

## Research Article

# Three-Step Epipolar-Based Visual Servoing for Nonholonomic Robot with FOV Constraint

Yang Xu,<sup>1,2</sup> Jun Peng,<sup>1,2</sup> Wentao Yu,<sup>1,2</sup> Yuan Fang,<sup>1,2</sup> and Weirong Liu<sup>1,2</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University, Changsha 410075, China

<sup>2</sup> Human Engineering Laboratory for Advanced Control and Intelligent Automation, Changsha 410075, China

Correspondence should be addressed to Jun Peng; pengj@csu.edu.cn

Received 28 March 2014; Revised 14 July 2014; Accepted 14 July 2014; Published 6 August 2014

Academic Editor: Guoqiang Hu

Copyright © 2014 Yang Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Image-based visual servoing for nonholonomic mobile robots using epipolar geometry is an efficient technology for visual servoing problem. An improved visual servoing strategy, namely, three-step epipolar-based visual servoing, is developed for the nonholonomic robot in this paper. The proposed strategy can keep the robot meeting FOV constraint without any 3D reconstruction. Moreover, the trajectory planned by this strategy is shorter than the existing strategies. The mobile robot can reach the desired configuration with exponential converge. The control scheme in this paper is divided into three steps. Firstly, by using the difference of epipoles as feedback, the robot rotates to make the current configuration and desired configuration in the same orientation. Then, by using a linear input-output feedback, the epipoles are zeroed so as to align the robot with the goal. Finally, by using the difference of feature points, the robot reaches the desired configuration. Simulation results and experimental results are given to illustrate the effectiveness of the proposed control scheme.

## 1. Introduction

With the development of computer vision and growing demand for robots intelligence, visual servoing is becoming a hot research field of robotics. Visual servoing is an extensive field where computer vision is used in the design of motion controller. The main task of visual servoing [1] is to regulate the pose (position and orientation) of the robot to reach the desired pose by using the image information obtained by a camera.

Different visual servoing (VS) approaches have been proposed to solve visual servoing problem. They can be classified into two main categories: position-based visual servoing (PBVS) [2–4] and image-based visual servoing (IBVS) [5, 6].

For the position-based visual servoing strategy, the desired pose is estimated on the basis of visual data and geometric models. For instance, an omnidirectional vision system [7] is used to determine the robot posture. A concept of 3D visible set for PBVS [3] is proposed. These strategies add some new concepts to overcome the shortcomings of

PBVS, but all these strategies require 3D reconstruction. 3D reconstruction, under normal circumstances, needs a large amount of computation which makes it difficult for PBVS to do real-time configuration.

To avoid the 3D reconstruction, the image-based visual servoing strategy is proposed. In IBVS, the errors between the initial and desired configuration of the feature points on the image plane are generated, and the feature points are controlled to move from current configuration to the desired configuration on the image plane. IBVS has been known to be more suitable for preventing the feature points from leaving the field of view (FOV) since the trajectories of the feature points are controlled directly on the image plane. An IBVS with Canny operator and line detecting strategy is proposed in [5]. However, image singularities and image local minima may exist due to the form of image Jacobian. This happens frequently as encountered in the use of general IBVS strategy. In order to solve this issue, an approach named homography-based visual servoing has been proposed in [8] for mobile robots, which needs camera calibration parameters and an adaptive estimation of a constant depth-related parameter.

In addition, this strategy cannot let the initial configuration converge to the desired configuration exponentially.

In some unknown environments, the calibration is not exactly known; the visual servoing is to be the uncalibrated visual servoing. So, [9] proposed a quaternion-based camera rotation estimate and a new closed loop error system to solve the robustness of vision-based control systems for the uncalibrated vision servoing. On the basis of [9], an adaptive homography-based visual servo tracking controller [10] is designed to compensate for the lack of unknown depth information using a quaternion formulation to represent rotation tracking error. While a robust adaptive uncalibrated visual servo controller [11] is put forward to asymptotically regulate a robot end-effector to a desired pose compensating for the unknown depth information and intrinsic camera calibration parameters.

Keeping the camera with FOV is an important problem in the visual servoing. Reference [12] presents a novel two-level scheme for adaptive active visual servoing of a mobile robot to provide a satisfactory solution for the field-of-view problem, while [13] introduces a novel visual servo controller that is designed to control the pose of the camera to keep multiple objects in the FOV of a mobile camera. A set of underdetermined task functions are developed to regulate the mean and variance of a set of image features. The nonlinear characteristic of mobile robot is another important problem in the visual servoing. Reference [14] presents a controller for locking a moving object in 3D space at a particular position on the image plane for both the highly nonlinear robot dynamics and unknown motion of the object.

Recently, a novel IBVS strategy [6] is proposed by computing the epipolar geometry between the current image and the desired one. When the angle between focus length  $f$  and  $x$ -axis of the initial configuration is larger than the desired configuration, this strategy can let the initial configuration converge to the desired configuration exponentially. But when the angle between  $f$  and  $x$ -axis of the initial configuration is smaller than the desired configuration, the trajectory planned by the strategy may be much longer, and the time cost by this IBVS strategy may be much increased; sometimes the feature points are out of field of view.

To overcome these shortcomings, we proposed a three-step strategy. Firstly, we add one step to rotate the robot from the initial configuration to the intermediate configuration, which has the same orientation as the desired configuration. With this step, we can guarantee that the angle between  $f$  and  $x$ -axis of the new configuration will never be smaller than the desired configuration. Thereby, the trajectory will always be smaller in our three-step strategy and the robot will keep the feature points with the FOV constraint. Then, a linear input-output feedback is used by the second step to let the epipoles be zero so as to align the robot with the goal. Finally, a proportional plus integral controller is introduced into the third step to take less time reaching the desired configuration.

This paper is organized as follows. Section 2 presents the main task of IBVS, the nonholonomic robot model, and the epipolar geometry. An outline of the system frameworks is given. Section 3 presents the control scheme. Simulations are provided in Section 4 and experiment results are illustrated

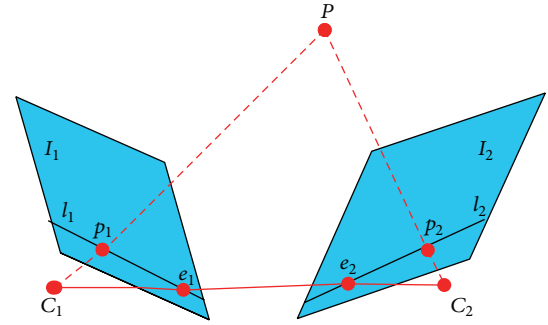


FIGURE 1: The schematic diagram of epipolar geometry.

in Section 5 to evaluate the effect of the proposed control scheme. Finally, Section 6 is a conclusion.

## 2. Problem Formulation

In this section, visual servoing, epipolar geometry, non-holonomic robot model, and general framework are briefly introduced.

*2.1. Visual Servoing.* Robot visual servoing is a strategy for driving mobile robot from current pose (position and orientation) to desired pose (position and orientation) by using feature points of current view and desired view as feedback input while keeping the feature points within the FOV.

*2.2. Epipolar Geometry.* Epipolar geometry describes the intrinsic geometry between two views and only depends on the relative location between cameras and their internal parameters. As shown in Figure 1, the points  $C_1$  and  $C_2$  are called focal length center, the line connecting points  $C_1$  and  $C_2$  is called the baseline, and the intersection of the baseline and the image plane is called epipole, that is,  $e_1$  and  $e_2$ . Let  $P$  be one of the points in 3D space.  $p_1$  and  $p_2$  are the projection points in image planes  $I_1$  and  $I_2$ . The lines  $l_1$  and  $l_2$  are called the epipolar line. From the geometry knowledge,  $e_1$  and  $e_2$  can represent the relative orientation between the image planes. These points will be used later.

The value of the epipoles can be directly computed by the geometrical relationship between the desired and current views. The common method is using fundamental matrix  $F$  to compute the epipoles. Following is the epipolar geometry:

$$\begin{aligned} l_1 &= F^T p_1, \\ l_2 &= F p_2, \\ F &= M^{-T} \times E M^{-1}, \end{aligned} \quad (1)$$

where  $E$  is an essential matrix.

Note that  $F$  can be estimated by some well-known algorithms, like Hartley's normalized 8-point algorithm [15] or others like RANSAC algorithm [16] and LMedS algorithm [17].

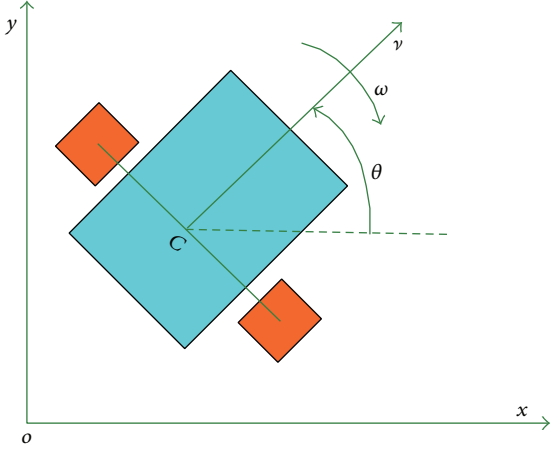


FIGURE 2: The nonholonomic robot with two independently derived wheels.

**2.3. Nonholonomic Robot.** The nonholonomic robot with two independently derived wheels is shown in Figure 2.  $C$  is the mass center of the robot, which is located in the middle of deriving wheels.  $\theta$  is the orientation angle of the robot.  $v$  and  $\omega$  are the linear and angular velocities of the robot. The kinematic model for the robot with the nonholonomic constraint of pure rolling and nonslipping is

$$\begin{aligned}\dot{x} &= v \cos \theta, \\ \dot{y} &= v \sin \theta, \\ \dot{\theta} &= \omega.\end{aligned}\quad (2)$$

**2.4. General Frameworks.** The initial configuration presented in Figures 3(a) and 3(b) is the first step, the main work of which is to take the current configuration in the same direction with the desired configuration. With the role of this step, we can keep the feature points in the field of view. Figure 3(c) is the second step, the main goal of which is to zero the epipoles with input-output linear feedback. With this step, the robot is aligned with the desired configuration. Figure 3(d) is the last step to reach the desired configuration by comparing the feature points in the two views.

The three-step control scheme will be described in detail in Section 3.

### 3. Three-Step Control Scheme

In this section, we will design a three-step strategy scheme for the mobile robot driving to the desired configuration. We will detail how to realize each step.

**3.1. Match the Orientation.** A derivation of epipole kinematics will be presented. First, we will derive the expression of the epipoles as a function of the robot configuration  $r$ .

Figure 4 shows the geometry relation with two views of the same scene. The current configuration is  $r = [x, y, \theta]^T$ , and the desired configuration is  $r_d = [0, 0, \pi/2]^T$ .  $f$  is the

focal distance.  $\varphi$  is the angle between the  $x$ -axis and the line joining the desired and current camera centers.  $\theta$  is the angle between  $f$  and the  $x$ -axis.  $d$  is the distance between the desired camera center and the current camera center.

First of all, we will rotate the robot, and the current configuration and the desired configuration will be in the same orientation. With the first step, the trajectory planned by the strategy in this paper is shorter than the existing strategy and the singular problem can be effectively avoided. The singular problem happened when  $f \perp d$  in red color in Figure 4. But  $f$  is never  $\perp d$  if after the first step the orientation of current configuration and desired configuration is the same orientation. From Figure 4, we have

$$e_{du} = f \frac{x}{y} = f \tan \left( \frac{\pi}{2} - \varphi \right), \quad (3)$$

$$e_{au} = f \tan(\theta - \varphi) = f \frac{\tan \theta - \tan \varphi}{1 + \tan \theta \tan \varphi}, \quad (4)$$

where  $e_{du}$  is the epipole of desired configuration and  $e_{au}$  is the epipole of current configuration. Solving  $e_{du} - e_{au}$  from (3) and (4) yields

$$e_{du} - e_{au} = f \left( \tan \left( \frac{\pi}{2} - \varphi \right) - \tan(\theta - \varphi) \right). \quad (5)$$

The time derivative of  $e_{du} - e_{au}$  is

$$\dot{e}_{du} - \dot{e}_{au} = f v_1 \frac{y \cos(\theta) - x \sin(\theta)}{y^2} - \frac{f(\dot{\theta} - \dot{\varphi})}{\cos^2(\theta - \varphi)}, \quad (6)$$

where  $v_1$  is the linear velocity. We set the control law as

$$v_1 = 0, \quad (7)$$

$$v_2 = g_x (e_{du} - e_{au}),$$

where  $g_x$  is the positive coefficient. Equation (6) can be simplified to be

$$\dot{e}_{du} - \dot{e}_{au} = -v_2 \frac{f}{\cos^2(\varphi - \theta)}, \quad (8)$$

where  $v_2$  is the angular velocity. We can use (7) to rewrite the (8) as

$$\dot{e}_{du} - \dot{e}_{au} = -g_x (e_{du} - e_{au}) \frac{f}{\cos^2(\varphi - \theta)}. \quad (9)$$

Equation (9) can be shortened to be

$$\dot{e}_{du} - \dot{e}_{au} = -g (e_{du} - e_{au}), \quad (10)$$

where

$$g = \frac{g_x f}{\cos^2(\varphi - \theta)} > 0. \quad (11)$$

From (10),  $e_{du} - e_{au}$  will converge to zero in a limited time. So, we take (7) as control law and then the mobile robot will go to the same orientation with the desired configuration. If the feature points of the desired configuration are in the field of view, then the next two steps will keep the feature points in the field of view. Note the following points.

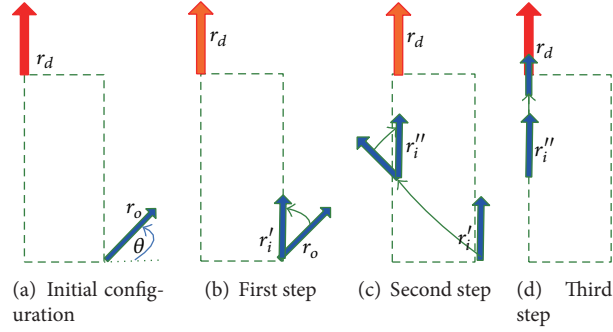


FIGURE 3: The scheme of three-step epipolar-based visual servoing system. (a) The initial configuration of the mobile robot; (b) the first step that the mobile robot will move from  $r_o$  to  $r'_i$ ; (c) the second step that the mobile robot will move from the  $r'_i$  to  $r''_i$ ; (d) the third step that the mobile robot will move from the  $r''_i$  to  $r_d$ .

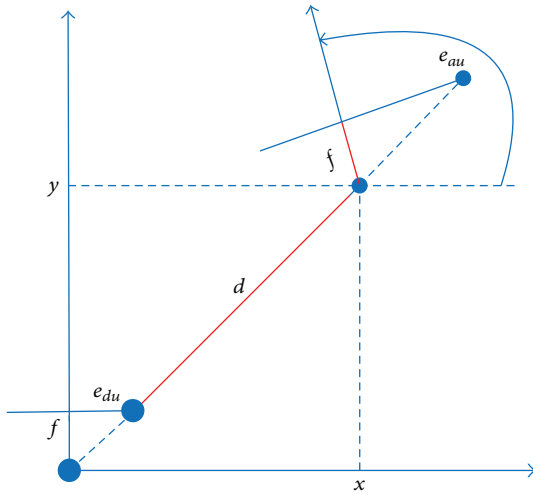


FIGURE 4: The geometry relation between current configuration and desired configuration.

- (i) The above control law (7) is image-based, since it only uses the measured epipoles. No information of the robot configuration or any other odometric data is used.
- (ii) The form of (10) is essential to guarantee that  $e_{du} - e_{au}$  is zero in finite time.
- (iii) From  $e_{du} - e_{au}$ , we know that the robot will converge to the same orientation from the desired configuration.

It remains to be shown how to avoid the proposed control law (7) to become singular. As is shown in the control law, the linear velocity ( $v_1$ ) is always defined and the angular velocity ( $v_2$ ) has a potential singularity when  $y = 0$  or  $(\varphi - \theta) = \pi/2$ . The following remarks are in order at this point.

*Remark 1.* We provide the control law just running under the condition that the epipolar geometry between the desired and current camera views can be defined, corresponding with the case of  $y \neq 0$  and  $(\varphi - \theta) \neq \pi/2$ . If  $y = 0$  (see (3), (6), and control law (7) that are undefined) or  $(\varphi - \theta) = \pi/2$  (see (4), (6), and control law (7) that are undefined), the homography

matrix  $H$  can be decomposed to design a replaced rotational controller to diminish the orientation error. Using stacking of the fundamental matrix [18]  $F$  to observe the norm of the 9-dimensional vector, estimate the homography matrix  $H$  which is still defined in this situation and decompose it to get the rotation matrix  $R$  between the desired and current camera views. At last, we provide a simple proportional rotational controller to diminish the orientation error.

*Remark 2.* Here, we provide how to choose the values of controller parameter.

- (i) *Choice of  $g_x$ .*  $g_x$  should be the positive value. The value of  $g_x$  determines the rate of convergence  $v_2$ .

*3.2. Zero the Epipoles.* For feedback-linearization purposes, we will need the kinematics of the epipoles with respect to the velocities  $v_1$  and  $v_2$ . From (3) and (4), the time derivative of  $e_{du}$  is obtained as follows:

$$\dot{e}_{du} = f \frac{v_1 y \cos \theta - v_1 x \sin \theta}{y^2}. \quad (12)$$

As shown in Figure 4

$$\begin{aligned} y &= d \sin(\varphi), \\ x &= d \cos(\varphi). \end{aligned} \quad (13)$$

So,  $\dot{e}_{du}$  and  $\dot{e}_{au}$  can be expressed as

$$\dot{e}_{du} = v_1 \frac{f \sin(\varphi - \theta)}{d \sin^2(\varphi)}, \quad (14)$$

$$\dot{e}_{au} = \frac{f}{\cos^2(\theta - \varphi)} (\dot{\theta} - \dot{\varphi}). \quad (15)$$

From Figure 4, we can obtain

$$\begin{aligned} \varphi &= \arctan\left(\frac{y}{x}\right), \\ \dot{\varphi} &= \frac{(\dot{y}x - \dot{x}y)}{d^2}. \end{aligned} \quad (16)$$

Taking the two equations above into (15), the time derivative of  $e_{au}$  can be written as

$$\dot{e}_{au} = v_1 \frac{f \sin(\varphi - \theta)}{d \cos^2(\varphi - \theta)} + v_2 \frac{f}{\cos^2(\varphi - \theta)}. \quad (17)$$

From some simple geometry knowledge, we have

$$\begin{aligned} \cos(\varphi - \theta) &= \text{sign}(e_{au}e_{du}) \frac{f}{\sqrt{e_{au}^2 + f^2}}, \\ \sin(\varphi - \theta) &= -\text{sign}(e_{au}e_{du}) \frac{e_{au}}{\sqrt{e_{au}^2 + f^2}}, \end{aligned} \quad (18)$$

where the sign functions guarantee that any position of the current configuration is taken into consideration in these equations. We take these into (14) and (15), and the relationship between the input and the output time derivatives is expressed as

$$\dot{e}_{au} = -v_1 \frac{\text{sign}(e_{au}e_{du}) e_{au} \sqrt{e_{au}^2 + f^2}}{d f} + v_2 \frac{e_{au}^2 + f^2}{f}, \quad (19)$$

$$\dot{e}_{du} = -v_1 \frac{\text{sign}(e_{au}e_{du}) e_{au} (f^2 + e_{du}^2)}{d f \sqrt{e_{au}^2 + f^2}}, \quad (20)$$

where  $d$  is the distance of two epipoles. In summary, the simple matrix form should be

$$\begin{bmatrix} \dot{e}_{au} \\ \dot{e}_{du} \end{bmatrix} = H \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad (21)$$

with

$$H = \begin{bmatrix} \frac{\text{sign}(e_{au}e_{du}) e_{au} \sqrt{e_{au}^2 + f^2}}{d f} & \frac{e_{au}^2 + f^2}{f} \\ \frac{\text{sign}(e_{au}e_{du}) e_{au} (e_{du}^2 + f^2)}{d f \sqrt{e_{au}^2 + f^2}} & 0 \end{bmatrix}. \quad (22)$$

Here, we are faced with a difficulty that  $d$  is unknown in the image-based control and the input-output is not linear. We set the control law as

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \widehat{H}^{-1} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad (23)$$

with

$$\widehat{H}^{-1} = \begin{bmatrix} 0 & \frac{-\text{sign}(e_{au}e_{du}) \widehat{d} f \sqrt{e_{au}^2 + f^2}}{e_{au} (e_{du}^2 + f^2)} \\ \frac{f}{e_{au}^2 + f^2} & \frac{-f}{e_{du}^2 + f^2} \end{bmatrix}. \quad (24)$$

Taking (23) into (21), we obtain

$$\begin{bmatrix} \dot{e}_{au} \\ \dot{e}_{du} \end{bmatrix} = H \widehat{H}^{-1} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 1 & \left( \frac{\widehat{d}}{d} - 1 \right) \frac{e_{au}^2 + f^2}{e_{du}^2 + f^2} \\ 0 & \frac{\widehat{d}}{d} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \quad (25)$$

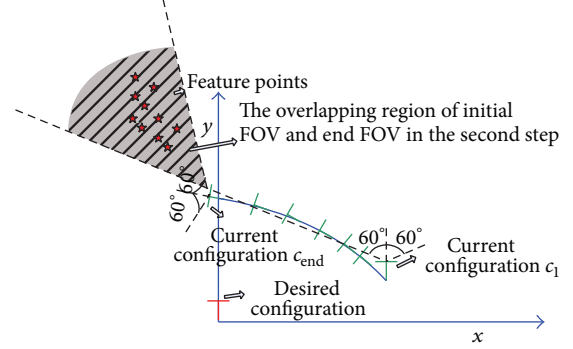


FIGURE 5: Find overlapping region of initial FOV and end FOV in the second step.

in which we set

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} -g_1 e_{au} \\ -g_2 e_{du}^{\beta/\gamma} \end{bmatrix}, \quad (26)$$

where  $g_1 > 0$ ,  $g_2 > 0$ ,  $\beta$ , and  $\gamma$  are the position odd integers with  $\beta < \gamma$ . Also, update the distance estimate  $d$  according to the following equation:

$$\dot{\widehat{d}} = g_2 f^2 \widehat{d} e_{du}^{\beta/\gamma}. \quad (27)$$

Taking (27) into (26), we obtain

$$\dot{e}_{au} = -g_1 e_{au} - \left( \frac{\widehat{d}}{d} - 1 \right) \frac{e_{au}^2 + f^2}{e_{du}^2 + f^2} g_2 e_{du}^{\beta/\gamma}, \quad (28)$$

$$\dot{e}_{du} = -\frac{\widehat{d}}{d} g_2 e_{du}^{\beta/\gamma}. \quad (29)$$

The control law can be written as follows:

$$v_1 = \text{sign}(e_{au}e_{du}) \frac{\widehat{d} f \sqrt{e_{au}^2 + f^2}}{e_{au} (e_{du}^2 + f^2)} g_2 e_{du}^{\beta/\gamma}, \quad (30)$$

$$v_2 = -\frac{f}{e_{au}^2 + f^2} g_1 e_{au} + \frac{f}{e_{au}^2 + f^2} g_2 e_{du}^{\beta/\gamma}.$$

Then,  $e_{au}$  and  $e_{du}$  will converge to zero in finite time.

It is assumed that the angle of camera view is  $120^\circ$ . For example, in Figure 5, the maximum differential angle of camera view is between the conditions  $c_1$  and  $c_{end}$ . In the circumstances, the feature points can be set in the shaded area. From Figure 5, it can be seen that when the robot moves to the second step, the feature points will keep in FOV.

*Remark 3.* It remains to be shown how to adjust the control if the proposed control law (30) becomes singular. As is shown in the control law (30), the angular velocity ( $v_2$ ) is always defined and the linear velocity ( $v_1$ ) has a potential singularity when  $e_{au}$  is equal to zero.

(i) If  $e_{au}$  is equal to zero at the beginning of the second step, we can perform a preliminary maneuver in

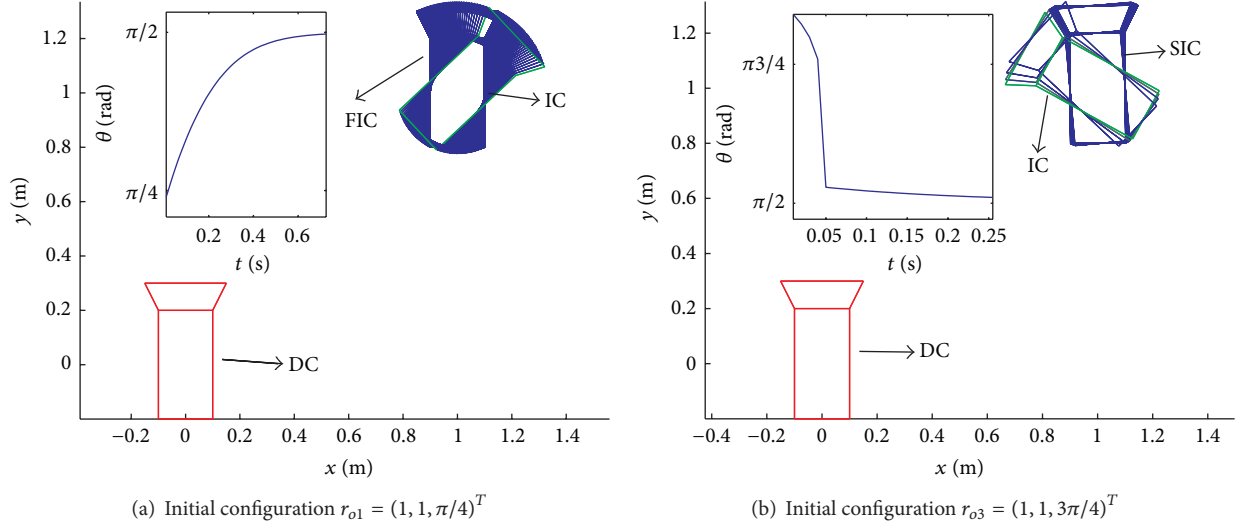


FIGURE 6: The first step: the robot trajectory. (a) The initial configuration is  $r_{o1} = (1, 1, \pi/4)^T$ ; (b) the initial configuration is  $r_{o3} = (1, 1, 3\pi/4)^T$ . DC is the desired configuration, FIC is the intermediate configuration of the first step, and IC is the initial configuration. Both figures show that the orientation of the robot is towards  $\pi/2$  (the orientation of desired configuration). In this step, the robot is moving from IC to SIC.

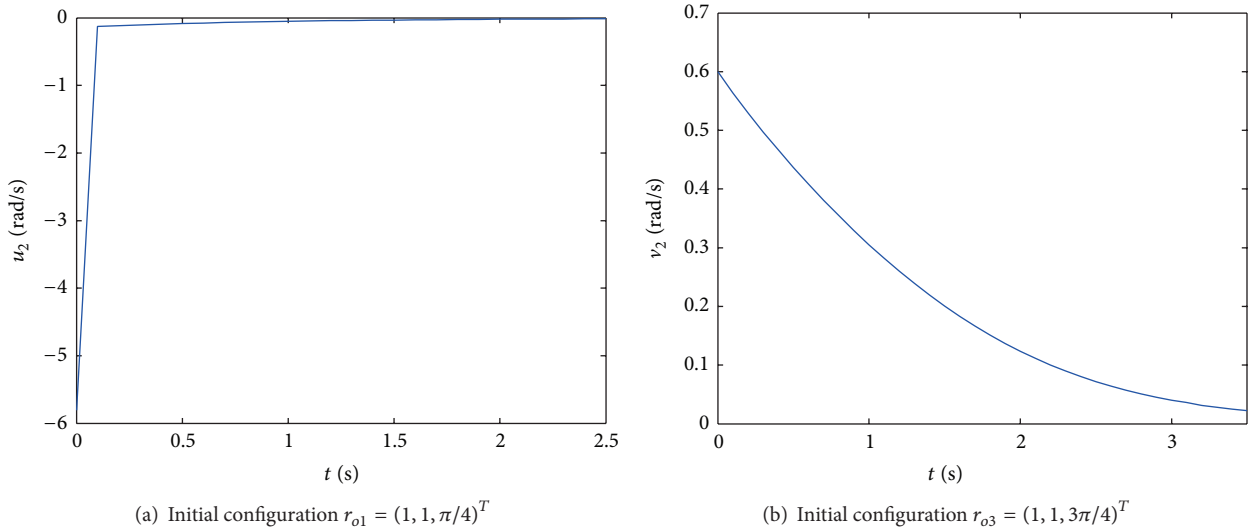


FIGURE 7: The first step. (a) is in the initial configuration  $r_{o1} = (1, 1, \pi/4)^T$ , (b) is in the initial configuration  $r_{o3} = (1, 1, 3\pi/4)^T$ , and these figures show the regular velocity  $v_2$  of the robot in the first step.

order to displace the current epipole to a nonzero value. Using a small certain value  $\widetilde{e}_{au}$ , the preliminary control is as follows:

$$\begin{aligned} v_1 &= 0, \\ v_2 &= -\frac{f}{e_{au}^2 + f^2} g_1 (e_{au} - \widetilde{e}_{au}). \end{aligned} \quad (31)$$

With this choice, we get from (19)  $\dot{e}_{au} = -g_1(e_{au} - \widetilde{e}_{au})$ , so  $e_{au}$  converges exponentially to  $\widetilde{e}_{au}$ , and then we can use our provided control law.

(ii) According to (29), as  $e_{du}^{\beta/\gamma}$  is bounded,  $e_{du}$  will converge to zero at finite time  $t_x$ . From  $t_x$ , (28)

becomes  $\dot{e}_{au} = -g_1 e_{au}$ ,  $e_{au}$  will converge to zero with exponential rate  $g_1$ ,  $v_1$  is equal to zero, and  $v_2 = -(f/(e_{au}^2 + f^2))g_1 e_{au}$ . So, the robot system will perform the pure rotation in this phase. Hence, convergence of the epipoles to zero is obtained at a finite distance.

(iii) As already noticed, after the transient ( $t > t_x$ ),  $e_{du}$  is equal to zero and  $v_1$  is equal to zero, which prevents the potential singularity.  $e_{au}$  can be bounded by bounding  $g_2$ . For sufficiently small  $g_2$ , the current epipole  $e_{au}$  cannot cross zero during the transient. With the desired epipole,  $e_{du}$  reaches zero before  $e_{au}$ , and the proposed control law is never singular. This



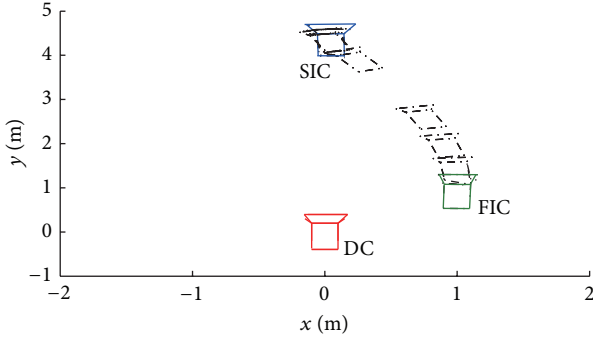


FIGURE 8: The second step: the robot trajectory. DC is the desired configuration, SIC is the intermediate configuration of the second step, and FIC is the intermediate configuration of the first step. In this step, the robot is moving from FIC to SIC.

kind of control law is also known as a terminal sliding mode.

*Remark 4.* We now present how to choose the control parameters, that is,  $\beta/\gamma$ ,  $g_1$ ,  $g_2$ , and initial estimate of the robot distance  $\hat{d}$ .

- (1) *Choice of  $\beta/\gamma$ .* The current and desired robot positions may increase during the second step. According to (30), when the  $e_{au}$  and  $e_{du}$  increase, we want to decrease the value of  $v_1$  and  $v_2$ , so  $\beta/\gamma$  should be chosen less than one and close to zero.
- (2) *Choice of  $g_1$ ,  $g_2$ .* Remark 3 requires  $g_2$  to be sufficiently small to guarantee that the proposed control law (30) is never singular. We can choose  $g_1$ , based on the characteristic that  $e_{au}$  and  $e_{du}$  never change sign. Now, we take the following into account.
  - (i) If the initial epipole values  $e_{au}$  and  $e_{du}$  are in different signs, the perturbation term in (28) pushes  $e_{au}$  into singular.  $g_2$  should be very sufficiently small. There is no special strategy for  $g_1$ .
  - (ii) If the initial epipole values  $e_{au}$  and  $e_{du}$  are in the same sign, the perturbation term in (28) pulls  $e_{au}$  away from singular. Any value of  $g_2$  is satisfiable.  $g_2$  only determines the rate of convergence  $e_{au}$  after the transient ( $t_x$ ).
  - (iii) In both simulation and experiments, the choice of  $g_1$ ,  $g_2$  is sufficient to achieve singularity avoidance.

- (3) *Choice of Initial Estimate of the Robot Distance  $\hat{d}$ .* According to Remark 3, it is necessary to initialize  $\hat{d}$  at a value  $\hat{d}_0 > d_0$ , where  $\hat{d}_0$  means the initial estimate value of  $d$  and  $d_0$  means the initial value of  $d$ . We can use an upper bound derived from the knowledge of the environment, where the robot moves.

**3.3. Match the Feature Points.** At the end of the second step, both  $e_{au}$  and  $e_{du}$  are zero and the intermediate configuration

is in the same orientation with the desired configuration. Now, we are facing the problem of how to use feature points to realize the control law into the robot from intermediate configuration to desired configuration. Similar to the above two steps, the third step control law works in the camera image plane. The basic idea is to make each feature point in the current image plane match the feature points in the desired image plane. In principle, only one feature point is needed to implement this idea. We can also use a number of feature points as a choice in case of noisy images. We set  $D = \|p_a\|^2 - \|p_d\|^2$ , the difference between the squared norms of the current feature  $p_a$  and the desired feature  $p_d$ .

If proportional control is used here to be the control law, then the system will cost much time to reach the desired configuration. So, we let the control law be proportional plus integral controller as

$$\begin{aligned} v_1 &= -g_t D - g_i * \int_0^t D dt, \\ v_2 &= 0, \end{aligned} \quad (32)$$

where  $g_t > 0$  and  $g_i > 0$ . Then, the current configuration will converge from the intermediate configuration  $r_i$  to the desired configuration  $r_d$  exponentially.

*Remark 5.* We provide how to choose the values of controller parameters.

- (i) *Choice of  $g_t$ ,  $g_i$ ,  $g_t$ , and  $g_i$*  should be the positive value. The value of  $g_t$  determines the rate of convergence  $v_1$ . The value of  $g_i$  determines the control precision of the system.

## 4. Simulation Results

In this section, simulation results are provided to validate the proposed approach. The scene consists of ten feature points which are random on the plane. Simulations have been performed using MATLAB and the Epipolar Geometry Toolbox [19]. Ten pairs of corresponding feature points are used in the desired and current image. They are used in all three steps. In the simulation, we use  $f = 0.04$  m. The initial and desired configurations are chosen as

$$\begin{aligned} r_{o1} &= \left(1, 1, \frac{\pi}{4}\right)^T, \\ r_{o2} &= \left(1, 1, \frac{\pi}{2}\right)^T, \\ r_{o3} &= \left(1, 1, \frac{3\pi}{4}\right)^T, \\ r_d &= \left(0, 0, \frac{\pi}{2}\right)^T. \end{aligned} \quad (33)$$

We use three initial configurations standing for different situations. In the first step, the parameter is  $g_x = 30$ . In the second step, the parameters are  $g_1 = 0.6$ ,  $g_2 = 0.4$ , and  $\beta/\gamma = 4/9$ . And in the third step, the parameters are  $g_t = 1000$ ,  $g_i = 10$ . We set the initial estimate of the robot distance  $\hat{d} = 3$  m.

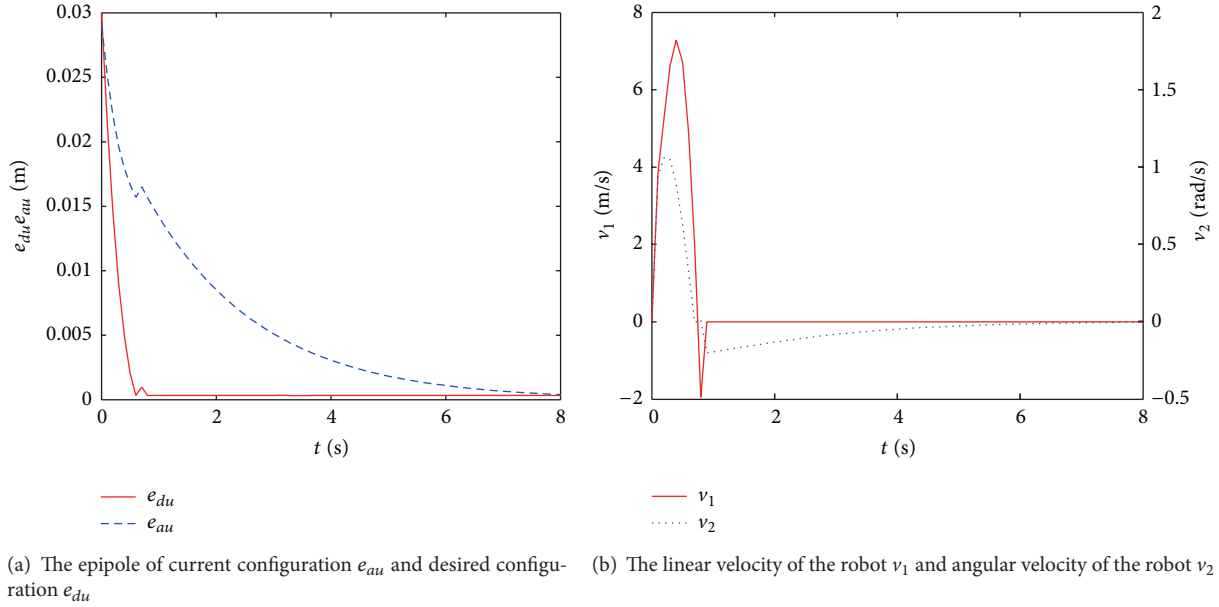


FIGURE 9: The second step. (a) The epipole of current configuration  $e_{au}$  and the epipole of desired configuration  $e_{du}$  in the second step. The  $e_{au}$  and  $e_{du}$  will be zero in finite time. (b) The linear velocity of the robot  $u_1$  and angular velocity of the robot  $u_2$  in the second step.

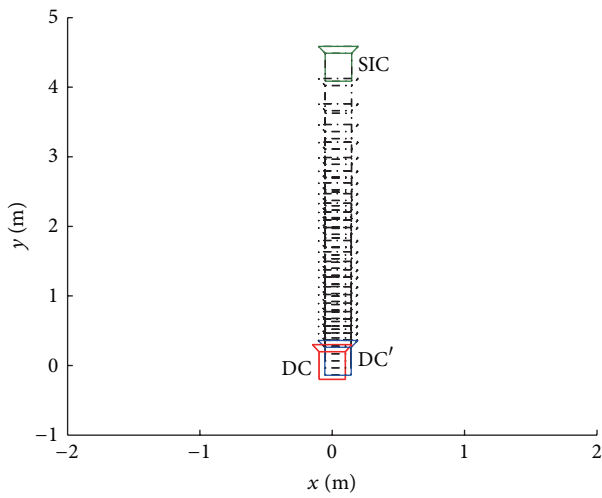


FIGURE 10: The third step: the robot trajectory. DC is the desired configuration, DC' is the final configuration, and SIC is the intermediate configuration of the second step. In this step, the robot is moving from SIC to DC'.

The trajectory of the robot is shown in Figures 6, 8, and 10.

In the first step, the robot takes  $r_{o1}$  as initial configuration. Figures 6(a) and 7(a) show the robot trajectory and the regular velocity (the linear velocity is zero). In the beginning, the robot is in the orientation of  $\pi/4$ . So, the difference between  $e_{du}$  and  $e_{au}$  is very large; the regular velocity is also the same. In this case  $e_{du}$  is smaller than  $e_{au}$ , so the regular velocity is positive. The robot will rotate clockwise and get closer to the orientation of  $\pi/2$ , and the regular velocity decreases. We can get exponential convergence as shown in Figure 7(a).

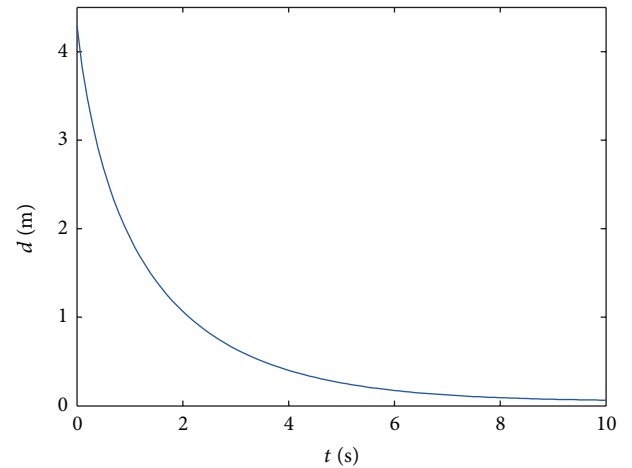


FIGURE 11: The third step. The distance between the current configuration and the desired configuration in the third step.

The robot takes  $r_{o3}$  as initial configuration; the result is shown in Figures 6(b) and 7(b). This case is just the opposite to the above. At first, the robot is in the orientation of  $3\pi/4$ .  $e_{du}$  is larger than  $e_{au}$ . So, the regular velocity is negative. The robot will rotate anticlockwise and the convergence will be exponential as shown in Figure 7(b).

The initial configuration is  $r_{o2}$ . In the beginning, the robot is in the orientation of  $\pi/2$ . So,  $e_{du}$  is equal to  $e_{au}$ ; this step is finished at first.

In the second step, together with effect of the first step, regardless of the initial configuration, at the beginning of the second step, the robot is in the orientation of  $\pi/2$ . And these three initial configurations are equal now. So, only one of



TABLE 1: Simulation results of final configuration.

Initial configuration	Desired configuration	Final configuration using [6] strategy	Final configuration using three-step strategy
$(2, 4, 0.7\pi)$	$(0, 0, 0.5\pi)$	$(0.0316, -0.1384, 0.5013\pi)$ $t = 26.9$ s	$(0.1114, -0.0227, 0.4999\pi)$ $t = 11.9$ s
$(2, 1, 0.3\pi)$	$(0, 0, 0.5\pi)$	$(-0.0360, 20.2318, 0.5021\pi)$ $t = 30$ s	$(0.0394, 0.0148, 0.5013\pi)$ $t = 12.2$ s
$(1, 1, 0.3\pi)$	$(0, 0, 0.5\pi)$	$(-0.0358, 139.0497, 0.5004\pi)$ $t = 30$ s	$(0.0392, 0.0411, 0.5013\pi)$ $t = 9.7$ s

The desired configuration coordinate is set by ourselves, so we set  $(0, 0, 0.5\pi)$  without loss of generality. The simulation time is 30 s.  $t$  means the whole visual servo processing time.

TABLE 2: Comparison of distance using different strategies.

Angle $(\theta)^1$	Distance using strategy of [6] $(d)^2$	Distance using three-step strategy $(d)^2$
120°	4.0272 m	4.1498 m
115°	4.0250 m	4.1498 m
110°	4.0188 m	4.1498 m
105°	4.0213 m	4.1498 m
100°	4.0271 m	4.1498 m
95°	4.0224 m	4.1498 m
90°	4.1498 m	4.1498 m
85°	5.4232 m	4.1498 m
80°	7.2354 m	4.1498 m
75°	9.1564 m	4.1498 m
70°	18.398 m	4.1498 m
65°	28.657 m	4.1498 m
60°	71.493 m	4.1498 m
55°	184.77 m	4.1498 m

<sup>1</sup> $\theta$  is the angle between  $f$  and  $x$ -axis.

<sup>2</sup> $d$  is the distance between the second intermediate configuration and desired configuration as in Figure 10 SIC  $\rightarrow$  DC.

them is needed to be discussed. In Figure 9(a), as expected,  $e_{du}$  and  $e_{au}$  are declined to zero. The  $e_{du}$  is zeroed at time  $t = 1$  s, and the  $e_{au}$  is zeroed at the time  $t = 7$  s. While the input control ( $v_1$  and  $v_2$ ) is shown in Figure 9(b), Figure 8 shows the robot trajectory.

In the third step, the robot trajectory shown in Figures 10 and 11 shows the distance between current configuration and desired configuration.

The desired configuration and final configuration are summarized in Table 1; it is clear to see that the final configurations are very close to the desired ones by using one three-step strategy. But, by using the strategy of [6], only the first group is close to the desired one, and the others cannot reach the desired one, and the processing time of three-step strategy is much shorter than the strategy of [6]. From Table 2, if  $\theta$  is less than 75°, the path planned by the strategy of [6] will be an error. But the path planned by three-step strategy will be of the same short distance, 4.1498 m. If  $\theta$  is more than 90°, the paths planned by the strategy of [6] and the three-step strategy are almost the same. You can see that the three-step strategy is more robust and efficient in this case than the strategy proposed by [6]. Figure 12 shows the trajectory of the second group in Table 1. Figure 13 shows

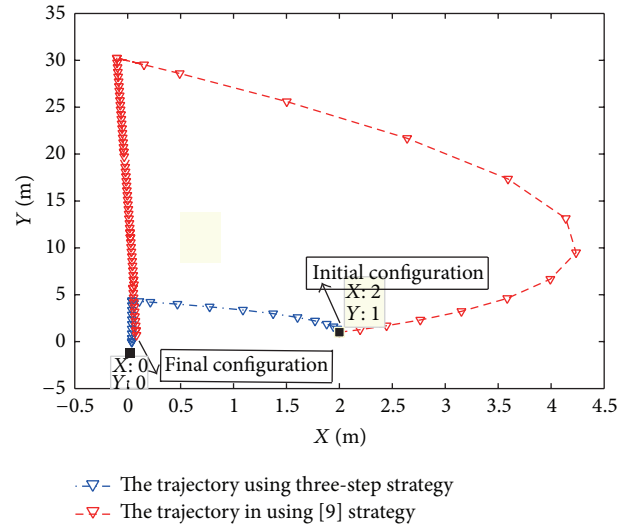


FIGURE 12: The trajectory of the second group in Table 1.

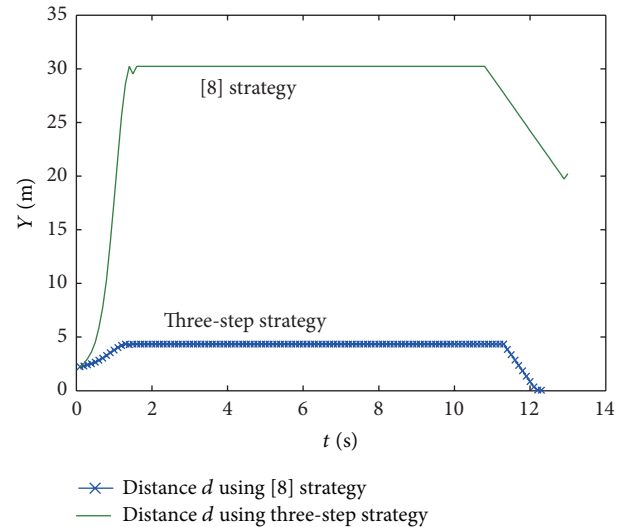


FIGURE 13: The simulation time is 13 s; the distance between the current configuration and the desired configuration of the second group in Table 1.

the distance between the current configuration and desired configuration of the second group in Table 1.

The movement of feature points in the first step is shown in Figure 14. The feature points in current configuration are

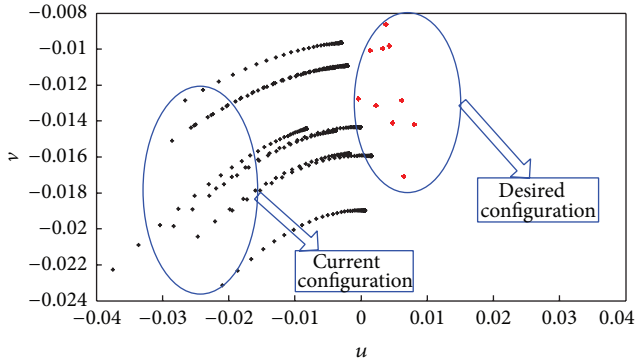


FIGURE 14: The first step: movement of feature points.

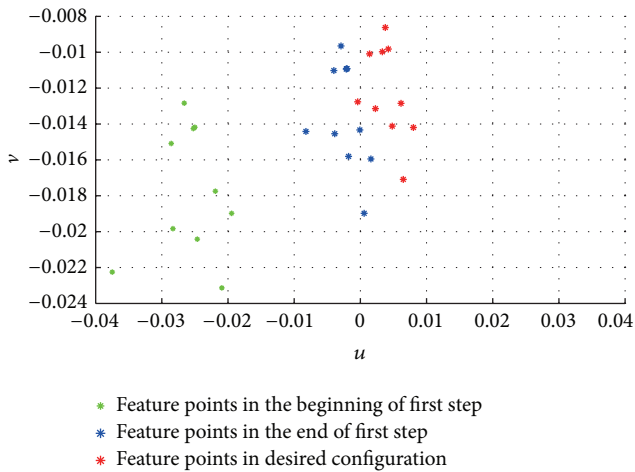


FIGURE 15: The first step: final result of feature points.

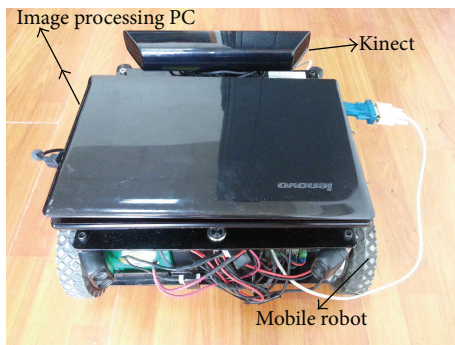


FIGURE 16: Testbed. Mobile robot used for the experiments equipped with a KINECT by Microsoft Company, Ltd.

moving close to the feature points in desired configuration, which is effective in keeping the camera with FOV.

The simulation results are shown in Figure 15. Feature points of current configuration are in close proximity to the desired ones.

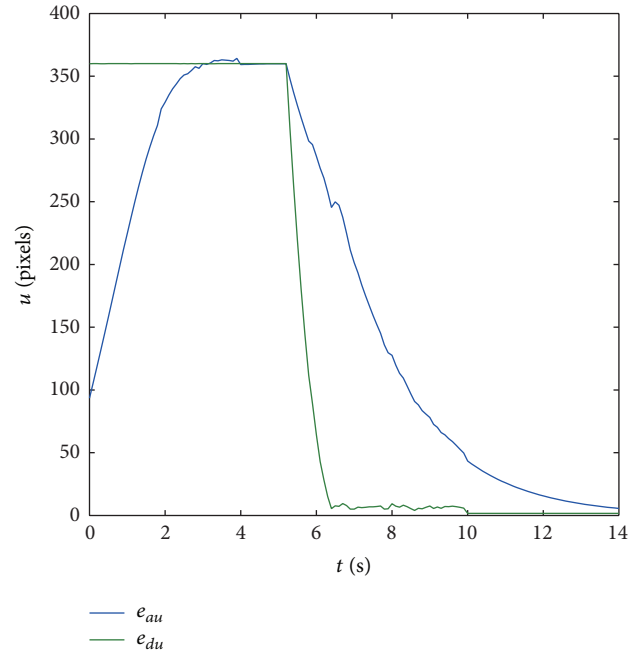


FIGURE 17: Experiment result. Epipole behavior.

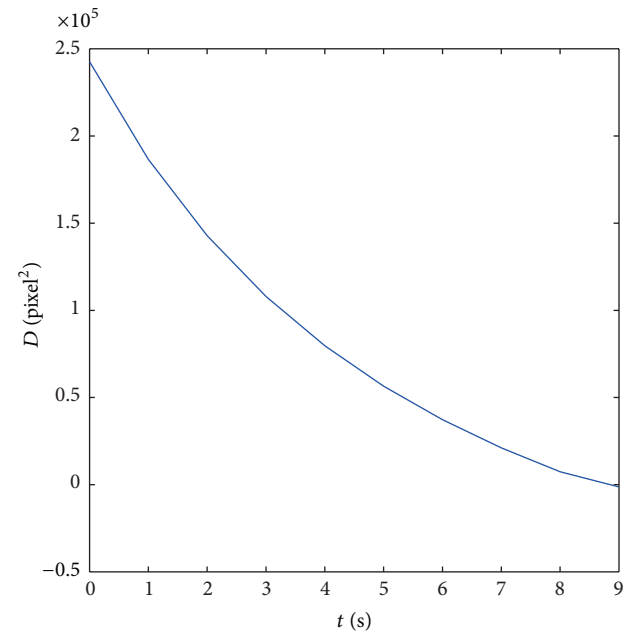


FIGURE 18: Experimental results. Exponential decrease of the difference  $D$  between the actual and desired features.

## 5. Experiment Results

**5.1. Testbed.** As shown in Figure 16, the testbed consists of the following components: a differential drive mobile robot (with an Samsung ARM S3C 2410 inside), a kinect camera that captures 30 frames per second with eight-bit RGB-image at a  $640 \times 480$  resolution, a first Intel core i5 inside PC (operating under the MS Windows 7 x64 operating system), and a second Intel core i5 inside PC (operating under the Ubuntu



FIGURE 19: Experimental results. Snapshots of the desired configuration of robot (a). Also, snapshots of the feature motion in desired configuration (b).

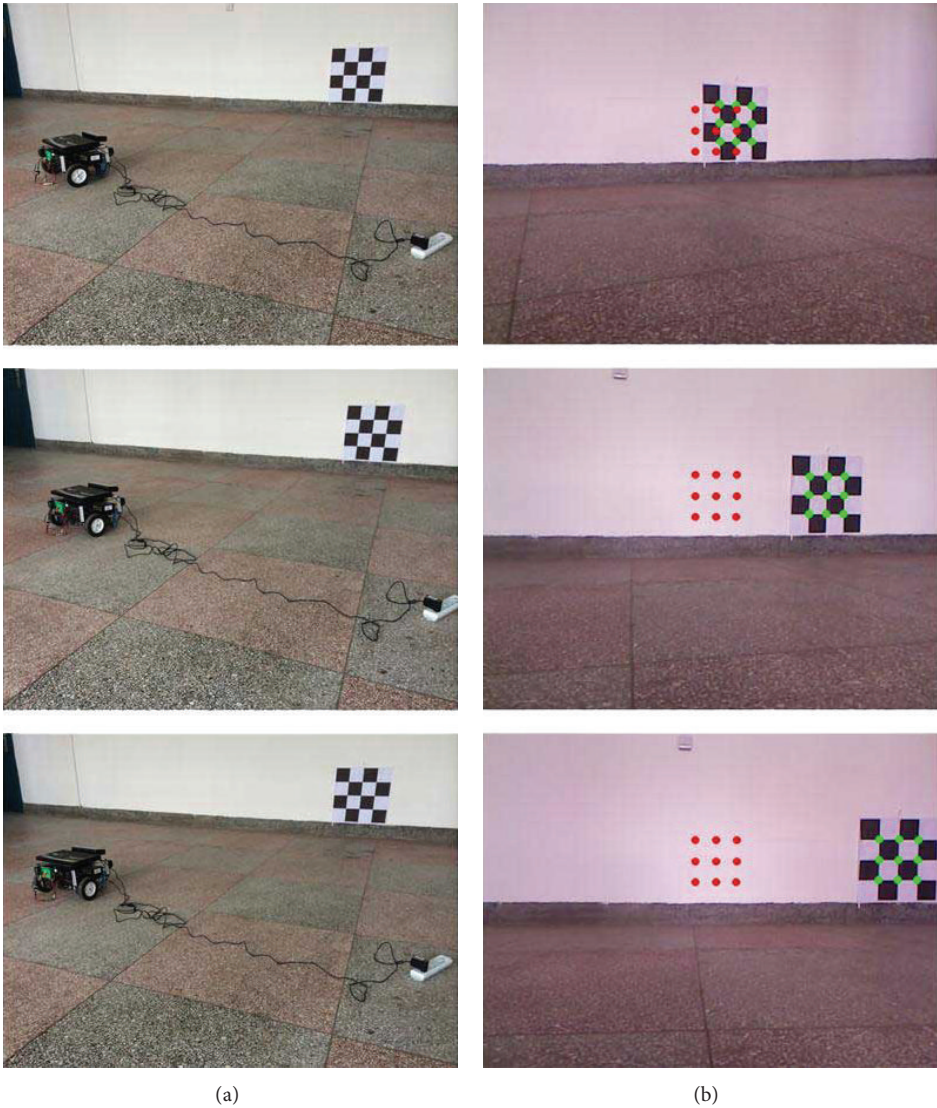


FIGURE 20: Experimental results. Snapshots of the robot motion (a) during the first step. Also, snapshots of the feature motion (b) superimposed on the current image.



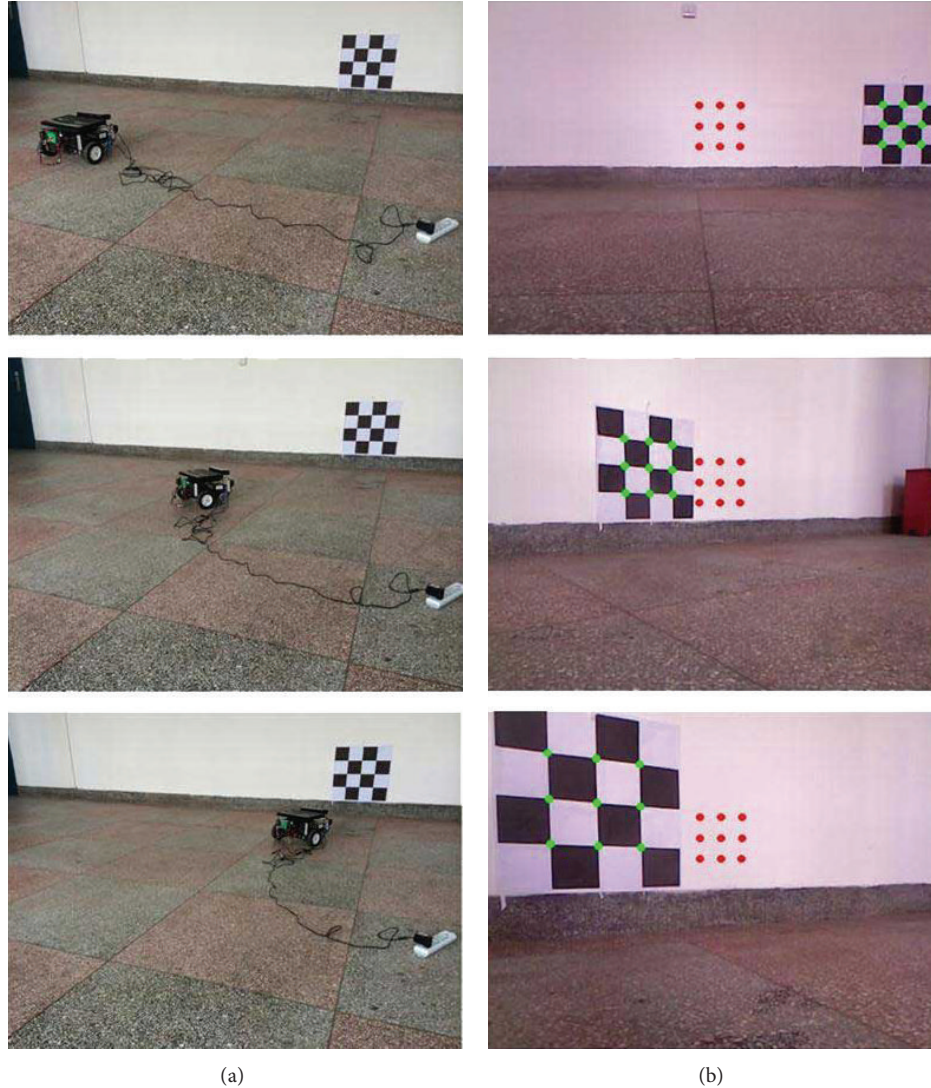


FIGURE 21: Experimental results. Snapshots of the robot motion (a) during the second step. Also, snapshots of the feature motion (b) superimposed on the current image.

10.04 operating system, a Linux kernel based operating system). The internal mobile robot controller (Samsung ARM S3C 2410) hosts the control algorithm that was written in Linux-C/C++. The first PC is used for image processing. IMAGE PROCESSING, the image processing algorithm, is written in MS visual studio MFC (C++ based with the aid of the OPENCV 2.4.1 library). The communication protocol between the image processing PC and internal mobile robot controller is the serial communication. The second PC is a remote PC, and it is used to remotely login to the internal mobile robot controller via Telnet. The remote PC can log the run data of mobile robot and can also debug the mobile robot. The chessboard is rigidly attached to a rigid structure that is used as the target. We use the OPENCV FindChessboard algorithm to determine the coordinates of each point in the chessboard. The mobile robot is controlled by a torque input. The torque controller requires the actual linear and angular velocity of the mobile robot. So, the mobile robot

is equipped with the steering motor encoders. The encoder data is performed by the DSP controller. The communication between DSP controller and internal mobile robot controller is CAN communication. Using the OPENCV camera calibration algorithm, the intrinsic calibration parameters of the KINECT are determined. The image center coordinates are determined to be  $u_0 = 319.5$  [pixels] and  $v_0 = 239.5$  [pixels], and the focal lengths are  $fk_u = 620.7$  [pixels] and  $fk_v = 620.7$  [pixels]. The intrinsic matrix is to be

$$K = \begin{bmatrix} 620.7 & 0 & 319.5 \\ 0 & 620.7 & 239.5 \\ 0 & 0 & 1 \end{bmatrix}. \quad (34)$$

5.2. Results. For the visual servoing task, a set of 9 corresponding feature points are chosen in the two images and tracked in real time by means of a FindChessboard algorithm. The robot moves under the three-step visual

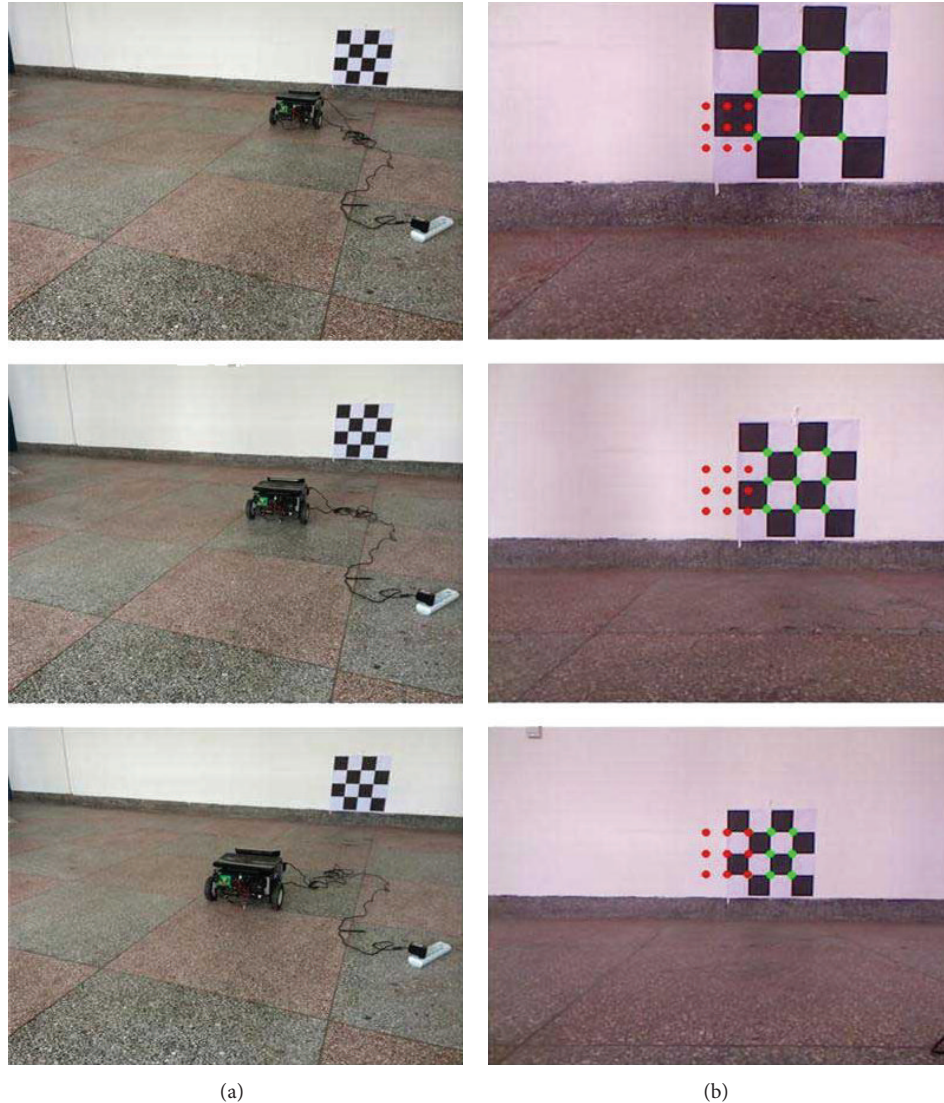


FIGURE 22: Experimental results. Snapshots of the robot motion (a) during the third step. Also, snapshots of the feature motion (b) superimposed on the current image.

servoing algorithm. In the first step, the parameter is  $g_x = 30$ . In the second step, the parameters are  $g_1 = 0.6$ ,  $g_2 = 0.4$ , and  $\beta/\gamma = 4/9$ . And, in the third step, the parameters are  $g_t = 1000$  and  $g_i = 10$ . We set the initial estimate of the robot distance  $\hat{d} = 7$  m.

First, control law (6) will take action, as shown in Figure 17. Both the epipoles  $e_{au}$  and  $e_{du}$  are driven to the same value. Due to the actuator deadzone, we just need  $e_{au}$  to be close to  $e_{du}$ .

Then, the second step is carried out under the action of the control law (30). Both the epipoles  $e_{au}$  and  $e_{du}$  are driven to zero. Due to the actuator deadzone, in Figure 17, the epipoles  $e_{au}$  and  $e_{du}$  are almost zero but not exactly zero.

Finally, the third step is executed under the influence of the control law (26); the  $D$  (the difference between the current configuration and desired configuration) will

decrease exponentially. Figure 18 shows that  $D$  will decrease nearly to zero.

Figures 20, 21, and 22 collect nine snapshots of the robot motion during the first, second, and third steps, respectively. On the right of each snapshot, the current (green) and desired (red) feature points are shown to be superimposed to the current image. Figure 19 shows the desired feature points (right) and desired configuration of robot (left). During the first step (Figure 20), the epipoles are driven to be the same value. The orientation of the current configuration and desired configuration will be the same. While the second step (Figure 21) is carried out, the epipoles are driven to the principal point. The third step (Figure 21) is then executed, and the current feature points converge to their target. The overall servoing performance is satisfying, resulting in a positioning error of about 3 cm left with respect to the target position.



## 6. Conclusions

In this paper, a new visual servoing strategy is proposed, named three-step epipolar-based visual servoing, which includes three steps. Firstly, by using the difference of epipoles as feedback, the robot rotates to make the current configuration and desired configuration in the same direction. Secondly, by using a linear input-output feedback, the epipoles are zeroed so as to align the robot with the goal. Thirdly, by using the feature points, the robot reaches the desired configuration.

The main advantages of the proposed control scheme includes: (1) importing the first step to rotate to desired configuration; (2) adding integral control to accelerate the convergence in the third step. The strategy is capable of solving the problem of keeping feature points in FOV and moreover planning the path correctly and shortly compared with [6] which can be evaluated through the simulation results and experiment results.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (nos. 61379111, 61071096, 61073103, 61003233, and 61202342), Specialized Research Fund for Doctoral Program of Higher Education (nos. 20100162110012 and 20110162110042), and China Postdoctoral Science Foundation, Postdoctoral Science Planning Project of Hunan Province, Postdoctoral Science Foundation of Central South University (120951).

## References

- [1] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 313–326, 1992.
- [2] F. Janabi-Sharifi and M. Marey, "A kalman-filter-based method for pose estimation in visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 939–947, 2010.
- [3] D.-H. Park, J.-H. Kwon, and I.-J. Ha, "Novel position-based visual servoing approach to robust global stability under field-of-view constraint," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4735–4752, 2012.
- [4] V. Lippiello, B. Siciliano, and L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 73–86, 2007.
- [5] W. Gang, M. Zhengda, and L. Jusan, "A method of error compensation in image based visual servo," in *Proceedings of the International Conference on Electrical and Control Engineering (ICECE '10)*, pp. 83–86, Wuhan, China, June 2010.
- [6] G. L. Mariottini, G. Oriolo, and D. Prattichizzo, "Image-based visual servoing for nonholonomic mobile robots using epipolar geometry," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 87–100, 2007.
- [7] Y. Do, G. Kim, and J. Kim, "Omnidirectional vision system developed for a home service robot," in *Proceedings of the 14th International Conference on Mechatronics and Machine Vision in Practice (M2VIP '07)*, pp. 217–222, Xiamen, China, December 2007.
- [8] Y. Fang, W. E. Dixon, D. M. Dawson, and P. Chawda, "Homography-based visual servo regulation of mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 35, no. 5, pp. 1041–1050, 2005.
- [9] G. Hu, N. Gans, and W. Dixon, "Quaternion-based visual servo control in the presence of camera calibration error," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 5, pp. 489–503, 2010.
- [10] G. Hu, N. Gans, N. Fitz-Coy, and W. Dixon, "Adaptive homography-based visual servo tracking control via a quaternion formulation," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 128–135, 2010.
- [11] G. Hu, W. MacKunis, N. Gans et al., "Homography-based visual servo control with imperfect camera calibration," *IEEE Transactions on Automatic Control*, vol. 54, no. 6, pp. 1318–1324, 2009.
- [12] Y. Fang, X. Liu, and X. Zhang, "Adaptive active visual servoing of nonholonomic mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 486–497, 2012.
- [13] N. R. Gans, G. Hu, K. Nagarajan, and W. E. Dixon, "Keeping multiple moving targets in the field of view of a mobile Camera," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 822–828, 2011.
- [14] H. Wang, Y. Liu, W. Chen, and Z. Wang, "A new approach to dynamic eye-in-hand visual tracking using nonlinear observers," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 2, pp. 387–394, 2011.
- [15] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.
- [16] M. Yang, "Estimating the fundamental matrix using Loo minimization algorithm," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA'08)*, pp. 9241–9246, Chongqing, China, June 2008.
- [17] L. Rui and W. Feng, "An algorithm for estimating fundamental matrix based on removing the exceptional points," in *Proceedings of the International Conference on Computational Intelligence and Natural Computing (CINC '09)*, pp. 88–90, June 2009.
- [18] E. Malis and F. Chaumette, "21/2D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement," *International Journal of Computer Vision*, vol. 37, no. 1, pp. 79–97, 2000.
- [19] G. L. Mariottini and D. Prattichizzo, "EGT for multiple view geometry and visual servoing: robotics and vision with pinhole and panoramic cameras," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 26–39, 2005.