IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 13, 2017, accepted December 17, 2017, date of publication January 5, 2018,
date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2787738

A Neuro-Fuzzy Visual Servoing Controller for an Articulated Manipulator

WEI PAN[1], MENGYANG LYU[1], KAO-SHING HWANG[2], (Senior Member, IEEE), MING-YI JU[3], (Member, IEEE), AND HAOBIN SHI[1]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2]Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
[3]Department of Computer Science and Information Engineering, National University of Tainan, Tainan 70005, Taiwan

Corresponding author: Kao-Shing Hwang (hwang@ccu.edu.tw)

This work was supported by the Aeronautical Science Foundation of China under Grant 2016ZC53022.

ABSTRACT The challenges of selecting appropriate image features, optimizing complex nonlinear computations, and minimizing the approximation errors always exist in visual servoing. A fuzzy neural network controller is developed for a six-degrees-of-freedom robot manipulator to perform visual servoing is proposed to tackle these problems. To increase the accuracy of the image preprocesses, a synthetic image process performs feature extraction for the controller. The method combines a support vector machine contour recognition algorithm and a color-based feature recognition algorithm. For visual servoing, a control method based on the fuzzy cerebellar model articulation controller with the Takagi–Sugeno framework is proposed to directly map an image feature error vector to a desired robot end-effector velocity. This approach achieves visual servoing control without the need of computing the inverse interaction matrix. The control variables are learned and updated by the T–S fuzzy inference. This simplifies the implementation of visual servoing in real-time applications. The proposed control method is used to control an articulated manipulator with an eye-in-hand configuration. The results of simulations and experiments demonstrate that the proposed visual servoing controller has good performance, in terms of stability and convergence.

INDEX TERMS CMAC, robotic manipulator, T-S fuzzy, visual servoing.

I. INTRODUCTION

Visual servoing is a visual control technology that uses visual information to control the motion of a robotic system. Visual servoing reduces the instability and uncertainty of the robotic system. The response speed and accuracy of the visual orientation or tracking is also increased. The robotic vision servoing system is highly accurate and robust. Detailed reviews of visual servoing can be found in [1] and [2].

In general, visual servoing can be classified into three categories: position-based visual servoing (PBVS) [3], image-based visual servoing (IBVS) [4] and hybrid servoing control [5]. The process of visual servoing involves image processing and servoing control. Typically, PBVS defines the error signal for the posture between the end-effector of a robot and the target object, in a Cartesian coordinate system, and calculates the position and posture of the end-effector against the target object using the robotic kinematics model and camera calibration. Because there is no direct control of the image, it is impossible to ensure that the object always stays within the camera's field of view for PBVS, especially if there are significant calibration errors. Perspective from multiple points also causes a problem if the object is badly posed, and the process is sensitive to perturbations. Small errors in the image measurements can induce quite different results. In position-based visual servoing, most of the control problems concern the pose estimation algorithm. It is difficult to determine the number, the nature and the configuration of features that are necessary to ensure optimal pose estimation. PBVS does not actually integrate the functions of vision and control into a feedback control system.

In comparison to PBVS, IBVS has a clearer control structure and is easier to implement. The control law is directly defined on the image plane space. IBVS establishes a non-linear mapping relationship between the image feature error and the posture of the robotic manipulator. Although IBVS is not sensitive to calibration error or the errors in workspace modeling, the existence of a singular point in the Jacobian matrix can lead to unstable system responses. IBVS is generally satisfactory, even if there are calibration errors in the configurations of a camera or a hand-eye device.

However, stability and convergence problems can occur if the Jacobian matrix becomes singular during the servoing, which leads to an unstable behavior, or there may be local minima because of unrealizable image motions. A hybrid servoing control method is proposed that has the advantages of both PBVS and IBVS. However, different visual servoing tasks use different switching rules for PBVS and IBVS so the generality of the hybrid method is limited.

Various image processing methods are used to extract and match features for IBVS, such as the RGB-based method [5]–[7], the contour-based method [8], the SIFT-based method [9] or the HOG-based method [10]. These methods rely on certain characteristic aspects of the images (e.g. color, shape, or point coordinates) so that there is a corresponding relationship between the camera image and the target objects in the workspace. However, image bias can occur when these methods are used separately. An effective control scheme and control method can solve the problem of uncertainty that is caused by the estimation of the image's Jacobian matrix and the robot's Jacobian matrix. Some studies propose closed-form solutions [11] or adaptive schemes [12]. An augmented image-based visual servoing is proposed in [13] to improve IBVS. However, changing environment has a significant effect on visual servoing. Effective control strategies address the problem of uncertainties that is caused by estimating the image's Jacobian matrix. Recent studies of visual servo control mainly concentrate on fuzzy control, neural network control, and other intelligent control methods. Using fuzzy logic inference, fuzzy control gives a good approximation of the nonlinear mapping between variations in the image features and the motion of the manipulator motion. Stability analysis and fuzzy control systems are proposed in [14] and [15]. The Takagi–Sugeno (T-S) fuzzy model can represent a general class of static or dynamic nonlinear systems [16]. An adaptive distributed fuzzy T–S controller is used in [17]. It has been proven that the T–S fuzzy model [18] gives better performance for a servoing system.

Neural networks uniformly approximate arbitrary input-output nonlinear mappings. In particular, a cerebellar model articulation controller (CMAC) is used for the real-time control of robotic manipulators because it allows simple computation and fast learning and has good generalization [19], [20]. The combination of fuzzy logic and a neural network allows easy expression of human knowledge of fuzzy logic and distributed information storage and learning ability for a neural network and allows effective modeling and control for a complex system. Reinforcement learning has also been used to improve the performance of visual servoing using online learning [21]–[23].

In this paper, a visual servoing system for a 6-DOF robotic manipulator is designed, implemented and verified. There are two main contributions: image feature point recognition and a visual servoing control. In order to extract image features quickly and accurately, a synthetic image processing method is proposed. This method combines an SVM-based feature point extraction algorithm and a color-based feature point recognition algorithm. It addresses the problem of unknown interference in the image background during feature point extraction. The main issue with visual servoing is controlled. This paper proposes a visual servoing method that combines CMAC and a fuzzy T-S model. The training data for the control model is derived using IBVS, so the pseudoinverse matrix of the Jacobin and the inverse interaction matrix for the robot do not have to be computed during servoing, which greatly reduces the computation cost. The robotic manipulator that is used is an articulated manipulator with an eye-in-hand configuration.

This paper has six sections. Following the introduction, Section II discusses the image-based visual servoing and visual servoing control system that uses FCMAC-T-S. The synthetic method for recognition of image feature points is discussed in Section III. The visual servoing controller based on an FCMAC-T-S model is illustrated in detail in Section IV. The experimental results are presented in Section V. Section VI gives the conclusion.

II. DESIGN OF THE VISUAL SERVOING CONTROL STRUCTURE

A. CLASSICAL IMAGE-BASED VISUAL SERVOING

The aim of visual servoing control is to minimize the error e (t) between the current image and the target image, which is defined as:

$$e\,(t) = \varepsilon - \varepsilon^* \tag{1}$$

where ε is a current vector for features and ε^* is the desired vector for features.

The motion of the camera is represented by the translation $\boldsymbol{T_c} = \left[v_x, v_y, v_z\right]^T$ and the rotation $\boldsymbol{R_c} = \left[\omega_x, \omega_y, \omega_z\right]^T$ in space and the velocity of the camera is expressed as $\boldsymbol{X_c} = [\boldsymbol{T_c}, \boldsymbol{R_c}]^T$. In robot kinematics, the interaction matrix $\boldsymbol{L}_\varepsilon$ is used to establish the relationship between the feature point vector ε and the camera velocity $\boldsymbol{X_c}$, which is defined as:

$$\varepsilon = \boldsymbol{L}_\varepsilon \boldsymbol{X_c} \tag{2}$$

The interaction matrix $\boldsymbol{L}_\varepsilon$ is related to chosen features ε by:

$$\boldsymbol{L}_\varepsilon = \begin{bmatrix} \frac{-1}{Z} & 0 & \frac{x}{Z} & xy & -\left(1+x^2\right) & y \\ 0 & \frac{-1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \end{bmatrix} \tag{3}$$

where Z is the depth between the target object and the end-effect and (x, y) is the coordinate of a feature point in the image.

The velocity of the camera $\boldsymbol{X_c}$ is expressed as:

$$\boldsymbol{X_c} = -\lambda \boldsymbol{L}_\varepsilon^{-1} \left(\varepsilon - \varepsilon^*\right) \tag{4}$$

where $\boldsymbol{L}_\varepsilon^{-1}$ is the inverse matrix of $\boldsymbol{L}_\varepsilon$ and λ is the servoing gain constant. When $\boldsymbol{L}_\varepsilon$ is not square, the inverse interaction matrix $\boldsymbol{L}_\varepsilon^{-1}$ does not exist. The Moore-Penrose Pseudoinverse is used to approximate the inverse value of $\boldsymbol{L}_\varepsilon$, which is recorded as $\boldsymbol{L}_\varepsilon^{+}$.

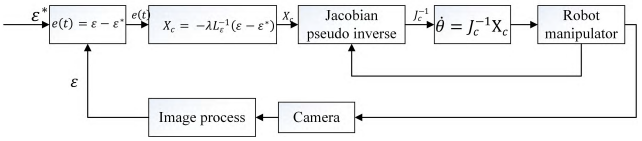

FIGURE 1. The IBVS control scheme.

According to robot kinematics, the relationship between the velocity of the camera $\boldsymbol{X_c}$ and the joint velocity $\dot{\theta}$ is defined by a Jacobian matrix $\boldsymbol{J_c}$:

$$\dot{\theta} = \boldsymbol{J_c^{-1}} \boldsymbol{X_c} \tag{5}$$

where $\boldsymbol{J_c^{-1}}$ is the pseudoinverse of Jacobian matrix $\boldsymbol{J_c}$.

$\boldsymbol{J_c^{-1}}$, $\boldsymbol{L_\varepsilon^+}$ and $\dot{\theta}$ are computed as:

$$\dot{\theta} = -\lambda \boldsymbol{J_c^{-1}} \boldsymbol{L_\varepsilon^+} \left(\varepsilon - \varepsilon^*\right) \tag{6}$$

The architecture of the IBVS is shown in Fig.1.

B. VISUAL SERVOING USING FCMAC-T-S

In IBVS, the inverse interaction matrix $\boldsymbol{L_\varepsilon^+}$ and the pseudoinverse of Jacobian matrix $\boldsymbol{J_c^{-1}}$ are calculated repeatedly. The IBVS control scheme must continuously estimate the parameters in the interaction matrix and compute the interaction matrix, which requires much computation time. To reduce the servoing control structure that uses CMAC and T-S Fuzzy is computation cost in a real-time control environment, a visual servoing control structure that uses CMAC and T-S Fuzzy is proposed in this paper. The control scheme is shown in Fig. 2.

In the proposed method, the image's Jacobian is numerically estimated during the camera's motion without taking into account the analytical form that is shown in (4). In this scheme, the inputs, which are feature errors, are quantified into the receptive field using a fuzzy basis function. The controller takes advantage of the generalization ability of CMAC. Hypercubes are formed by overlapping receptive fields. In the original CMAC, the values of weights are directly mapped to the memory space but the FCMAC-T-S controller uses a T-S fuzzy inference method that maps a linear combination of the inputs to the weight values. The output for the entire network is the sum of the activated weights, which is the angular velocity of each joint of the manipulator. The learning law for the controller learns and updates the parameter values according to the reference model so that the controller can ensure adequate stability for visual servoing, for example, the IBVS uses a PID controller.

III. THE SYNTHETIC FEATURE POINT RECOGNITION ALGORITHM

An image feature point is a piece of distinct information that is immune to image rotation, translation or scaling. Feature point recognition can directly affect the accuracy and efficiency of visual servoing. Obtaining the coordinate vector of these feature points quickly and accurately from

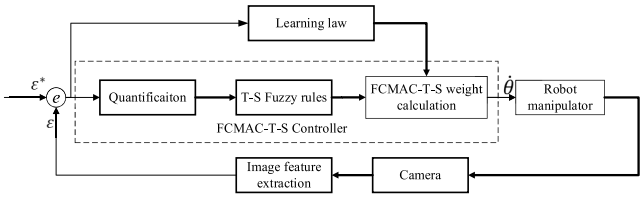

FIGURE 2. The FCMAC-T-S control scheme.

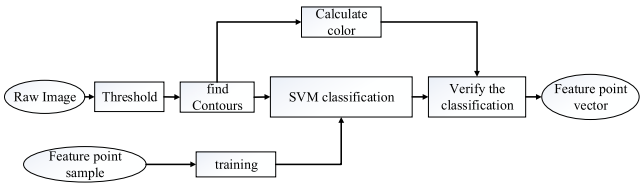

FIGURE 3. Feature point extraction.

continuously changing images is the main feature extraction problem for visual servoing. By matching the image with the prepared template, the computer can automatically identify the target feature points. In the visual servoing system, the servoing input is only a matrix of the feature points, so the raw image requires further processing, including image segmentation and feature point extraction. Because of the effect of external disturbance and the working environment, a single image processing method has difficulty in obtaining a matrix accurately so it is necessary to combine several methods. Using an edge extraction algorithm, this paper uses circles, squares, triangles, and crosses as the feature points of the target image. An SVM classification algorithm combined with a color-based image recognition algorithm is used to match the feature points of the target image. The basic process is shown in Fig.3.

A. EXTRACTION OF THE IMAGE BOUNDARY

Images are captured by a camera that is mounted on the end-effector of the manipulator. To locate objects and boundaries in an image, the threshold operation proposed in [24] is applied to the images transformed into binary ones. In order to separate out the ideal boundaries of one target object, an image boundary extraction algorithm (IB-EA) based on border following is used in [25]. For an input binary image, $f\ (i, j)$ represents the pixel value (either 0 or 1). To determine the start points of the boundaries, a scanning operation is executed line-by-line. The start points are located if and only if (i, j) statisfies either of the following conditions: if $f\ (i, j-1) = 0$ and $f\ (i, j) = 1$, $f\ (i, j)$ is the starting point of the outer boundary, or if $f\ (i, j) \geq 1$ and $f\ (i, j+1) = 0$, $f\ (i, j)$ is the starting point of the hole boundary.

From the starting point, the pixels on the boundary are marked. A unique identifier, called NBD, is assigned to represent the newly discovered boundary and the process is initialized by setting NBD=1. When a new boundary is found, its number is NBD plus 1. If and only if $f\ (i, j) = 1$ and $f\ (i, j+1) = 0$, $f\ (i, j)$ is assigned as -NBD, which means that a termination point for a right boundary has been reached.

The boundaries that are determined form a contour collection $T = (t_1, t_2, \cdots t_m)$, where t_i is the boundary matrix that have been found by IB-EA.

B. CONTOUR RECOGNITION

The image contour collections must match the desired feature templates. A support vector machine (SVM) algorithm [26] is used to recognize and classify the contour collection. The desired feature templates are marked using labels $\Gamma = (\Gamma_1, \Gamma_2, \cdots, \Gamma_n)$. For the SVM classifier, there is a training dataset $D = \{(t_1, \Gamma_1), (t_2, \Gamma_2) \cdots (t_n, \Gamma_n)\}$. A kernel function $K(t_i, t_j)$ is selected to map data to high-dimensional space.

To address the problem of nonlinear classification, SVM constructs separating hyperplanes as a linear classifier to separate datasets. The decision function can be defined as:

$$f(x) = \sum_{i=1}^{n} \Gamma_i a_i K(t_i, t_j) + b \tag{7}$$

where a_i and b are classification parameters needed to obtain, $0 \ll i, j \ll n$.

In order to get the best classification results, the cost function for the optimal separating hyperplanes is defined as follows:

$$min \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \Gamma_i \Gamma_j a_i a_j K(t_i, t_j) - \sum_{j=1}^{n} a_j \tag{8}$$

where $\sum_{i=1}^{n} \Gamma_i a_i = 0, 0 \ll a_i \ll \xi, i = 1, 2, \cdots n$. ξ is a penalty parameter and $\xi > 0$.

According to the Lagrangian multiplier method, the desired optimal parameter a^* can be calculated as $a^* = (a_1^*, \cdots a_n^*)^T$, so the desired optimal parameter b^* is denoted as:

$$b^* = \Gamma_j - \sum_{i=1}^{n} \Gamma_i a_i^* K(t_i, t_j) \tag{9}$$

The corresponding decision function $f(x)$ is calculated as:

$$f(x) = \sum_{i=1}^{n} \Gamma_i a_i^* K(t_i, t_j) + b^* \tag{10}$$

After getting the optimal classification surface, the shape of the feature points can be predicted in details.

C. SYNTHETIC FEATURE RECOGNITION ALGORITHM (S-FRA)

However, the SVM-based feature recognition algorithm (SVM-FRA) may yield incorrect recognition results, which reduces the degree of accuracy, if image feature extraction is performed in a noisy environment. To improve the accuracy of SVM-FRA, a color-based feature recognition process imposed to SVM-FRA, called S-FRA, is proposed to increase the classification ability of the SVM-FRA. The pixel points in a contour region t_i form a pixel collection Pr_{t_i}. For a defined feature point with a specific color, the color set is denoted as $Q_{Pr_{t_i}} = \{R, G, B\}$, where $R = \frac{\sum_{i=0}^{n} r}{n}$, $G = \frac{\sum_{i=0}^{n} g}{n}$, $B = \frac{\sum_{i=0}^{n} b}{n}$, n is the number of pixels in a pixel collection Pr_{t_i}, and r, g ,b are the strength of the three channels.

The centroid of Pr_{t_i} is defined as $X_i(x_i, y_i)$:

$$\begin{cases} x_i = \dfrac{\sum_{l=1}^{n} x_l^i}{n} - IL/2 \\ y_i = \dfrac{\sum_{l=1}^{n} y_l^i}{n} - IW/2 \end{cases} \tag{11}$$

Where (x_l^i, y_l^i) denotes the coordinate of the pixel in the collection Pr_{t_i}, IL and IW respectively represent the length and width of the image.

In the experiment, for all contour collection $T = (t_1, t_2, \cdots t_m)$, where $m >= 4$, there are only four desired feature point labels $\Gamma = (\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4)$.

If these have the relationship:

$$\begin{cases} 0 < |Q_{Pr_{t_i}} - Q_{\Gamma j}| < \chi \\ t_i \in \Gamma_i \end{cases} \tag{12}$$

Then t_i is defined as the jth feature point. where χ is a color identifier constant, $Q_{\Gamma j}$ represents the color set of the label Γ_j. Ultimately, $X_i(x_i, y_i)$ can be considered as the coordinate of the ith feature point.

IV. THE FCMAC-T-S CONTROLLER

A. STRUCTURE OF THE FCMAC-T-S CONTROLLER

In image-based visual servoing, $\boldsymbol{J_c^{-1}}$ and $\boldsymbol{L_\varepsilon^+}$ are constantly changing with the end-effector of the robotic manipulator so the computation cost is very high and the visual servoing is inefficient. The entire servoing system may even fail. $\boldsymbol{L_\varepsilon}$ is the matrix of camera depth Z. The height (depth) of the end-effecter changes so it is difficult to estimate without increasing the number of sensors in real-time, which significantly reduces the precision of visual servoing. Neural networks allow nonlinear fitting, memory and self-learning, so they are widely used for intelligent control systems. This paper uses a Cerebellar Model Articulation Controller (CMAC) neural network as the base controller. The concept of fuzzy control theory is used to transform the servoing input into fuzzy variables, which better reflects the fuzziness and continuity of human cognition. The Takagi-Sugeno model can transform non-linear problems into a linear correlation, which allows the possibility of there being no solution in a nonlinear system and increases the stability of the system. In this section, a controller based on FCMAC-T-S replaces classical image-based visual servoing (IBVS). The control system that is presented in this paper is shown in Fig.4.

B. SERVOING ADJUSTMENT WITH FCMAC-T-S CONTROL

1) MAPPING FROM THE INPUT DOMAIN TO THE STATE SPACE

The input domain for the controller is a continuous N-dimensional space, which is denoted by $\boldsymbol{I} = [I_1, I_2, \ldots, I_N]^T, \boldsymbol{I} \subset \mathbf{R}^N$, where I_i represents the coordinate error between current image features and desired image features. Depending on the vector $\boldsymbol{I}$, a state space is generated,

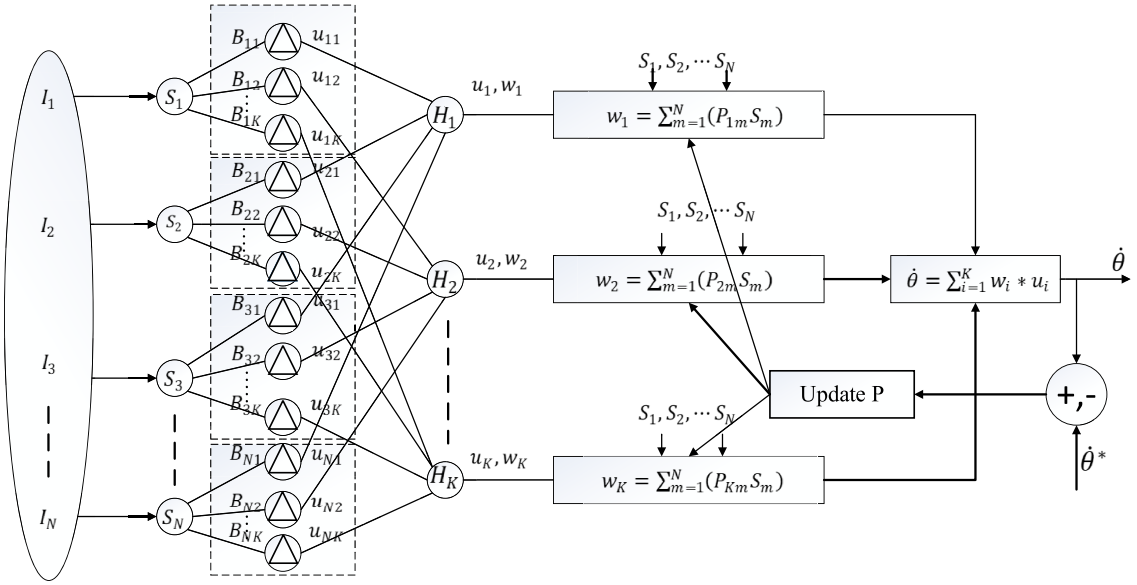

FIGURE 4. The structure of the FCMAC-T-S controller.

which is denoted by $\boldsymbol{S}= [S_1, S_2, \ldots, S_N]$, $\boldsymbol{S} \subset \mathbf{R}^N$, where N is the state dimension.

The mapping from the input domain to the state space is described as:

$$Q{:}\boldsymbol{I} \rightarrow \boldsymbol{S} \tag{13}$$

For a given input variable I_i, the value range can be divided into small discrete intervals, called elements. The value of a state variable S_i is relevant to these elements. Assuming that the width of one element of input I_i is Δd, and $I_i \in [\mathrm{V}_{L_i}, V_{U_i}]$, the state variable S_i is calculated as:

$$S_i = \left\lceil \frac{I_i - \mathrm{V}_{L_i}}{\Delta d} \right\rceil \tag{14}$$

2) FUZZY QUANTIZATION IN A RECEPTIVE FIELD SPACE

Each state variable S_i must be quantified into several local receptive fields, called blocks. Blocks are denoted by $B_i = \{B_{i1}, B_{i2}, \cdots, B_{iK},\}$, where i indicates the ith state variable and K is the quantization level. Each block consists of several successive elements. It is noted that the number of elements in a block affects the generalization capability of the FCMAC-T-S model. The notation C is the generalization parameter. The quantization level combined with the generalization parameter determines the quality of the network.

The number of blocks for each state variable is determined by the quantization level and the generalization parameter. For a quantization level K, there are K layers with the same value range. Each layer is divided into blocks, according to the generalization parameter C. By shifting an element, different blocks are obtained and there is partial overlapping between directly adjacent layers.

In the receptive field space of the FCMAC-T-S model for this paper, each block uses a fuzzy receptive-field basis function, which is formulated as a symmetrical triangular membership function, rather than a rectangular basis function. The definition of a symmetrical triangular membership function is:

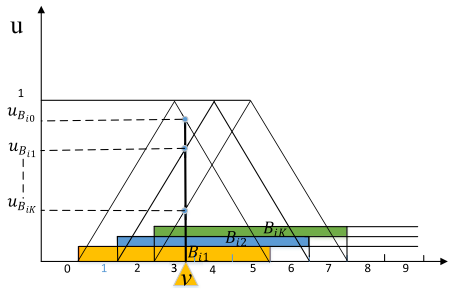

FIGURE 5. Fuzzy quantization of the FCMAC.

$$\mu_{B_{ij}}(S_i) = \begin{cases} \dfrac{(I_i - a)}{(b-a)/2} & a \le I_i < \dfrac{(b+a)}{2} \\ 1 & I_i = \dfrac{(b+a)}{2} \\ \dfrac{(b - I_i)}{(b-a)/2} & \dfrac{(b+a)}{2} < I_i \le b \\ 0 & otherwise \end{cases} \tag{15}$$

where $\mu_{B_{ij}}(S_i)$ denotes the basis function of the jth block of the ith state variable, $\mu_{B_{ij}}(S_i) \in [0, 1]$ and a and b respectively denote the maximum value and the minimum value of one block.

If the value of an input variable I_i is v and the quantization level $K = 3$ and the generalization parameter $C = 5$, the corresponding state variable S_i is quantified into three blocks and each block includes five elements. Three membership grades are calculated using the fuzzy triangular membership function. The process for block division and fuzzy quantization is illustrated in Fig.5.

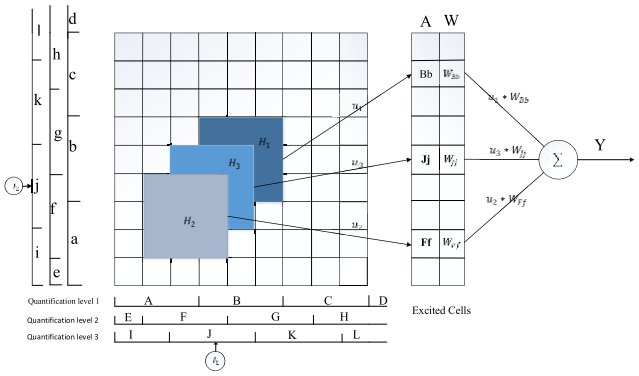

FIGURE 6. The process for forming the hypercube.

The blocks of different state variables can combine hypercubes, which are denoted by $H = \{H_1, H_2, \cdots, H_K\}$. Only blocks on the same layer can form a hypercube and the dimensions of a hypercube must be consistent with the state space. The number of hypercubes determines the size of the receptive field space. The mechanism for the overlapped hypercubes is shown using a 2-dimensional example in Fig.6. This example has an input vector $\boldsymbol{I}= [I_1, I_2]$, with a quantization level $K = 3$ and a generalization parameter $C = 3$.

3) T-S FUZZY IN WEIGHT MEMORY

IN the FCMAC-T-S model for this paper, each hypercube is associated with a weight vector using a Takagi-Sugeno fuzzy inference. A fuzzy rule R^l for the lth layer is expressed as:

$$\begin{aligned} &\textit{If } s_1 \textit{ is } B_{1l} \textit{ and } s_2 \textit{ is } B_{2l} \textit{ and } \cdots s_i \textit{ is } B_{il} \cdots \textit{ and } s_N \textit{ is } B_{Nl} \\ &\textit{Then } w_l = p_0^l s_0 + p_1^l s_1 + p_2^l s_2 + \cdots + p_i^l s_i + \cdots + p_N^l s_N \end{aligned} \tag{16}$$

where the *If*-part represents fuzzy receptive fields that are associated with the state variables, the *Then*-part denotes that a weight value w_l is mapped to an affine linear function of the state variables and p_i^l represents a linear parameter.

The fuzzy logic *AND* operation is performed to compute the activation strength of a rule, which is expressed as:

$$A_l(\boldsymbol{s}) = \mu_{B_{1l}}(s_1) \cdot \mu_{B_{2l}}(s_2) \cdot \ldots \cdot \mu_{B_{Nl}}(s_N) = \prod_{i=1}^{N} \mu_{B_{il}}(s_i) \tag{17}$$

where $A_l(\boldsymbol{s})$ denotes the activation strength of the lth fuzzy rule.

The weight space $w_l(\boldsymbol{s})$ is expressed as:

$$w_l(\boldsymbol{s}) = \left[p_0^l p_1^l \cdots p_N^l\right] \cdot [s_0 s_1 \cdots s_N]^T = \boldsymbol{p_l} \cdot \boldsymbol{s} \tag{18}$$

where $w_l(\boldsymbol{s})$ denotes the weight of the *Then*-part of the lth fuzzy rule and s_0 is a constant with a value of 1.

The activation strength of each fuzzy rule determines the weight's proportion in the weight space, which is expressed by the effective function rules as:

$$\Phi_l(\boldsymbol{s}) = \frac{A_l(\boldsymbol{s})}{\sum_{l=1}^{K} A_l(\boldsymbol{s})} \tag{19}$$

where $\Phi_l(\boldsymbol{s})$ denotes the proportion of each rule in all of the rules.

4) THE OUTPUT SPACE

The final output for the FCMAC-T-S model represents the variation in the angle of each joint Θ, which is determined as a weighted mean value over all fuzzy rules as:

$$\Theta = \frac{\sum_{l=1}^{K} A_l(\boldsymbol{s}) w_l(\boldsymbol{s})}{\sum_{l=1}^{K} A_l(\boldsymbol{s})} \tag{20}$$

C. LEARNING AND UPDATING

The FCMAC-T-S model converts the output weights to a linear combination of T-S fuzzy rules so the learning and updating of weights are equivalent to the learning and updating of the parameters in a linear combination. The updating of the parameters in the linear combination is given by:

$$w_l(t) = \boldsymbol{p_l}(t-1) \cdot \boldsymbol{s} + \Delta \boldsymbol{p_l} \cdot \boldsymbol{s} \tag{21}$$

where $\Delta \boldsymbol{p_l}$ is the increment of the vector $\boldsymbol{p_l}(t-1)$.

The gradient descent method is used to update the linear parameters $\boldsymbol{p_l}(t)$. The mean square covariance between the desired output Θ_d and the actual output Θ_s is defined as the cost function E:

$$\begin{aligned} E &= \frac{1}{2}(\Theta_d - \Theta_s)^2 = \frac{1}{2}\left(\Theta_d - \frac{\sum_{l=1}^{K} A_l(\boldsymbol{s}) w_l(\boldsymbol{s})}{\sum_{l=1}^{K} A_l(\boldsymbol{s})}\right)^2 \\ &= \frac{1}{2}(\gamma)^2 \end{aligned} \tag{22}$$

where γ is the difference between Θ_d and Θ_s.

The adjustment of the weight value is denoted by:

$$\begin{aligned} \boldsymbol{p_l}(\mathrm{t}) &= \boldsymbol{p_l}(\mathrm{t}-1) + \Delta \boldsymbol{p_l} = \boldsymbol{p_l}(\mathrm{t}-1) - \eta \cdot \frac{\partial E}{\partial \boldsymbol{p_l}(\mathrm{t}-1)} \\ &= \boldsymbol{p_l}(\mathrm{t}-1) - \eta \cdot \frac{\partial E}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \boldsymbol{p_l}(\mathrm{t}-1)} \end{aligned} \tag{23}$$

According to the cost function E, $\boldsymbol{p_l}(t)$ is also expressed as:

$$\begin{aligned} \boldsymbol{p_l}(\mathrm{t}) &= \boldsymbol{p_l}(\mathrm{t}-1) - \eta \cdot \gamma \cdot \frac{\partial \left(\Theta_d - \frac{\sum_{l=1}^{K} A_l(\boldsymbol{s}) \boldsymbol{p_l}(\mathrm{t}-1) \cdot \boldsymbol{s}}{\sum_{l=1}^{K} A_l(\boldsymbol{s})}\right)}{\partial p_r(\mathrm{t}-1)} \\ &= \boldsymbol{p_l}(\mathrm{t}-1) + \eta \cdot \gamma \cdot \frac{\sum_{l=1}^{K} A_l(\boldsymbol{s}) \cdot \boldsymbol{s}}{\sum_{l=1}^{K} A_l(\boldsymbol{s})} \end{aligned} \tag{24}$$

where $\boldsymbol{p_l}(t)$ denotes the linear combination parameters of the lth rule in the tth step and η is the learning rate.

D. THE CONTROL ALGORITHM

The proposed controller algorithm based on FCMAC-T-S is described using **Algorithm 1**.

V. EXPERIMENTAL RESULTS

The experiments consist of two parts: simulations and real experiments. The simulations were performed in Webots7.0.3. The simulation environment, as shown in Fig.7, consisted of a HIWIN Articulated Robot-RA605 model and

Algorithm 1 FCMAC-T-S Controller Algorithm

Step 1: Divide the state space according to the input domain. Complete mapping from the input domain to the state space.

Step 2: Quantify each state variable S_i into several blocks according to the quantization level and the generalization parameter.

Step 3: For all blocks in each layer, calculate the basis function $\mu_{B_{il}}$.

Step 4: Calculate the activation strength $A_l(\boldsymbol{s})$ for each hypercube. The values of weights in the hypercube are calculated by:

$$w_l = p_0^l s_0 + p_1^l s_1 + p_2^l s_2 + \cdots + p_i^l s_i + \cdots + p_N^l s_N$$

Step 5: Calculate the final output Θ

$$\Theta = \frac{\sum_{l=1}^{K} A_l(\boldsymbol{s}) w_l(\boldsymbol{s})}{\sum_{l=1}^{K} A_l(\boldsymbol{s})}$$

Step 6: According to the difference between the desired output and the actual output, update the linear parameters using the gradient descent method.

Step7: When learning is complete, repeat Steps 1-5 to calculate the actual output.

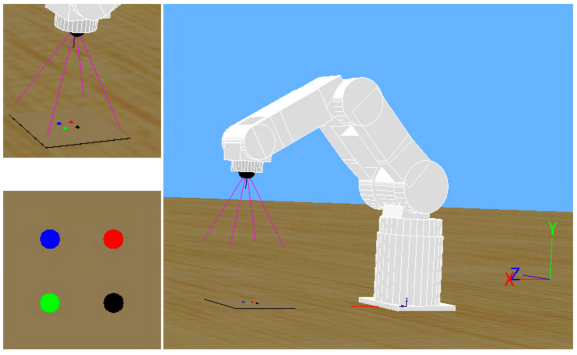

FIGURE 7. Simulation of the HIWIN RA605 robotic manipulator.

a camera that was mounted on the end-effector of the manipulator and four balls as the target. The camera provided color images with a resolution of 640 × 480 pixels. Four balls of the same radius size (5mm) but different colors were placed on the floor. The real environment, as shown in Fig.8, included a HIWIN Articulated Robot-RA605, a high density (HD) camera that was mounted on the end-effector of the manipulator and an industrial PC. Four differently shaped target objects (triangle, circle, square and cross) were fixed on the workbench.

Four experiments were conducted to demonstrate the efficiency of the proposed method. Firstly, the synthetic feature recognition algorithm (S-FRA) was compared with the SVM-based feature recognition algorithm (SVM-FRA). The accuracy and precision of feature extraction for both methods was tested in a real environment. Secondly, the proposed

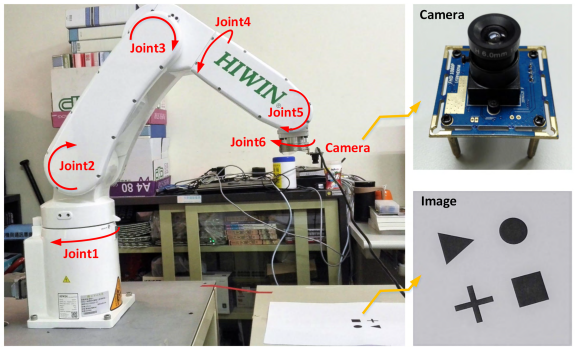

FIGURE 8. A real environment and the HIWIN RA605 robotic manipulator.

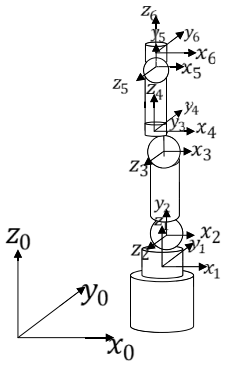

FIGURE 9. The HIWIN RA605 coordinate system.

control method that uses FCMAC-T-S and classical IBVS were tested using the simulation platform. The proposed control method that uses FCMAC-T-S and the original control method using CMAC were then compared in the real environment to demonstrate the efficiency of the proposed method. Finally, to illustrate the effect of generalization using FCMAC-T-S, the proposed method was compared with the original CMAC.

A. IMPLEMENTATION OF THE HIWIN RA605 KIINEMATICS MODEL

The HIWIN Articulated Robot-RA605 manipulator is an agile 6-axis jointed-arm robot that has six degrees of freedom and all six joints are rotational joints. The robot's coordinate system is s shown in Fig.9. The first three joints mainly affect the position of the end effector and the last three joints determine the posture of the end effector.

To ensure accurate visual servoing, the HIWIN RA605 kinematics model was correctly implemented in the simulation platform and the real environment. Using a homogeneous transformation matrix method that was proposed by Denavit and Hartenberg, the kinematics equations for the manipulator was established. The Denavit-Hartenberg (DH) convention for assigning coordinate frames uses four parameters to define the reference coordinate frame for each link in the HIWIN RA605 manipulator, as shown in Table. 1. The coordinate system for the J_{n-1} joint was rotated and translated

TABLE 1. HIWIN RA605 D-H parameters.

Joint	θ_i	d_i	a_i	α_i
1	θ_1	0.375	0.03	90
2	θ_2	0.340	0	0
3	θ_3	0	-0.04	-90
4	θ_4	0.338	0	90
5	θ_5	0.0865	0	-90
6	θ_6	0	0	0

TABLE 2. The transformation matrices for the HIWIN RA605.

joint	transformation matrices
${}^0_1T$	$R(Z,\theta_1)\cdot T(0.03,\ 0{,}0.375)\cdot R(X,90°)$
${}^1_2T$	$R(Z,\theta_2)\cdot T(0{,}0.34{,}0)$
${}^2_3T$	$R(Z,\theta_3)\cdot T(-0.04{,}0{,}0)\cdot R(X,-90°)$
${}^3_4T$	$R(Z,\theta_4)\cdot T(0,\ 0{,}338)\cdot R(X,90°)$
${}^4_5T$	$R(Z,\theta_5)\cdot T(0,\ 0.0865{,}0)\cdot R(X,-90°)$
${}^5_6T$	${}^5_6T = R(Z,\theta_6)$

to the coordinate system for the J_n joint. The transformation matrices are formulated in Table. 2.

The D-H parameters are substituted into the matrices

$$
{}^0_1\mathrm{T} = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & a_1\cos(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & a_1\sin(\theta_1) \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^1_2\mathrm{T} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & -d_2\sin(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & d_2\cos(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^2_3\mathrm{T} = \begin{bmatrix} \cos(\theta_3) & 0 & -\sin(\theta_3) & a_3\cos(\theta_3) - d_3\sin(\theta_3) \\ \sin(\theta_3) & 0 & \cos(\theta_3) & a_3\sin(\theta_3) - d_3\cos(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^3_4\mathrm{T} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ \sin(\theta_4) & 0 & -\cos(\theta_4) & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^4_5\mathrm{T} = \begin{bmatrix} \cos(\theta_5) & 0 & -\sin(\theta_5) & -d_5\sin(\theta_5) \\ \sin(\theta_5) & 0 & \cos(\theta_5) & d_5\cos(\theta_5) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^5_6\mathrm{T} = \begin{bmatrix} \cos(\theta_6) & -\sin(\theta_6) & 0 & 0 \\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

The multiplication of the coordinate transformation matrices ${}^0_6T = {}^0_1T(\theta_1){}^1_2T(\theta_2){}^2_3T(\theta_3){}^3_4T(\theta_4){}^4_5T(\theta_5){}^5_6T(\theta_6)$ is the forward kinematics equation, which shows the relationship between the object's coordinate system and the world coordinate system. The corresponding inverse matrix is derived using the forward kinematics equation.

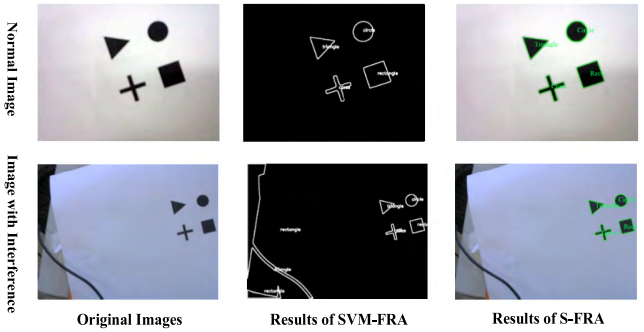

FIGURE 10. Image processing results.

B. IMAGE FEATURE RECOGNITION TESTS

In this experiment, four differently shaped target objects (triangle, circle, cross, rectangle) were stationary on the workbench and the camera was moved by the manipulator to observe the objects at a fixed depth, as shown in Fig.8.

The S-FRA algorithm proposed in this paper and the SVM-FRA algorithm are used to extract and recognize feature points. The tests were performed in environments with no interference and interference. When there is no interference, the image that is captured by the camera contains only four target objects. When there is interference, there are some interference items, as well as the target objects. The recognition accuracy and bias were recorded. The accuracy rate is calculated as:

$$P = TP/(TP + FP) \tag{25}$$

where True Positive(TP) = 4 indicates the number of feature points that are recognized correctly and False Positive (FP) represents the number of feature points that are recognized incorrectly. In order to achieve a more concrete comparison, the recognition process was repeated 100 times using both methods. Images were captured using different angles and positions.

In Fig.10, the recognition results are compared when there is no interference. The S-FRA and SVM-FRA identify the feature points correctly. Then Interference items (a piece of the cable and the edges of the workbench) appear in the image. It is obvious that more than four objects are recognized by the SVM-FRA. The interference items affect the accuracy of the SVM-FRA, but the S-FRA recognizes four targets objects accurately. To a certain degree, the S-FRA conducts a color-based compensation to avoid a "False Positive" and improve the accuracy of feature extraction.

As Fig.11 (a) shows, the accuracy rate for the S-FRA is significantly better than the SVM-FRA. In Fig.11 (b), the range of bias for the coordinates that are calculated using S-FRA is 0-9 pixels and the range of bias for the coordinates that are calculated using SVM-FRA is 0-20 pixels. The bias for S-FRA is significantly less than that for SVM-FRA. It is seen that S-FRA increases the accuracy of feature recognition and allows the controller to perform better.

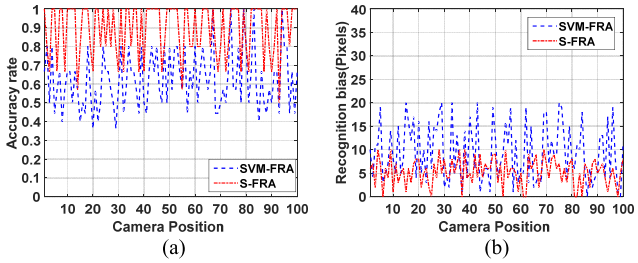

FIGURE 11. A comparison of different methods to extract feature points: (a) feature point recognition accuracy and (b) total feature point bias for a different camera position.

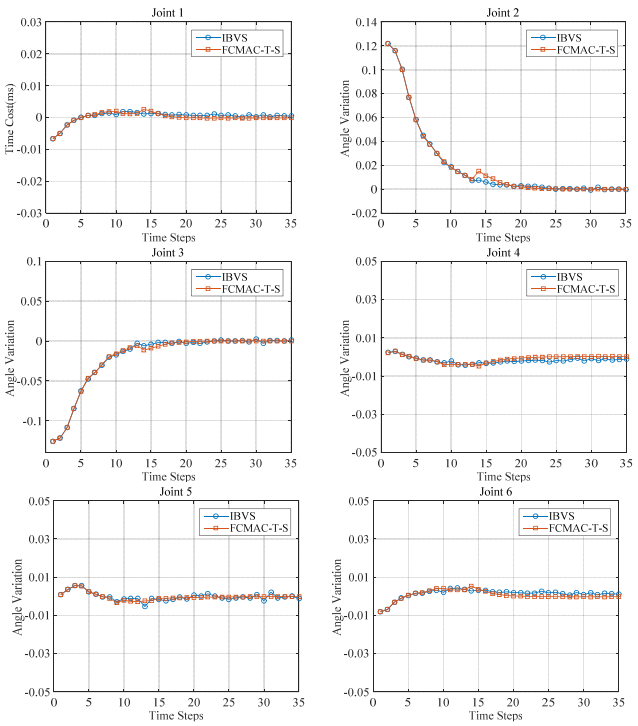

FIGURE 12. Comparisons of the joint velocity for the FCMAC-T-S and the IBVS.

C. SIMULATION TESTS

In the simulation platform, the coordinate of four feature points is recorded as $X_1 = [x_1, y_1]^T$, $X_2 = [x_2, y_2]^T$, $X_3 = [x_3, y_3]^T$, $X_4 = [x_4, y_4]^T$. The desired coordinates of the four feature points are $X_1^* = \left[x_1^*, y_1^*\right]^T$, $X_2^* = \left[x_2^*, y_2^*\right]^T$, $X_3^* = \left[x_3^*, y_3^*\right]^T$, $X_4^* = \left[x_4^*, y_4^*\right]^T$. Therefore, the feature error is

$$e\,(t) = \varepsilon - \varepsilon^* = \left[X_1 - X_1^*, X_2 - X_2^*, X_3 - X_3^*, X_4 - X_4^*\right]^T \tag{26}$$

To demonstrate the viability of the proposed controller, it is necessary to determine whether the proposed control model can imitate the IBVS process well. In the experiment, IBVS produces the training data for the FCMAC-T-S, which contains the error information $e\,(t)$ and the angle variation $\dot{\theta}$ for each joint of the manipulator. FCMAC-T-S learns the data that is produced by IBVS and then controls the manipulator.

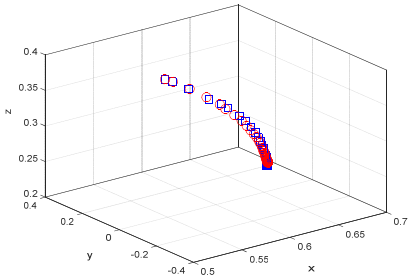

FIGURE 13. A comparison of the moving trajectory in the simulation.

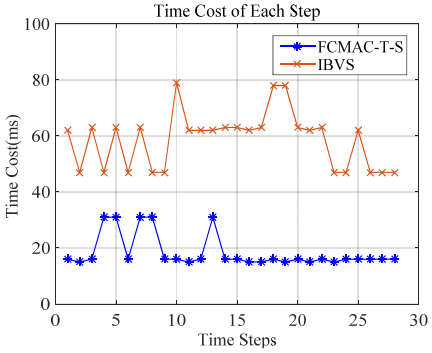

FIGURE 14. A comparison of time cost of the FCMAC-T-S with the IBVS.

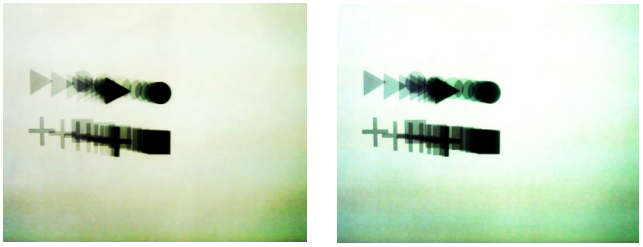

FIGURE 15. A comparison of vision servoing in a real environment: (a) a movement trajectory for the IBVS and (b) a movement trajectory for the FCMAC-T-S.

Comparisons of the angular variation $\dot{\theta}$ are shown in Fig.12. The movement trajectory of the end-effector for the IBVS and the FCMAC-T-S is shown in Fig.13. Fig.12 and Fig.13 show that the proposed controller that uses FCMAC-T-S can learn IBVS well and performs visual servoing independently and smoothly.

The pseudoinverse of Jacobian matrix $\boldsymbol{J_c^{-1}}$ and the inverse matrix of interaction matrix $\boldsymbol{L_\varepsilon^{-1}}$ both need to calculate repeatedly during the process of the IBVS, that is a very time-consuming process. Nevertheless, the FCMAC-T-S control algorithm directly maps an image feature error vector to a desired robot end-effector velocity, which omitted the calculation process of pseudoinverse matrix. Therefore, a comparison experiment between FCMAC-T-S and conventional IBVS is conducted to demonstrate time efficiency of the proposed method. A metrics of time cost is defined as the time consumed for each servoing cycle (a time step as shown in Fig. 14). Fig.14 shows that it takes about 65 milliseconds

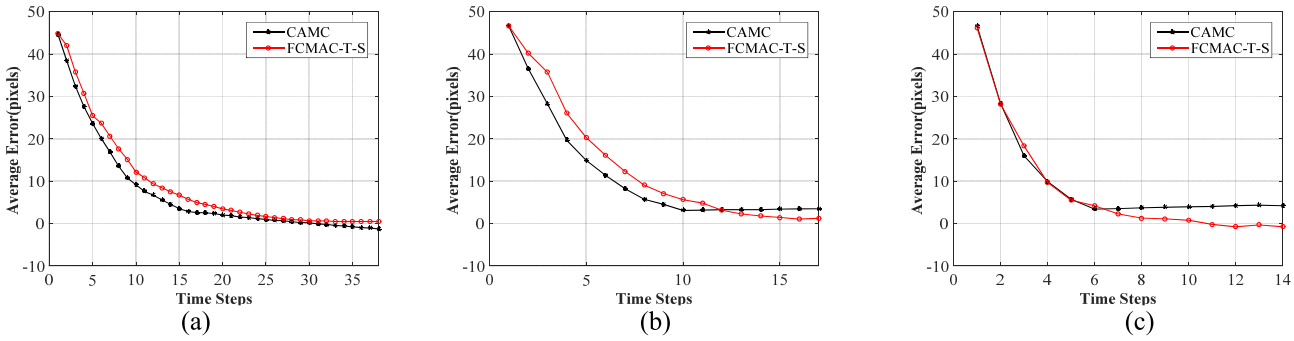

FIGURE 16. Comparisons of image feature error for the FCMAC-T-S and the CMAC for different servoing gains. (a) K= 0.4, (b) K=0.8 and (c) K=1.2.

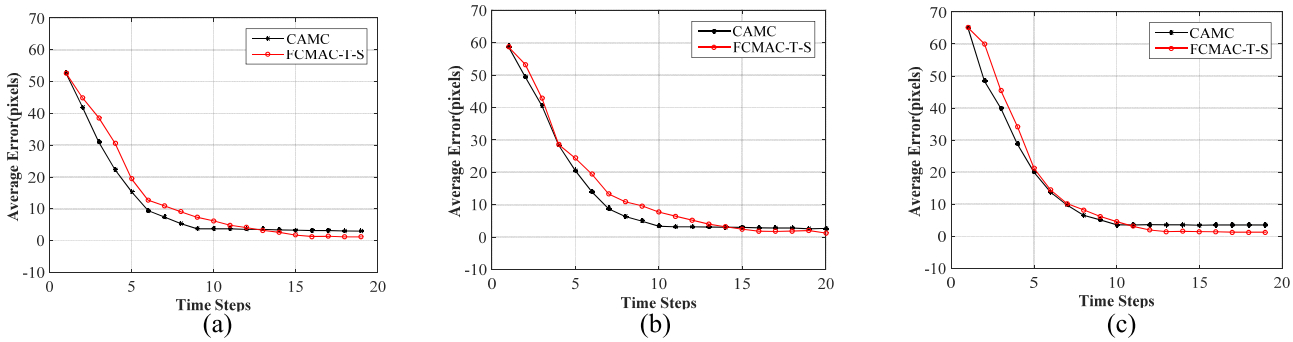

FIGURE 17. Comparisons of the generalization ability for the FCMAC-T-S and the CMAC.

to take one step in IBVS and the time is about 16 milliseconds in FCMAC-T-S. That means that time efficiency of the FCMAC-T-S is four times better than the IBVS.

D. A COMPARISON OF THE CONTROL AND THE REAL EXPERIMENT

To verify the efficiency and practicality of the proposed controller based on FCMAC-T-S, it was compared in a real environment with the IBVS and a controller that uses classical CMAC. Four differently shaped target objects (triangle, circle, cross, and rectangle) were fixed on the workbench. During the tests, the camera for the robot end-effector maintained a fixed depth to the image plane. Using visual servoing, the camera was moved from its initial position to the desired position.

The trajectories of the continuous movement towards the desired position are shown in Fig.15. Using the learning mechanism, the proposed controller that uses FCMAC-T-S can control the manipulator stably and smoothly and performs visual servoing as well as the IBVS.

The proposed controller that uses FCMAC-T-S and the controller that uses classical CMAC were compared to demonstrate the convergence and stability of the proposed method. Before the experiment, both controllers were trained using the training datasets that were generated by IBVS, for which the values of the servoing gain K were 0.4, 0.8 and 1.2. The average error between the actual output and the desired output was calculated to illustrate the servoing precision. In Fig.16, the average error for the controller that uses FCMAC-T-S converges to 0 more quickly than that for the controller that uses classical CMAC. When the servoing gain K is increased, the convergence and stability of the classical CMAC controller become worse. The proposed FCMAC-T-S performs better, in terms of convergence and stability.

E. GENERALIZATION EFFECT TEST

To verify the generalization of the proposed controller that uses FCMAC-T-S, real experimental tests were performed. In the workspace of the manipulator, eight points in the same plane were selected as the training points, for which the center was an endpoint and other points were start points. Eight training lines were formed from the start points to an endpoint. It is seen that the manipulator that is controlled using IBVS with a servoing gain $K = 0.8$ moves along the training lines and generates the training datasets. The FCMAC-T-S controller and the classical CMAC controller were also trained. After training, three random points adjacent to the training lines were selected as start points for visual servoing. As the error convergence curves for FCMAC-T-S and CMAC in Fig.17 shows, the proposed FCMAC-T-S controller has a better generalization ability than the classical CMAC controller. If the number of training data sources increases continuously, FCMAC-T-S allows visual servoing from any point in the workspace.

VI. CONCLUSION

The visual servoing actually consists of two stages: the feature extraction followed by the visual servoing control process. For the feature extraction, we proposed the synthetic feature recognition algorithm (S-FRA) which is based on a well-known SVM-based feature recognition algorithm but taking color attributes into consideration so as that the proposed method can increase the capability of robustness to noise disturbance. In the serving control process, which is actually the main theme of this work, we proposed an FCMAC-T-S to tackle the visual servoing problem. From the results of the experience, our method has the advantages of time-efficiency, accuracy, and fast stability over the conventional image-based visual servoing. Besides, the proposed controller with a T-S fuzzy framework is enabled to generate smoother outputs than the CMAC.

The proposed control method that uses an FCMAC-T-S model learns and mimics an ideal control law by approximating the nonlinear mapping between the variation in the image features and the motion of the manipulator. Compared with a classical image-based servoing method, the proposed controller allows visual servoing without the need for complex computation of an inverse matrix or a pseudoinverse matrix. In the viewpoint of computational complexity, the computational complexity of the proposed method is $O(m * n)$ only during the training process, and $O(1)$ during the servoing process; where m denoted training frequency and n is the size of training data. Future work will address several issues in the proposed visual servo controller. An initial coarse estimation of the interaction matrix can produce unstable results, especially at the beginning of the servoing, because some visual features might not be within the camera's field of view. If a better reference control law is used as the learning object, the proposed controller could be more properly trained and updated for the same scenario settings for manipulator applications. The proposed control scheme will be extended a single controller by adding a patronizing compensator that provides fine online tuning when there exist external disturbances.

REFERENCES

[1] K. Hashimoto, "A review on vision-based control of robot manipulators," *Adv. Robot.*, vol. 17, no. 10, pp. 969–991, Dec. 2003.
[2] F. Chaumette, "Potential problems of unstability and divergence in image-based and position-based visual servoing," in *Proc. Eur. Control Conf. (ECC)*, Karlsruhe, Germany, Aug./Sep. 1999, pp. 4549–4554.
[3] V. Lippiello, B. Siciliano, and L. Villani, "Position-based visual servoing in industrial multirobot cells using a hybrid camera configuration," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 73–86, Feb. 2007.
[4] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *IEEE Robot. Autom. Mag.*, vol. 13, no. 4, pp. 82–90, Dec. 2006.
[5] C. Y. Tsai, C. C. Wong, C. J. Yu, C. C. Liu, and T. Y. Liu, "A hybrid switched reactive-based visual servo control of 5-DOF robot manipulators for pick-and-place tasks," *IEEE Syst. J.*, vol. 9, no. 1, pp. 119–130, Mar. 2015.
[6] C. Teulière and E. Marchand, "A dense and direct approach to visual servoing using depth maps," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1242–1249, Oct. 2014.
[7] B. Driessen, F. Liefhebber, T. T. Kate, and K. Van Woerden, "Collaborative control of the MANUS manipulator," in *Proc. IEEE Int. Conf. Rehabil. Robot. (ICORR)*, Chicago, IL, USA, Jun./Jul. 2005, pp. 247–251.
[8] W.-C. Chang, M.-Y. Cheng, and H.-J. Tsai, "Image feature command generation of contour following tasks for SCARA robots employing image-based visual servoing—A PH-spline approach," *Robot. Comput.-Integr. Manuf.*, vol. 44, pp. 57–66, Apr. 2017.
[9] W. G. Pence, F. Farelo, R. Alqasemi, Y. Sun, and R. Dubey, "Visual servoing control of a 9-DoF WMRA to perform ADL tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Saint Paul, MN, USA, May 2012, pp. 916–922.
[10] H. Pandya, K. M. Krishna, and C. V. Jawahar, "Discriminative learning based visual servoing across object instances," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 3447–3454.
[11] F. Chaumette and S. Hutchinson, "Visual servo control. II. Advanced approaches [Tutorial]," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 109–118, Mar. 2007.
[12] C. C. Cheah, S. P. Hou, Y. Zhao, and J. J. E. Slotine, "Adaptive vision and force tracking control for robots with constraint uncertainty," *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 3, pp. 389–399, Jun. 2010.
[13] M. Keshmiri, W.-F. Xie, and A. Mohebbi, "Augmented image-based visual servoing of a manipulator using acceleration command," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5444–5452, Oct. 2014.
[14] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, no. 2, pp. 135–156, Jan. 1992.
[15] N. G. Adar, A. E. Tiryaki, and R. Kozan, "Real time visual servoing of a 6-DOF robotic arm using fuzzy-PID controller," *Acta Phys. Polonica A*, vol. 128, no. 2B, pp. B-348–B-351, Jul. 2015.
[16] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116–132, Jan./Feb. 1985.
[17] I. Siradjuddin, L. Behera, T. M. McGinnity, and S. Coleman, "Image-based visual servoing of a 7-DOF robot manipulator using an adaptive distributed fuzzy PD controller," *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 2, pp. 512–523, Apr. 2014.
[18] Z. H. Xiu and G. Ren, "Stability analysis and systematic design of Takagi–Sugeno fuzzy control systems," *Fuzzy Sets Syst.*, vol. 151, no. 1, pp. 119–138, Apr. 2006.
[19] J. S. Albus, "A new approach to manipulator control: The cerebellar model articulation controller (CMAC)," *J. Dyn. Syst., Meas., Control*, vol. 97, no. 3, pp. 220–227, Sep. 1975.
[20] J. S. Albus, "Data storage in the cerebellar model articulation controller (CMAC)," *J. Dyn. Syst., Meas., Control*, vol. 97, no. 3, pp. 228–233, Sep. 1975.
[21] R. J. Wai and P. C. Chen, "Robust neural-fuzzy-network control for robot manipulator including actuator dynamics," *IEEE Trans. Ind. Electron.*, vol. 53, no. 4, pp. 1328–1349, Jun. 2006.
[22] Z. Miljković, M. Mitić, M. Lazarević, and B. Babić, "Neural network Reinforcement Learning for visual control of robot manipulators," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1721–1736, Apr. 2013.
[23] H. Shi, X. Li, K. S. Hwang, W. Pan, and G. Xu, "Decoupled visual servoing with fuzzy Q-learning," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1 pp. 241–252, Jan. 2018.
[24] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[25] S. Suzuki, "Topological structural analysis of digitized binary images by border following," *Comput. Vis. Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, Apr. 1985.
[26] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

WEI PAN received the Ph.D. degree from Northwestern Polytechnical University, China, in 2008. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University, and also a Visiting Scholar with the Electrical Engineering Department, National Sun Yat-sen University, Taiwan. His research interests include intelligent robots, machine learning, decision support systems, and multi-agent systems.

MENGYANG LYU received the B.S. degree from the School of Software and Microelectronics, Northwestern Polytechnical University, China, in 2011, where he is currently pursuing the master's degree with the School of Computer Science. His research interests include intelligent control, machine learning, electrical engineering, and embedded system design.

KAO-SHING HWANG (M'93–SM'09) received the M.M.E. and Ph.D. degrees in electrical and computer engineering from Northwestern University, Evanston, IL, USA, in 1989 and 1993, respectively. He was with National Chung Cheng University, Taiwan, from 1993 to 2011. He was the Deputy Director of the Computer Center from 1998 to 1999, the Chairman of the Electrical Engineering Department from 2003 to 2006, and the Director of the Opti-Mechatronics Institute, National Chung Cheng University, from 2010 to 2011. He is currently a Professor of the Electrical Engineering Department, National Sun Yat-sen University, and also an Adjunct Professor with the Department of Healthcare Administration and Medical Informatic, Kaohsiung Medical University, Taiwan. His research interests include methodologies and analysis for various intelligent robot systems, machine learning, embedded system design, and ASIC design for robotic applications. He is a fellow of the Institution of Engineering and Technology.

MING-YI JU (M'10) received the B.S. degree in electrical engineering from Tatung University, Taipei, Taiwan, in 1993, and the Ph.D. degree from the Electrical Engineering Department, National Chung Cheng University, Chia-Yi, Taiwan, in 2001. He was an Assistant Professor with the Department of Information Engineering, I-Shou University, Kaohsiung, Taiwan, from 2003 to 2006. Since 2006, he has been with the Department of Computer Science and Information Engineering, National University of Tainan, Tainan, Taiwan, where he is currently an Associate Professor. His research interests include intelligent robotic systems, robot vision, soft computing, and optimization.

HAOBIN SHI received the Ph.D. degree from Northwestern Polytechnical University, China, in 2008. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University, and also a Visiting Scholar of the Electrical Engineering Department, National Sun Yat-sen University, Taiwan. 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.