Adaptive Image-based Visual Servoing with Temporary Loss of the Visual Signal

Haobin Shi, Gang Sun, Yuanpeng Wang, and Kao-Shing Hwang, *Senior Member*, *IEEE*

***Abstract*—Image-based visual servoing (IBVS) can reach a desired position for a relatively stationary target using continuous visual feedback. Proper feature extraction and appropriate servoing control laws are essential to performance for IBVS. IBVS control can be interrupted or interfered abruptly if no features are extracted when the observed object is occluded. To address the problem of missing feature points in current images during a visual navigation task, a homography method that uses a priori visual information is proposed to predict all of the missing feature points and to ensure the execution of IBVS. The mixture parameter for the image Jacobian matrix can also affect the control of IBVS. The settings for the mixture parameter are heuristic so there is no a systematic approach for most IBVS applications. An adaptive control approach is proposed to determine the mixture parameter. The proposed method uses a reinforcement learning (RL) method to adaptively adjust the mixture parameter during the robot movement, which allows more efficient control than a constant parameter. A logarithmic interval state space partition for RL is used to ensure efficient learning. The integrated visual servoing control system is validated by several experiments that involve wheeled mobile robots reaching a target with a desired configuration. The results for simulation and experiment demonstrate that the proposed method has a faster convergence rate than other methods.**

***Index Terms*—Image-based visual servoing (IBVS), Mobile robot, Reinforcement learning (RL), *Q*-learning, Occlusion prediction.**

I. Introduction

STUDIES of autonomous wheeled mobile robots (WMRs) are important within the robotics field [1][2]. Rapid developments in computer vision (CV), image processing and control theory mean that the techniques of visual servoing have been introduced into robotic systems to improve the flexibility and intelligence of robots. Visual servoing is used for vertical take-off and landing (VTOL) of a quad-rotor helicopter [3] and for an unmanned aerial vehicle (UAV) that track power lines [4]. By the definition of system errors, the current visual servoing methods can be divided into position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [5][6]. Position-based visual servoing control uses image features to estimate the current pose of the target such that the accuracy of the control system depends on the estimation of the target's pose, which is significantly affected by the target geometric model and the camera calibration. PBVS methods usually perform well in a Cartesian space, but are sensitive to camera calibration errors, kinematic model errors, noises, and so on [7][8]. In contrast, error signals for IBVS are defined in the image space. The geometrical information for image features that is acquired by visual sensors is used directly to feedback and control the motion of a robot [9]. IBVS control systems are less sensitive to camera and robot calibration errors and initial conditions, because calibration errors only affect the rate of convergence for reaching the desired position [10]. Therefore, IBVS is more suitable for WMRs because some random errors are inevitable, such as slipping or sliding in operations.

Proper feature extraction and appropriate servoing control laws are essential to the performance of IBVS [11]. Some unexpected factors also need to be taken into considerations to avoid the failure of servoing, such as the problems of occasional occlusion. Especially, in an unstructured environment, IBVS can be interrupted or interfered abruptly, if features are not extracted successively [12][13]. This problem is addressed by eliminating unseen features directly from the pool of candidate features that are extracted from images, in order to alleviate computational disturbances [14], but occluded features sometimes cannot be simply eliminated when there are too few visible features to provide sufficient information. The method in [13] calculates a homography relationship between two images, one of which suffers temporary loss of visual signal, to predict the positions of features in an occluded image. However, this method must estimate prior transformations for translations and rotations, which are acquired by a variety of sensors, such as an odometer.

Methods that address the problem of occlusion for WMRs must have a rapid response time, in order to prevent hazardous situations where robots do not receive servoing commands. A variety of efficient image processing methods have been applied to feature extraction and matching, such as RGB-based methods [15][16], scale invariant feature transform (SIFT)-based methods [17][18], and speeded up robust features

Manuscript received February 19, 2018; revised May 03, 2018 and June 28, 2018; accepted August 04, 2018. This work is supported in part by the Shaanxi Province Key Research and Development Program of China under Grant 2018GY-187, in part by the Aeronautical Science Foundation of China under Grant 2016ZC53022, and in part by the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University under Grant ZZ2018026. Paper no. TII-18-1110.R1. (Corresponding author: Kao-Shing Hwang.)

Haobin Shi is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China. (e-mail: shihaobin@nwpu.edu.cn).

Gang Sun and Yuanpeng Wang are with the School of Software and Microelectronics, Northwestern Polytechnical University, Xi'an 710129, China. (e-mail: sung@mail.nwpu.edu.cn; wangyp@mail.nwpu.edu.cn).

Kao-Shing Hwang is with the Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan. He is also an adjunct professor of the Department of Healthcare Administration and Medical Informatic, Kaohsiung Medical University, Taiwan (e-mail: hwang@ccu.edu.tw).

(SURF)-based methods [19][20]. However, there are some drawbacks for RGB-based methods in terms of changes in the image size and orientation. The Oriented FAST and Rotated BRIEF (ORB) algorithm combine and improve the algorithms for FAST and BRIEF to combine the merits of both methods and performs better in terms of rotation invariance and computational complexity [21][22][23]. The ORB operator is two orders of magnitude faster than SURF and SIFT so the ORB allows real-time prediction because it has less computational complexity. The shortcomings of ORB in terms of scale invariance are effectively avoided because of the short time interval between two frame images during the WMRs movement.

An appropriate servoing control law is the key to the performance of IBVS. Different forms of image Jacobian matrix s derive different control laws and produce different control effects. Three universal forms of the image Jacobian matrix have been proposed [5]. These are expressed as: $\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s(t)}}$, $\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s}^*}$, and $\hat{\mathbf{J}}=(\mathbf{J}_{\mathbf{s(t)}}+\mathbf{J}_{\mathbf{s}^*})/2$. A more general expression for the image Jacobian matrix uses a parameter β and is written as $\hat{\mathbf{J}}=\beta\hat{\mathbf{J}}_{\mathbf{s(t)}}+(1-\beta)\hat{\mathbf{J}}_{\mathbf{s}^*}$, where the range of β is from 0 to 1. This general expression of $\hat{\mathbf{J}}$ is adaptable to more general cases with a time varying β. The value of β is related to the displacement of the camera and must take account of the displacement [24]. When the camera approaches closely to the target, a small value of β is assigned to induce a smooth convergence, and vice versa. However, not studies have proposed a systematic method to adaptively select a series of β values during servoing control to ensure good performance.

This paper proposes a homography method with a priori visual information to predict all missing target features in occluded images, in order to ensure continuous IBVS. The proposed homography method only needs coordinates of image feature points instead of sensors and utilizes un-occluded feature points to calculate more accurate homography relationships. The value of β is usually artificially set to a constant, which results in unsatisfactory control. As we know, there is no systematic way to obtain adaptive β. Therefore, a reinforcement learning (RL) method is used to adaptively select a series of values for β during servoing control, in order to improve IBVS control performance. Taking into account the characteristic of the WMRs movement, the state space partition for RL is designed to achieve rapid learning convergence.

The remainder of the paper is organized as follows. Section II presents the background for the proposed methods, such as the kinematics model of a WRM, the image Jacobian matrix, and a general IBVS framework. Section III presents a method to predict missing target feature points that compensates for a break in the sequential feature chain. Section IV presents an adaptive selection method that uses RL to select a series of β values. Simulations and experiments are conducted to demonstrate the performance of the proposed methods in Section V. Finally, conclusions are drawn in the last section.

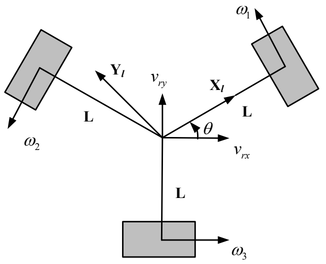


Fig. 1. The kinematics model of WMR with three omni-directional wheels.

II. Background

A. Kinematics model for WRMs

A complete IBVS for a mobile robot with three omni-directional wheels is implemented. The top of the WMRs is equipped with a Kinect with a RGB-D camera that measure depth information. The transformation between the camera's speed and robot's speed is:

$$\mathbf{v}_\mathbf{r}=\begin{bmatrix} v_{rx} & v_{ry} & 0 & 0 & 0 & \omega_r \end{bmatrix}^T=\begin{bmatrix} \mathbf{R}_\mathbf{c}^\mathbf{r} & \mathbf{s}\left(\mathbf{d}_\mathbf{c}^\mathbf{r}\right)\mathbf{R}_\mathbf{c}^\mathbf{r} \\ \mathbf{0}_{3\times3} & \mathbf{R}_\mathbf{c}^\mathbf{r} \end{bmatrix}\mathbf{v}_\mathbf{c}=\mathbf{G}\mathbf{v}_\mathbf{c} \quad (1)$$

where $\mathbf{v}_\mathbf{r}$ is the robot velocity vector with respect to the robot coordinate system, v_{rx}, v_{ry}, ω_r are linear velocities in the x-direction, the y-direction, and the yaw angular velocity for the WMR, respectively, $\mathbf{v}_\mathbf{c}$ is the camera velocity vector with respect to the camera coordinate system, $\mathbf{R}_\mathbf{c}^\mathbf{r}$ is the rotation matrix for the two coordinate systems, $\mathbf{d}_\mathbf{c}^\mathbf{r}=\begin{bmatrix} d_x & d_y & d_z \end{bmatrix}^T$ is the initial position vector of the camera coordinate system with respect to the robot coordinate system, and $\mathbf{s}\left(\mathbf{d}_\mathbf{c}^\mathbf{r}\right)$ is the skew symmetric matrix for the vector $\mathbf{d}_\mathbf{c}^\mathbf{r}$.

The kinematics model for a WMRs is shown in Fig. 1, where $\mathbf{X}_l$-$\mathbf{Y}_l$ represents a local frame that is attached to the robot itself. The center of this local frame coincides with the center of gravity of the robot. The three omnidirectional wheels are located at an angle relative to the local frame of $\alpha_i(i=1,2,3)$. If the local axis $\mathbf{X}_l$ is used as a starting point and the degrees in the clockwise direction are measured as positive, $\alpha_1=0^\circ$, $\alpha_2=120^\circ$, and $\alpha_3=240^\circ$.

The relationship between the velocities of the three wheels and the linear velocity and the angular velocity of the WMR is given in (2):

$$\boldsymbol{\omega}=\mathbf{P}\times\mathbf{v}_\mathbf{r} \quad (2)$$

where $\boldsymbol{\omega}=\begin{bmatrix} \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T$ are the velocities of the three wheels, $\mathbf{v}_\mathbf{r}=\begin{bmatrix} v_{rx} & v_{ry} & \omega_r \end{bmatrix}^T$ is the linear velocity and the angular velocity of the WMRs, and $\mathbf{P}$ is the transformation matrix between $\boldsymbol{\omega}$ and $\mathbf{v}_\mathbf{r}$, given by:

$$\mathbf{P}=\frac{1}{R}\begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\sin(\theta+\alpha_2) & \cos(\theta+\alpha_2) & L \\ -\sin(\theta+\alpha_3) & \cos(\theta+\alpha_3) & L \end{bmatrix} \quad (3)$$

where R is the radius of the wheels, L is the distance from the wheel centers to the robot center, and θ is the yaw angle of the robot in the global coordinate system.

The transformation between the camera speed and the speed of the three wheels is shown in (4).

$$\boldsymbol{\omega}=\mathbf{PGv_c} \tag{4}$$

B. Image-based Visual Servoing

If M feature points are obtained by the feature extraction algorithm, the M feature points are denoted as $\mathbf{f_i}\left(i=1,2,\ldots\right.$. Their coordinates in the camera's coordinate system are (X_i, Y_i, Z_i). The projection point $\mathbf{f_i}$ of $\mathbf{F_i}$ on the image plane is defined as (x_i, y_i). The relationship between the feature point $\mathbf{F_i}$ and its projection point $\mathbf{f_i}$ is:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \frac{f}{Z_i}\begin{bmatrix} a_x & 0 \\ 0 & a_y \end{bmatrix}\begin{bmatrix} X_i \\ Y_i \end{bmatrix} \tag{5}$$

where f is the camera's focal length. a_x and a_y are the respective numbers of pixels per unit distance (mm) in the horizontal and vertical directions.

The feature error $\mathbf{e}$ measures the difference between the current and the desired positions of image features and is expressed as:

$$\mathbf{e}=\left(\mathbf{f}_1-\mathbf{f}_1^*, \mathbf{f}_2-\mathbf{f}_2^*, \ldots, \mathbf{f}_M-\mathbf{f}_M^*\right)^T \in \Re^{2M\times 1} \tag{6}$$

where $\mathbf{f}_i^* = \left(x_i^*, y_i^*\right)^T$ is the desired image coordinate of the i^{th} feature point.

There is a projection relationship between the 2-D point in the image plane and the 3-D point in the camera coordinate system. Kinematics shows that the image feature coordinate changes depending on the motion of the robot. The time derivative for the feature error $\mathbf{e}$ is expressed as

$$d\mathbf{e}/dt = \hat{\mathbf{J}}\begin{bmatrix} v_{tx} & v_{ty} & \omega_r \end{bmatrix}^T \tag{7}$$

Since the movement of WMRs involves a 3-DOF model, $\hat{\mathbf{J}}$ is a $2M*3$ matrix. This consists of the 1st, 2nd and 6th columns of the image Jacobian matrix for all M feature points. $\hat{\mathbf{J}}_\mathbf{i}$ is the image Jacobian matrix for the i^{th} point.

$$\hat{\mathbf{J}}_\mathbf{i} = \begin{bmatrix} \dfrac{-f\alpha_x}{Z_i} & 0 & \dfrac{\alpha_x y_i}{\alpha_y} \\ 0 & \dfrac{-f\alpha_y}{Z_i} & \dfrac{-\alpha_y x_i}{\alpha_x} \end{bmatrix} \tag{8}$$

To ensure an exponential decoupled decrease in the feature error $\mathbf{e}$, i.e. $d\mathbf{e}/dt = -\lambda\mathbf{e}$. The relationship between the feature error and the mobile robot's velocity is given by:

$$\begin{bmatrix} v_{tx} & v_{ty} & \omega_r \end{bmatrix}^T = -\lambda\hat{\mathbf{J}}^+\mathbf{e} \tag{9}$$

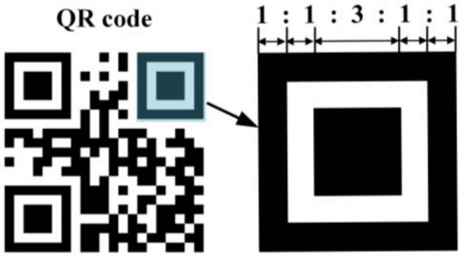
where λ is the servoing gain. The speeds of the three wheel are given by the (4) and (9).

$$\boldsymbol{\omega} = \mathbf{PG}\begin{bmatrix} v_{tx} & v_{ty} & \omega_r \end{bmatrix}^T = -\lambda\mathbf{PG}\hat{\mathbf{J}}^+\mathbf{e} \tag{10}$$

where $\hat{\mathbf{J}}^+ \in \Re^{2M\times 3}$ is the approximation of the pseudo-inverse of the image Jacobian matrix $\mathbf{J}$.

There are usually three forms of Jacobian matrix:

$$\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s(t)}} \tag{11}$$



(a) (b)

Fig. 2. (a) QR code and (b) Finder pattern.

$$\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s}^*} \tag{12}$$

$$\hat{\mathbf{J}}=(\mathbf{J}_{\mathbf{s(t)}}+\mathbf{J}_{\mathbf{s}^*})/2 \tag{13}$$

$\hat{\mathbf{J}}$ is expressed as general IBVS (GIBVS) by introducing a specific parameter β.

$$\hat{\mathbf{J}} = \beta\hat{\mathbf{J}}_{\mathbf{s(t)}} + \left(1-\beta\right)\hat{\mathbf{J}}_{\mathbf{s}^*} \tag{14}$$

where the value of β ranges from 0 to 1.

III. Prediction of Occluded Features

The WMR that is used for this study as an experimental platform that is a 3-DOF model, such that at least two target feature points must be extracted from the current image. Occasional target occlusion is an inevitable problem due to uncertainty or noise that is induced by an unstructured working environment. This study predicts occluded features in image sequences during servoing. A homography method with a priori visual information is proposed to predict all of the missing target feature points when target detection fails.

A. Extraction of feature points

Quick response (QR) codes have been widely used in various industries. Therefore, in this study, a QR code is used as the target template and four vertex coordinates of the QR code are used as the feature points in IBVS. The QR code and its finder pattern are shown in Fig. 2. Extracting the region of interest (ROI) can greatly reduce the complexity of the calculation for the vertex coordinates because the target does not occupy a large area in the image. The finder pattern area has an obvious contrast to the background. The black and white pixels along any line through the center of the finder pattern has a special ratio of 1:1:3:1:1. The approximate location of the QR code and ROI can be easily extracted from the images using three finder patterns. The vertex coordinates for the QR code are obtained through information of edge lines and used to calculate the image Jacobian matrix. During IBVS control, the nearest image that is successfully detected is saved as a key frame image $\mathbf{Frame^{key}}$ when the target is occluded. $\mathbf{Frame^{current}}$ is the current frame image with missing feature points. $\mathbf{Frame^{key}}$ is used for subsequent target prediction.

B. Feature prediction in occluded images

The proposed homography method requires $\mathbf{Frame^{current}}$ and $\mathbf{Frame^{key}}$ for feature prediction. The shortcomings of ORB in terms of scale invariance are effectively avoided because of the short time interval between two frame images. If the target is not occluded, the image feature extraction algorithm extracts all M pairs of feature points. When the target is occluded, only $N(N<M)$ pairs of feature points $\mathbf{F^{target}} = \{(\mathbf{f_j^{target}}, \mathbf{\bar{f}_j^{target}}) \mid 1 \le j \le N\}$ match between the two frames, so some feature points in $\mathbf{Frame^{current}}$ are lost. For this study, $M=4$, when $M<4$, the target is occluded. $\mathbf{f_j^{target}}$ and $\mathbf{\bar{f}_j^{target}}$ are respectively the target feature points in $\mathbf{Frame^{key}}$ and $\mathbf{Frame^{current}}$. The ORB operator is used to detect and match two frame images to obtain L pairs of features $\mathbf{F^{orb}} = \{(\mathbf{p_k}, \mathbf{\bar{p}_k}) \mid 1 \le k \le L\}$. $\mathbf{p_k}(x_k, y_k, 1)$ and $\mathbf{\bar{p}_k}(\bar{x}_k, \bar{y}_k, 1)$ are respectively the pixel coordinates of the two frame images for the same $\mathbf{P_k}$ in the world coordinate system. The mapping between the two images is expressed by a homography matrix $\mathbf{H}$.

$$\mathbf{\bar{p}_k} = \mathbf{Hp_k} \tag{15}$$

The homogenous homography matrix $\mathbf{H} \in \Re^{3\times 3}$ has 8 unsolved values because one of parameters h_{22} can be equal to 1. The homography matrix $\mathbf{H}$ is calculated by solving (16), which requires four or more non-collinear feature points.

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -\bar{x}_1 x_1 & -\bar{x}_1 y_1 & -\bar{x}_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -\bar{y}_1 x_1 & -\bar{y}_1 y_1 & -\bar{y}_1 \\ & & & & \vdots & & & & \\ x_j & y_j & 1 & 0 & 0 & 0 & -\bar{x}_j x_j & -\bar{x}_j y_j & -\bar{y}_j \\ 0 & 0 & 0 & x_j & y_j & 1 & -\bar{y}_j x_j & -\bar{y}_j y_j & -\bar{y}_j \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{12} \\ h_{20} \\ h_{21} \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{16}$$

However, the number of matching features is much more than four and there are some mismatch. A random sample and consensus (RANSAC) algorithm is used to address the mismatch problem.

The RANSAC algorithm has three steps for each iteration. The first randomly selects four pairs of feature points from $\mathbf{F^{orb}}$. The second uses the selected feature points to calculate a homography matrix $\mathbf{H}$ as the model. The third divides the remaining $L-4$ pairs of feature points into inliers and outliers using (17). Finally, the model with the largest number of inliers is used as the prediction model.

$$\left\| \begin{bmatrix} \bar{x}_j \\ \bar{y}_j \\ 1 \end{bmatrix} - \mathbf{H} \begin{bmatrix} x_j \\ y_j \\ 1 \end{bmatrix} \right\| \le \zeta \quad (j < L-4) \tag{17}$$

where x_j and $\bar{x}_j$ ($j < L-4$) are a pair of feature points obtained by the ORB operator. H is calculated using the selected four pairs of feature points. ζ is a threshold that distinguishes between inliers and outliers.

To reduce the prediction error, the method of selecting feature points in RANSAC is improved. Some feature points use prior visual information instead of random selection. Since $\mathbf{F^{target}}$ are more reliable than $\mathbf{F^{orb}}$, in each round of iteration, $\mathbf{F^{target}}$ are used as a prior information, and $4-N$ pair features are selected randomly from $\mathbf{F^{orb}}$. The model with the most inliers is the final homography matrix $\mathbf{H}$ until the maximum number of iterations is reached or the calculation model converges.

Algorithm 1 The proposed homography method

```
Definition
 $\mathbf{F^h}$ : = the 4 pairs of points to calculate homography matrix
 $T_{\max}$ : = maximal iteration number
get_homography() : = calculate homography matrix by $\mathbf{F^h}$
get_inliers() : = calculate number of inliers by $\mathbf{Model}$ and $\mathbf{F^{orb}}$
Initialization
    $\mathbf{F^{orb}} \leftarrow$ calculate $\mathbf{Frame^{key}}$ and $\mathbf{Frame^{current}}$ by ORB operator;
    $\mathbf{F^{target}} \leftarrow$ calculate $\mathbf{Frame^{key}}$ and $\mathbf{Frame^{current}}$ by target detection
    $t \leftarrow 1$;
Repeat $t{+}{+}$
    $\mathbf{F^h} \leftarrow \mathbf{F^{target}}$; //a priori knowledge//
    $\max\_inliers \leftarrow 0$; //record the optimal condition//
    $k \leftarrow 1$;
    Repeat $k{+}{+}$
        $(\mathbf{p}, \mathbf{\bar{p}}) \leftarrow$ point pairs are selected randomly from $\mathbf{F^{orb}}$;
        add $(\mathbf{p}, \mathbf{\bar{p}})$ into $\mathbf{F^h}$;
    Until $k > 4-N$
    $\mathbf{Model} \leftarrow$ get_homography( $\mathbf{F^h}$ );
    $inliers^t \leftarrow$ get_inliers( $\mathbf{Model}, \mathbf{F^{orb}}$ );
    If $\max\_inliers < inliers^t$ then //update//
        $\mathbf{H} \leftarrow \mathbf{Model}$;
        $\max\_inliers \leftarrow inliers^t$
    End if
    If model has converged then
        break;
    End if
Until $t > T_{\max}$
For $1 \le i \le M$ ; $\mathbf{f_i} \leftarrow \mathbf{Hf_i^{key}}$
```

The target features $\mathbf{f_i}(x_i, y_i, 1)$ in $\mathbf{Frame^{current}}$ are predicted using the matrix $\mathbf{H}$ and $\mathbf{f_i^{key}}$. All missing target feature points for $\mathbf{Frame^{current}}$ are calculated to ensure that there is the IBVS control. The proposed homography method with a prior visual information is presented in **Algorithm 1**.

IV. Adaptive Image-based Visual Servoing by RL

The value of β is related to the displacement of the camera. A small value of β is assigned to ensure a smooth convergence when the camera approaches closely to the target, and vice versa. The selection for the value β is critical. A method for adaptively adjusting the value β using Q-learning is proposed. The convergence rate and the stability of the RL algorithm is affected by the state space partition and the action selection strategy, so a logarithmic interval state space partitioning method and an action selection strategy that uses simulated annealing (SA) and statistical probability are used.

A. Updating Iteration

Q-learning is a model-free and value-based reinforcement learning algorithm in a Markov process. The formula for updating Q values is:

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q_t(s_t, a_t) + \alpha(r + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1})) \tag{18}$$

where (s_t, a_t) is a state-action pair at the t^{th} learning cycle, γ is the discount rate, α is the learning rate and r is the real-time reward.

The mobile robot's sojourn time in each state is a general random variable. This situation can be regarded as a semi-Markov decision process. Traditional *Q*-learning does not give a good solution for a semi-Markov problem [25][26]. In a semi-Markov process, the current state s_c transits into the next state s_n after *T* learning cycles. The *Q*-value is not updated and a series of real-time rewards are obtained as $\{r_{t+i} \mid i = 0,1,\ldots$. The iterative formula for action-value function is defined as:

$$V_{t+i}(s_c, a_{t+i}) = r_{t+i} + \gamma V_{t+i+1}(s_c, a_{t+i+1}), \quad 1 \le i \le T-2 \tag{19}$$

$$V_{t+T-1}(s_c, a_{t+T-1}) = r_{t+T\text{-}1} + \gamma \max_a Q_t(s_n, a_{t+T}) \tag{20}$$

$$Q_{t+T}(s_c, a) = (1-\alpha)Q_t(s_c, a_t) + \alpha(r_t + \sum\nolimits_{k=1}^{T-1} V_{t+k}) / T \tag{21}$$

where $r_t + \sum\nolimits_{k=1}^{T-1} V_{t+k}$ is the cumulative reward for the *T* learning cycles.

B. State Selection and Partition

The selection and partition of the state is particularly important for the reinforcement learning model. Image features cannot be used directly as a state space because the computational complexity for RL increases exponentially as the number of state dimensions increases when the number of feature points is large, which is, the curse of dimensionality. The β value is also related to the displacement of the camera. Therefore, the rotation and translation of the camera between the current image and the desired image are taken as state space, which is a more general method.

The rotation and translation of the camera are obtained using singular value decomposition (SVD). $\mathbf{F}_i$ and $\bar{\mathbf{F}}_i$ $(i = 1, 2, \ldots, M)$ are respectively the coordinates of the target feature point in the camera coordinate system for the robot at the current position and the desired position. The corresponding image coordinates are f_i and $\bar{\mathbf{f}}_i$. The relationship between $\mathbf{F}_i$ and $\bar{\mathbf{F}}_i$ is:

$$\mathbf{F}_i = (\mathbf{R} + \frac{\mathbf{T}}{\bar{d}} \bar{\mathbf{n}}^T)\bar{\mathbf{F}}_i \tag{22}$$

where $\mathbf{R}$ and $\mathbf{T}$ are respectively the rotation matrix and the translation vector, $\bar{\mathbf{n}}$ and $\bar{d}$ are respectively the normal vector of the target plane and its distance with respect to the camera coordinate system. The relationship between the points in the image coordinate system and the points in the camera coordinate system is:

$$\mathbf{f}_i = \frac{1}{z_i}\mathbf{K}\mathbf{F}_i,\ \bar{\mathbf{f}}_i = \frac{1}{\bar{z}_i}\mathbf{K}\bar{\mathbf{F}}_i \tag{23}$$

where $\mathbf{K}$ is the intrinsic camera matrix that is obtained by camera calibration. By (14) and (22), (21) is expressed as:

$$\mathbf{F}_i = (\mathbf{K}^{-1}\mathbf{H}\mathbf{K})\bar{\mathbf{F}}_i \tag{24}$$

This matrix $\mathbf{H}$ is estimated by four or more non- collinear matched target feature points that are obtained by the feature extraction algorithm. $\mathbf{R}$ and $\mathbf{T}/\bar{d}$ are obtained by SVD for $(\mathbf{K}^{-1}\mathbf{H}\mathbf{K})$. The yaw angle θ of the robot is calculated using $\mathbf{R}$ and the translations t_x and t_y in two directions which is obtained by $\mathbf{T}/\bar{d}$. Therefore, $[t_x, t_y, \theta]$ is used as the state space, so the dimension of the state space is always three, even if there are many feature points.

The global asymptotic stability of the IBVS still cannot be ensured, even with a perfect knowledge of the image Jacobian matrix. Only a local asymptotic stability around the small neighborhood of the desired position is ensured. Therefore, a logarithmic interval state space partitioning (LISSP) method is required for more dexterous control in the vicinity of the target position.

The partitioning method for each dimension of continuous state space $[t_x, t_y, \theta]$ is similar. For example, s^1 donates the first dimension t_x. The upper limit $s^1_{\max}$ and the lower limit $s^1_{\min}$ depend on the operating environment. If the interval $\left[0, s^1_{\max}\right]$ is divided into n segments, then every segment is $\left[s_i^{1+}, s_{i+1}^{1+}\right](i = 0,1,\ldots,n-1)$. In a similar way, the interval $\left[s^1_{\min}, 0\right]$ is divided into $\left[s_i^{1-}, s_{i+1}^{1-}\right]$ (*i* =0,1,…,*n*-1). The calculation formula for s_i^{1+} and s_i^{1-} are

$$s_i^{1+} = s^1_{\max} \times \left(1 - \log_{n+1}\left(n-i+1\right)\right) \tag{25}$$

$$s_i^{1-} = s^1_{\min} \times \left(1 - \log_{n+1}\left(n-i+1\right)\right) \tag{26}$$

The partitioning method is similar for the other two dimensions, but the upper limits and the lower limits are different.

In theory, the yaw angle θ of the robot ranges from $-\pi/2$ to $\pi/2$ and the translations t_x and t_y range from $-\infty$ to $+\infty$. However, when the yaw angle θ is close to $-\pi/2$ or $\pi/2$, or the translation is large, the target in the image will be lost. In addition, if the state partition uses the upper and lower limits for each state dimension, the fineness of the partition interval is adversely affected. Therefore, $\left[-\pi/2, \theta_{\min}\right]$, $\left[\theta_{\max}, \pi/2\right]$, $\left[-\infty, t_{x\min}\right]$, $\left[t_{x\max}, +\infty\right]$, $\left[-\infty, t_{y\min}\right]$, and $\left[t_{y\max}, +\infty\right]$ are regarded as extra states. For example, a dimensional translation range is $\left[0, 5\right]$ meters, of which $\left[4, 5\right]$ meters is used as an extra state and $\left[0, 4\right]$ is non-uniformly divided. Finally, each dimension state space is divided into $2n+2$ states.

C. Action Set

If the β value is directly selected as the action set, the division is small and the uncertainty in the range of the value of β results in a very large number of action set numbers. This involves a great learning cost for a RL system. Reasonable initial parameters are also important for RL. Therefore, an initial value β^{initial} that is acquired by experience is set artificially. This initial value β^{initial} is fine-tuned by RL, which greatly reduces the learning cost. In this study, $\beta^{\text{initial}} = 0.7$. Different states have different β values. Initialization is:

$$\beta(\mathbf{s}) {=} \beta^{\text{initial}}(\mathbf{s} \in \mathbf{S}) \tag{27}$$

If the action set number is $2n+1$ and the change unit is ρ, the action set $\mathbf{A} {=} \{a_i \mid i = 1, 2, \ldots, 2n+1\}$ is calculated as

$$\left(-n\rho, -(n-1)\rho, \ldots, -\rho, 0, \rho, \ldots, (n-1)\rho, n\rho\right) \tag{28}$$

The formula for updating $\beta(\mathbf{s})$ is

$$\beta(\mathbf{s}) = \beta(\mathbf{s})\left(1 + a_i\right) \in \left[0, 1\right] \tag{29}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2865004, IEEE Transactions on Industrial Informatics

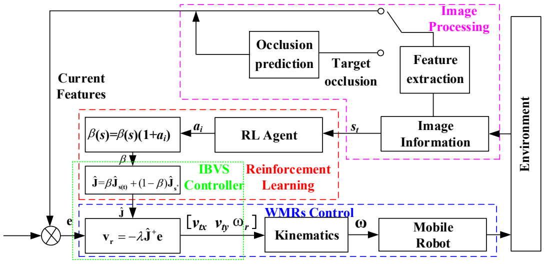


Fig. 3. The functional block diagram for the proposed IBVS system.

where $\mathbf{s}$ is the current state and a_i is the current selected action for $\mathbf{A}$.

Boltzmann used simulated annealing (SA) and statistical probability to address the conflicting requirements of exploration and utilization. The Boltzmann policy determines the probability of executing an action a_i in the current state and the probability is:

$$P(\mathbf{s},a_i)=\frac{\exp\left[Q(\mathbf{s},a_i)/T_t\right]}{\sum_{a_k\in A}\exp\left[Q(\mathbf{s},a_k)/T_t\right]} \tag{30}$$

where $T_t>0$ is the temperature coefficient. This controls the probability of choosing action a_i. If T_t tends to zero, the action selection is no longer random. T_t decays from the fixed initial value $Tem_{\max}$ and the updating formula is:

$$\begin{cases}T_0=Tem_{\max}\\ T_{t+1}=\mu T_t\end{cases} \tag{31}$$

where $0<\mu<1$ is the annealing factor. The smaller the value of μ, the faster is the attenuation of the temperature T_t.

D. Reward Function

The design of reward function is based on the goal of the tasks. The reward function designed in this work is composed of three sub-goals: reaching the desired position, outside the field of view, and reducing the feature errors as fast as possible.

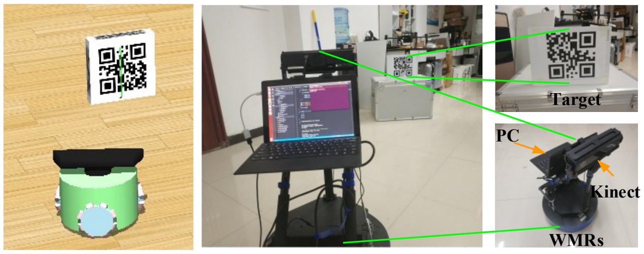
1) Reaching the desired position: If the pixel errors for all features in each dimension are all less than a threshold number of pixels, the WMR is recognized to have arrived at the desired position and a good reward, a positive constant *R*, is given.

2) Outside the field of view: If the number of features that is extracted is less than the preset number *M*, it is assumed that some target features may be outside the camera field of view and a bad reward, a negative constant *R*, is given.

3) Reducing the feature error: the reward for the accelerated reduction of errors uses the feature error $\sum_{i=1}^{M}\left\|\mathbf{f_i}-\mathbf{f_i^*}\right\|$, which is defined by Equation (6). The closer the distance to the desired position, the greater is the reward is granted. Therefore the reward function is established as Equation (32), which is used in both simulations and experiments.

$$r=\begin{cases}R & \text{,reach desired position}\\ -R & \text{,outside the field of view}\\ -R\sum_{i=1}^{M}\left\|\mathbf{f_i}-\mathbf{f_i^*}\right\|\Big/\left(M\sqrt{row^2+col^2}\right) & \text{,reduce the error}\end{cases} \tag{32}$$

where R is a positive constant that is determined by the actual situation, M is the number of feature points, *row* and *col* are respectively the height and width of the image plane.



(a) (b)

Fig. 4. The experimental environment: (a) the simulation platform and (b) the real environment.

TABLE I
Experimental parameters

Parameter	Value
ε	4 pixels
γ	0.9
η	0.95
h	7
n	10
ρ	0.05
R	100
α	0.4
λ	0.3
$T_{\max}$	40
$Tem_{\max}$	500
ζ	4

E. Functional Block Diagram for the Proposed IBVS System

The functional block diagram for the proposed IBVS system is shown in Fig. 3. The proposed method has three phases: the image processing phase, the RL phase, and the IBVS control phase. For image processing, the feature coordinates and the distances are used to calculate feature errors and the image Jacobian matrix. Occlusion prediction is performed when the target is occluded. For the RL phase, the β value is adjusted by using action a_i according to the current state s_t. Different β values are used for the control process because there are different states. Finally, in the IBVS control phase, the control quantity $[v_{rx}\quad v_{rx}\quad \omega_r]$ is obtained by the IBVS controller. The linear and the angular velocity of the WMRs transforms into the speed of the three wheels of the WMRs according to Kinematics.

V. Experiments and Simulations

The experiments were conducted to verify the superiority of proposed methods. Firstly, the proposed homography method and the homography method with RANSAC were tested in a real environment where the target is occluded. Secondly, in terms of the convergence rate for RL, the LISSP method and the uniform partitioning method for the state space were tested on the simulation platform. Thirdly, the agent was tested on the simulation platform at different stages of learning, which proves that the agent has learning ability. Finally, the proposed IBVS method, $\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s(t)}}$, $\hat{\mathbf{J}}=\mathbf{J}_{\mathbf{s}^*}$, and $\hat{\mathbf{J}}=(\mathbf{J}_{\mathbf{s(t)}}+\mathbf{J}_{\mathbf{s}^*})/2$ were tested on a simulation platform and in a real environment to demonstrate the efficiency of the proposed IBVS method.

The experiments were simulated in a commercial robotic simulator, which provides a WMR with the same dynamics as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2018.2865004, IEEE Transactions on Industrial Informatics

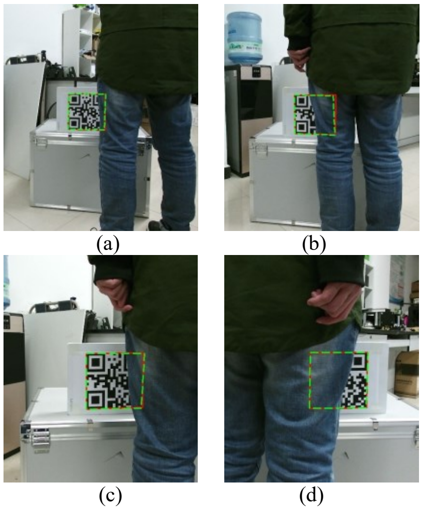
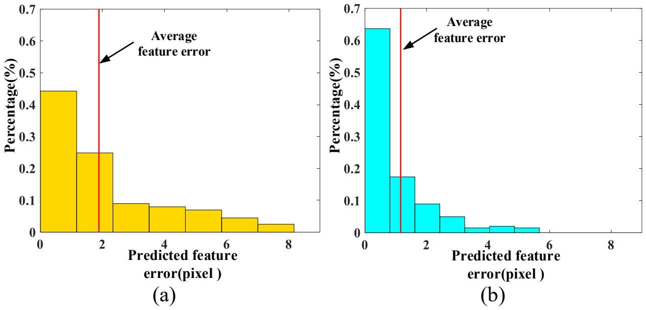

(a) (b) (c) (d)

Fig. 5. The experiment of occlusion prediction for the proposed homography method when QR code is occluded.



(a) (b)

Fig. 6. The predicted feature error distribution: (a) the homography method with RANSAC and (b) the proposed homography method.

the real one that was used in the experiments, as shown in Fig. 4(a). The experiences (the *Q*-table) that are learned in simulations are directly transferred to the physical robot before the real experiment, in order to reduce the times of real experiments. The experiments were conducted in a real environment to demonstrate the practicality of the proposed method. The real environment is shown in Fig. 4(b). The simulation and experimental parameters in this paper are shown in Table I.

A. Target occlusion test

The RGB camera for Kinect was calibrated and the result is $f_x = 551.70$ pixels, $f_y = 550.69$ pixels, $C_x = 465.85$ pixels, $C_y = 266.35$ pixels. The images observed by the RGB camera of the Kinect were sent to a computer for conducting the image process. A QR code was the target.

During the WMR movement, the test of QR code occlusion is performed. In Fig. 5, the red solid line represents the actual position of the QR code, the green dotted line represents the predicted position of the QR code. The experimental results show that the predicted feature information for the proposed homography method can be used for IBVS control instead of missing feature information.

The homography method with RANSAC and the proposed homography method were tested 50 times using an occluded QR code. The pixel coordinate errors for all occluded vertexes was calculated. The distribution of the predicted feature errors is shown Fig. 6, where the width of each bar represents the range of predicted feature errors in the pixel, the height represents the frequency of errors in the pixel, and the red line represents the average value of the predicted feature errors for all missing feature points. However, the average value of the predicted feature errors for the homography method with RANSAC is 1.89 pixels. The average value for the predicted feature error using the proposed homography method is 1.28 pixels. This experimental result shows that the model that is trained using the proposed homography method predicts more accurately. When the QR code is occluded and the feature information that is acquired by the camera is lost, the proposed homography method uses the context of the sequential image information to achieve a reasonable estimation for the lost feature information during visual servoing.

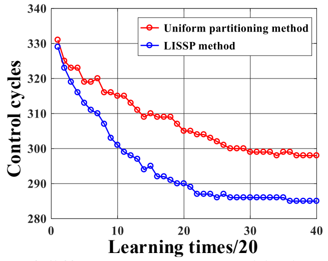


Fig. 7. A comparison of different state space partitioning methods.

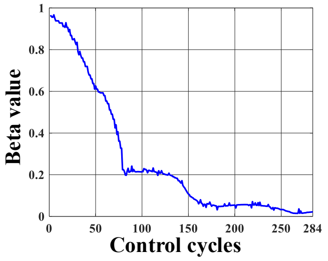


Fig. 8. The RL output adaptive β values after learning.

B. A comparison of methods for partitioning state space

There are some *Q*-Learning rules in the experiments:

1). If the WMR does not arrive at the desired position after 400 control cycles, the episode is terminated and this episode's control cycle is 400.

2). If some features are lost during servoing control, the episode is terminated and this episode's control cycle is ignored.

3). If each dimensional pixel error for all features are less than 4 pixels. The episode is terminated and the current control cycle is recorded.

For the uniform state space partitioning method, the $3m \times 3m$ experimental space is divided into 20×20 cells, so the area of each cell is $0.15m \times 0.15m$. The yaw angle of the robot is divided into 20 ranges, so the partition unit is 6°. Finally, $20 \times 20 \times 20$ states are obtained.

The LISSP method and the uniform state space partitioning method were tested on the simulation platform. The initial position and the orientation for the WMRs were ($0m, 0m, 0^\circ$). The desired position and orientation were ($0.5m, 1.5m, 0^\circ$). The position and orientation of the QR code were ($0.5m, 2.0m, 0^\circ$). Both partitioning methods were tested 20 times using *Q*-Learning with the Boltzmann action selection strategy. The experiments ran with control cycles of 0.05s.

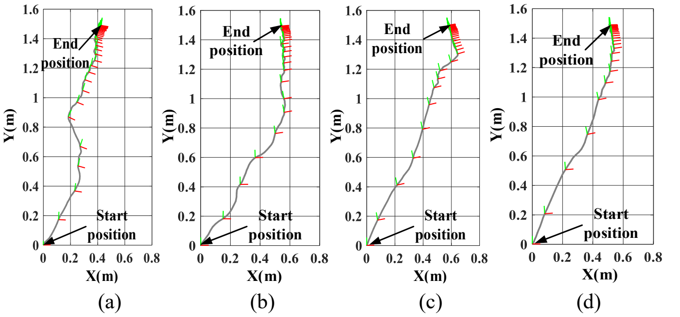


Fig. 9. The movement trajectory of the WMR for different learning cycles: (a) 0 learning cycles, (b) 200 learning cycles, (c) 400 learning cycles and (d) 800 learning cycles.

The learning experiment was repeated 800 times. Fig. 7 shows that the number of control cycles decreases during learning because of the learning behavior. However, the curve for the uniform partitioning method converges to 298 control cycles after about 720 episodes. The curve for LISSP eventually converges to 284 control cycles after about 700 episodes. The LISSP converges faster than the uniform partitioning method. Fig. 8 shows that the RL output adaptive beta values after learning, which means that the change for β is not drastic.

C. *Learning test*

To demonstrate the learning ability of the agent, the outcomes for before learning, after 200 learning cycles, after 400 learning cycles and after 800 learning cycles were demonstrated on the simulation platform. The movement trajectory for the WMR after different numbers of learning cycles is shown in Fig. 9. This experiment uses the same initial position as Experiment B.

In Fig. 9, the red and green lines respectively represent the X and Y axes of the WMR, the yaw angle of which is measured every ten control cycles. The gray line represents the movement trajectory of the WMR. Fig. 9 shows that the trajectory is not smooth in the early stages of learning because action selections are being explored, but this improves as learning time increases. The results of the experiment show that the agent learns well.

D. *A comparison of the control methods*

To demonstrate the efficiency and applicability of the proposed method, the proposed IBVS method is compared with $\hat{\mathbf{J}}=\mathbf{J}_{s(t)}$, $\hat{\mathbf{J}}=\mathbf{J}_{s^*}$, and $\hat{\mathbf{J}}=(\mathbf{J}_{s(t)}+\mathbf{J}_{s^*})/2$ using a simulation platform and in a real environment. Several methods are compared in terms of real-time feature errors for the same initial conditions.

In the simulation platform and the real environment, the initial position and the orientation for the WMR are ($0m, 0m, -20^{\circ}$). The desired position and the orientation are ($0.7m, 2m, 0^{\circ}$). The position and the orientation of the QR code are ($0.7m, 2.5m, 0^{\circ}$). Each method was tested 50 times. Average values for the feature errors in the pixel for these 50 tests are given in Fig. 10, and Fig. 11. Since the average errors for four feature points are similar, only the feature errors for two diagonal feature points are shown. Fig. 10 shows that the settings for $\hat{\mathbf{J}}=\mathbf{J}_{s(t)}$ requires about 339 control cycles to converge, $\hat{\mathbf{J}}=\mathbf{J}_{s^*}$ converges in 364 control cycles, and $\hat{\mathbf{J}}=(\mathbf{J}_{s(t)}+\mathbf{J}_{s^*})/2$ converges in about 327 control cycles, but the proposed IBVS method only requires about 284 control cycles.

In a real environment with various noise, it is necessary to put the learned beta values into the real scene to continue the learning. Fig. 11 shows that $\hat{\mathbf{J}}=\mathbf{J}_{s^*}$, $\hat{\mathbf{J}}=\mathbf{J}_{s(t)}$, $\hat{\mathbf{J}}=(\mathbf{J}_{s(t)}+\mathbf{J}_{s^*})/2$ respectively require about 313 control cycles, 331 control cycles and 288 control cycles to converge. The proposed IBVS method requires only about 266 control cycles. The experimental results show that the proposed IBVS method converges faster than the compared methods.

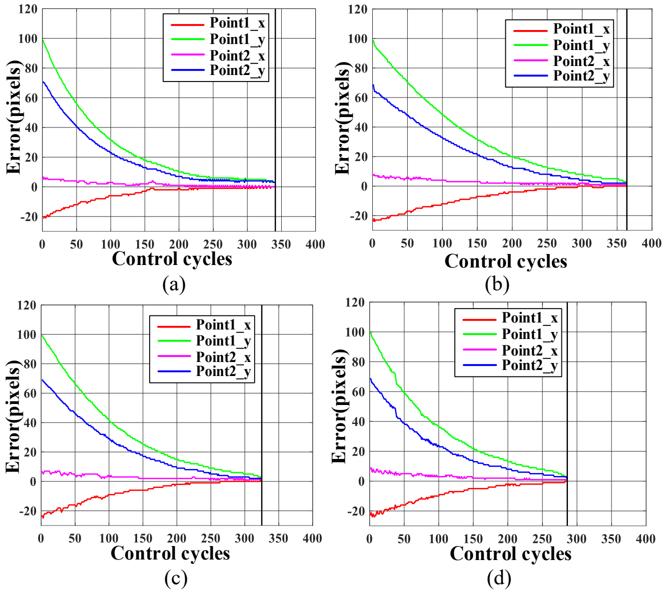


Fig. 10. A comparison of the feature error in the simulation platform: (a) $\hat{\mathbf{J}}=\mathbf{J}_{s(t)}$, (b) $\hat{\mathbf{J}}=\mathbf{J}_{s^*}$, (c) $\hat{\mathbf{J}}=(\mathbf{J}_{s(t)}+\mathbf{J}_{s^*})/2$ and (d) the proposed IBVS method.

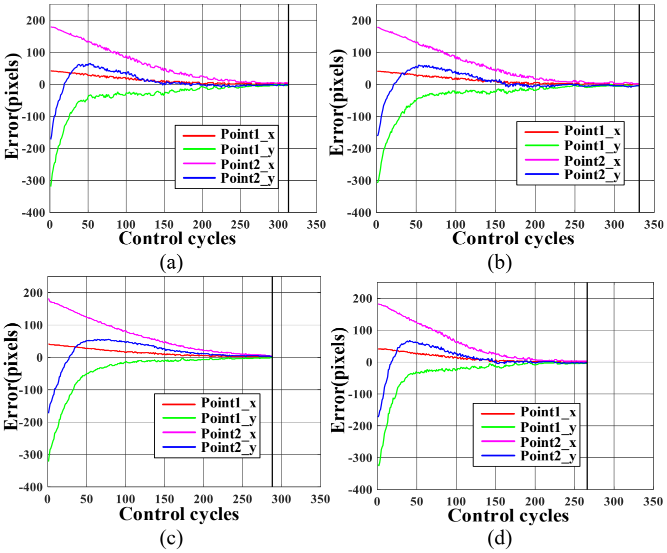


Fig. 11. A comparison of the feature error in the real environment. (a) $\hat{\mathbf{J}}=\mathbf{J}_{s(t)}$ (b) $\hat{\mathbf{J}}=\mathbf{J}_{s^*}$, (c) $\hat{\mathbf{J}}=(\mathbf{J}_{s(t)}+\mathbf{J}_{s^*})/2$ and (d) the proposed IBVS method.

VI. Conclusions

Proper feature extraction and appropriate servoing control law are essential to performance for IBVS. IBVS control can be interrupted or interfered abruptly if no features are extracted when the observed object is occluded. A homography method that uses a priori visual information is proposed to predict the image coordinates for missing feature points, in order to ensure the continuous execution of IBVS. The homography relationship that is defined using the proposed method is more accurate than the homography method without prior information. Meanwhile, an adaptive approach is proposed to determine the mixing parameter β for the IBVS control law. A series of values for β is calculated using *Q*-learning to improve control performance. The performance of the proposed controller is evaluated and validated in simulations and experiments, in terms of the movement characteristics of the WMRs under servoing control. In the future, will determine how the proposed method can be extended to other robots, such as under-actuated quadrotor drones.

References

[1] H. Cho and S. W. Kim, "Mobile Robot Localization Using Biased Chirp-Spread-Spectrum Ranging," in *IEEE Transactions on Industrial Electronics*, vol. 57, no. 8, pp. 2826-2835, Aug. 2010.

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

[3] X. Hou, R. Mahony and F. Schill, "Comparative Study of Haptic Interfaces for Bilateral Teleoperation of VTOL Aerial Robots," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 10, pp. 1352-1363, Oct. 2016.

[4] O. Araar and N. Aouf, "Visual Servoing of a Quadrotor UAV for the Tracking of Linear Structured Infrastructures," *2013 IEEE International Conference on Systems, Man, and Cybernetics*, Manchester, 2013, pp. 3310-3315.

[5] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," in *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82-90, Dec. 2006.

[6] F. Janabi-Sharifi, L. Deng and W. J. Wilson, "Comparison of Basic Visual Servoing Methods," in *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 5, pp. 967-983, Oct. 2011.

[7] 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," in *IEEE Transactions on Industrial Electronics*, vol. 59, no. 12, pp. 4735-4752, Dec. 2012.

[8] R. Ozawa and F. Chaumette, "Dynamic visual servoing with image moments for a quadrotor using a virtual spring approach," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 5670-5676.

[9] M. Keshmiri, W. F. Xie and A. Mohebbi, "Augmented Image-Based Visual Servoing of a Manipulator Using Acceleration Command," in *IEEE Transactions on Industrial Electronics*, vol. 61, no. 10, pp. 5444-5452, Oct. 2014.

[10] Serra P, Cunha R, Hamel T, et al, "Nonlinear Image-Based Visual Servo Controller for the Flare Maneuver of Fixed-Wing Aircraft Using Optical Flow," in *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 570-583, 2015.

[11] H. Shi, X. Li, K. S. Hwang, W. Pan and G. Xu, "Decoupled Visual Servoing With Fuzzy *Q*-Learning," in *IEEE Transactions on Industrial Informatics*, vol. 14, no. 1, pp. 241-252, Jan. 2018.

[12] N. Cazy, P. B. Wieber, P. R. Giordano and F. Chaumette, "Visual servoing when visual information is missing: Experimental comparison of visual feature prediction schemes," *2015 IEEE International Conference on Robotics and Automation* (ICRA), Seattle, WA, 2015, pp. 6031-6036.

[13] W. Fu, H. Hadj-Abdelkader and E. Colle, "Visual servoing based mobile robot navigation able to deal with complete target loss," *2013 18th International Conference on Methods & Models in Automation & Robotics* (MMAR), Miedzyzdroje, 2013, pp. 502-507.

[14] V. Lippiello, B. Siciliano and L. Villani, "Position-Based Visual Servoing in Industrial Multirobot Cells Using a Hybrid Camera Configuration," in *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 73-86, Feb. 2007.

[15] I. Siradjuddin, S. P. Tundung, A. S. Indah and S. Adhisuwignjo, "A real-time Model Based Visual Servoing application for a differential drive mobile robot using Beaglebone Black embedded system," *2015 IEEE International Symposium on Robotics and Intelligent Sensors* (IRIS), Langkawi, 2015, pp. 186-192.

[16] W. Pan, M. Lyu, K. S. Hwang, M. Y. Ju and H. Shi, "A Neuro-Fuzzy Visual Servoing Controller for an Articulated Manipulator," in *IEEE Access*, vol. PP, no. 99, pp. 1-1. DOI:10.1109/ACCESS.2017.2787738.

[17] H. Lang, Y. Wang and W. d. S. Clarence, "Vision based object identification and tracking for mobile robot visual servo control," *2010 IEEE International Conference on Control & Automation* (ICCA), Xiamen, China, 2010, pp. 92-96.

[18] W. G. Pence, F. Farelo, R. Alqasemi, Y. Sun and R. Dubey, "Visual servoing control of a 9-DoF WMRA to perform ADL tasks," *2012 IEEE International Conference on Robotics and Automation* (ICRA), Saint Paul, MN, 2012, pp. 916-922.

[19] N. Djelal, N. Saadia and A. Ramdane-Cherif, "Target tracking based on SURF and image based visual servoing," *International Conference on Communications*, Marseilles, 2012, pp. 1-5.

[20] T. La Anh, J.-B. Song, "Robotic grasping based on efficient tracking and visual servoing using local feature descriptors", *International Journal of Precision Engineering & Manufacturing*, vol. 13, no. 3, pp. 387-393, 2012.

[21] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision* (ICCV), Barcelona, 2011, pp. 2564-2571.

[22] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” *2006 European Conference on Computer Vision* (ECCV), Graz, Austria, May 2006, pp. 430–443

[23] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” *2010 European Conference on Computer Vision* (ECCV), Hersonissos, Greece, September 2010, pp.

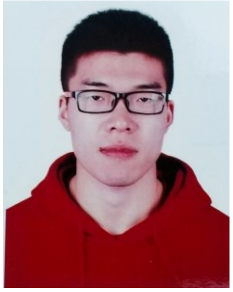
[24] M. Marey and F. Chaumette, "Analysis of classical and new visual servoing control laws," 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, 2008, pp. 3244-3249.

[25] H. Shi, Z. Lin, S. Zhang, X. Li, K. S. Hwang, “An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer,” *Information Sciences*, vol. 436, pp. 267-281, April 2018.

[26] H. Shi, Z. Lin, K. S. Hwang, S. Yang and J. Chen, “An Adaptive Strategy Selection Method With Reinforcement Learning for Robotic Soccer Games,” in *IEEE Access*, vol. 6, pp. 8376-8386, 2018.

Haobin Shi is an Associate Professor in the School of Computer Science at Northwestern Polytechnical University, China, and visiting scholar of Electrical Engineering Department at National Sun Yat-sen University, Taiwan. He received his Ph.D. degree from Northwestern Polytechnical University, China, in 2008. He is the director of the Chinese Association for Artificial Intelligence. His research interests include intelligent robots, decision support systems, artificial intelligence, multi-agent systems, and machine learning.

Gang Sun received his B.S. degree in the School of Electronics and Information Engineering, Xi'an Technology University, China, in 2012. He is currently working toward master degree in the School of Software and Microelectronics, Northwestern Polytechnical University, China. His research interests include intelligent robots, intelligent control, machine learning.



Yuanpeng Wang received his B.S. degree in the School of Software and Microelectronics, Northwestern Polytechnical University, China, in 2011. He is currently working toward master degree in the School of Software and Microelectronics, Northwestern Polytechnical University, China. His research interests include mobile robot, intelligent control, reinforcement learning.

Kao-Shing Hwang (M'93–SM'09) is a professor of Electrical Engineering Department at National Sun Yat-sen University, and an adjunct professor of the department of Healthcare Administration and Medical Informatic, Kaohsiung Medical University, Taiwan. He received the M.M.E. and Ph.D. degrees in Electrical and Computer Engineering from Northwestern University, Evanston, IL, U.S.A., in 1989 and 1993, respectively. He had been with National Chung Cheng University in Taiwan from 1993-2011. He was the deputy director of Computer Center (1998-1999), the chairman of the Electrical Engineering Department (2003~2006), and the director of the Opti-mechatronics Institute of the university (2010~2011). He has been a member of IEEE since 1993 and a Fellow of the Institution of Engineering and Technology (FIET). His research interest includes methodologies and analysis for various intelligent robot systems, machine learning, embedded system design, and ASIC design for robotic applications.