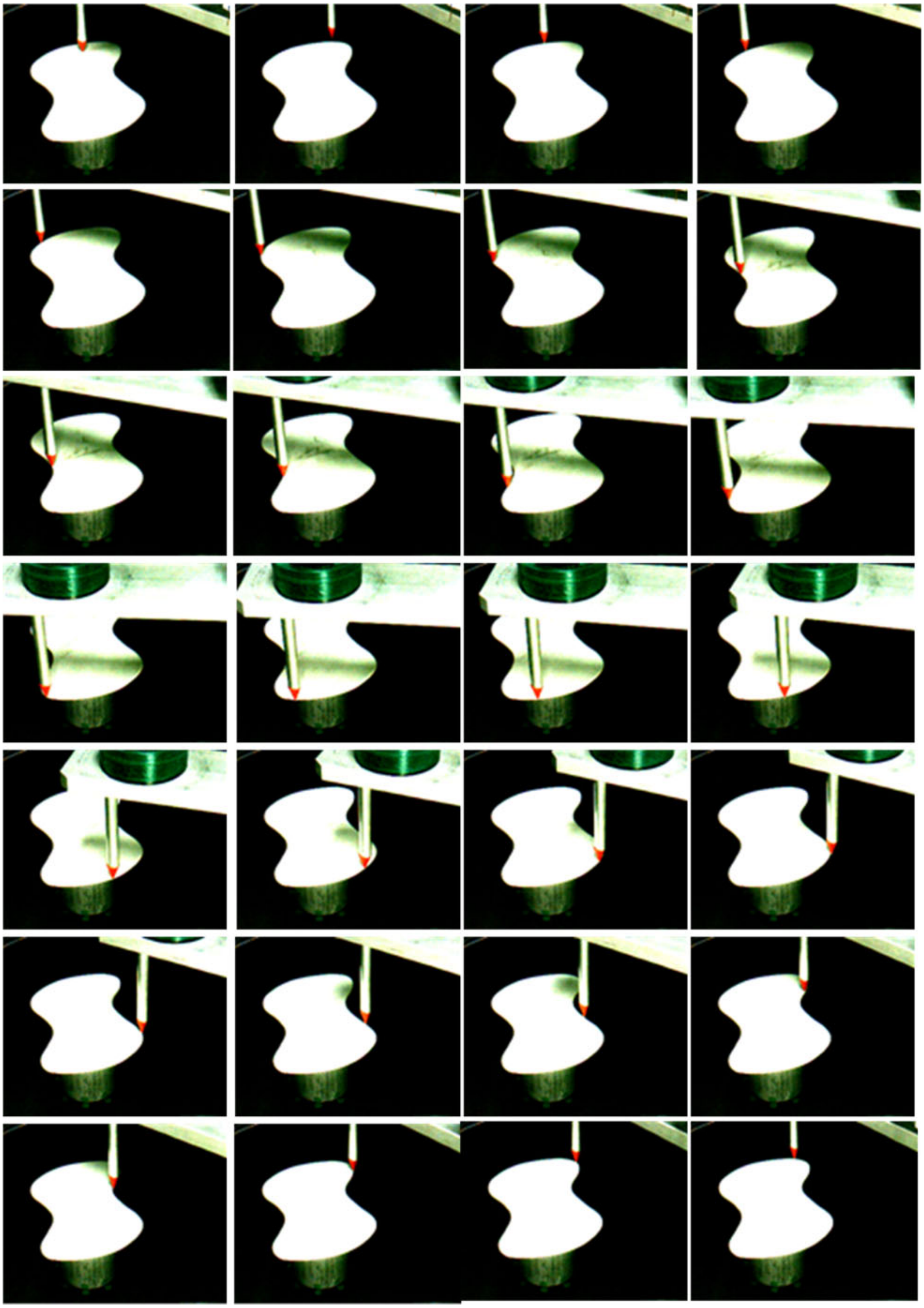


Figure 12. Snapshots of an image sequence of the contour following experiment.

information is essential when implementing the interaction matrix. One way to estimate the current depth value is to use the depth observer. The aim of this experiment is to assess the performance of the depth observer in contour following applications. In addition, the sliding mode controller is adopted as the visual loop controller throughout this experiment. Figures 9 and 10 show the experimental results of tracking error and contour error, respectively. Figure 11 shows the average tracking error and contour errors of ten contour following experiments using a depth observer or a fixed depth value. The results shown in Figure 11 indicate that the sliding mode IBVS with a depth observer slightly outperforms the one with a fixed depth value, especially in the tracking error of image feature. Figure 12 shows snapshots of an image sequence of the contour following experiment taken for every 0.62 s.

Conclusions

This article focuses on the dynamics performance improvement of a visual servoing system employing an IBVS structure in contour following applications. In particular, a sliding mode IBVS structure is developed in this article to deal with the external disturbance caused by modeling error and the noise in image feature measurement. In addition, in this article, the desired image feature commands are generated using a parametric curve interpolator based on the image captured by an eye-to-hand camera. Moreover, the depth information essential in the calculation of the interaction matrix is provided by a nonlinear depth observer so as to converge image feature tracking error. Results of contour following experiments of objects without known geometric models verify the effectiveness of the proposed approach.

This article focuses on the case of unknown two-dimensional planar contour following only. An extension to a more general case of 3D contour following in industrial applications is a research topic deserving of in-depth investigation in the future. In addition, in this article, a dual control loop consisting of a visual loop and robot control loop is adopted. The pros and cons of using single sampling rate or dual sampling rate for the control scheme in a visual servoing system need to be further addressed as well.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The project is supported by the National Science Council of the Republic of China, Taiwan, under grant no. MOST 103-2221-E-006-185-MY2.

References

1. Hutchinson S, Hager GD and Corke PI. A tutorial on visual servo control. *IEEE Trans Rob Autom* 1996; 12(5): 651–670.
2. Golnabi H and Asadpour A. Design and application of industrial machine vision systems. *Robot Comput Integr Manuf* 2007; 23(6): 630–637.
3. Bolotnikova A, Chappellet K, Paolillo A, et al. A circuit-breaker use-case operated by a humanoid in aircraft manufacturing. In: *13th IEEE conference on automation science and engineering*, Xi'an, China, 20–23 August 2017.
4. Serra P, Cunha R, Hamel T, et al. Landing of a quadrotor on a moving target using dynamic image-based visual servo control. *IEEE Trans Rob* 2016; 32(6): 1524–1535.
5. Ozawa R and Chaumette F. Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach. In: *2011 IEEE international conference on robotics and automation (ICRA)*, Shanghai, China, 09–13 May 2011, pp.5670–5676. New York: IEEE.
6. Cherubini A, Chaumette F and Oriolo G. Visual servoing for path reaching with nonholonomic robots. *Robotica* 2011; 29(7): 1037–1048.
7. Wang Y, Lang H and de Silva CW. A hybrid visual servo controller for robust grasping by wheeled mobile robots. *IEEE ASME Trans Mechatron* 2010; 15(5): 757–769.
8. Janabi-Sharifi F, Deng L and Wilson WJ. Comparison of basic visual servoing methods. *IEEE ASME Trans Mechatron* 2011; 16(5): 967–983.
9. Chaumette F and Hutchinson S. Visual servo control. I. Basic approaches. *IEEE Robot Autom Mag* 2006; 13(4): 82–90.
10. Chaumette F. Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman DJ, Hager GD and Morse AS (eds), *The confluence of vision and control. Lecture notes in control and information sciences*. Vol. 237. London: Springer, 1998, pp. 66–78.
11. Corke PI. Dynamic issues in robot visual-servo systems. In: Giralt G and Hirzinger G (eds), *Robotics research-international symposium*. Vol. 7. London: Springer, 1996, pp. 488–498.
12. Kazemi M, Gupta K and Mehrandezh M. Path-planning for visual servoing: a review and issues. In: Chesi G and Hashimoto K (eds) *Visual servoing via advanced numerical methods*. London: Springer, 2010, pp. 189–207.
13. Chang WC, Cheng MY and Tsai HJ. Image feature command generation of contour following tasks for SCARA robots employing image-based visual servoing—a PH-spline approach. *Robot Comput Integr Manuf* 2017; 44: 57–66.
14. Chang WC. Binocular vision-based 3-D trajectory following for autonomous robotic manipulation. *Robotica* 2007; 25(5): 615–626.
15. Mezouar Y and Chaumette F. Path planning for robust image-based control. *IEEE Trans Rob Autom* 2002; 18(4): 534–549.
16. Mezouar Y and Chaumette F. Optimal camera trajectory with image-based control. *Int J Rob Res* 2003; 22(10–11): 781–803.
17. Schramm F and Morel G. Ensuring visibility in calibration-free path planning for image-based visual servoing. *IEEE Trans Rob* 2006; 22(4); 848–854.
18. Kyrki V, Kragic D and Christensen HI. New shortest-path approaches to visual servoing. In: *2004 IEEE international conference on intelligent robots and systems (IROS)*, Edmonton, Canada, 28 September – 2 October 2004, pp.349–354. New York: IEEE.
19. Chesi G and Hung YS. Global path-planning for constrained and optimal visual servoing. *IEEE Trans Rob* 2007; 23(5): 1050–1060.

20. Kazemi M, Gupta K and Mehrandezh M. Global path planning for robust visual servoing in complex environments. In: *2009 IEEE international conference on robotics and automation (ICRA)*, Kobe, Japan, 12–17 May 2009, pp.326–332.
21. Sharma R and Hutchinson S. Motion perceptibility and its application to active vision-based servo control. *IEEE Trans Rob Autom* 1997; 13(4): 607–617.
22. De Luca A, Oriolo G and Robuffo Giordano P. Feature depth observation for image-based visual servoing: theory and experiments. *Int J Rob Res* 2008; 27(10): 1093–1116.
23. Keshmiri M, Xie WF and Mohebbi A. Augmented image-based visual servoing of a manipulator using acceleration command. *IEEE Trans Ind Electron* 2014; 61(10): 5444–5452.
24. Becerra HM, López-Nicolás G and Sagüés C. A sliding-mode-control law for mobile robots based on epipolar visual servoing from three views. *IEEE Trans Rob* 2011; 27(1): 175–183.
25. Gans NR and Hutchinson SA. Stable visual servoing through hybrid switched-system control. *IEEE Trans Rob* 2007; 23(3): 530–540.
26. Jiang P, Bamforth LC, Feng Z, et al. Indirect iterative learning control for a discrete visual servo without a camera-robot model. *IEEE Trans Syst Man Cybern B Cybern* 2007; 37(4): 863–876.
27. Chaumette F and Hutchinson S. Visual servo control. II. Advanced approaches. *IEEE Robot Autom Mag* 2007; 14(1): 109–118.
28. Cheng MY, Tsai MC and Kuo JC. Real-time NURBS command generators for CNC servo controllers. *Int J Mach Tools Manuf* 2002; 42(7): 801–813.
29. Cheng MY and Wang YH. Velocity field construction for contour following tasks represented in NURBS form. *IEEE Trans Automat Control* 2009; 54(10): 2405–2410.
30. Piegl L and Tiller W. *The NURBS book*. Berlin, Germany: Springer Science & Business Media, 2012.
31. Biswas S and Lovell BC. *Bézier and splines in image processing and machine vision*. London, England: Springer Science & Business Media, 2007.
32. Farin GE. *Curves and surfaces for CAGD: a practical guide*. Massachusetts, USA: Morgan Kaufmann, 2002.
33. Chen CY, Shieh SS, Cheng MY, et al. Vision-based Pythagorean hodograph spline command generation and adaptive disturbance compensation for planar contour tracking. *Int J Adv Manuf Technol* 2013; 65(5–8): 1–15.
34. Farouki RT and Sakkalis T. Pythagorean hodographs. *IBM J Res Dev* 1990; 34(5): 736–752.
35. Farouki RT and Shah S. Real-time CNC interpolators for Pythagorean-hodograph curves. *Comput Aided Geom Des* 1996; 13(7): 583–600.
36. Young KD, Utkin VI and Ozguner U. A control engineer's guide to sliding mode control. In: *1996 IEEE international workshop on variable structure systems (VSS)*, Tokyo, Japan, 5–6 December 1996, pp.1–14. New York: IEEE.
37. Khalil HK. *Nonlinear systems*. New Jersey: Prentice-Hall, 2.5 1996, 5–1
38. Jean-Jacques ES and Weiping L. *Applied nonlinear control*. Englewood Cliffs: Prentice Hall, 1991.