Appearance-based Robot Visual Servo via a Wavelet Neural Network

Qingjie Zhao, Zengqi Sun, Fuchun Sun, and Jihong Zhu

Abstract: This paper proposes a robot visual servo approach based on image appearance and a wavelet function neural network. The inputs of the wavelet neural network are changes of image features or the elements of image appearance vector, and the outputs are changes of robot joint angles. Image appearance vector is calculated by using eigen subspace transform algorithm. The proposed approach does not need a priori knowledge of the robot kinematics, hand-eye geometry and camera models. The experiment results on a real robot system show that the proposed method is practical and simple.

Keywords: Eigen transform, image appearance, visual servo, wavelet neural network.

1. INTRODUCTION

Robot visual servo with one or more CCD cameras has received increasing attention in recent years. The topic spans several disciplines including robotics, computer vision and control theory. There are still many difficult problems both in theory researches and in applications, such as the problems of feature extraction, system calibration and servo strategy.

Various neural network approaches have been studied for the control of visually guided robots. Miller [1] proposes a learning control approach that utilizes a CMAC (Cerebellar Model Articulation Controller) neural network to track an object on a conveyor based on an eye-on-hand configuration. In [2,3], self organizing neural networks are proposed to learn the inverse kinematics of a visual-robotic system. Hashimoto [4] simulates the use of neural networks for image-based visual servo. They relate image data to desired changes of robot joint angles based on an eye-on-hand system. Rosenblum [5] uses a radial basis function network in visual autonomous road following. Wells [6,7] proposes neural servo method with several kinds of image features. Lee [8] uses a neural network to control a mobile manipulator to track given trajectories in a workspace. Spiking neural networks are utilized in [9] to control autonomous mobile robots. Wen Yu [10] uses a RBF neural network to compensate the unknown gravity and friction for PD-like visual servo.

On the other hand, in robot visual servo local geometric characteristics are usually used as image features, such as points, lines or other geometric elements [4,5,11-13]. Point features appear to be the most common, which correspond to corners [4], holes or region centers [11], or specially designed points on targets [12,13]. These local geometric features can not always be obtained reliably because of surface reflectance, illumination, and other disturbing factors, and it is difficult to extract or match local geometric features. Moreover, artificial marks are not always practical in real environments. Therefore, some global image features are introduced such as Fourier descriptors, geometric moments [6,7], and appearance from subspace methods [14,15]. In [15], we use image features from eigen subspace transform to produce fictitious robot motions.

Based on the ideas of eigen image features and wavelet neural networks in [15], this paper proposes a practical robot visual servo method, in which our goal is to control a robot moving to a desired position according to vision information, instead of producing fictitious robot motions. The difference between our approach and the others is that we combine image appearance and a wavelet function neural network in robot visual servo system. Experiments are carried out on a real robot-vision system.

The next section briefly introduces the problems of image appearance, wavelet neural networks and learning algorithms, and will prove that the wavelet neural network is superior to the standard Sigmoid neural network by an intense nonlinear example. Section 3 describes the approach of vision-based robot servo based on a wavelet neural network. In Section 4, experiment results are shown. Finally Section 5 gives a conclusion to this paper.

Manuscript received January 22, 2006; revised December 30, 2006, December 25, 2007, and February 14, 2008; accepted April 1, 2008. Recommended by Editor Jae-Bok Song. This work was supported by the National Natural Science Foundation of China: 60772063.

Qingjie Zhao is with Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: zhaoqj@bit.edu.cn).

Zengqi Sun, Fuchun Sun, and Jihong Zhu are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mails: {Szq-dcs, fcsun, jhzhu}@mail.tsinghua.edu.cn).

2. SEVERAL PROBLEMS

2.1. Image appearance

Just as in [15], suppose we acquire P sample images with size $q_1 \times q_2$. Each image is expressed as a N-dimension random vector $F_p (p = 1, 2, \cdots, \mathrm{P})$, $\mathrm{N} = q_1 \times q_2$. The average image $\bar{F}$ can be estimated from the P sample images by using $\bar{F} = \mathrm{P}^{-1} \sum_{p=1}^{\mathrm{P}} F_p$. The image difference matrix is defined as:

$$A = (F_1 - \bar{F} \quad F_2 - \bar{F} \quad \cdots \quad F_{\mathrm{P}} - \bar{F}). \tag{1}$$

In order to reduce the computing burden, we substitute $A^{\mathrm{T}} A$ for AA^{T} in the formula of covariance matrix $C = (AA^{\mathrm{T}})/\mathrm{P}$, that is, $C = (A^{\mathrm{T}} A)/\mathrm{P}$, where T denotes transfer operator.

We can calculate the eigen-values of the matrix C, and the M largest eigenvalues $\lambda_i (i = 1, 2, \cdots, \mathrm{M})$ and the corresponding eigenvectors $u_i (i = 1, 2, \cdots, \mathrm{M})$.

These M eigenvectors form an orthogonal basis for the original image set. A dimension reduction is achieved by using $\mathrm{M} (\ll \mathrm{N})$ eigenvectors.

The transform matrix is

$$U = (u_1 \; u_2 \; \ldots \; u_{\mathrm{M}}). \tag{2}$$

An image F_{i} can be represented by image appearance ξ as

$$\xi = U^{\mathrm{T}} (F_i - \bar{F}), \quad \xi = (\xi_1 \quad \xi_2 \quad \cdots \quad \xi_{\mathrm{M}})^{\mathrm{T}}. \tag{3}$$

The image appearance ξ is a vector, whose M elements are called image features. From the P image appearance vectors related to the above P sample images, the average appearance $\bar{\xi}$ can be proved to be zero.

2.2. Neural network

In robot visual servo, wavelet neural networks [16,17], inspired by feed-forward neural networks and wavelet decompositions, can be used to learn the nonlinear relation between image data and control signals to change joint angles, not needing the robot kinematics and camera models.

The wavelet function neural network used in this paper is similar to a feed-forward network except using a wavelet function $\psi(s) = (1 - s^2) \cdot \exp(-s^2/2)$ instead of a Sigmoid function in selecting the activation function of hidden layers, as shown in Fig. 1. In Section 2.4, we will show why we choose the wavelet neural network and the Levenberg-Marquardt algorithm.

The outputs of the neural network can be expressed

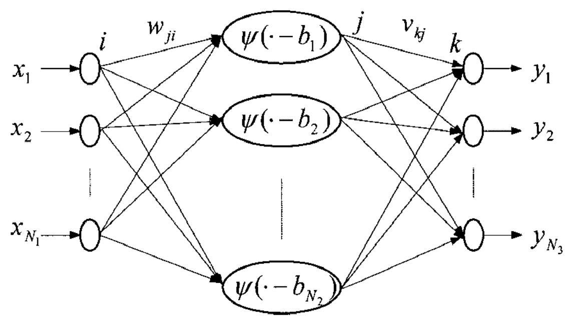


Fig. 1. Structure of the neural network.

as

$$y_k = \sum_{j=1}^{N_2} v_{kj} \psi\left(\sum_{i=1}^{N_1} w_{ji} x_i - b_j\right) + c_k, \; k = 1, 2, \cdots, N_3. \tag{4}$$

The nomenclature of the network is shown as following.

i: $1, \cdots, N_1$, node index for the input layer

j: $1, \cdots, N_2$, node index for the hidden layer

k: $1, \cdots, N_3$, node index for the output layer

N_1: number of input notes

N_2: number of hidden notes

N_3: number of output notes

w_{ji}: ith input weight of the jth note in the hidden layer, related to wavelet scaling

v_{kj}: jth input weight of the kth note in the output layer

b_j: parameter related to wavelet shifting

c_k: bias

$x_1, \cdots, x_{N_1}$: input variables

$y_1, \cdots, y_{N_3}$: output variables

2.3. Learning algorithm

The basic back-propagation (BP) algorithm adjusts the network parameters in the steepest descent direction, with the first order training speed. It is well known the method is very slow and inefficient, so based on it, comes many improved approaches. For example, momentum can be added to back-propagation learning by making parameter changes equal to the sum of a fraction of the last parameter change and the new change suggested by the back-propagation rule, which can avoid the network getting stuck in a shallow local minimum. In addition, an adaptive learning rate will attempt to keep the learning step size as large as possible while keeping learning stable.

However second order minimization methods are far superior to the first order ones with respect to learning time, especially in accurately approximating smooth functions. Newton's second order method often converges faster than the gradient method. Unfortunately, it is complex and expensive to compute the Hessian matrix. In this paper we use Quasi-Newton algorithm with Levenberg-Marquardt modifications [15] to approach second-order training speed without having to compute the Hessian matrix.

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k - \alpha_k (\boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{J}_k + \mu \boldsymbol{I})^{-1} \boldsymbol{J}_k^{\mathrm{T}} \boldsymbol{e}_k \tag{5}$$

In (5), $\boldsymbol{X}$ consists of all the weights and biases of the neural network, $\boldsymbol{J}$ is the Jacobian matrix, which contains first derivatives of the network errors with respect to the parameters, $\boldsymbol{e}$ is the error vector of image features, $\boldsymbol{I}$ denotes identity matrix, α is the learning rate, and $\mu \geq 0$ is a scalar coefficient.

2.4. An application example

The local characteristic of wavelets means that the wavelet neural network can properly approximate a function with intense change or discontinuity. These neural networks have strong abilities to map a nonlinear relation and extract some details of a signal. They are immune to noise for their ability of filtering low frequency. Compared with the standard Sigmoid networks, they converge much more rapidly.

In the following, we use a Sigmoid network and a wavelet network respectively to approximate the intense nonlinear function $y = \sin(2x)\cos(x)$, by using different learning algorithms. The Sigmoid function and the wavelet function used respectively are:

$$\psi_1(s) = 1/[1+\exp(-s)],$$

$$\psi_2(s) = (1-s^2)\exp(-s^2/2).$$

We first sample 61 points for x in [0, 6] with the interval 0.1, then we get 61 sample data. $[x_1, x_3, \ldots, x_{61}]$ are samples for learning, and $[x_2, x_4, \ldots, x_{60}]$ are samples for testing. The total iteration is 200. The results are shown in Fig. 2, Fig. 3 and Table 1, where

$$msel = \sum_i (y_{di} - y_i)^2 / 61, \quad i = 1,3,\cdots,61,$$

$$mset = \sum_j (y_{dj} - y_j)^2 / 60, \quad j = 2,4,\cdots,60.$$

We can find that, in the several cases, the wavelet neural network with Levenberg-Marquardt algorithm has the fasted convergence (Curve f in Fig. 2) and the best effect (the last case in Table 1). After 200 iterations, this wavelet neural network is closest to the ideal curve (see Fig. 3).

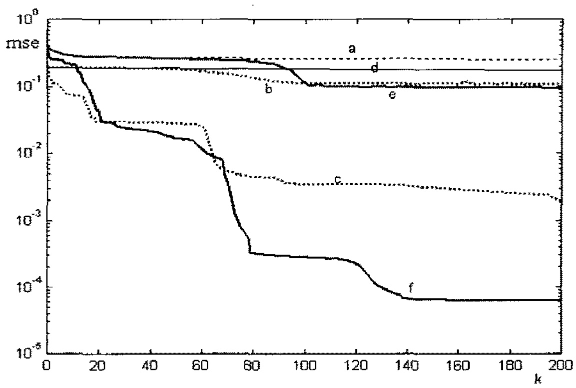


Fig. 2. Errors during the learning stage for different cases.

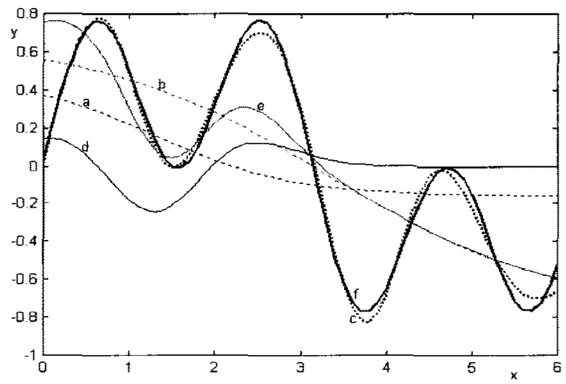


Fig. 3. Trained results of different cases.

3. ROBOT VISUAL SERVO VIA A WAVELET NEURAL NETWORK

According to vision information, a manipulator with six degrees of freedom is controlled to move based on a wavelet function neural network. The structure of the network is similar to that in Fig. 1, where the inputs are the changes of six image features: $\Delta\xi_1, \Delta\xi_2, \cdots \Delta\xi_6$ ($\Delta\xi_i = \xi_i - \xi_i^d$, $i = 1,\cdots,6$), and the outputs are the changes of six robot joint angles: $\Delta\theta_1, \Delta\theta_2, \cdots \Delta\theta_6$ ($\Delta\theta_i = \theta_i - \theta_i^d$, $i = 1,\cdots,6$). ξ_i^d and $\theta_i^d (i = 1,\cdots,6)$ are image features and robot joint angles at the desired position. Feature selection for control should consider not only the requirement

Table 1. Results of both networks with different learning algorithms.

Function	Learning algorithm	*msel*	*mset*	Learning (See Fig. 2)	Results (See Fig. 3)
Sigmoid	BP+momentum	0.2498	0.2542	Curve a	Curve a
Sigmoid	BP+adaptive rate	0.1108	0.1071	Curve b	Curve b
Sigmoid	LM	0.0021	0.0016	Curve c	Curve c
Wavelet	BP+momentum	0.1723	0.1732	Curve d	Curve d
Wavelet	BP+adaptive rate	0.0993	0.0903	Curve e	Curve e
Wavelet	LM	0.000079	0.000044	Curve f	Curve f

of image recognition, but also the controllability, sensitivity and computational cost. Therefore, in the system with six degrees of freedom we extract six primary components from an image as image features.

The experiment process consists of two stages: learning stage and servo stage. The learning stage includes acquiring a set of images, calculating an eigen subspace transform matrix and image features, and learning the parameters of the wavelet neural network. The servo stage includes acquiring an image, transforming it into image appearance vector, computing robot joints by the trained wavelet neural network, communicating and controlling the robot to move, until the desired image appearance are achieved. The detailed steps of learning and servo are described as follows.

3.1. Learning stage

1) Control the robot moving to different positions near to the desired point, capture pictures, save the pictures and the corresponding robot joint angles.
2) Compute average image $\bar{\boldsymbol{F}}$, transform matrix $\boldsymbol{U}$, and image appearance $\xi=(\xi_1,\cdots,\xi_6)^{\mathrm{T}}$ at every sampling position, using the technique introduced in section 2, and save these results.
3) Given the desired image appearance and robot joint angle: $\xi^d=(\xi_1^d,\cdots,\xi_6^d)^{\mathrm{T}}$, $\boldsymbol{\theta}^d=(\theta_1^d,\cdots,\theta_6^d)^{\mathrm{T}}$, at every sampling position compute
$$\Delta\xi=\xi-\xi^d=(\Delta\xi_1,\cdots,\Delta\xi_6)^{\mathrm{T}},$$
$$\Delta\boldsymbol{\theta}=\boldsymbol{\theta}-\boldsymbol{\theta}^d=(\Delta\theta_1,\cdots,\Delta\theta_6)^{\mathrm{T}}.$$
4) Normalize the data of $\Delta\xi_1,\Delta\xi_2,\cdots\Delta\xi_6$ and $\Delta\theta_1,\Delta\theta_2,\cdots\Delta\theta_6$ to the range [-1,+1], using $x'=2(x-\min)/(\max-\min)-1$.
5) Train the weights and the biases of the wavelet neural network based on the normalized data.
6) Save the results into the computer, including the connection weights and biases of the neural network, and the maxima and the minima of input data and output data.

3.2. Servo stage

According to different feedback information, robot visual servo approaches can be classified into two kinds: imaged-based visual servo, in which an error signal measured directly in the image is mapped to actuator commands; and position-based visual servo, in which actuator commands are computed with respect to the 3-degree workspace [18]. According to this, our system is an imaged-based or feature-based visual servo system, which may reduce computational delay, eliminate the necessity for image interpretation and eliminate errors due to sensor modeling and camera calibration.

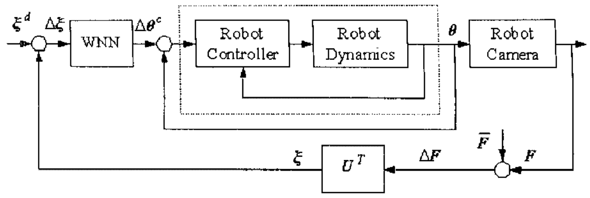


Fig. 4. Block diagram of robot visual servo.

The principle is given as Fig. 4. The trained wavelet neural network and the transform matrix are acquired during the learning stage. During the servo stage, the error of image appearance is transferred to the input of the wavelet neural network, and the desired control command is computed by the wavelet neural network. Then the controller gives a signal to motivate the robot until it gets to the goal position with a zero image feature error.

The steps of servo stage are concluded as follows.

1) Load $\boldsymbol{U}$, $\bar{\boldsymbol{F}}$, parameters of the neural network, the maxima and the minima.
2) Capture current image $\boldsymbol{F}$, read the corresponding robot joint position $\boldsymbol{\theta}=(\theta_1,\cdots,\theta_6)^{\mathrm{T}}$, and calculate image appearance $\xi=(\xi_1,\cdots,\xi_6)^{\mathrm{T}}=\boldsymbol{U}(\boldsymbol{F}-\bar{\boldsymbol{F}})$.
3) If $\xi\neq\xi^d$, then go to (4); else end.
4) $\Delta\xi=2(\Delta\xi-\Delta\xi_{\min})/(\Delta\xi_{\max}-\Delta\xi_{\min})-1$, $\Delta\xi=\xi-\xi^d$.
5) Compute $\Delta\boldsymbol{\theta}$ from the trained wavelet neural network.
6) $\Delta\boldsymbol{\theta}^c=(\Delta\boldsymbol{\theta}+1)(\Delta\boldsymbol{\theta}_{\max}-\Delta\boldsymbol{\theta}_{\min})/2+\Delta\boldsymbol{\theta}_{\min}$.
7) $\boldsymbol{\theta}^c=\boldsymbol{\theta}-\Delta\boldsymbol{\theta}^c$.
8) $\boldsymbol{\theta}^c$ is an input of the robot controller to control the robot moving to the desired position. During the robot moving, pictures are taken every 50ms, and the image appearance ξ is calculated.
9) If $\xi\neq\xi^d$, then go to (4); else end.

4. EXPERIMENT RESULTS

4.1. Experiment system

The experiment system includes an UP6 industry manipulator (see Fig. 5) with six revolute joints, XRC robot controller, human-robot cooperation unit, sensors and processing units, PC controller, and et al. A CCD camera is mounted on the end-effector and has a focal length of 6mm or so. Objects are put on the table in the workspace.

A Matrox Meteor frame grabber is connected with the personal computer by the PCI bus, used to acquire images with size 320×240.

The system provides function libraries for image capturing and robot operating. The soft modules involve image processing, feature computing, mathematical arithmetic, neural network class and

Fig. 5. The experiment system.

control algorithms.

4.2. Experiment results

Learning process is programmed in MATLAB, including computing the average image $\bar{\boldsymbol{F}}$ and the transform matrix $\boldsymbol{U}$, and training parameters of the wavelet function neural network. Servo program is written in Visual C++, including capturing images, computing image appearance by eigen space inverse transform, calculating joints by the trained neural network, communicating and controlling the robot to move. The sampling period is 50ms.

Figs. 6 and 7 are the results of an experiment with a toy plane. The goal of controlling is to make the error between current image appearance and desired image appearance less than a small numeral, which means the robot has arrived at the anticipant pose. The mean of squared errors of six image features is shown in Fig. 6, $mse=\Delta\xi^T\Delta\xi/6$. Fig. 7(a) is the image showing the anticipant relative pose between the robot end-effector and the scene; Fig. 7(b) corresponds to the initial relative pose; Fig. 7(c) is the 103th image during the robot visual servo; and Fig. 7(d) illuminates the terminal relative pose which shows the robot has approached to the anticipant position.

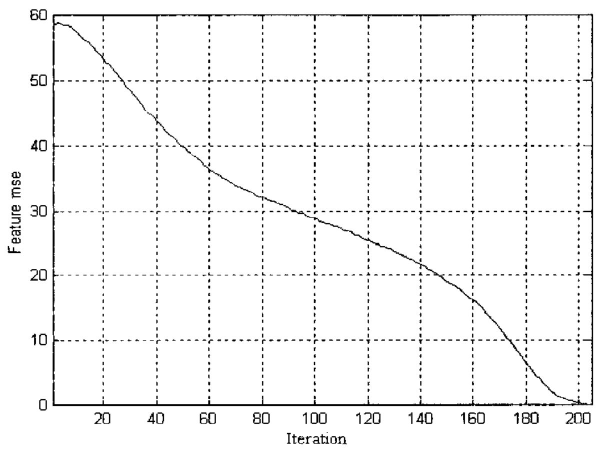
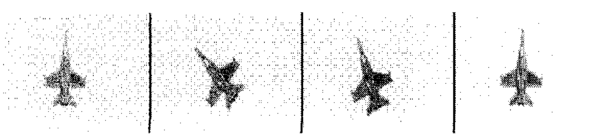


Fig. 6. The mean of squared errors of six image features.



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

Fig. 7. Images from the camera on the end-effector. (a) Anticipant image (b) Initial image (c) 103th image (d) Terminal image.

For different objects or a little more complex static backgrounds, the algorithms and the control process are all the same.

5. CONCLUSION

The paper proposes an appearance-based visual servo approach. A wavelet function neural network is utilized to learn the nonlinear relation between image features and control signals, where the image features are calculated by using eigen subspace transform algorithm. The advantages include: (1) Image features from eigen subspace transform can be utilized for different objects or complex backgrounds, not requiring special points or artificial marks. (2) The approach does not need any a priori knowledge of robot kinematics, hand-eye geometry and camera model. (3) The proposed approach can be easily used in practice, especially for structured environments.

REFERENCES

[1] W. T. Miller, "Sensor based control of robotic manipulators using a general learning algorithm," *IEEE Journal of Robotics Automation*, vol. 3, no. 2, pp. 157-165, 1987.

[2] R. M. Martinetz, H. J. Ritter, and K. J. Schulten, "Three-dimensional neural net for learning visuo-motor coordination of a robot arm," *IEEE Trans. on Neural Networks*, vol. 1, pp. 131-136, Mar. 1990.

[3] J. A. Walter and K. J. Schulten, "Implementation of self-organizing neural networks for visuo-motor control of an industrial robot," *IEEE Trans. on Neural Networks*, vol. 4, pp. 86-95, Jan. 1993.

[4] K. Hashimoto, T. Kimoto, and T. Ebine, "Manipulator control with image-based visual servo," *Proc. of IEEE Conf. Robotics and Automation*, pp. 2267-2272, 1991.

[5] M. Rosenblum and L. S. Davis, "An improved radial basis function network for visual autonomous road following," *IEEE Trans. on Neural Networks*, vol. 7, pp. 1111-1120, Sept. 1996.

[6] G. Wells, C. Venaille, and C. Torras, "Promising research vision-based robot servo using neural networks," *Image and Vision Computing*, vol. 14, pp. 715-732, 1996.

[7] G. Wells and C. Torras, "Selection of image features for robot servo using mutual information," *Proc. of IEEE Conf. Robotics and Automation*, pp. 2819-2826, 1998.

[8] C. Y. Lee, I. K. Jeong, I.-H. Lee, and J. J. Lee, "Motion control of mobile manipulator based on neural networks and error compensation," *Proc. of IEEE Int. Conf. Robotics and Automation*, pp. 4627-4632, 2004.

[9] H. Hagras, A. P. Cornish, M. Colley, V. Callaghan, and G. Clarke, "Evolving spiking neural network controllers for autonomous robots," *Proc. of IEEE Conf. Robotics and Automation*, pp. 4620-626, 2004.

[10] W. Yu, "Stability analysis of visual servoing with sliding-mode estimation and neural compensation," *International Journal of Control, Automation, and Systems*, vol. 4, no. 5, pp. 545-558, 2006.

[11] J. Feddema and O. Mitchell, "Vision-guided servo with feature-based trajectory generation," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 691-700, 1989.

[12] B. Espiau, F. Chaumette, and F. Rives, "A new approach to visual servo in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 313-326, 1992.

[13] E. Malis, F. Chaumette, and S. Boudet, "2-1/2-D visual servo," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238-250, 1999.

[14] S. K. Nayar, S. A. Nene, and H. Murase, "Subspace methods for robot vision," *IEEE Trans. on Robotics and Automation*, vol. 12, pp. 750-758, 1996.

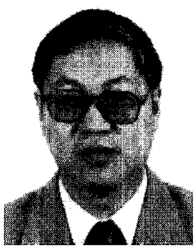
[15] Q. Zhao and Z. Sun, "Image-based robot motion simulation," *Optics Communications*, vol. 205, pp. 257-263, 2002,

[16] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Trans. on Neural Networks*, vol. 3, pp. 889-898, 1992.

[17] J. Zhang, G. G. Walter, and Y. Miao, "Wavelet neural networks for function learning," *IEEE Trans. on Signal Processing*, vol. 43, pp. 1458-1497, 1995.

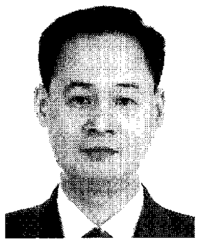
[18] P. Corke and S. Hutchinson, "Real-time vision, tracking and control," *Proc. of IEEE Conf. Robotics and Automation*, pp. 2267-2272, 2000.

Qingjie Zhao received the B.S. and M.S. degrees in Control Science and Technology, from China University of Petroleum in 1988, and Shandong University in 1996, respectively. She received the Ph.D. degree in Computer Science and Technology from Tsinghua University, in 2003. From 1988 to 1996 she was an Engineer in Jinan instrument factory. She is currently an Associate Professor in the School of Computer Science and Technology, Beijing Institute of Technology, China. Her current research interests include computing intelligence, machine vision, system theory and application. She is a Member of China Computer Federation.



Zengqi Sun received the B.S. degree from the Dept. of Automatic Control, Tsinghua University, in 1966, and the Ph.D. degree from Chalmas University of Technology, Sweden, in 1981. He is currently a Professor in the Dept. of Computer Science and Technology, Tsinghua University. He is also a Vice-President of the Chinese Association of Artificial Intelligence, Standing Council Member of the Chinese Association of Automation, the Chair of RoboCup China Committee, the Chair of Intelligent Automation Society of the Chinese Association of Automation, the Chair of the Robot Competition Committee of Chinese Association of Automation, Editors for Journal of China Science, Journal of Control Theory and Applications, Journal of Robotics, International Journal of Control, Automation and Systems. His current research interests include intelligent control, robotics, fuzzy systems, neural networks and evolution computing.

Fuchun Sun received the B.S. and M.S. degrees from Naval Aeronautical Engineering Academy in 1986 and 1989, and then worked there over four years. He received the Ph.D. degree from the Dept. of Computer Science and Technology, Tsinghua University in 1998. From 1998 to 2000 he was a Postdoctoral Fellow in the Dept. of Automation, Tsinghua University. He is currently a Professor in the Dept. of Computer Science and Technology, Tsinghua University. His research interests include intelligent control, neural networks, fuzzy systems, and robotics. He is a Member of the Technical Committee on Intelligent Control of the IEEE Control System Society, an associated editor of IEEE Trans. on Neural Netqworks and a Member of the Editorial Board of the International Journal of Soft Computing-A Fusion of Foundations, Methodologies and Applications.



Jihong Zhu received the B.S. degree from Jiangsu University of Science and Technology in 1990, the Ph.D. degree from Nanjing University of Aeronautics and Astronautics in 1995. From 1996 to 1997 he worked in Nanjing University of Aeronautics and Astronautics, and from 1998 to 1999 he was in Tsinghua University as a post-doctoral fellow. He is currently a professor in the Dept. of Computer Science and Technology, Tsinghua University. He is Youth Committee Member of Chinese Automation Association, Member of Intelligent Robot Institute of Chinese Artificial Intelligence Association. His research interests include robust control, filter of non-definite system.