International Journal of Control, Automation and Systems 17(9) (2019) 2297-2309
http://dx.doi.org/10.1007/s12555-019-0006-8
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Adaptive Trajectory Tracking of Wheeled Mobile Robots Based on a Fish-eye Camera

Zhaobing Kang, Wei Zou*, Hongxuan Ma, and Zheng Zhu

Abstract: This paper presents a novel adaptive trajectory tracking control method, which can precisely control wheeled mobile robots only using an uncalibrated fish-eye camera fixed on the ceiling. Different from existing approaches, the inertial device, distorted image correction, and the trajectory expression are not required in the control system. The position and orientation of the mobile robot in the camera coordinate system are estimated by the extended POSIT (Pose from Orthography and Scaling with Iteration) algorithm in real-time. Based on estimation results, the controller considering both tracking errors and parameter estimated errors is designed by linear parameterization, where the camera intrinsic parameters are online updated. The asymptotic convergence of the tracking error and the estimated error to zero is proved by the Barbalat lemma. Circular trajectory and irregular trajectory tracking experiments have been conducted to verify the performance of our controller.

Keywords: Adaptive control, pose estimation, visual trajectory tracking, wheeled mobile robots.

1. INTRODUCTION

Visual servoing plays an important role in the precise motion control of the mobile robot, where the pinhole camera and the inertial device are the most commonly used sensors to get the mobile robot position and orientation. In some areas like autonomous driving, a wide angle of view is necessary, where the pinhole camera cannot meet the requirement while the fish-eye camera is a good choice. However, precise motion control taking the fish-eye image as feedback is a big challenge due to its severe radial distortion. Moreover, most controllers rely on the inertial device to accurately detect position and orientation of the mobile robot. However, the measurement values become unreliable due to accumulative errors of the inertial device. To solve these problems, this paper explores mobile robots visual trajectory tracking without using other sensors except an uncalibrated fish-eye camera which is fixed on the ceiling.

Object pose estimation methods are employed in this article to determine the position and orientation of the mobile robot relative to the camera coordinate system, which can be summarized into two categories. The first one applies filter as the pose estimator, such as Kalman filter (KF) or particle filter (PF). In extended Kalman filter (EKF) methods, incorrect estimation of the dynamic noise covariance will degrade estimated performance. To solve this problem, an adaptive extended Kalman filter (AEKF) [1] technique is proposed, which can be used to update the dynamic noise covariance matrix adaptively. Inspired by AEKF, an iterative AEKF (IAEKF) [2] is proposed to solve the issues of uncertain dynamic noise and poor filter initialization, which integrates noise adaptation mechanism with iterative-measurement linearization. The performance of this category relies on parameter initialization, whose value is difficult to be determined appropriately in most cases. The second category formulates the pose estimation as the Perspective-n-Point (PnP) problem, which can be solved by using at least three 3D points in the world and their 2D projections in the image. Works in [3] and [4] summarize major solutions for this category and analyze their robustness to image noise and numerical stability. To make the algorithms more robust to measurement errors and image noises, more than four image points and iterative methods are employed [5–11], where the pose from orthography and scaling with iteration (POSIT) [12] algorithm is one of the most commonly used pose estimation methods. In this algorithm, perspective projection model is employed to describe the imaging principle of the pinhole camera, the approximations of the rotation matrix and the translation vector are obtained by the scaled orthographic projection model, and the algo-

Manuscript received January 4, 2019; revised March 10, 2019; accepted March 22, 2019. Recommended by Associate Editor Pinhas Ben-Tzvi under the direction of Editor Euntai Kim. This journal was supported by the National Key Research and Development Program of China (Grant No. 2018YFB1306303) and the National Natural Science Foundation of China (Grant No. 61773374).

Zhaobing Kang, Hongxuan Ma, and Zheng Zhu are affiliated to Institute of Automation Chinese Academy of Sciences and University of Chinese Academy of Sciences, China (e-mails: {kangzhaobing2017, mahongxuan2016, zhuzheng2014}@ia.ac.cn). Wei Zou is affiliated to Institute of Automation Chinese Academy of Sciences and TianJin Intelligent Tech. Institute of CASIA Co., Ltd. Address: No. 95, Zhongguancun Road, Haidian, Beijing, China, 100190. (e-mail: wei.zou@ia.ac.cn).
* Corresponding author.

rithm can converge to accurate pose in a few iterations. However, POSIT algorithm cannot be applied to the fish-eye camera directly due to its unsuitability to the perspective projection model.

The methods commonly used to establish the imaging model for the fish-eye camera are summarized by Han *et al*. [13], which can be divided into two ways. Some methods try to build the model according to the projection relationship between the fish-eye camera and the pinhole camera [14–16]. The others try to find a function that can fit the projection principle precisely. A classic method in the second category is proposed by Kannala *et al*. [17], which is suitable for both the fish-eye camera and the pinhole camera.

Using visual input to control robots allows more flexible and robust behaviors than traditional position based control [18]. Different from robot manipulators [19–21], the nonholonomic problem of the mobile robot makes precise motion control difficult. To solve this problem and provide a smooth and stable state feedback law, a control method is proposed by Masutani *et al*. [22]. Besides the nonholonomic problem, visual servo regulation [23–26] and tracking [27–30] are two important issues for the mobile robot that have been widely studied. Visual servo regulation mainly focuses on the strategies that can drive the robot to the target pose while the visual tracking is interested in trajectory tracking [31,32] or object tracking [33]. An adaptive trajectory tracking controller eligible for the wide-angle camera is proposed by Liang *et al*. [30], where the uncalibrated camera is fixed on the ceiling. By linear parameterization, an adaptive law is designed to update the unknown parameters. However, the position and orientation of the mobile robot in the world coordinate system should be well initialized beforehand and accurately estimated on-line during controlling by using the inertial device.

In some occasions, the control system is too complex to achieve precise control using the ordinary adaptive control method. Some excellent works [34, 35] provide a good way to solve this problem. They realize synchronization control of complex dynamical networks subject to nonlinear couplings and uncertainties. Different from [34] and [35], the neural network is introduced in [36] and [37] to precise control of the nonlinear systems with unknown constant or variable control gains.

From the above analysis, most of the control methods only fit for the pinhole camera. For the methods applicable to the fish-eye camera or the wide-angle camera, sensors besides camera are usually required to get essential information of the mobile robot.

This paper presents a novel adaptive trajectory tracking control method for mobile robots, where an uncalibrated fish-eye camera is the only used sensor. The pose of the robot in the camera coordinate system is estimated by an extended POSIT algorithm. Based on linear parameterization, an adaptive controller taking both the tracking errors and the parameter estimated errors into consideration is designed to control the mobile robot and update the intrinsic parameters of the camera. The large field of view of a fish-eye camera is useful in navigation and path planning, where the research results in our paper are helpful to realize precise control of the mobile robot.

The novel contributions of this paper can be summarized as follows: (i) A new adaptive trajectory tracking controller is proposed. The novelty lies in that distorted image correction and trajectory expression are not required in the control system. (ii) The fish-eye camera is the only used sensor in the system, and the POSIT algorithm is extended to fit for the fish-eye camera which is employed to estimate the position and orientation of the mobile robot. Therefore, the accumulative errors lying in the method using the inertial device do not exist. (iii) Both circular trajectory and irregular trajectory tracking results are given to show the effectiveness of our controller.

2. EXTENDED POSIT ALGORITHM

Compared with other PnP methods, accurate camera calibration is not necessary in POSIT algorithm, and the initial value for iteration can always be set to zero, which solves the initialization problem. Since the perspective projection is not suitable for the fish-eye camera, the estimation results of POSIT algorithm are inaccurate using the fish-eye image. To overcome this drawback, this paper presents an extended POSIT algorithm which can obtain precise estimation results for the fish-eye camera.

Fig. 1 is used to clarify the projection models used in this paper. M_0 and M_i $(i = 1,2,3)$ are feature points on an object, whose relative poses are known. Suppose that the focal length f of the camera is known, and the corresponding image points of M_0 and M_i in the pinhole camera and the fish-eye camera are denoted as m_0, m_i and $m_0^{'}$, $m_i^{'}$, respectively. Without loss of generality, M_0 is chosen as the origin of the object coordinate system whose coordinate axes x_b, y_b and z_b are shown in Fig. 1. The camera coordinate system is denoted as $O_c x_c y_c z_c$, and the unit vectors of three axes are denoted as $\vec{i}$, $\vec{j}$ and $\vec{k}$. The rotation matrix ${}^c\boldsymbol{R}_o$ and the translation vector ${}^c\boldsymbol{T}_o$ of the frame $M_o x_b y_b z_b$ relative to the frame $O_c x_c y_c z_c$ can be expressed as

$$ {}^c\boldsymbol{R}_o = \begin{bmatrix} i_{xb} & i_{yb} & i_{zb} \\ j_{xb} & j_{yb} & j_{zb} \\ k_{xb} & k_{yb} & k_{zb} \end{bmatrix}, \quad {}^c\boldsymbol{T}_o = \overrightarrow{O_c M_0}, $$

where i_{xb}, i_{yb} and i_{zb} are the coordinates of $\vec{i}$ in the object coordinate system.

To estimate ${}^c\boldsymbol{R}_o$ and ${}^c\boldsymbol{T}_o$, we draw a plane K that passes through point M_0, parallels to the image plane G and intersects with z_c-axis at point H, and the distance from the point O_c to K is denoted as Z_0. The line of sight for M_i

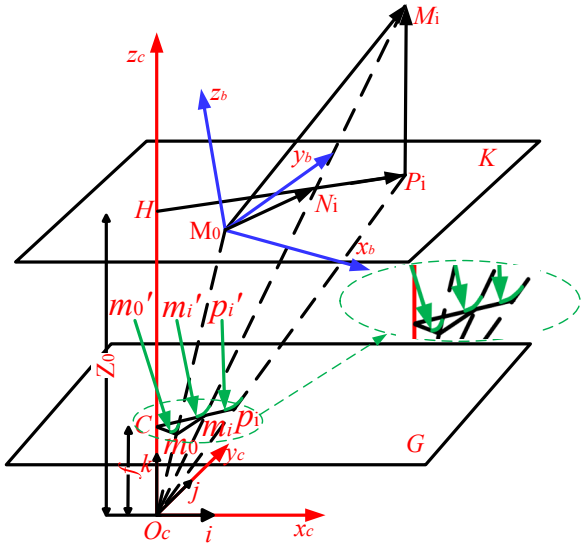


Fig. 1. Projection models of the pinhole camera and the fish-eye camera. Object points M_0, M_i and P_i are respectively projected as m_0, m_i and p_i for the pinhole camera and $m_0^{'}$, $m_i^{'}$ and $p_i^{'}$ for the fish-eye camera.

intersects with the plane K at point N_i. The scaled orthographic projection (SOP) is employed to solve the pose estimation problem, which is an approximation to the perspective projection. In SOP, all the feature points are assumed at the same depth Z_0, and the point M_i is orthographically projected onto the plane K at P_i whose corresponding image point is p_i in the pinhole image and $p_i^{'}$ in the fish-eye image. Therefore, x and y coordinates of P_i and M_i are the same while z coordinate of P_i is Z_0. The coordinates of points M_0, M_i and P_i in the camera coordinate system are denoted as $(X_{M0}, Y_{M0}, Z_0)^T$, $(X_{Mi}, Y_{Mi}, Z_i)^T$ and $(X_{Mi}, Y_{Mi}, Z_0)^T$ respectively. Their corresponding coordinates in the pinhole image plane are denoted as $(x_{M0}, y_{M0})^T$, $(x_{Mi}, y_{Mi})^T$, $(x_{Pi}, y_{Pi})^T$ and those in the fish-eye image plane are denoted as $\left(x_{M0'}, y_{M0'}\right)^T$, $\left(x_{Mi'}, y_{Mi'}\right)^T$, $\left(x_{Pi'}, y_{Pi'}\right)^T$ respectively. According to the perspective projection model, the coordinates of M_0, M_i and P_i in the pinhole image plane can be calculated by

$$x_{M0} = X_{M0} f / Z_0, \quad y_{M0} = Y_{M0} f / Z_0, \tag{1}$$
$$x_{Mi} = X_{Mi} f / Z_i, \quad y_{Mi} = Y_{Mi} f / Z_i, \tag{2}$$
$$x_{Pi} = X_{Mi} f / Z_0, \quad y_{Pi} = Y_{Mi} f / Z_0. \tag{3}$$

Since the perspective projection model is not suitable for the fish-eye camera, the radially symmetric projection model [17] is employed for both the pinhole camera and the fish-eye camera, which is drawn in Fig. 2. Based on this model, the pinhole camera and the fish-eye camera

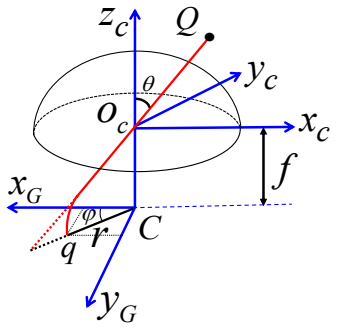


Fig. 2. The radially symmetric projection model. For the fish-eye camera, object point Q is projected as q. For the pinhole camera, the intersection of the red dash line and the black dash line represents the corresponding image point of Q.

can be respectively expressed as

$$r_p = f \tan\theta, \tag{4}$$
$$r_f = f \sin\theta, \tag{5}$$

where θ is the incident angle between the line of sight for the point Q and the principal axis z_c, r_p and r_f is the distance between the principal point C and the image point q. Therefore, the position relationship between q in the fish-eye image and in the pinhole image is

$$\frac{r_f}{r_p} = \cos\theta. \tag{6}$$

According to (1)-(3) and (6), the coordinates of M_0, M_i and P_i in the fish-eye image plane can be expressed as

$$x_{M0'} = x_{M0}\cos\theta_{M0}, \; y_{M0'} = y_{M0}\cos\theta_{M0}, \tag{7}$$
$$x_{Mi'} = x_{Mi}\cos\theta_{Mi}, \; y_{Mi'} = y_{Mi}\cos\theta_{Mi}, \tag{8}$$
$$x_{Pi'} = x_{Pi}\cos\theta_{Pi}, \; y_{Pi'} = y_{Pi}\cos\theta_{Pi}. \tag{9}$$

Assumption 1: In Fig. 1, without loss of generality, suppose $0 \le \theta_{M0} \le \theta_{Pi} < 90°$. For the fish-eye camera, if $\left|\overrightarrow{M_0M_i}\right| \le 0.1Z_0$, $\cos\theta_{M0}$ and $\cos\theta_{Pi}$ can be regarded as the same i.e., $\cos\theta_{M0} \approx \cos\theta_{Pi}$.

Illustration: We explain how the condition $\left|\overrightarrow{M_0M_i}\right| \le 0.1Z_0$ is determined as following. Suppose that if $\cos\theta_{Pi} \ge 0.95\cos\theta_{M0}$ then $\cos\theta_{Pi} \approx \cos\theta_{M0}$. According to Fig. 1, it can be seen that $\theta_{M0} = \angle HOM_0$ and $\theta_{Pi} = \angle HOP_i$, and let $\theta_{Pi} = \theta_{M0} + \Delta\theta$.

In the interval $[0,90°)$, $\cos\theta$ is a monotonically decreasing function. Therefore, for a given $\theta_{M0} \in [0,90°)$, $\Delta\theta$ reaches the maximum value when $\cos\theta_{Pi} = 0.95\cos\theta_{M0}$ and points H, M_0 and P_i are collinear, where the corresponding value of $\left|\overrightarrow{M_0P_i}\right|$ can be calculated by

$$\left|\overrightarrow{M_0P_i}\right| = Z_0\tan(\theta_{M0} + \Delta\theta) - Z_0\tan\theta_{M0}. \tag{10}$$

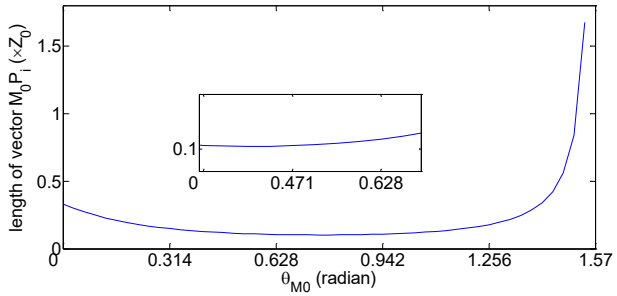


Fig. 3. The maximum value of $\left|\overrightarrow{M_0P_i}\right|$ for a given incident angle θ_{M0}.

Based on the above analysis, the values of $\left|\overrightarrow{M_0P_i}\right|$ are shown in Fig. 3, where $\left|\overrightarrow{M_0P_i}\right| = 0.1Z_0$ is the maximum value that can make $\cos\theta_{Pi} = 0.95\cos\theta_{M0}$ for the fish-eye camera at every point on the interval $[0, 90^\circ)$.

From the above analysis, we can conclude that if $\left|\overrightarrow{M_0P_i}\right| \le 0.1Z_0$ then $\cos\theta_{M0} \approx \cos\theta_{Pi}$. Since $\overrightarrow{P_iM_i}$ is perpendicular to the plane K, therefore, $\left|\overrightarrow{M_0P_i}\right| \le \left|\overrightarrow{M_0M_i}\right|$, and if $\left|\overrightarrow{M_0M_i}\right| \le 0.1Z_0$, we can conclude $\cos\theta_{M0} \approx \cos\theta_{Pi}$. □

Based on this hypothesis, the expression of (9) can be rewritten as

$$\begin{aligned} x_{Pi'} &= X_{Mi}\frac{f}{Z_0}\cos\theta_{Pi} \\ &= \frac{f}{Z_0}\left(X_{Mi}\cos\theta_{Pi} - X_{M0}\cos\theta_{M0}\right) + x_{M0'} \\ &\approx x_{M0'} + \frac{f}{Z_0}\cos\theta_{M0}\left(X_{Mi} - X_{M0}\right) \\ &= x_{M0'} + s\left(X_{Mi} - X_{M0}\right), \end{aligned} \tag{11}$$

$$\begin{aligned} y_{Pi'} &= Y_{Mi}\frac{f}{Z_0}\cos\theta_{Pi} \\ &= \frac{f}{Z_0}\left(Y_{Mi}\cos\theta_{Pi} - Y_{M0}\cos\theta_{M0}\right) + y_{M0'} \\ &\approx y_{M0'} + \frac{f}{Z_0}\cos\theta_{M0}\left(Y_{Mi} - Y_{M0}\right) \\ &= y_{M0'} + s\left(Y_{Mi} - Y_{M0}\right), \end{aligned} \tag{12}$$

where $s = \frac{f}{Z_0}\cos\theta_{M0}$ is the scaling factor of the SOP. From Fig. 1, it can be seen that

$$\overrightarrow{M_0M_i} = \overrightarrow{M_0N_i} + \overrightarrow{N_iP_i} + \overrightarrow{P_iM_i}. \tag{13}$$

In perspective projection, points M_i and N_i are projected at the same image point, therefore, (8) is rewritten as

$$\left(x_{Mi'}, y_{Mi'}\right)^T = f\frac{\left(X_{Ni}, Y_{Ni}\right)^T}{Z_0}\cos\theta_{Mi}, \tag{14}$$

where $\left(X_{Ni}, Y_{Ni}\right)^T$ represents x and y coordinates of N_i in the camera coordinate system. According to (1), 7 and 14, $\overrightarrow{M_0N_i}$ can be expressed as

$$\begin{aligned} \overrightarrow{M_0N_i} &= \left(X_{Ni}, Y_{Ni}, Z_0\right)^T - \left(X_{M0}, Y_{M0}, Z_0\right)^T \\ &= \frac{Z_0}{f}\left(\frac{\left(x_{Mi'}, y_{Mi'}, 0\right)^T}{\cos\theta_{Mi}} - \frac{\left(x_{M0'}, y_{M0'}, 0\right)^T}{\cos\theta_{M0}}\right). \end{aligned} \tag{15}$$

Since $\triangle Cm_iO_c$ is similar to $\triangle P_iN_iM_i$, it is obtained that

$$\frac{\left|\overrightarrow{P_iM_i}\right|}{\left|\overrightarrow{O_cC}\right|} = \frac{\overrightarrow{M_0M_i}\cdot\overrightarrow{k}}{f} = \frac{\left|\overrightarrow{N_iP_i}\right|}{\left|\overrightarrow{Cm_i}\right|}. \tag{16}$$

Substituting (2) and (8) into (16), $\overrightarrow{N_iP_i}$ can be expressed as

$$\overrightarrow{N_iP_i} = \frac{\overrightarrow{M_0M_i}\cdot\overrightarrow{k}}{f}\cdot\frac{\overrightarrow{Cm_{i'}}}{\cos\theta_{Mi}}. \tag{17}$$

Substituting (15) and (17) into (13), $\overrightarrow{M_0M_i}$ equals to

$$\begin{aligned} \overrightarrow{M_0M_i} = \frac{Z_0}{f}&\left(\frac{\left(x_{Mi'}, y_{Mi'}, 0\right)^T}{\cos\theta_{Mi}} - \frac{\left(x_{M0'}, y_{M0'}, 0\right)^T}{\cos\theta_{M0}}\right) \\ &+ \frac{\overrightarrow{M_0M_i}\cdot\overrightarrow{k}}{f}\cdot\frac{\overrightarrow{Cm_{i'}}}{\cos\theta_{Mi}} + \overrightarrow{P_iM_i}. \end{aligned} \tag{18}$$

From Fig. 1, $\overrightarrow{P_iM_i}\cdot\overrightarrow{i} = 0$ and $\overrightarrow{P_iM_i}\cdot\overrightarrow{j} = 0$. (18) is respectively multiplied by $\frac{f}{Z_0}\overrightarrow{i}$ and $\frac{f}{Z_0}\overrightarrow{j}$, it is obtained that

$$\frac{f}{Z_0}\overrightarrow{M_0M_i}\cdot\overrightarrow{i} = \frac{1+\varepsilon_i}{\cos\theta_{Mi}}x_{Mi'} - \frac{1}{\cos\theta_{M0}}x_{M0'}, \tag{19}$$

$$\frac{f}{Z_0}\overrightarrow{M_0M_i}\cdot\overrightarrow{j} = \frac{1+\varepsilon_i}{\cos\theta_{Mi}}y_{Mi'} - \frac{1}{\cos\theta_{M0}}y_{M0'}, \tag{20}$$

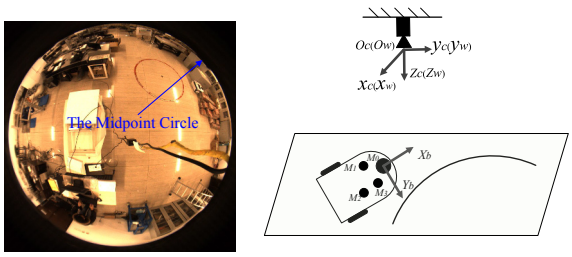
where $\varepsilon_i = \frac{1}{Z_0}\overrightarrow{M_0M_i}\cdot\vec{k}$ and $\vec{k} = \vec{i}\times\vec{j}$. Let $\boldsymbol{I} = \frac{f}{Z_0}\vec{i}$, $\boldsymbol{J} = \frac{f}{Z_0}\vec{j}$, we have

$$\xi_i = \overrightarrow{M_0M_i}\cdot\boldsymbol{I}, \tag{21}$$

$$\eta_i = \overrightarrow{M_0M_i}\cdot\boldsymbol{J}, \tag{22}$$

where ξ_i and η_i represent the right-hand sides of (refeq22) and (20) respectively. For a given point in the fish-eye image, its corresponding incident angle θ can be obtained by (5), therefore, $\cos\theta_{M0}$ and $\cos\theta_{Mi}$ can be calculated based on the image points of M_0 and M_i. Only ε_i, $\boldsymbol{I}$, and $\boldsymbol{J}$ are unknown in (19) and (20), so ${}^c\boldsymbol{R}_o$ and ${}^c\boldsymbol{T}_o$ can be calculated according to the iterative method in the POSIT algorithm.

Remark 1: In the extended POSIT algorithm, the image principal point C is required. For the fish-eye camera, the principal point is the center of the Midpoint Circle shown in Fig. 4(a) and can be determined by circle detection algorithms.



(a) (b)

Fig. 4. The fish-eye image and the trajectory tracking system. (a) The fish-eye image. (b) The trajectory tracking system.

3. THE IMAGE-BASED KINEMATICS MODEL

3.1. System description

The trajectory tracking system is shown in Fig. 4(b), where the world coordinate system $O_w x_w y_w z_w$ is coincide with the camera coordinate system, the meanings of the body coordinate system $M_0 x_b y_b z_b$ and the camera coordinate system $O_c x_c y_c z_c$ are the same as the definition in Fig. 1. The mobile robot is driven by two rear wheels with differential mode and balanced by a front caster wheel which has no influence on its kinematic properties. A fish-eye camera is mounted on the ceiling and parallel to the ground. Four black circle are fixed on the mobile robot, whose centers are noncoplanar and relative poses are known. During tracking the robot is represented by the circle center M_0 which is on the symmetry axis of the mobile robot and has a distance d to the rear axis in the x_b direction. The coordinates of the point M_0 and the rear axis midpoint in the world coordinate system are respectively denoted as $x_0^w(t)$, $y_0^w(t)$, z_0^w and $x(t)$, $y(t)$, z. The orientation of the mobile robot is denoted as $\theta_o(t)$, which is the angle between the x_b-axis and the x_w-axis. With regard to the system configuration, the tracking system satisfies the following assumptions.

Assumption 2: The ground where the robot moves on is flat and can be regarded as a plane.

Assumption 3: The mobile robot has no slippage while moving on the ground.

Assumption 4: These four black circles on the robot can be online detected in the image during tracking.

3.2. Image-based kinematics model

According to the nonholonomic constraints, we can get the equation

$$\dot{x}(t)\sin\theta_o(t) - \dot{y}(t)\cos\theta_o(t) = 0, \tag{23}$$

where $\dot{x}(t)$, $\dot{y}(t)$, and $\theta_o(t)$ can be respectively expressed as

$$\begin{aligned} \dot{x}(t) &= v\cos\theta_o(t), \\ \dot{y}(t) &= v\sin\theta_o(t), \\ \dot{\theta}_o(t) &= \omega, \end{aligned} \tag{24}$$

and v and ω denote the linear velocity and the angular velocity of the mobile robot respectively. The rotation matrix ${}^c\boldsymbol{R}_o$ and the translation vector ${}^c\boldsymbol{T}_o$ of the frame $M_0 x_b y_b z_b$ relative to the frame $O_c x_c y_c z_c$ can be expressed as

$${}^c\boldsymbol{R}_o = \begin{bmatrix} \cos\theta_o(t) & -\sin\theta_o(t) & 0 \\ \sin\theta_o(t) & \cos\theta_o(t) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad {}^c\boldsymbol{T}_o = \begin{bmatrix} x_0^c(t) \\ y_0^c(t) \\ z_0^c \end{bmatrix},$$

where $x_0^c(t)$, $y_0^c(t)$ and z_0^c denote the coordinates of the point M_0 in the camera coordinate system. ${}^c\boldsymbol{R}_o$ and ${}^c\boldsymbol{T}_o$ can be estimated by the extended POSIT algorithm by observing the circle centers M_0, M_1, M_2 and M_3. Since the world coordinate system coincides with the camera coordinate system, the translation vector ${}^w\boldsymbol{T}_o$ of the frame $M_0 x_b y_b z_b$ relative to the frame $O_w x_w y_w z_w$ can be expressed as

$${}^w\boldsymbol{T}_o = \begin{bmatrix} x_0^w(t) \\ y_0^w(t) \\ z_0^w \end{bmatrix} = \begin{bmatrix} x_0^c(t) \\ y_0^c(t) \\ z_0^c \end{bmatrix}.$$

According to the position relationship between M_0 and the midpoint of the rear axis, the coordinates of M_0 in the world coordinate system can be calculated by

$$\begin{bmatrix} x_0^w(t) \\ y_0^w(t) \\ z_0^w \end{bmatrix} = \begin{bmatrix} x(t) + d\cos\theta_o(t) \\ y(t) + d\sin\theta_o(t) \\ z - h \end{bmatrix}, \tag{25}$$

where h is the height from the rear axis to the top of the mobile robot. According to (24) and (25), the velocity of the point M_0 in the frame $O_w x_w y_w z_w$ can be expressed as

$$\begin{bmatrix} \dot{x}_0^w(t) \\ \dot{y}_0^w(t) \end{bmatrix} = \begin{bmatrix} \cos\theta_o(t) & -d\sin\theta_o(t) \\ \sin\theta_o(t) & d\cos\theta_o(t) \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{26}$$

According to Fig. 2 and (7), M_0 is projected into the fish-eye image with coordinates

$$\begin{aligned} \begin{bmatrix} u_x \\ u_y \end{bmatrix} &= \frac{\cos\theta_{M0}(t)}{z_0^c} \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \begin{bmatrix} x_0^c(t) \\ y_0^c(t) \end{bmatrix} + \begin{bmatrix} u_{x0} \\ u_{y0} \end{bmatrix} \\ &= \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \sin\theta_{M0}(t) \begin{bmatrix} \cos\varphi(t) \\ \sin\varphi(t) \end{bmatrix} + \begin{bmatrix} u_{x0} \\ u_{y0} \end{bmatrix}, \end{aligned} \tag{27}$$

where u_x and u_y denote the row and column coordinates in the image, and k_x, k_y, u_{x0} and u_{y0} denote intrinsic parameters of the fish-eye camera, and $\varphi(t)$ is the angle between $\overrightarrow{Cq}$ and the x_G-axis. The parameters $\theta_{M0}(t)$ and $\varphi(t)$ can be respectively calculated as follows:

$$\begin{aligned} \sin\theta_{M0}(t) &= \frac{\sqrt{x_0^c(t)^2 + y_0^c(t)^2}}{\sqrt{x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}}}, \\ \cos\varphi(t) &= \frac{x_0^c(t)}{\sqrt{x_0^c(t)^2 + y_0^c(t)^2}}. \end{aligned} \tag{28}$$

Differentiating (27) with respect to time and substituting (26) and (28) into the results, then it can be obtained

$$\begin{bmatrix} \dot{u}_x \\ \dot{u}_y \end{bmatrix} = \frac{\boldsymbol{M}_0 \boldsymbol{M}_1}{\left(x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}\right)^{3/2}} \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix}, \tag{29}$$

where

$$\boldsymbol{M}_0 = \begin{bmatrix} k_x\left(y_0^c(t)^2 + z_0^{c2}\right) & -k_x x_0^c(t) y_0^c(t) \\ -k_y x_0^c(t) y_0^c(t) & k_y\left(x_0^c(t)^2 + z_0^{c2}\right) \end{bmatrix},$$
$$\boldsymbol{M}_1 = \begin{bmatrix} \cos\theta_o(t) & -d\sin\theta_o(t) \\ \sin\theta_o(t) & d\cos\theta_o(t) \end{bmatrix}.$$

Equation (29) can be rewritten as

$$\begin{bmatrix} \dot{u}_x \\ \dot{u}_y \end{bmatrix} = \boldsymbol{J}_P(t) \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix}, \tag{30}$$

where $\boldsymbol{J}_P(t)$ is the image Jacobian matrix.

4. ADAPTIVE TRACKING CONTROLLER

The principal point coordinates mentioned in Remark 1 are an approximation to the real values, which is good enough for the extended POSIT algorithm. However, using the approximation values directly in the controller is not rigorous. Therefore, the adaptive control method is employed to design the controller, and the camera intrinsic parameters are updated by the adaptive law.

4.1. Adaptive controller design

At time t, the desired trajectory position and the real trajectory position in the image are denoted as $\boldsymbol{y}_{pd}(t)$ and $\boldsymbol{y}_p(t)$ respectively, the image position tracking errors can be expressed as

$$\Delta\boldsymbol{y}_p(t) = \boldsymbol{y}_p(t) - \boldsymbol{y}_{pd}(t). \tag{31}$$

The image velocity errors can be derived

$$\Delta\dot{\boldsymbol{y}}_p(t) = \dot{\boldsymbol{y}}_p(t) - \dot{\boldsymbol{y}}_{pd}(t). \tag{32}$$

The aim of the controller design is to find a control law that can make the image position tracking errors converge to zero. Since the intrinsic parameters cannot be acquired for the uncalibrated fish-eye camera, the linear parameterization method is employed to design the controller.

Linear parameterization method can be described as: for a vector $\boldsymbol{q}$, the matrix $\boldsymbol{Y}$ can be linearly parameterized as

$$\boldsymbol{Yq} = \boldsymbol{La}, \tag{33}$$

where $\boldsymbol{Y} \in \Re^{n\times n}$, $\boldsymbol{q} \in \Re^{n\times 1}$, $\boldsymbol{L} \in \Re^{n\times s}$, and $\boldsymbol{a} \in \Re^{s\times 1}$. $\boldsymbol{L}$ is the regressor matrix and composed by known parameters, and $\boldsymbol{a}$ is the vector that contains all unknown parameters.

Based on this method, (30) can be parameterized in a linear form

$$\boldsymbol{J}_P(t) \begin{bmatrix} v \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{N}({}^c\boldsymbol{T}_o, v, \boldsymbol{\omega}, \theta_o(t))\boldsymbol{\rho}, \tag{34}$$

where $\boldsymbol{N}({}^c\boldsymbol{T}_o, v, \boldsymbol{\omega}, \theta_o(t))$ is the regressor matrix whose parameters are all known, and $\boldsymbol{\rho} = \left[k_x, k_y, u_{x0}, u_{y0}\right]^T$ is the vector that contains all the intrinsic parameters. The image position errors between the real pixel coordinates and the estimated ones of M_0 at time t_j are expressed as

$$\boldsymbol{e}(t_j, t) = \begin{bmatrix} u_x \\ u_y \end{bmatrix} - \begin{bmatrix} \hat{u}_x \\ \hat{u}_y \end{bmatrix}, \tag{35}$$

where $\hat{u}_x$ and $\hat{u}_y$ denote the estimated pixel coordinates that can be expressed as

$$\begin{bmatrix} \hat{u}_x \\ \hat{u}_y \end{bmatrix} = \begin{bmatrix} \hat{k}_x & 0 \\ 0 & \hat{k}_y \end{bmatrix} \sin\theta_{M0}(t) \begin{bmatrix} \cos\varphi(t) \\ \sin\varphi(t) \end{bmatrix} + \begin{bmatrix} \hat{u}_{x0} \\ \hat{u}_{y0} \end{bmatrix},$$

and $\hat{\boldsymbol{\rho}} = \left[\hat{k}_x, \hat{k}_y, \hat{u}_{x0}, \hat{u}_{y0}\right]^T$ is the vector that contains all the estimated intrinsic parameters of the camera and updated by a reasonably designed adaptive law. Substituting (27) and (28) into (35) and parameterizing (35) in a linear form, it can be obtained

$$\boldsymbol{e}(t_j, t) = \boldsymbol{W}_e({}^c\boldsymbol{T}_o)\Delta\boldsymbol{\rho}, \tag{36}$$

where $\boldsymbol{W}_e({}^c\boldsymbol{T}_o)$ denotes the regressor matrix and $\Delta\boldsymbol{\rho} = \hat{\boldsymbol{\rho}} - \boldsymbol{\rho}$. ${}^c\boldsymbol{T}_o$ can be estimated by the extended POSIT algorithm, therefore, the element values in $\boldsymbol{W}_e({}^c\boldsymbol{T}_o)$ are constant at time t_j.

Theorem 1: For the visual trajectory tracking system shown in Fig. 4(b), the inverse of the image Jacobian matrix in (30) always exist.

Proof: According to (29) and (30), the determinant of $\boldsymbol{J}_P(t)$ can be derived as

$$\begin{aligned} |\boldsymbol{J}_P(t)| &= \left| \frac{\boldsymbol{M}_0\boldsymbol{M}_1}{\left(x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}\right)^{3/2}} \right| \\ &= \frac{dk_x k_y z_0^{c2}}{\left(x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}\right)^2}. \end{aligned} \tag{37}$$

For the commonly used cameras, k_x and k_y are far greater than zero. The distance d between the point M_0 and the rear axis can be set to a nonzero value, and the height z_0^c from the camera to the point M_0 is also far greater than zero according to the system description. Therefore, the determinant of $\boldsymbol{J}_P(t)$ is nonzero, and the inverse of the image Jacobian matrix $\boldsymbol{J}_P(t)$ always exists. □

The inverse of $\boldsymbol{J}_P(t)$ is

$$\boldsymbol{J}_P^{-1}(t) = \left(x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}\right)^{3/2} \boldsymbol{M}_1^{-1}\boldsymbol{M}_0^{-1}, \tag{38}$$

where

$$\boldsymbol{M}_0^{-1} = \frac{\boldsymbol{M}_0^{'}}{z_0^{c2}\left(x_0^c(t)^2 + y_0^c(t)^2 + z_0^{c2}\right)},$$

$$\boldsymbol{M}_1^{-1} = \begin{bmatrix} \cos\theta_o(t) & -\sin\theta_o(t) \\ \frac{\sin\theta_o(t)}{d} & \frac{\cos\theta_o(t)}{d} \end{bmatrix},$$

$$\boldsymbol{M}_0^{'} = \begin{bmatrix} \frac{x_0^c(t)^2+z_0^{c2}}{k_x} & \frac{x_0^c(t)^2 y_0^c(t)^2}{k_y} \\ \frac{x_0^c(t)^2 y_0^c(t)^2}{k_x} & \frac{y_0^c(t)^2+z_0^{c2}}{k_y} \end{bmatrix}.$$

Based on the above equations, the tracking controller can be designed as below according to (31)

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \hat{\boldsymbol{D}}_P \dot{\boldsymbol{y}}_{pd}(t) - \hat{\boldsymbol{D}}_P \boldsymbol{K}_P \Delta \boldsymbol{y}_p, \tag{39}$$

where $\hat{\boldsymbol{D}}_P = \hat{\boldsymbol{J}}_P^{-1}$ denotes the estimation of $\boldsymbol{J}_P^{-1}$ and $\boldsymbol{K}_P = k_p \boldsymbol{I}_{2\times 2}$ with k_p being a positive constant. Though the inverse of the image Jacobian matrix always exists, singularity of matrix $\hat{\boldsymbol{J}}_P^{-1}$ may happen when the estimated values of k_x and k_y are equal to or close to zero. In this situation, the values are replaced by valid estimated results whose update time is nearest to now.

The camera intrinsic parameters are online updated by the following adaptive law:

$$\begin{aligned} \dot{\hat{\boldsymbol{\rho}}} =& \Gamma^{-1}\{\boldsymbol{N}^T\left({}^c\boldsymbol{T}_o, v, \omega, \theta_o(t)\right)\boldsymbol{K}_P \Delta \boldsymbol{y}_p \\ & - \boldsymbol{W}_e^T\left({}^c\boldsymbol{T}_o\right)\boldsymbol{K}_e \boldsymbol{e}\left(t_j, t\right)\}, \end{aligned} \tag{40}$$

where $\Gamma^{-1} \in \Re^{4\times 4}$ and $\boldsymbol{K}_e \in \Re^{2\times 2}$ denote symmetric positive definite matrices with corresponding dimensions.

Remark 2: The first term in the tracking controller (39) is feedforward term that is necessary to efficiently solve the trajectory tracking problem, and the second term is the feedback term. The first term in adaptive law (40) is used to compensate for the error in the system caused by the estimated parameters, and the second term is used to minimize the estimated projection error in (35). Larger values of k_p and k_e can make the mobile robot converge to the desired trajectory faster, but the controller outputs may out of the dynamic limitation of the mobile robot. Based on our experimental results, $0.1 \le k_p \le 0.5$ and $0.1 \le k_e \le 0.3$ are reasonable intervals.

4.2. Stability Proof

The Barbalat lemma [38] is first introduced.

Barbalat lemma: If $V(t)$ satisfies following conditions:

1) $V(t)$ is lower bounded.
2) $\dot{V}(t)$ is negative semi-definite.
3) $\dot{V}(t)$ is uniformly continuous in time (satisfied if $\ddot{V}(t)$ is finite).

Then $\dot{V}(t) \to 0$ as $t \to \infty$.

Theorem 2: The controller with the law in (39) and accompanied with the update law in (40) can ensure the tracking error $\Delta \boldsymbol{y}_p(t)$ and the estimated error $\boldsymbol{e}(t_j, t)$ asymptotically converge to zero in the sense

$$\lim_{t\to\infty} \Delta \boldsymbol{y}_p(t) = \boldsymbol{0},\ \lim_{t\to\infty} \boldsymbol{e}(t_j, t) = \boldsymbol{0}.$$

Proof: Introduce the Lyapunov function

$$V(t) = \frac{1}{2}\Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \Delta \boldsymbol{y}_p(t) + \frac{1}{2}\Delta \boldsymbol{\rho}^T \Gamma \Delta \boldsymbol{\rho}. \tag{41}$$

Differentiating (41) with respect to time, it is obtained that

$$\dot{V}(t) = \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \Delta \dot{\boldsymbol{y}}_p(t) + \Delta \boldsymbol{\rho}^T \Gamma \dot{\hat{\boldsymbol{\rho}}}. \tag{42}$$

Substituting (30) and (32) into (42) results in

$$\begin{aligned} \dot{V}(t) =& \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \left(\dot{\boldsymbol{y}}_p(t) - \dot{\boldsymbol{y}}_{pd}(t)\right) + \Delta \boldsymbol{\rho}^T \Gamma \dot{\hat{\boldsymbol{\rho}}} \\ =& \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \left(\boldsymbol{J}_P - \hat{\boldsymbol{J}}_P\right)\begin{bmatrix} v \\ \omega \end{bmatrix} \\ & + \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \hat{\boldsymbol{J}}_P \begin{bmatrix} v \\ \omega \end{bmatrix} + \Delta \boldsymbol{\rho}^T \Gamma \dot{\hat{\boldsymbol{\rho}}} \\ & - \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \dot{\boldsymbol{y}}_{pd}(t). \end{aligned} \tag{43}$$

Substituting (39) and (40) into (43) yields

$$\begin{aligned} \dot{V}(t) =& \Delta \boldsymbol{\rho}^T \boldsymbol{N}^T \boldsymbol{K}_P \Delta \boldsymbol{y}_p(t) \\ & + \Delta \boldsymbol{\rho}^T \Gamma \dot{\hat{\boldsymbol{\rho}}} + \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \dot{\boldsymbol{y}}_{pd}(t) - k_p^2 \Delta \boldsymbol{y}_p(t)^T \Delta \boldsymbol{y}_p(t) \\ & - \Delta \boldsymbol{y}_p(t)^T \boldsymbol{K}_P \dot{\boldsymbol{y}}_{pd}(t) \\ =& -k_p^2 \Delta \boldsymbol{y}_p(t)^T \Delta \boldsymbol{y}_p(t) - \boldsymbol{e}^T \boldsymbol{K}_e \boldsymbol{e} \le 0. \end{aligned} \tag{44}$$

The Lyapunov function (41) has upper bound. Therefore, $\Delta \boldsymbol{y}_p$ and $\Delta \boldsymbol{\rho}$ have bounds. To confirm the third condition of the Barbalat lemma, differentiating (44) with respect to time, we have

$$\ddot{V}(t) = -k_p^2 \Delta \boldsymbol{y}_p(t)^T \Delta \dot{\boldsymbol{y}}_p(t) - \boldsymbol{e}^T \boldsymbol{K}_e \dot{\boldsymbol{e}}. \tag{45}$$

Substituting (32) and (36) into (45) yields

$$\begin{aligned} \ddot{V}(t) =& -2k_p^2 \Delta \boldsymbol{y}_p(t)^T \left(\dot{\boldsymbol{y}}_p(t) - \dot{\boldsymbol{y}}_{pd}(t)\right) \\ & - 2\boldsymbol{e}^T \boldsymbol{K}_e \boldsymbol{W}_e \dot{\hat{\boldsymbol{\rho}}}. \end{aligned} \tag{46}$$

As $\dot{\boldsymbol{y}}_p(t)$ and $\dot{\boldsymbol{y}}_{pd}(t)$ have bounds, based on the above description, other parameters in (46) also have bounds. Therefore, $\ddot{V}(t)$ has bound. According to the Barbalat lemma, $\lim_{t\to\infty} \dot{V}(t) \to 0$, therefore, $\lim_{t\to\infty} \Delta \boldsymbol{y}_p(t) \to \boldsymbol{0}$ and $\lim_{t\to\infty} \boldsymbol{e}(t_j, t) \to \boldsymbol{0}$. The system is globally asymptotically stable. □

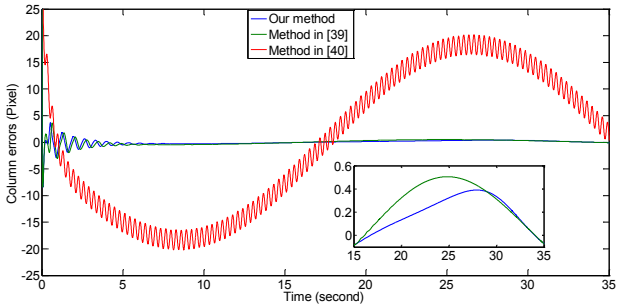
5. SIMULATION AND EXPERIMENTS

5.1. Simulation results

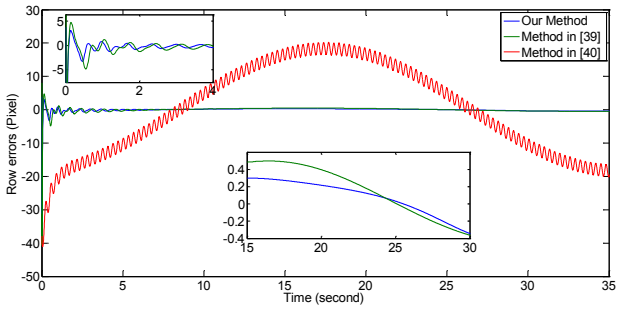
In simulation, we compare our method with the controllers proposed in [39] and [40]. The procedure of the

controller in [39] is similar to the one in [30], while the feedforward term in the control law and the estimated projection error term in the adaptive law are ignored in [39]. The method in [40] proposes a visual trajectory tracking controller, which is composed of a kinematic controller and a sliding mode controller. Since our controller is designed by the kinematic model, the kinematic controller in [40] is employed to compare with our method. The pose of the mobile robot can be accurately calculated in simulation, therefore, the extended POSIT algorithm is abandoned in this part without affecting the comparative results. The coordinate systems are set according to the description of Fig. 4(b). Since the methods in [39] and [40] are both designed for the pinhole camera, the pinhole camera model is used in their simulations, while the fish-eye camera model is used in our method. The desired trajectory is a circle with a radius of 700 mm, and the coordinates of the circle center in the camera coordinate system are 1000 mm, 1000 mm and 2000 mm respectively. To make the comparative results reliable, the parameters are adjusted to make all the controllers achieve their best performance. For our method and the method in [39], parameter values are set to $\boldsymbol{K}_p = 0.3\boldsymbol{I}_{2\times2}$, $\boldsymbol{K}_e = 0.1\boldsymbol{I}_{2\times2}$ and $\Gamma^{-1} = 10^{-7}\boldsymbol{I}_{4\times4}$. For the method in [40], k_1=4 and k_2=90 (k_3 used in [40] is denoted as k_2 in our paper) are the best choice. The tracking results are shown in Fig. 5(a) and Fig. 5(b). Fig. 5(a) shows the tracking error in the column direction, and Fig. 5(b) shows the tracking error in the row direction. From these two subfigures, it can be seen that the tracking error of the method in [40] is the largest, and it cannot achieve stable control performance only using the kinematic controller. From Fig. 5(a), we can see that the tracking error of the controller in [39] is nearly the same as our method at the beginning. In this period, the mobile robot does not attain the desired trajectory, therefore, tracking errors of our method and the method in [39] are large. During smooth motion period, the tracking error of method in [39] is larger than our method. According to Fig. 5(b), it can be seen that the tracking error of the method in [39] is larger than our method both at the beginning period and at the smooth motion period.

From the above analysis, we can conclude that the controller proposed in this paper can achieve good trajectory tracking performance.



(a)



(b)

Fig. 5. Simulation results. (a) The tracking errors in the column direction. (b) The tracking errors in the row direction.

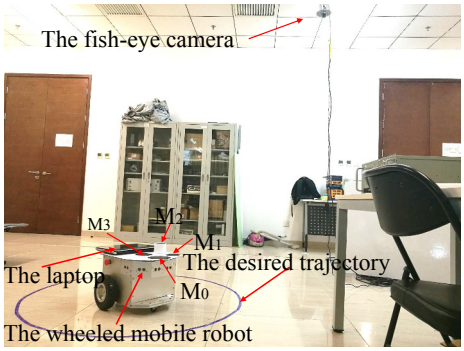


Fig. 6. The experimental system setup.

5.2. Experimental system setup

To verify the effectiveness of our controller, the control method proposed by Liang *et al*. [30] is employed to compare with our method, which can achieve good performance in both circular trajectory and irregular trajectory tracking using a wide-angle camera. In this method, the robot position and orientation must be measured by an inertial device, therefore, control accuracy decreases in a long-term tracking task due to enlarged accumulative errors. In the compared method, the camera is not required to parallel to the ground, which will be studied in our future work. The experimental system setup is shown in Fig. 6. In the system, the fish-eye camera is mounted on the ceiling, whose optical axis is perpendicular to the ground. Four black circles with different radii are fixed on the robot, whose centers are noncoplanar. The coordinate systems are established according to Fig. 4(b). The centers of the four black circles lie on $(0,0,0)$, $(0,159,0)$ mm, $(-135,30,0)$ mm and $(-135,172,-60)$ mm in the object coordinate system. The distance from the rear axis to the origin of the object coordinate system is 300 mm and to the top of the mobile robot is 450 mm. Without loss of generality, for these two methods, the initial values

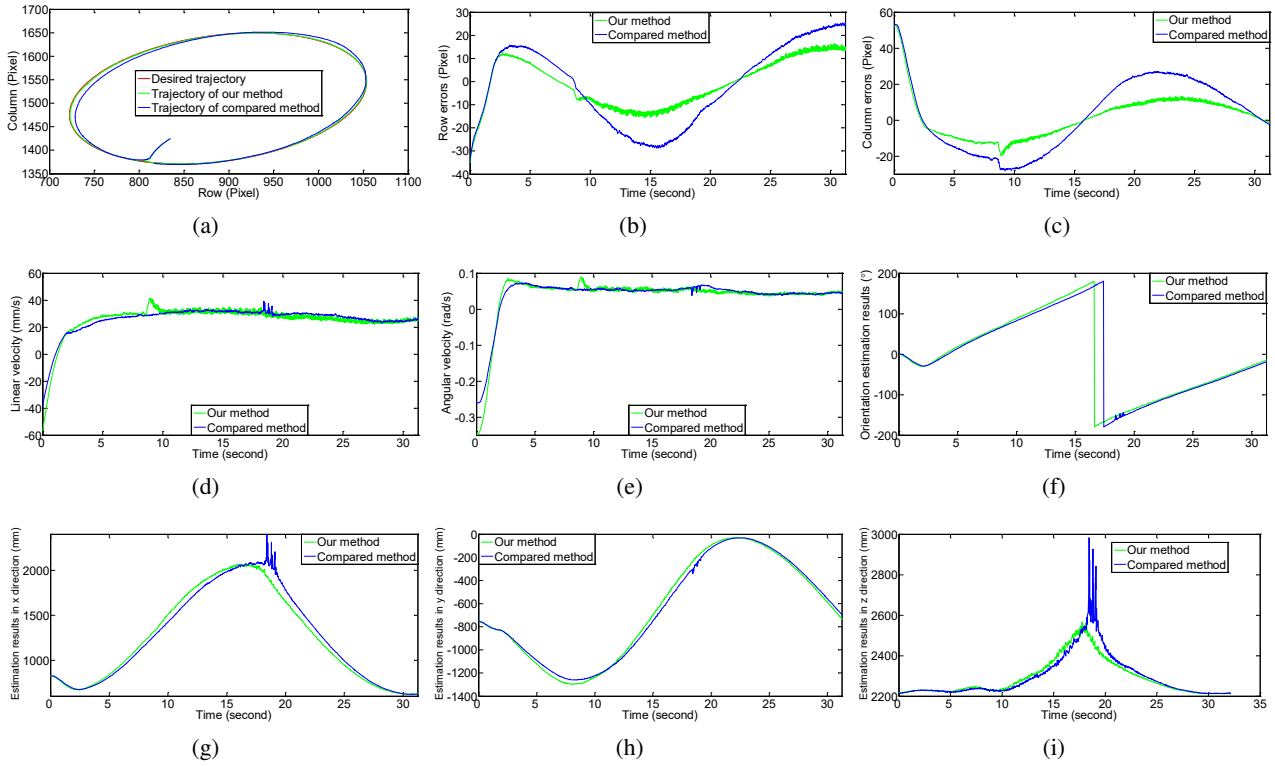


Fig. 7. Circle trajectory experimental results: (a) The desired trajectory and the real trajectories in the image. (b) The tracking errors in the row direction. (c) The tracking errors in the column direction. (d) The linear velocities of the mobile robot. (e) The angular velocities of the mobile robot. (f) The orientation value of the mobile robot estimated by the extended POSIT algorithm. (g) The position values of the mobile robot in the x direction estimated by the extended POSIT algorithm. (h) The position values of the mobile robot in the y direction estimated by the extended POSIT algorithm. (i) The position values of the mobile robot in the z direction estimated by the extended POSIT algorithm.

of the camera intrinsic parameters are set to $k_x = k_y = 350$ and $u_{x0} = u_{y0} = 500$, the controller gains are given by $\boldsymbol{K}_p = 0.2\boldsymbol{I}_{2\times2}$, $\boldsymbol{K}_e = 0.1\boldsymbol{I}_{2\times2}$, $\Gamma^{-1} = 10^{-7}\boldsymbol{I}_{4\times4}$, and the position and orientation of the mobile robot are acquired by the extended POSIT algorithm.

5.3. Circle trajectory experimental results

The radius of the blue circle trajectory is set to 700 mm. To make the results more persuasive, the camera is located far away from the circle center. The desired trajectory in the image can be predetermined by the image processing algorithm, which is discretized using the step of 1 pixel along the tangent direction of the trajectory. For these two methods, the position and orientation of the mobile robot are chosen randomly and are the same at start, and their values are estimated by the extended POSIT algorithm during tracking. The compared results are shown in Fig. 7.

Fig. 7(a)-(c) shows the image trajectories, the tracking errors in the row direction and the tracking errors in the column direction. From Fig. 7(b), it can be seen that the row error of our controller changes in the interval $[-15, 15]$ pixel while the error range of the compared controller is $[-30, 25]$ pixel, and their mean values are 9.3 pixels and 15.4 pixels respectively. In the column direction, the error intervals are $[-20, 10]$ pixel and $[-30, 30]$ pixel respectively, and their mean error values are 9.4 pixels and 17.1 pixels respectively. From these three subfigures, we can conclude that the control method proposed in this paper is effective for the precise control of the mobile robot using a fish-eye camera. The position and orientation of the mobile robot in the world coordinate system are estimated by the extended POSIT algorithm, and the estimation results are shown in Fig. 7(f)-(i). From these subfigures, we can see that the estimation results of our method are smooth, which means that the extended POSIT algorithm is robust to the image noise and the measurement error. Note that there exist severe noises in the estimation results of compared method, which emerges at the vertices position of the major axis of the image trajectory. This reason can be explained according to Fig. 7(d)-(e). From this two subfigures, to adjust the bear angle of the mobile

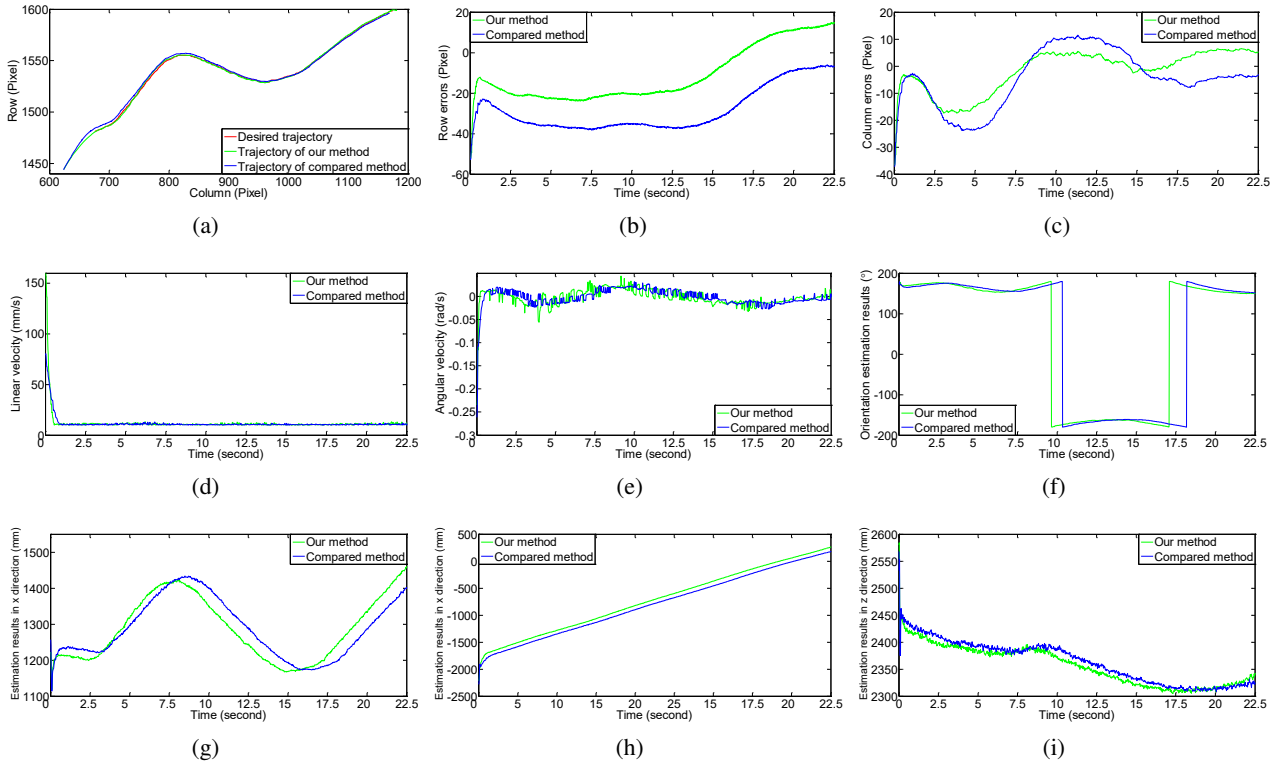


Fig. 8. Irregular trajectory tracking experimental results: (a) The desired trajectory and the real trajectories in the image. (b) The tracking errors in the row direction. (c) The tracking errors in the column direction. (d) The linear velocities of the mobile robot. (e) The angular velocities of the mobile robot. (f) The orientation value of the mobile robot estimated by the extended POSIT algorithm. (g) The position values of the mobile robot in the x direction estimated by the extended POSIT algorithm. (h) The position values of the mobile robot in the y direction estimated by the extended POSIT algorithm. (i) The position values of the mobile robot in the z direction estimated by the extended POSIT algorithm.

robot, v and ω of the compared controller become large, which decreases the accuracy of the object detection due to image blur. Therefore, the estimation results are also affected. Note that at 10th seconds our controller outputs also have a suddenly change, but their values become normal after slow decrease. Therefore, the object can be accurately detected, and estimation results are smooth. From Fig. 7(d)-(e) and Fig. 7(i), it can be seen that, after 10th seconds, the linear velocity v and the angular velocity ω of our controller keep stable for different estimation results in the z direction. This result confirms the robustness of our controller to estimation errors.

From the above analysis, we can conclude that our method can precisely control the mobile robot using a fisheye camera and is robust to estimation errors.

5.4. Irregular trajectory experimental results

The irregular trajectory tracking experiments are also conducted to show the performance of our method, and the desired trajectory in the image also can be predetermined based on the method mentioned in Section 5.2. The experimental results are shown in Fig. 8. Fig. 8(a)-(c) shows the irregular trajectories in the image, the tracking errors in the row direction and the tracking errors in the column direction. Note that the row error intervals of our method and the compared method are $[-20, 15]$ pixel and $[-40, -10]$ pixel respectively, and mean values are 15.9 pixels and 26.2 pixels. In the column direction, the error intervals of our method and the compared method are $[-20, 10]$ pixel and $[-15, 5]$ pixel, and mean values are 5.9 pixels and 8.5 pixels respectively. From the above analysis, it can be concluded that our controller achieves a better control performance than the compared method. Fig. 8(f)-(i) shows the estimation results of the extended POSIT algorithm. From these subfigures, it can be seen that the curve shape and value of two methods are nearly the same, which proves the repeatability of the extended POSIT algorithm. The good tracking performance of our controller demonstrates that the estimation results of the extended POSIT algorithm are reliable.

This experiment shows that our method has a good control performance in tracking the trajectory whose expres-

sion is unknown. This result indicates that our controller can also be used for the areas where the path is planned dynamically.

6. CONCLUSION

In this paper, a new adaptive visual trajectory tracking controller for the wheeled mobile robot is designed, where a fish-eye camera fixed on the ceiling is the only used sensor. In this method, distorted image correction, the inertial device and the expression of the trajectory are not needed. The position and orientation of the mobile robot in the world coordinate system are calculated by the extended POSIT algorithm using four noncoplanar points. According to estimation results and linear parameterization, the adaptive controller is given considering both the tracking errors and the parameter estimated errors, and the camera intrinsic parameters are updated by the controller. Moreover, the stability of the control system is proved. The circular trajectory experiment shows that the proposed controller has a good control performance and is robust to the estimation error of the extended POSIT algorithm. The irregular trajectory experiment shows that the controller can also precisely control the mobile robot tracking a trajectory whose expression is unknown. In the future work, we will focus on improving the robustness of the extended POSIT algorithm to larger image detection error, and how to precisely control the mobile robot using a casual fixed camera, which may not parallel to the ground will also be studied.

REFERENCES

[1] M. Ficocelli and F. J. Sharifi, "Adaptive filtering for pose estimation in visual servoing," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 19-24, 2001.

[2] F. J. 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, September 2010.

[3] R. M. Haralick, H. Joo, and C. Lee, "Pose estimation from corresponding point data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 6, pp. 1426-1446, Nov/Dec 1989.

[4] R. M. Haralick, C. Lee, K. Ottenberg, and M. Nölle, "Review and analysis of solutions of the three point perspective pose estimation problem," *International Journal of Computer Vision*, vol. 13, no. 3, pp. 331-356, December 1994.

[5] E. Mair, K. H. Strobl, M. Suppa, and D. Burschka, "Efficient camera-based pose estimation for real-time applications," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 2696-2703, 2009.

[6] O. Tahri, H. Araujo, Y. Mezouar, and F. Chaumette, "Efficient decoupled pose estimation from a set of points," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 1608-1613, 2014.

[7] Y. Kuang and K. Aström, "Pose estimation with unknown focal length using points, directions and lines," *Proc. of IEEE International Conference on Computer Vision*, pp. 529-536, 2013.

[8] T. Zhou and X. Jing, "Surface-based detection and 6-DoF pose estimation of 3-D objects in cluttered scenes," *IEEE Transactions on Robotics*. vol. 32, no. 6, pp. 1347-1361, August 2016.

[9] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2011-2018, 2017.

[10] S. Ramalingam, S. Bouaziz, and P. Sturm, "Pose estimation using both points and lines for geo-localization," *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4716-4723, 2011.

[11] J. Bimbo, L. D. Seneviratne, K. Althoefer, and H. Liu, "Combining touch and vision for the estimation of an object's pose during manipulation," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4022-4026, 2013.

[12] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15 no. 1-2, pp. 123-141, June 1995.

[13] S. B. Han, J. H. Kim, and H. Myung, "Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system," *IEEE Transactions on Mechatronicss*, vol. 18, no. 6, pp. 1745-1756, December 2013.

[14] C. Brauer-Burchardt and K. Voss, "A new algorithm to correct fish-eye and strong wide-angle-lens-distortion from single images," *Proc. of International Conference on Image Processing*, pp. 225-228, 2001.

[15] F. Devernay and O. Faugeras, "Straight lines have to be straight," *Machine Vision and Applications*, vol. 13, no. 1, pp. 14-24, August 2001.

[16] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern Recognition Letters*, vol. 16, no. 4, pp. 433-441, April 1995.

[17] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no.8, pp. 1335-1340, June 2006.

[18] M. Jägersand, "Visual servoing using trust region methods and estimation of the full coupled visual-motor jacobian," *Lasted Applications of Control and Robotics*, pp. 105-108, October 1996.

[19] M. Keshmiri, W. F. Xie, and A. Ghasemi, "Visual servoing using an optimized trajectory planning technique for a 4 DOFs robotic manipulator,"*International Journal of Control, Automation, and Systems*, vol. 15, no. 3, pp. 1362-1373, June 2017.

[20] C. S. Kim, E. J. Mo, S. M. Han, M. S. Jie, and K. W. Lee, "Robust visual servo control of robot manipulators with uncertain dynamics and camera parameters," *International Journal of Control, Automation, and Systems*, vol. 8, no. 2, pp. 308-313, April 2010.

[21] S. Jung, "Improvement of tracking control of a sliding mode controller for robot manipulators by a neural network,"*International Journal of Control, Automation, and Systems*, vol. 16, no. 2, pp. 937-943, April 2018.

[22] Y. Masutani, M. Mikawa, N. Maru, and F. Miyazaki, "Visual servoing for non-holonomic mobile robots," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 1133-1140, 1994.

[23] B. Li, Y. Fang, and X. Zhang, "Visual servo regulation of wheeled mobile robots with an uncalibrated onboard camera," *IEEE Transactions on Mechatronics*, vol. 21, no. 5, pp. 2330-2342, November 2016.

[24] X. Zhang, Y. Fang, B. Li, and J. Wang, "Visual servoing of nonholonomic mobile robots with uncalibrated camera-to-robot parameters," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 1, pp. 390-400, August 2017.

[25] B. Li, X. Zhang, Y. Fang, and W. Shi, "Visual servo regulation of wheeled mobile robots with simultaneous depth identification," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 460-469, June 2018.

[26] X. Zhang, Y. Fang, and N. Sun, "Visual servoing of mobile robots for posture stabilization: from theory to experiments," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 1, pp. 1-15, August 2013.

[27] M. Kobbilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 557-562, 2016.

[28] R. Kelly, E. Bugarin, and V. Sanchez, "Image-based visual control of nonholonomic mobile robots via velocity fields: Case of partially calibrated inclined camera," *Proc. of IEEE Conference on Decision and Control*, pp. 3071-3076, 2016.

[29] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Behal, "Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 31, no. 3, pp. 341-352, June 2001.

[30] X. Liang, H. Wang, W. Chen, and T. Liu, "Adaptive image-based trajectory tracking control of wheeled mobile robots with an uncalibrated fixed camera," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 6, pp. 2266-2282, March 2015.

[31] M. Gianni, F. Ferri, M. Menna, and F. Pirri, "Adaptive robust three-dimensional trajectory tracking for actively articulated tracked vehicles," *Journal of Field Robotics*, vol. 33, no. 7, pp. 901-930, April 2016.

[32] S. R. Bista, P. R. Giordano, and F. Chaumette, "Combining line segments and points for appearance-based indoor navigation by image based visual servoing," *Proc. of IEEE International Conference on Intelligent Robots and Systems*, pp. 2960-2967, 2017.

[33] K. Zhang, J. Chen, and B. Jia, "Asymptotic moving object tracking with trajectory tracking extension: a homography-based approach," *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4664-4685, April 2017.

[34] X. J. Li and G. H. Y, "FLS-based adaptive synchronization control of complex dynamical networks with nonlinear couplings and state-dependent uncertainties," *IEEE Transactions on Cybernetics*, vol. 46, no. 1, pp. 171-180, February 2015.

[35] X. J. Li and G. H. Y, "Adaptive fault-tolerant synchronization control of a class of complex dynamical networks with general input distribution matrices and actuator faults," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 559-569, December 2015.

[36] D. D. Zheng, Y. P. Pan, K. Guo, and H. Y. Yu, "Identification and control of nonlinear systems using neural networks: A singularity-free approach," *IEEE Transactions on Neural Networks and Learning Systems*, 2019. DOI: 10.1109/TNNLS.2018.2886135

[37] K. Guo, Y. P. Pan, and H. Y. Yu, "Composite learning robot control with friction compensation: A neural network-based approach", *IEEE Transactions on Industrial Electronics*, 2019. DOI: 10.1109/TIE.2018.2886763

[38] V. M. Popov, *Hyperstability of Control Systems*, Springer-Verlag, New York, 1973.

[39] X. Liang, H. Wang, and W. Chen, "Adaptive image-based visual servoing of wheeled mobile robots with fixed camera configuration," *Proc. of IEEE International Conference on Robotics and Automation*, pp. 6199-6204, September 2014.

[40] F. Yang and C. Wang, "Adaptive tracking control for uncertain dynamic nonholonomic mobile robots based on visual servoing," *Journal of Control Theory and Application*, vol. 10, no. 1, pp. 56-63, February 2012.

Zhaobing Kang received his B.Eng. degree in mechanical engineering and automation from Dezhou University, China, in 2008, the M.Eng. degree in mechanical and electronic engineering from Harbin Institute of Technology, China. Currently, he is a Ph.D. degree candidate in Institute of Automation at Chinese Academy of Science, China. He is also with University of Chinese Academy of Science, China. His research interests include visual servoing and robot location and navigation.

Wei Zou received his B.Eng. degree in control science and engineering from Baotou University of Iron and Steel Technology, China, in 1997, an M.Eng. degree in control science and engineering from Shandong University of Technology, China, in 2000, and a Ph.D. degree in control science and engineering from IACAS, China, in 2003. Currently, he is a professor at the Research Center of Precision Sensing and Control, IACAS. His research interests include intelligent robotics, visual servoing, and robot localization and navigation.

Hongxuan Ma received his B.Sc. degree from Central South University, China, in 2016. He is currently a Ph.D. degree candidate in Institute of Automation at Chinese Academy of Sciences, China. He is also with University of Chinese Academy of Sciences, Beijing. His research interests include computer vision and robotics.

Zheng Zhu received his B.Sc. degree from Zhen Zhou University,China, in 2014. He is currently a Ph.D. degree candidate in Institute of Automation at Chinese Academy of Sciences, China. He is also with University of Chinese Academy of Sciences, Beijing. His research interests include computer vision, deep learning and robotics.