Journal of Southeast University (English Edition) Vol.19 No.1 Mar. 2003 ISSN 1003—7985

3-D visual tracking based on CMAC neural network and Kalman filter

Wang Huaming[1] Luo Xiang[2] Zhu Jianying[1]

([1] College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)
([2] Department of Mechanical Engineering, Southeast University, Nanjing 210096, China)

Abstract: In this paper, the Kalman filter is used to predict image feature position around which an image-processing window is then established to diminish feature-searching area and to heighten the image-processing speed. According to the fundamentals of image-based visual servoing (IBVS), the cerebellar model articulation controller (CMAC) neural network is inserted into the visual servo control loop to implement the nonlinear mapping from the error signal in the image space to the control signal in the input space instead of the iterative adjustment and complicated inverse solution of the image Jacobian. Simulation results show that the feature point can be predicted efficiently using the Kalman filter and on-line supervised learning can be realized using CMAC neural network; end-effector can track the target object very well.

Key words: visual tracking; CMAC; neural network; Kalman filter

Vision is a useful robotic sensor since it mimics the human sense of vision to guide the end-effector to track and handle the target objects in the environment. Shirai and Inoue[1] first proposed that a visual feedback loop can be used to correct the position of a robot to increase task accuracy, but early visual sensing and manipulation are combined in an open-loop fashion, "looking" then "moving". Later, Hill and Park[2] introduced the concept "visual servoing"— machine vision providing closed-loop position control for a robot end-effector.

According to the space in which the error signal is defined, visual servoing can be categorized into two groups: position-based visual servoing (PBVS) and image-based visual servoing (IBVS). In PBVS, features are extracted from the image, and then used to compute a 3-D reconstruction of the Cartesian space. The error signal is computed in the Cartesian task space. The principal advantage of PBVS is that tasks can be described in Cartesian coordinates, and the prime disadvantage is its high calibration dependence. In IBVS, the error signal is defined in the image space, i.e. based on the image features. IBVS is less sensitive to errors in camera calibration, requires smaller computational effort and is more suitable to tasks where no prior model of the task is available[3].

In IBVS, two important problems must be solved. One is the feature tracking in the image plane, and the other is the mapping from the error signal to the control signal.

The measurement of the motion of the features on the image plane must be done continuously and quickly. The traditional method used to measure this motion is based on an optical flow technique called sum-of-squared-difference (SSD). While searching in whole plane is time-consuming, to decrease the search space, a pyramidal search scheme is presented. The pyramidal scheme reduces the time required for the computation of the SSD algorithm, but reliability can be sacrificed when the selected feature loses its tracking properties at the coarser image resolution[4]. In this paper, the Kalman filter and SSD algorithm are combined. the Kalman filter is used to predict the feature points of the end-effector and target object, and then image-processing windows in which the SSD algorithm is applied are established around the predicted positions while the area we are not interested in is discarded. With this method, we can heighten the image-processing speed and diminish searching time.

As the feature points of the end-effector and target object are obtained, error signal can be defined in the image space. Mapping from error signal to control signal should be implemented quickly to guarantee the real-time performance. Conventional method is confronted with inverse matrix calculation. Furthermore, the image Jacobian should be adjusted dynamically, so the calculation load is very large. Here we adopt the CMAC neural network to implement

Received 2002-08-29.
Foundation item: The National Natural Science Foundation of China (59990470).
Biographies: Wang Huaming (1973—), male, graduate; Zhu Jianying (corresponding author), male, professor, zjyao@nuaa.edu.cn.

the nonlinear mapping. CMAC does not require the prior knowledge of the target's movements in the 3-D space and can take into account unexpected events such as system disturbance, so it is very appropriate for real-time robot control problems. In the end, simulation results are provided.

1 Feature Tracking

1.1 SSD method

The SSD method is based on optical flow technique, which assumes the intensities around a feature point remain constant as that point moves across the image plane. The displacement of a point $p = (u, v)$ at the next sampling time to $p' = (u + \Delta u, v + \Delta v)$ is found by finding the displacement $\Delta p = (\Delta u, \Delta v)$, which minimizes the SSD measure:

$$e(p, \Delta p) = \sum_{W} [I(u+i, v+j) - I'(u+i+\Delta u, v+j+\Delta v)]^2 \tag{1}$$

where I and I' are the intensity functions from two successive images and W is the window centered about the feature point which makes up the feature template.

While searching in whole image plane is time-consuming, here, the Kalman filter and SSD algorithm are combined to search the feature point.

1.2 Kalman filter

The discrete-time Kalman filter can be defined by

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{D}\boldsymbol{w}_k \tag{2}$$

$$\boldsymbol{y}_k = \boldsymbol{C}\boldsymbol{x}_k + \boldsymbol{v}_k \tag{3}$$

where $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are the process noise sequence with covariance $\boldsymbol{Q}$ and the measurement noise sequence with covariance $\boldsymbol{R}$, respectively. The Kalman filter assumes zero mean Guassian process and measurement noise sequences.

We obtain the following recursion equations that calculate the linear-least-square estimate of $\boldsymbol{x}_{k+1}$ given $\boldsymbol{y}_1$ through $\boldsymbol{y}_{k+1}$:

Predictor step

$$\hat{\boldsymbol{x}}_{k+1|k} = \boldsymbol{A}\hat{\boldsymbol{x}}_{k|k} + \boldsymbol{B}\boldsymbol{u}_k \tag{4}$$

$$\boldsymbol{P}_{k+1|k} = \boldsymbol{A}\boldsymbol{P}_{k|k}\boldsymbol{A}^{\mathrm{T}} + \boldsymbol{D}\boldsymbol{Q}\boldsymbol{D}^{\mathrm{T}} \tag{5}$$

Filter step

$$\hat{\boldsymbol{x}}_{k+1|k+1} = \hat{\boldsymbol{x}}_{k+1|k} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{K+1} - \boldsymbol{C}\hat{\boldsymbol{x}}_{k+1|k}) \tag{6}$$

$$\boldsymbol{P}_{k+1|k+1} = \boldsymbol{P}_{k+1|k} - \boldsymbol{K}_{k+1}\boldsymbol{C}\boldsymbol{P}_{k+1|k} \tag{7}$$

In (6) and (7)

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1|k}\boldsymbol{C}^{\mathrm{T}}(\boldsymbol{C}\boldsymbol{P}_{k+1|k}\boldsymbol{C}^{\mathrm{T}} + \boldsymbol{R})^{-1} \tag{8}$$

where $\boldsymbol{K}$ is called the gain matrix; $\boldsymbol{P}$ is called the covariance of the prediction error.

2 Image Based Visual Servo Control

2.1 Image Jacobian

In IBVS system, the pose estimation is solved implicitly — if the current view of the object matches the desired view then the object must be in the desired relative pose[5]. Let $\boldsymbol{X}_{\mathrm{t}}$ represent coordinates of the end-effector in the task space, and $\dot{\boldsymbol{X}}_{\mathrm{t}}$ represents the corresponding velocity. Let $\boldsymbol{X}_{\mathrm{i}}$ represent coordinates of image feature points and $\dot{\boldsymbol{X}}_{\mathrm{i}}$ the corresponding velocities. The image Jacobian, $\boldsymbol{J}_{\mathrm{v}}$, is a linear transformation that maps end-effector velocity to velocities of image feature points,

$$\dot{\boldsymbol{X}}_{\mathrm{i}} = \boldsymbol{J}_{\mathrm{v}}(r)\dot{\boldsymbol{X}}_{\mathrm{t}} \tag{9}$$

The most common image Jacobian is based on the motion of points in the image. A complete deduction process is detailed in Ref.[6]. The Jacobian can also be determined by other methods: estimation, updated with predictive methods, or solved experimentally.

2.2 Visual servo controller

The state equation is obtained by discretizing (9) as

$$\boldsymbol{X}_{\mathrm{i}}(k+1) = \boldsymbol{X}_{\mathrm{i}}(k) + T\boldsymbol{J}_{\mathrm{v}}(k)\dot{\boldsymbol{X}}_{\mathrm{t}}(k) \tag{10}$$

where T is the sampling period of the vision system.

To make the image feature points of the end-effector consistent to the feature points of the target that can be stationary or moving, an objective function that places a cost on error in feature positions and a cost on providing control energy is created[7]:

$$F(k+1) = [\boldsymbol{X}_{\mathrm{i}}(k+1) - \boldsymbol{X}_{\mathrm{d}}(k+1)]^{\mathrm{T}}\boldsymbol{Q} \cdot [\boldsymbol{X}_{\mathrm{i}}(k+1) - \boldsymbol{X}_{\mathrm{d}}(k+1)] + \dot{\boldsymbol{X}}_{\mathrm{t}}^{\mathrm{T}}(k)\boldsymbol{L}\dot{\boldsymbol{X}}_{\mathrm{t}}(k) \tag{11}$$

where $\boldsymbol{Q}$ and $\boldsymbol{L}$ are the weight matrices on feature error and control input, respectively.

By minimizing (11) with respect to current control input $\dot{\boldsymbol{X}}_{\mathrm{t}}$, the control input is obtained:

$$\dot{\boldsymbol{X}}_{\mathrm{t}}(k) = -(T\boldsymbol{J}_{\mathrm{v}}^{\mathrm{T}}(k)\boldsymbol{Q}T\boldsymbol{J}_{\mathrm{v}}(k) + \boldsymbol{L})^{-1}T\boldsymbol{J}_{\mathrm{v}}^{\mathrm{T}}(k) \cdot \boldsymbol{Q}[\boldsymbol{X}_{i}(\mathrm{k}) - \boldsymbol{X}_{\mathrm{d}}(k+1)] \tag{12}$$

As Eq.(12) shows, $\boldsymbol{Q}$ and $\boldsymbol{L}$ allow the user to place more or less emphasis on the feature error or the control input; their selection will affect the stability and response of the tracking system while there is not a standard procedure for the selection of the elements of $\boldsymbol{Q}$ and $\boldsymbol{L}$. Moreover, because of the Jacobian's time-varying nature, it should be adjusted continuously, which increases the computation expense. Another disadvantage of the Jacobian is the presence of singularity, which will make the system with an inverse Jacobian control law unstable.

To avoid the inverse calculation and singularity of image Jocobian, the CMAC is used to implement the nonlinear relation. Convergence velocity of the CMAC neural network is much faster than that of artificial NN with the back propagation (BP) algorithm, especially when mapping a vector from a multi-dimensional space to a smaller space. Control solution can be gained only by training the CMAC with a small portion of samples. Compared with the IBVS shown above, system calibration and complicated computation are transformed into simple neural network mapping.

3 CMAC Controller Design

3.1 CMAC neural network

The CMAC neural network is a neural network that models the structure and function of the part of the brain known as the cerebellum, which controls the musculature system[8]. The CMAC is designed to provide motor control for robotic manipulators. The basic structure of the CMAC algorithm regarding its function learning capability is illustrated in Fig.1.The key portions of the CMAC algorithm are two mappings.

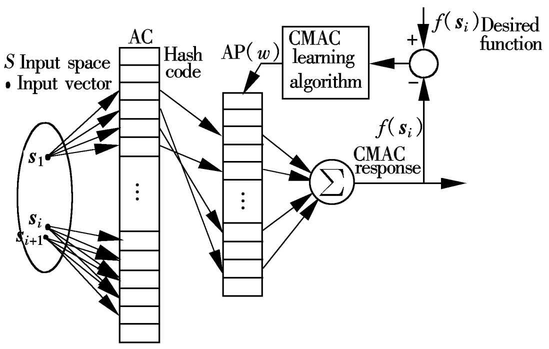

Fig.1 Architecture of a CMAC neural network

The two mappings realize the nonlinear relationship between input and output. The first one is the mapping from input space S to conceptual memory AC. The quantified input vector is defined by $\boldsymbol{s}_i = \{s_{i1}, s_{i2}, \cdots, s_{in}\}^{\mathrm{T}}$, which is mapped to c memory addresses. The mapped vector $\boldsymbol{R}_i$ (which is also called the receptive function) is defined by

$$\boldsymbol{R}_i = R(\boldsymbol{s}_i) = \{r_1(\boldsymbol{s}_i), r_2(\boldsymbol{s}_i), \cdots, r_c(\boldsymbol{s}_i)\}^{\mathrm{T}} \tag{13}$$

The number of addresses mapped for a sample of input space, c, is called the generalization size of the CMAC. As Fig.1 shows, similar inputs excite many of the same cells in the conceptual memory, thus produce similar outputs, while different inputs will produce different outputs.

The second one is the mapping from large conceptual memory space into a smaller physically realized memory space using hash code. Since the CMAC is used to learn the control of a robot arm, only a small portion of the input space corresponding to the desired trajectory is used, allowing for the input space to be hashed with a small possibility of collisions.

3.2 CMAC controller

Fig.2 is the block diagram of the visual servo the control system with the CMAC. In the diagram, the CMAC neural network and PD controller (the fixed PD controller) are combined together in parallel. A similar parallel implementation was first proposed by Miller for their robotic control application[9]. The total control signal to the end-effector is the sum of the CMAC response and the PD response. The learning algorithm uses the output of the PD controller to train the CMAC neural network. The goal of the learning algorithm is to minimize the PD control output, since the PD control output is directly proportional to the error signal. The CMAC will continue to learn until the system errors are within the noise levels of the system[10,11]. Thus, the learning algorithm is

$$\Delta \boldsymbol{w}_i = \beta(\boldsymbol{u}_{\text{total}} - \boldsymbol{u}_{\text{CMAC}}) = \beta \cdot \boldsymbol{u}_{\text{PD}} \tag{14}$$

where $\boldsymbol{u}_{\text{total}}$ is total control signal; $\boldsymbol{u}_{\text{CMAC}}$ is CMAC response and $\boldsymbol{u}_{\text{PD}}$ is PD output.

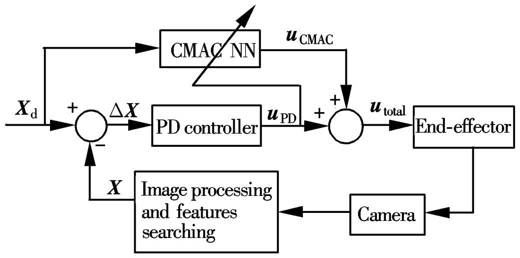

Fig.2 Block diagram of visual servo control using CMAC

The CMAC controller works as follows: the CMAC input space is composed of desired image features position vector $\boldsymbol{X}_{\text{d}}$ of the end-effector. The whole memory of the CMAC is initialized to zero. At the first run, CMAC response $\boldsymbol{u}_{\text{CMAC}}$ is zero, and the end-effector is controlled by $\boldsymbol{u}_{\text{PD}}$ entirely. At the same time, the corresponding weights in the CMAC are trained by $\boldsymbol{u}_{\text{PD}}$ as (14), so weights in memory are no longer zero, thus in the later time, $\boldsymbol{u}_{\text{CMAC}}$ and $\boldsymbol{u}_{\text{PD}}$ are summed to control the end-effector. The learning speed of the CMAC is very high. After several periods of learning, the end-effector's trajectory is nearly consistent with that of the target object. At this time, the end-effector is almost controlled by $\boldsymbol{u}_{\text{CMAC}}$, while $\boldsymbol{u}_{\text{PD}}$ is near zero. With the CMAC neural network, controlling while learning can be realized and real-time performance is ensured.

4 Simulation

4.1 Visual servo control system

The configuration of the simulated system is shown in Fig.3. The end-effector is actuated to move along the x, y, z axes to track the moving object. The objective of visual servo control is to map the error signal in image space to the control signal in x, y, z directions, which drive the end-effector to track the target object moving in a random trajectory with an average speed of 0.12 m/s. The workspace is a cuboid with $0 < x < 200$ mm, $-100 < y < 100$ mm, $0 < z < 200$ mm. To reflect the position change of feature points adequately in the image plane induced by the movement in workspace, two cameras are used. One is parallel to the x axis and the other to the y axis. This kind of configuration is based on x and y coordinates of object obtained from two cameras, respectively, which compose the input vector of the CMAC NN as described in section 4.2. In this system, a Sony XC-77RR black-and-white CCD camera is selected, which contains 768(H) × 493(V) sensor elements of size 11 μm(H) × 13 μm(V). The frame rate is 30 frame/s and the focus length is 50 mm. In this paper, perspective projection model is used.

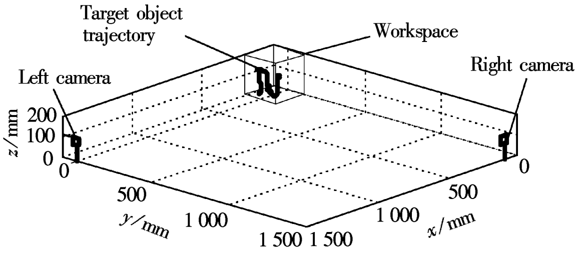

Fig.3 Configuration of simulated system (unit: mm)

Fig.4 shows the block diagram of the visual servo control system using a CMAC NN and Kalman filter. The Kalman filter is used to predict the visual features position of both the target object and end-effector, and then two windows of proper size are established around the predicted positions in which the image is processed; the SSD method is used to search the feature points. The Kalman filter can also serve to compensate the time delay caused by the image acquisition and processing and save us precious time for real-time implementation[12]. The principle of the CMAC controller is described in section 3.2.

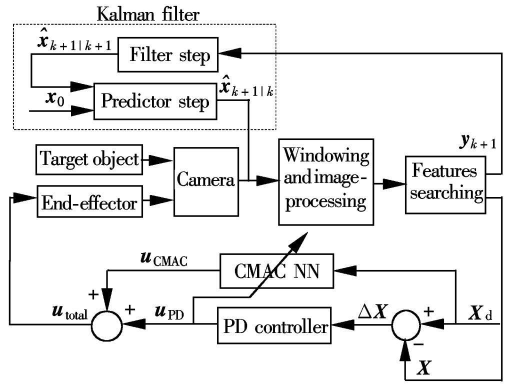

Fig.4 The visual servo control using CMAC and Kalman predictor

4.2 Simulation results

In this paper, the program is run under Matlab 5.3 or higher. Functions are written in C and are provided as CMEX-functions for speed.

The state vector for a single feature element consists of its position and velocity. We take the image coordinate x_{ld} of the target moving in random trajectory for example. The state equation is

$$\begin{bmatrix} \boldsymbol{p}_{k+1} \\ \boldsymbol{v}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_k \\ \boldsymbol{v}_k \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_k \tag{15}$$

where a_k is the random, time-varying acceleration and T is the sampling interval.

The measurement equation is

$$\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{v}_k \tag{16}$$

where $\boldsymbol{v}_k$ is random measurement noise.

Fig.5 shows the predicting process and the error between predicted position and measured position. As it shows, the curve of predicted position is near to that of measured position, so it is applicable to use a window of proper size around the predicted position to search the feature point.

After image feature points are searched, tracking a target object moving in random trajectory is simulated. Input to CMAC NN is the desired position vector of the end-effector image feature, i.e. position vector $\boldsymbol{X}_{\text{d}} = \{x_{\text{ld}}, x_{\text{rd}}, y_{\text{ld}}\}$ of the target object image feature, where $x_{\text{ld}}, x_{\text{rd}}, y_{\text{ld}}$ are the feature coordinates in the image planes of the left and right cameras. The generalization parameter c is 20 and the learning rate β is 0.5. Fig.6 shows the process of tracking a target object moving in random trajectory and the tracking error.

As shown in Fig.6, the weights can be adjusted on-line while off-line training is no longer needed, so the preparation of a large number of samples is avoided and the real-time performance is guaranteed. Another

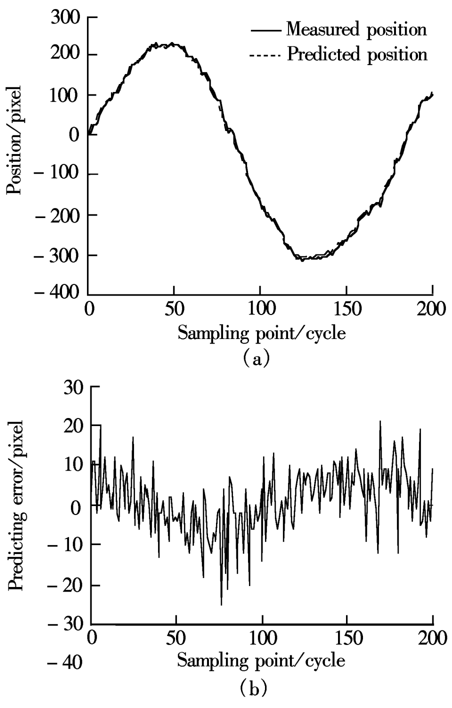

Fig.5 The predicting process and prediction error

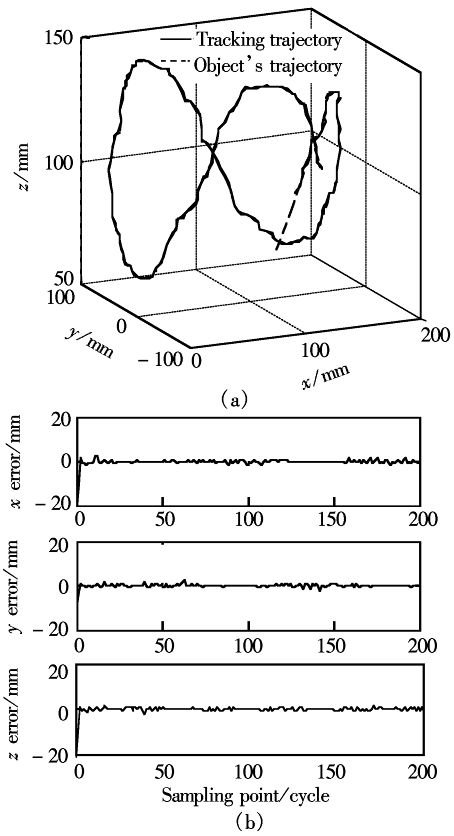

Fig.6 Tracking object moving in trajectory and the tracking error (unit: mm)

advantage is that the learning rate of CMAC NN is very high, and after several periods of learning and controlling, the tracking trajectory is almost consistent with the target object's trajectory. At this time, the output of the PD controller is near zero, which can be seen from the tracing errors in x, y, z directions, and the end-effector is almost controlled by the CMAC NN.

5 Conclusion

We use the Kalman filter to predict the positions of image features and then windows are created in which feature points are searched. Using a windowing technique, only small regions we are interested in are processed, so precious time is saved for real-time implementation. After feature points are obtained, error signal is defined in the image space, then the CMAC NN is used to map the error signal to the control signal to avoid the inverse solution of image Jacobian. Simulation results show that controlling while learning can be realized using the CMAC NN and the CMAC NN control scheme is proved to be a powerful tool for real-time nonlinear visual servo control applications.

References

[1] Shirai Y, Inoue H. Guiding a robot by visual feedback in assembling tasks [J]. *Pattern Recognition*, 1973,**5**(2):99 - 108.

[2] Hill J, Park W T. Real time control of a robot with a mobile camera [A]. In: *Proc of the 9th International Symposium on Industrial Robots* (*ISIR*) [C]. Washington DC, 1979.233 - 246.

[3] Kallio Pasi, Zhou Quan, Koivo Heikki N. Control issues in micromanipulation [A]. In: *International Symposium on Micromechatronics and Human Science*[C]. 1998.135 - 141.

[4] Nelson B J, Papanikolopoulos N P, Khosla P K. Visual servoing for robotic assembly[A]. In: Hashimoto, ed. *Visual Servoing-Real-Time Control of Robot Manipulator Based on Visual Sensory Feedback* [C]. River Edge, NJ: World Scientific, 1993.139 - 164.

[5] Corke P I, Hutchinson S A. Real-time vision, tracking and control [A]. In: *Proceedings of the 2000 IEEE International Conference on Robotics and Automation* [C]. San Francisco, CA, 2000.622 - 629.

[6] Hutchinson S A, Hager G, Corke P I. A tutorial on visual servo control [J]. *IEEE Trans on Robotics and Automation*, 1996,**12**(5):651 - 670.

[7] Vikramaditya Barmeshwar, Nelson Bradley J. Visually guided microassembly using optical microscopes and active vision techniques [A]. In: *IEEE International Conf on Robotics and Automation* [C]. Albuquerque, New Mexico, 1997. 3172 - 3177.

[8] Albus J S. A new approach to manipulator control: The cerebellar model articulation controller (CMAC)[J]. *Trans of the ASME*: *Journal of Dynamic Systems*, *Measurement and Con-*

trol, 1975,**97**(3):220 - 227.

[9] Miller W T, Glanz F H, Kraft L G. Application of a general learning algorithm to the control of robotic manipulators [J]. *International Journal of Robotics Research*, 1987,**6**(2): 84 - 98.

[10] Cetinkunt S, Donmez A. CMAC learning controller for servo control of high precision machine tools [A]. In: *Proc American Control Conf* [C]. San Francisco, CA, 1993. 1976 - 1980.

[11] Ku Sang Soon, Pinsopon Unnat, Cetinkunt Sabri, et al. Design, fabrication, and real-time neural network control of a three-degree-of-freedom nanopositioner [J]. *IEEE/ASME Transactions on Mechatronics*, 2000, **5**(3):273 - 280.

[12] Kara R, Wira P, Kihl H. *Hierarchical neural controller and adaptive Kalman filter for visual robot tracking tasks* [R]. University of Mulhouse, France: Technical Report EEA-TROP-TR-00-01, 2000.

基于 CMAC 神经网络和 Kalman 滤波器的三维视觉跟踪

王化明[1]　罗　翔[2]　朱剑英[1]

([1] 南京航空航天大学机电学院, 南京 210016)
([2] 东南大学机械工程系, 南京 210096)

摘　要　采用 Kalman 滤波器来预测图像特征点的位置,然后在其周围建立图像处理窗口,以达到减小特征点搜索区域及提高图像处理速度的目的.根据基于图像的视觉伺服 IBVS (image based visual servoing) 的原理,在视觉伺服控制环中加入 CMAC (cerebellar model articulation controller) 神经网络来实现从图像空间的误差信号向输入空间的控制信号的非线性映射,从而避免了图像雅可比矩阵的不断调整及其复杂的求逆过程.模拟结果表明:采用 Kalman 滤波器能有效预测特征点的位置,同时采用 CMAC 神经网络能实现在线有导师学习,末端执行器能较好地对目标物体进行跟踪.

关键词　视觉跟踪;CMAC; 神经网络; Kalman 滤波器

中图分类号　TP24