Nonlinear Visual Mapping Model for 3-D Visual Tracking With Uncalibrated Eye-in-Hand Robotic System

Jianbo Su, Yugeng Xi, Uwe D. Hanebeck, and G. Schmidt

***Abstract*—A new control scheme for uncalibrated robotic visual tracking problem is proposed that compromises the computational expenses of overall system with offline modeling and online control. A nonlinear visual mapping model for the uncalibrated hand-eye coordination is first proposed with an artificial neural network implementation. An online visual tracking controller is then developed together with a real-time motion planner. To improve the system performance, the control scheme is also integrated with a feedforward controller to compensate unknown object motions. Extensive simulations and experiments demonstrate the effectiveness of the proposed control scheme.**

***Index Terms*—Hand/eye coordination, image Jacobian matrix, neural networks, nonlinear mapping.**

I. Introduction

The essential problem in uncalibrated hand-eye coordination is how to map the error information in the sensing space of the vision system onto the control space of the robot without knowledge of the eye-hand relationship [1], [8]. Equivalently, it is to study how the camera model and the hand-eye relation model, which are static, globally nonlinear, and of high dimensions in visual sensing and robot control spaces, can easily be dealt with and realized in engineering sense. A well-known and widely accepted solution to this problem is based on the image Jacobian matrix introduced in [2]. The image Jacobian matrix is a local and linear approximation to the global nonlinear hand-eye relations and the camera model. Thus, it is time- and spatial variant and should be estimated online via smartly designed process [4], [7].

There have been many specific techniques developed so far for estimating image Jacobian matrix under different conditions [3], [6]. Online estimation of the image Jacobian matrix is computationally expensive for real-time control. Thus almost all successful applications of the uncalibrated hand-eye coordination are from systems fulfilling static tasks, such as grasping static object [6], pin-to-hole operation [5], or static positioning [3], etc. In addition, researches so far illustrate that the image Jacobian matrix-based methods cannot be used in the case that the object and the camera are moving simultaneously, i.e., case of dynamic tracking. The authors of [9] and [15] propose the ARMAX and ARX model for dynamic tracking and achieve good results in association with adaptive control law. However, the scheme actually deals with unknown object-related parameters, and at least rough calibration of the eye-hand relationship is still required for control.

There also have been ways to estimate the image Jacobian matrix in real-time control of robotic hand-eye coordination system, such as by using the tool of artificial neural networks (ANNs) [11], [12]. The offline training of the adopted ANN extraordinarily reduces computational complexity of online control in comparison with that of online estimation of image Jacobian matrix. However, due to the inherent limitation of the image Jacobian matrix that is linear and local, capacity of the ANN is far from being sufficiently exploited in this discipline. Thus, this scheme is only effective in limited cases.

The advantage of the ANN to approximate any nonlinear function with arbitrary preciseness should be sufficiently utilized in uncalibrated hand-eye coordination to achieve good performance, but the image Jacobian matrix model, which is local and linear, is weak in serving as the base to approximate the nonlinearities and complexities of camera models and hand-eye relations. Thus, a new nonlinear visual mapping model is first proposed with the ANN realization. A visual tracking controller is then designed based on the ANN to achieve robotic three-dimensional (3-D) tracking.

Traditional calibration-based coordination requires offline calibration to obtain the globally static but complex hand-eye relationships as well as the camera model for system control [13], [19]. Although online control is easy to implement, the offline calibration process is complicated and error-prone to realize in engineering [20]–[22]. The image Jacobian matrix-based uncalibrated coordination avoids the offline system calibration [16]. All modeling and control rely on online computation [23] and require intelligent estimation and control skills so that the model should be as simple as possible. This scheme is not very efficient in that a prior knowledge of the system model cannot be utilized and combined with the online control [17]. The new hand-eye coordination scheme proposed in this paper not only makes use of a prior knowledge of the system configuration but also adapt to changes of environments and applications via online learning. From the new scheme, a proper compromise between the computational complexities of offline training and online learning and control may be obtained. Since the whole computation burden for uncalibrated robotic hand-eye coordination is divided into two parts, better coordination performance and wide applications may be expected to achieve under this scheme.

This paper is organized as follows. Section II describes the problem to be investigated in this paper. Section III proposes the nonlinear visual mapping model, whereas Section IV addresses the control scheme of the system based on the neural network realization of the proposed model. A feedforward controller is suggested and discussed in this section to improve the system performance while dealing with system disturbance and environmental noises. Simulation results in Section V and the experiments results in Section VI demonstrate the effectiveness of the proposed scheme. Conclusions and future work are provided in Section VII.

II. Problem Description

For an eye-in-hand robotic system, the visual tracking problem studied in this paper is defined as moving the robot hand to locate the projection of a moving object in the image feature space as expected all the time. Fig. 1 shows the system configuration. The camera is fixed above the hand. The eye-hand relationship and the camera model are totally unknown. An object translates freely in the 3-D workspace. The vision controller does the motion planning in the image feature space and maps the planned motion to the robotic control space to instruct the manipulator servo controller. Consequently, translational tracking of the hand movement is generated until the object tracking process is stabilized and/or the object is grasped.

Fig. 2 shows the control and coordination structure we would adopt for the system depicted in Fig. 1. The desired state of the system is described in image plane. By a comparison with the true state of the system, the system errors are obtained. A sequence of movements of robot is planned from motion planning to eliminate the system errors. The nonlinear visual mapping model transforms the planned motion

Manuscript received January 24, 2002; revised July 31, 2002. This work was supported in part by the National Natural Science Foundation of China under Grant 69875010 and by the 2001 cooperative project between Deutsche Forschungsgemeinshaft (DFG) and the Ministry of Education (MoE) of China. This paper was recommended by Associate Editor I. Gu.

J. Su and Y. Xi are with the Department of Automation, Shanghai Jiaotong University, Shanghai, 200030, China (e-mail: jbsu@ sjtu.edu.cn; ygxi@ sjtu.edu.cn).

U. D. Hanebeck and G. Schmidt are with Institute of Automatic Control Engineering, Technical University of Munich, Munich, 80290, Germany (e-mail: uwe.hanebeck@ ei.tum.de; guenther.Schmidt@ ei.tum.de).

Digital Object Identifier 10.1109/TSMCB.2002.805813

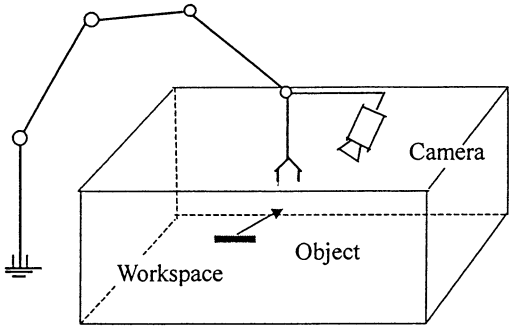

Fig. 1. Eye-in-hand system configuration.

from image plane to the robot servo control and yields the control instructions for the robot to move its hand as well as the hand-mounted camera. Since the control purpose is to drive the camera so that the object is located at an expected position in camera's image plane, the object motion is thus considered as the external disturbance to the camera motion in image plane. In this scheme, we can see that the nonlinear visual mapping model plays a critical role for the system coordination.

III. Visual Mapping Model

Define the image feature space $\Omega = span\{x, y, \lambda\}$, where x and y are the coordinates of the projected position of the object in the image plane, respectively. The third parameter λ is the feature measurement of the object in the image plane, which might be the length in the image for a rod-like object or the diameter or area in the image for a circular object. Note that here, the image feature space is extended in dimensionality by including physical feature of the object in image plane in addition to the dimensions for describing its projected position. With this definition, the dimension of the image feature space can be increased to be the same as, or even higher than, the degrees of freedom of robot control. Consequently, an invertible nonlinear visual mapping model may be obtained that can uniquely transform errors from image feature space to robot control space, or vice versa. Detailed explanations for the policy we adopt here will be given later. Since we restrict ourselves in this paper to 3-D translational movements of the robot hand, a 3-D image feature space is enough for this purpose.

In this paper, we use $\boldsymbol{p}$, $\boldsymbol{v}$, and $\boldsymbol{a}$ to denote position, velocity, and acceleration, respectively. The subscripts f, o, or c denote image feature, object, or camera, whereas the subscripts i or w denote the image feature space or robot control space, respectively. Hereby, we consider the mapping model from the robotic movement space to the image feature space. Suppose that at time instant k, the instant velocity (the changing rate) of the image features is $\boldsymbol{v}_{fi}(k)$. From [9], we have

$$\boldsymbol{v}_{fi}(k) = \boldsymbol{v}_{oi}(k) + \boldsymbol{v}_{ci}(k) \tag{1}$$

where $\boldsymbol{v}_{oi}(k)$ and $\boldsymbol{v}_{ci}(k)$ are the velocity components of the object in image feature space due to the object's translational motion and the camera's translational motion in robot control space, respectively.

The instant acceleration $\boldsymbol{a}_{fi}(k)$ of the object in image feature space at instant k is decided by the relative movement between the object and the camera. In the robot control space, the relative movement between the object and the camera at instant k is caused by the following three entities, i.e., the object's instant acceleration $\boldsymbol{a}_{ow}(k)$, the instant relative velocity $\boldsymbol{v}_{co,w}(k)$ between the object and the camera, and the camera's instant acceleration $\boldsymbol{a}_{cw}(k)$. Thus, the relative movement between the object and the camera can be decomposed to $\boldsymbol{a}_{ow}(k)$ with a zero object velocity plus a constant relative velocity $\boldsymbol{v}_{co,w}(k)$ and plus $\boldsymbol{a}_{cw}(k)$ with a zero camera velocity. Therefore, correspondingly, $\boldsymbol{a}_{fi}(k)$ can be decomposed into three parts, i.e.,

$$\boldsymbol{a}_{fi}(k) = \boldsymbol{a}_{oi}(k) + \boldsymbol{a}_{co,i}(k) + \boldsymbol{a}_{ci}(k) \tag{2}$$

where $\boldsymbol{a}_{oi}(k)$ is the acceleration component of the image features caused by $\boldsymbol{a}_{ow}(k)$. Since the motion parameters of the object is not measurable, we assume $\boldsymbol{a}_{ow}(k) = 0$ here, and thus, $\boldsymbol{a}_{oi}(k) = 0$ (true nonzero motions of the object will be regarded as external disturbances to the controller and compensation means will be discussed later). $\boldsymbol{a}_{co,i}(k)$ is the acceleration component of the image features caused by $\boldsymbol{v}_{co,w}(k)$ after a nonlinear perspective projection. The physical meaning is that even though $\boldsymbol{v}_{co,w}(k)$ is a constant, the motion of the projection point in the image plane is not a constant after the nonlinear perspective projection. Since $\boldsymbol{a}_{co,i}(k)$ can be uniquely decided by $\boldsymbol{p}_{fi}(k)$ and $\boldsymbol{v}_{fi}(k)$ in the image feature space, $\boldsymbol{a}_{co,i}(k)$ can be expressed as

$$\boldsymbol{a}_{co,i}(k) = g_{co}(\boldsymbol{p}_{fi}(k), \boldsymbol{v}_{fi}(k)). \tag{3}$$

In (2), $\boldsymbol{a}_{ci}(k)$ is the acceleration component of the image features caused by $\boldsymbol{a}_{cw}(k)$. Since the hand is exercising translational tracking, $\boldsymbol{a}_{cw}(k)$ should be exactly the same as the hand translational acceleration $\boldsymbol{a}_{hw}(k)$. Thus, $\mathbf{a}_{ci}(k)$ can be expressed as

$$\boldsymbol{a}_{ci}(k) = g_c(\boldsymbol{p}_{fi}(k), \boldsymbol{a}_{hw}(k)) \tag{4}$$

where the hand acceleration $\boldsymbol{a}_{hw}(k)$ can be obtained from the hand position coordinates $\boldsymbol{p}_{hw}(k)$ with a second-order differentiation (although errors may be involved by the differentiation, we will see later that this can be overcome in realization). Note that since the mapping from the robotic movement space to the image feature space is nonlinear, the functions $g_{co}(*,*)$ and $g_c(*,*)$ in (3) and (4), respectively, are nonlinear functions and are related. Putting (3) and (4) into (2), we obtain the mapping model

$$\boldsymbol{a}_{fi}(k) = g'(\boldsymbol{p}_{fi}(k), \boldsymbol{v}_{fi}(k), \boldsymbol{a}_{hw}(k)). \tag{5}$$

Equation (5) has the same dimensions of input and output. All coefficient matrices generally have full rank. Thus, the input-output relationship can then be exchanged to obtain an inverse mapping model from the image feature space to the robotic movement space, i.e.,

$$\boldsymbol{a}_{hw}(k) = g(\boldsymbol{p}_{fi}(k), \boldsymbol{v}_{fi}(k), \boldsymbol{a}_{fi}(k)). \tag{6}$$

This model, which is called the visual mapping model, nonlinearly relates the hand movement to the motion of the object in image feature space. Notice that this visual mapping model has the same dimensions for input and output spaces. Since, here, we only consider translational movements of the robot hand, which is 3-D, we also have a 3-D image feature space. If we have more degree-of-freedoms for hand movements, we should adopt more independent image feature parameters of the object, which leads to higher dimension of image feature space.

The characteristic that the visual mapping model has the same dimensions of input and output spaces is very important for practical realization and pursuing satisfactory tracking performance. In image Jacobian matrix model-based methods, there exists a problem of tracking singularity [6], [10], [18], which means that one or some of degree-of-freedoms for hand movements might become uncontrollable or uncontrolled during the tracking procedure. Deep research shows that this happens in the degenerate case when image features employed are not sufficient to reflect robot motion in one or some direction(s). If the dimension of image feature is equal to or higher than that of robot motion, then the tracking singularity may most probably not occur. Specifically, we choose the scheme that robot control space and the image feature

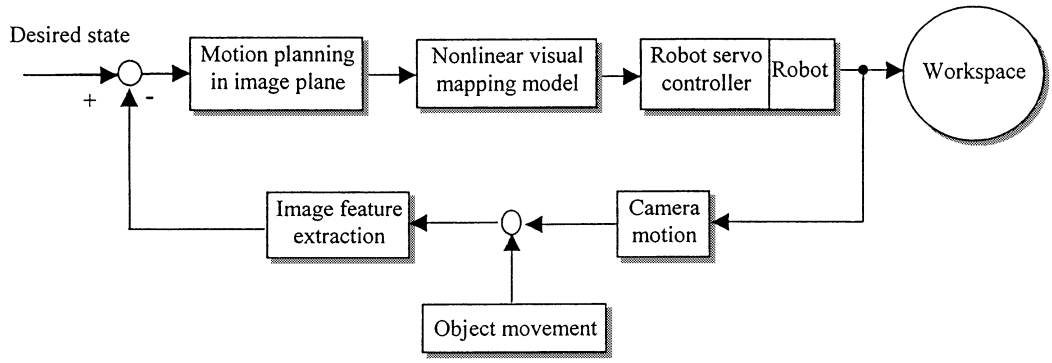

Fig. 2. Control and dynamic coordination structure of the system depicted in Fig. 1.

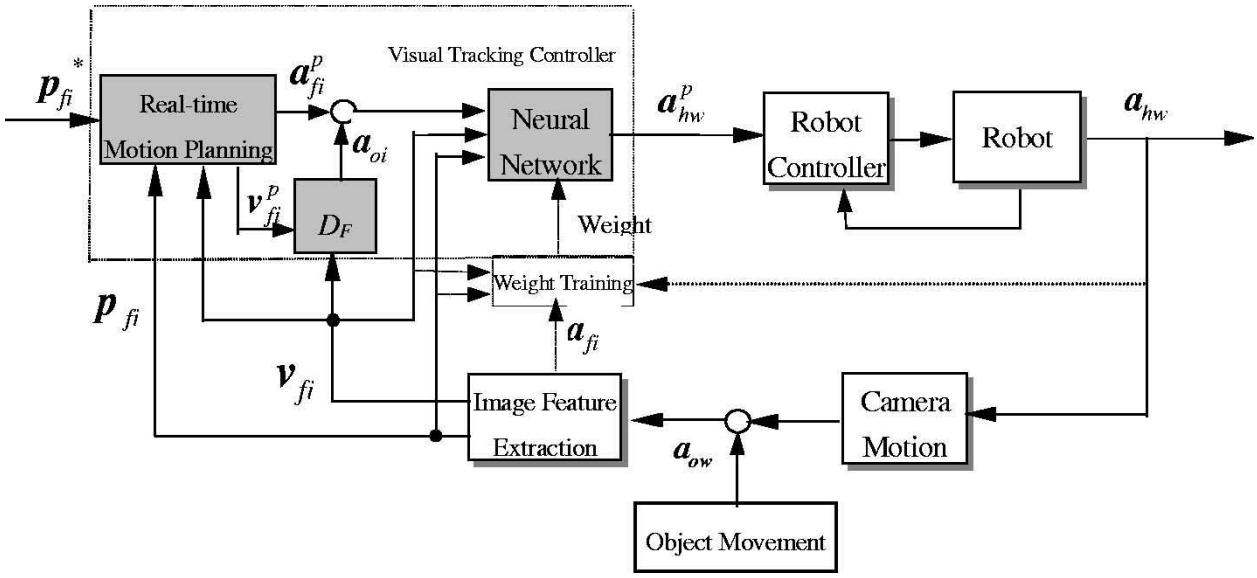

Fig. 3. Overall control system of the robotic 3-D visual tracking.

space are of the same dimensions. This scheme is taken into account when deriving the nonlinear visual mapping model, instead of being used as an additional modification in image Jacobian matrix- based methods. In this sense, the nonlinear visual mapping model is an extension of the image Jacobian matrix model that makes (6) nontrivial. It can guarantee that the tracking singularity is eliminated practically, although not theoretically. In addition, the same input–output dimension policy is very helpful for obtaining good convergence ability in training if the model is realized with ANN.

It is worth mentioning that the proposed visual mapping model is easy to extend to higher dimension ones or to the ones that have higher dimension of image feature space than robot control space. This can be done by invoking more independent image features from single image or more simultaneous images from multicameras to characterize object state and motion in image feature space.

IV. Control Scheme

In the last section, we presented the nonlinear visual mapping model described by (6). Since it is nonparametric, an ANN is constructed to realize it so that the difficulty in parameter recognition is avoided via offline training. We should point out that the capacity of ANN could sufficiently be exploited for uncalibrated visual servoing only by taking advantage of the nonlinear visual mapping model, which is more powerful than the image Jacobian matrix for describing direct mapping from visual space to robot control space.

For visual tracking, motion planning for robot hand is necessary for pursuing stable performance. In the following subsections, we first address the issue of motion planning and then the issue of construction and training of the neural network to realize the nonlinear visual mapping model. Control instructions obtained from motion planning are fed to the neural network to obtain robot movements that achieves dynamic visual tracking. Control structure of the whole system is shown in Fig. 3.

Since the motion of the object $\boldsymbol{a}_{ow}$ is not known and cannot be measured online, it serves as the external disturbance to the tracking controller. Thus, a feedforward controller D_F is used to compensate the unknown object movements and improves the tracking performance of the whole system. Design of D_F will be discussed in Section IV-C.

A. Real-Time Motion Planning

The motion planning is done in the 3-D image feature space. The input and output of the real-time motion planning module is shown in Fig. 3. Given the global expected position $\boldsymbol{p}_{fi}^{*}$ of the object and its present motion parameters $\boldsymbol{p}_{fi}(k)$, $\boldsymbol{v}_{fi}(k)$ in the image feature space, motion of the features are planned in real-time for the next time instant, including the planned position of the object $\boldsymbol{p}_{fi}^{p}(k+1)$, the velocity $\boldsymbol{v}_{fi}^{p}(k+1)$, and the acceleration $\boldsymbol{a}_{fi}^{p}(k+1)$, to achieve a quick, error-free tracking of the moving object. Meanwhile, the tracking error caused by object velocity variation and visual mapping model inaccuracy is expected to overcome. Suppose that the Euclidean distance be-

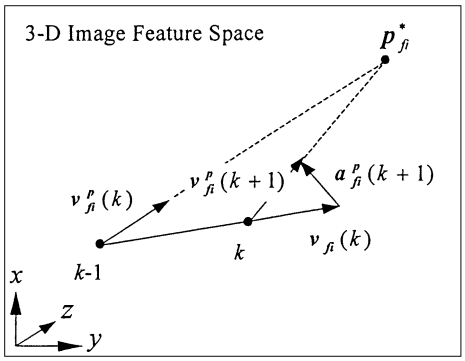

Fig. 4. Motion planning.

tween the position of the object $\boldsymbol{p}_{fi}(k)$ and the expected position $\boldsymbol{p}^*_{fi}$ at the kth instant in the image feature space is $d(k)$. In order to have the hand approaching the object as soon as possible, the direction of $\boldsymbol{v}^p_{fi}(k+1)$ should always be pointing to $\boldsymbol{p}^*_{fi}$, as shown in Fig. 4, and then, we have the following.

1) When $d(k) > d_1$, the magnitude of $\boldsymbol{v}^p_{fi}(k+1)$, $|\boldsymbol{v}^p_{fi}(k+1)|$ is increased until the hand is moving with the maximally allowed velocity.
2) When $d_1 > d(k) > d_2$, $|\boldsymbol{v}^p_{fi}(k+1)|$ is decreased until the hand is moving with a low velocity.
3) When $d(k) < d_2$, in accordance with the error between the current position and the expected position in the image feature space, $\boldsymbol{v}^p_{fi}(k+1)$ could be calculated by using the PI control algorithm so as to achieve error-free tracking

$$\boldsymbol{v}^p_{fi}(k+1) = \boldsymbol{v}_{fi}(k) + c_1[(\boldsymbol{p}^*_{fi} - \boldsymbol{p}_{fi}(k)) - c_2(\boldsymbol{p}^*_{fi} - \boldsymbol{p}_{fi}(k-1))]. \quad (7)$$

In the tracking scheme, the thresholds satisfy $d_1 > d_2 > 0$, where d_1 and d_2 are thresholds for switching to different schemes described above. c_1 and c_2 are the proportional and integral coefficients of the PI controller.

The output of the motion planner is the planned value of the object feature acceleration $\boldsymbol{a}^p_{fi}(k+1)$, which is approximately estimated from

$$\boldsymbol{a}^p_{fi}(k+1) = \boldsymbol{v}^p_{fi}(k+1) - \boldsymbol{v}_{fi}(k). \quad (8)$$

B. Neural Network Mapping

The function of the ANN is to realize the visual mapping model proposed in Section III that transforms the planned motion in the image feature space to the robot control space so that the motion instructions are fed to the robotic servo controller. The ANN should be trained offline before it is used in online control. The dotted-line arrows and their confluent module in Fig. 3 show the input–output data and the training phase prior to control. Based on the invertible mapping model described in (6), a three-layer ANN with nine inputs and three outputs is constructed. The back-propagation (BP) algorithm with a momentum term as an accelerator is utilized here for offline training of the ANN [24].

The training-samples are obtained as follows. Let an object move along a straight line with a constant velocity but different initial positions and initial velocities in the working space. Try to have the trajectory to span the whole working space as much as possible. A more practical case is that the object velocity is zero, i.e., the object is still. The hand randomly translates in 3-D space, i.e., its acceleration is a random variable. At each visual sampling instant, the position $\boldsymbol{p}_{fi}(k)$, the velocity $\boldsymbol{v}_{fi}(k)$, and the acceleration $\boldsymbol{a}_{fi}(k)$ of the object projection in the image feature space as well as the hand acceleration $\boldsymbol{a}_{hw}(k)$ in the robot control space are recorded as a group of training data. Note that the hand acceleration $\boldsymbol{a}_{hw}(k)$ is obtained by a second-order differentiation of its position coordinates $\boldsymbol{p}_{hw}(k)$. Although each single training sample is prone to the image quantization errors and other noises, such as noises from different order of differentiation, the visual mapping model can still be approximated due to a large amount of training samples.

When training, $\boldsymbol{p}_{fi}(k)$, $\boldsymbol{v}_{fi}(k)$, and $\boldsymbol{a}_{fi}(k)$ are stacked to be a nine-dimensional input vector. $\boldsymbol{a}_{hw}(k)$ is the desired 3-D output vector. After the training phase, the resultant weights of the trained ANN are recorded and used for online control. When the ANN is used in discrete visual tracking control, the inputs are the current position $\boldsymbol{p}_{fi}(k)$ and the current velocity $\boldsymbol{v}_{fi}(k)$ of the object projection in the image feature space as well as the planned image feature acceleration $\boldsymbol{a}^p_{fi}(k+1)$ obtained from motion planning. The output of the ANN is the acceleration instruction for the robotic servo controller that is to be executed in the coming visual sampling period, i.e., $\boldsymbol{a}^p_{hw}(k+1)$.

C. Feedforward-Feedback Controller

In deriving the nonlinear mapping model of (6), we assumed that $\boldsymbol{a}^p_{ow}(k) = 0$, which means $\boldsymbol{a}_{ow}(k) = 0$ in (2), but the object is actually moving with unknown velocity, i.e., $\boldsymbol{a}_{oi}(k) \neq 0$. Thus, an acceleration

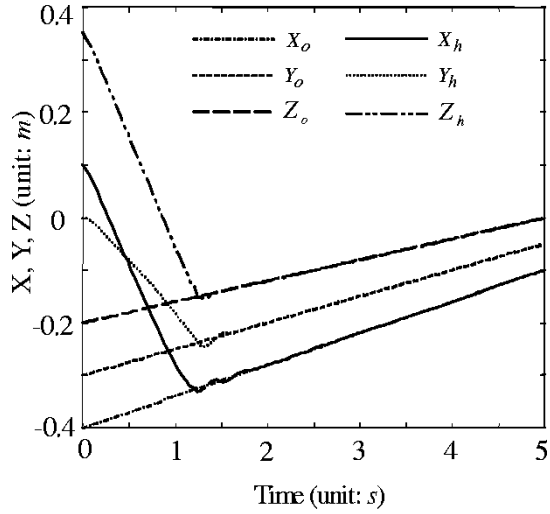

Fig. 5. Tracking an object with a constant velocity.

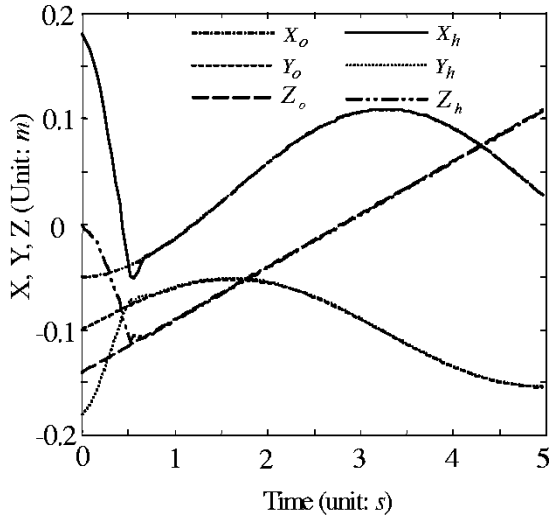

Fig. 6. Tracking an object with a swirl movement.

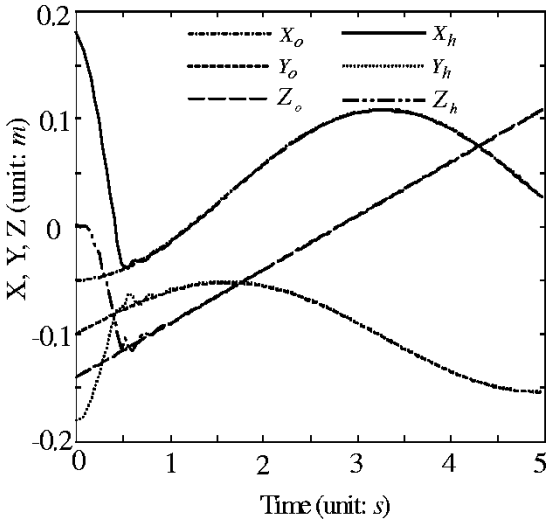

Fig. 7. Tracking with a change of the eye-hand relationship.

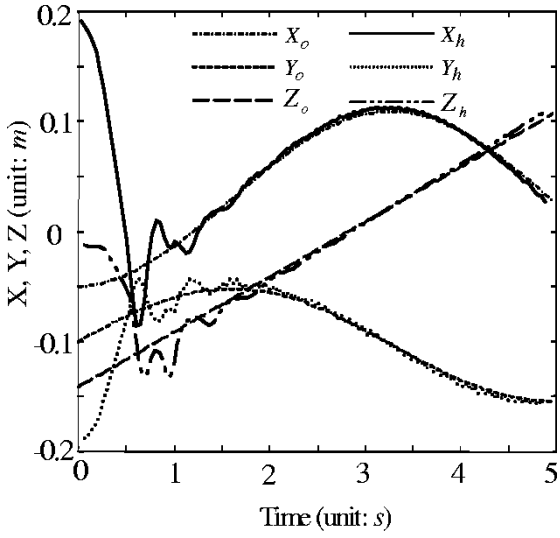

Fig. 8. Tracking after a rough training.

feedforward compensation controller D_F is incorporated into the visual tracking controller to estimate the unknown motion of the object and compensate for tracking control, as shown in Fig. 3.

Since the whole system is working in the discrete form by visual sampling moment, we discuss the design of the compensation controller D_F in its discrete form $D_F(z)$. The function of $D_F(z)$ is to estimate $\boldsymbol{a}_{oi}$ caused by the object's movement $\boldsymbol{a}_{ow}$ that cannot be measured directly. Since $\boldsymbol{a}_{oi}$ acts as the external disturbance to the controller, it is easy to infer that $\boldsymbol{a}_{oi}$ is related to the difference between the true velocity $\boldsymbol{v}_{fi}(k)$ and the planned velocity $\boldsymbol{v}^p_{fi}(k)$ of the image features. This difference can be measured in image feature space and is thus used to estimate $\boldsymbol{a}_{oi}$ by an AR model with the iterative least square estimation method.

We choose the feedforward controller as

$$D_F(z) = K_F \frac{1}{A(z^{-1})}. \tag{9}$$

Thus, we have

$$A\left(z^{-1}\right)\boldsymbol{a}_{oi}(k) = K_F\left[\boldsymbol{v}_{fi}(k) - \boldsymbol{v}^p_{fi}(k-1)\right] + \xi(k) \tag{10}$$

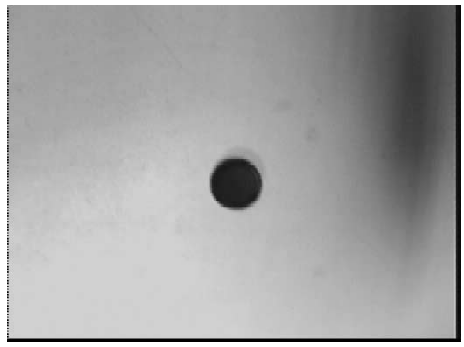

Fig. 9. Expected object image.

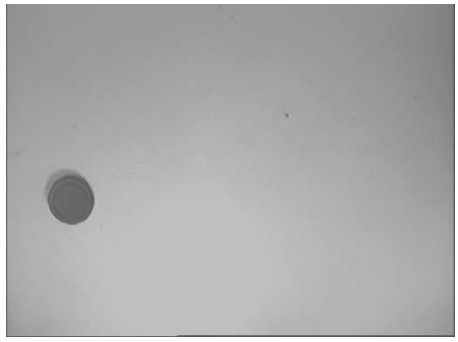

Fig. 10. Initial position of the object in the image plane.

where

$$A\left(z^{-1}\right) = 1 + a_1 z^{-1} + \cdots + a_m z^{-m} \tag{11}$$

and $\xi(k)$ in (10) is a white noise sequence with a zero mean and a variance σ used for characterizing image quantization error and other noises. K_F and a_i $(i = 1, \ldots, m)$ in (9) and (11) are adjustable according to applications to control the performance of $D_F(z)$. In (10), a delay of one visual sampling period (we use $\boldsymbol{v}^p_{fi}(k-1)$ instead of $\boldsymbol{v}^p_{fi}(k)$ in (10)) is taken into account for detection motions of image feature.

Substituting (11) into (10) and defining

$$\Phi(k) = \left[-\boldsymbol{a}_{oi}(k-1), \ldots, \boldsymbol{a}_{oi}(k-m), \boldsymbol{v}_{fi}(k) - \boldsymbol{v}^p_{fi}(k-1)\right]^T$$
$$\Theta(k) = \left[a_1, \ldots, a_m, K_F\right]^T$$

we have

$$\boldsymbol{a}_{oi}(k) = \Phi(k)^T \Theta(k) + \xi(k). \tag{12}$$

A typical estimation procedure is thus adopted to estimate $\Theta(k)$ with the normal least square estimation method with forgetting factor. The estimation value $\hat{\boldsymbol{a}}_{oi}(k)$ of $\boldsymbol{a}_{oi}(k)$ can be obtained from (12). $\hat{\boldsymbol{a}}_{oi}(k)$ is then added to the output of the motion planner to be the input of the ANN. Thus, the effect of the unknown object motion on object tracking can be compensated.

V. Simulations

The system configurations adopted in simulations are the same as shown in Fig. 1. A single camera is mounted at the end link of the robot manipulator. A rod-like object is moving in robot's 3-D workspace. The image feature space is formed by the 2-D positions of the object and the length of the object in the image plane. Thus, the image feature space is three dimensional, which is important for 3-D visual tracking control of the robot. Simulations are done for the above control scheme by

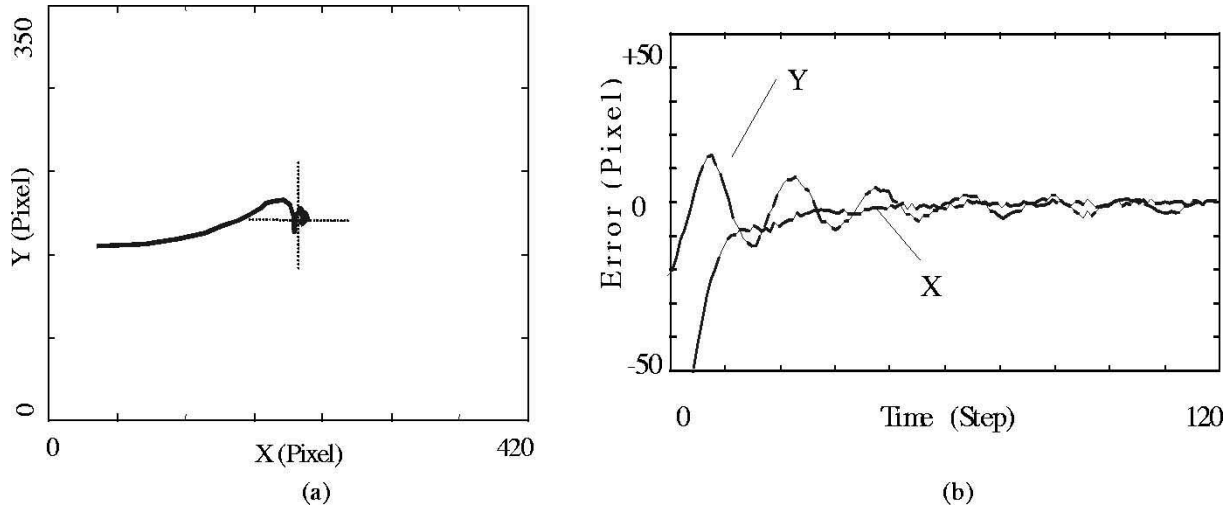

Fig. 11. (a) Projection trajectory of the object in the image plane (the cross-point is the expected position). (b) Error curve of the object's position in the image plane.

using the NN toolbox in Matlab5.1. The sampling space in robot control space for training the ANN is a cube of $[\,-0.2 \quad 0.2\,]^3$ (in meters). An ANN is constructed with nine inputs, three outputs, and 40 nodes in the hidden layer. The weights of the ANN are obtained with 3601 training samples and 20 000 iterations before being used in control. Parameters for motion planning in all simulations are empirically chosen as $c_1 = 40/\text{s}$, $c_2 = 0.5/\text{s}$, $d_1 = 0.003$ m, and $d_2 = 0.001$ m.

Figs. 5 and 6 are the transient processes of the robot hand approaching the object, where the position of the object is shown by $\boldsymbol{p}_{ow} = (X_o, Y_o, Z_o)$ and position of the robot hand $\boldsymbol{p}_{hw} = (X_h, Y_h, Z_h)$, both in the robot control space. In Fig. 5, the object is moving in a constant velocity along a straight line in space, whereas in Fig. 6, the object is moving with a swirl movement. The roll angle and the pitch angle between the camera coordinate system and the hand coordinate system are 30° and 20°, respectively. It is seen from the figures that the tracking is satisfactory, and the steady-state position errors of all axis directions in robot control space converge to zero rapidly. Moreover, since the ANN has good generalization ability, even though the object moves out of the training range in the working space, the tracking controller is still effective (see Fig. 5).

Fig. 7 illustrates the tracking when the roll angle and the pitch angle between the camera coordinate system and the hand coordinate system are changed to 0 and -20°, respectively. It is seen from the figure that even though no new training has been done for the ANN (i.e., the same weights obtained from the old training are used.), there is still a good tracking accomplished. It demonstrates that the controller based on ANN has a strong ability in environmental adaptation.

Fig. 8 demonstrates the tracking curve in the case of an insufficient training (638 training-samples, 1000 iterations, and 16 nodes in the hidden-layer). In this simulation, white Gaussian noise with a mean-square deviation of 0.36 (0.6 pixel of magnitude in each direction of image grid) is further added at the position coordinates of the object in image feature space to simulate quantization noise. We can see that effective tracking is still achieved, even though the ANN model used is undertrained and thus has quite rough realization for its nonlinear visual mapping model. Of course, the transient process is longer and vibrates more and the steady errors are larger when tracking is stable, compared with those in Figs. 5–7.

VI. Experiments

The experiments are to show how the visual mapping model and the whole control scheme proposed in this paper works for uncalibrated visual tracking. For simplicity of the system configurations, the object is moving in a 2-D working plane.

Fig. 12. Expected object image after a change of the pitch angle.

An Adept 604s robotic manipulator is used in our experiments to achieve object tracking. This robot arm has four degrees of freedom, with the first three rotational joints and the last one prismatic joint. Since the first two rotational joints of the robot can sufficiently accomplish 2-D movements for its hand, its last two degrees of freedom are locked to facilitate robot control. Thus, the robotic manipulator is reduced to a two-link system.

The camera is fixed on the hand, and the object moves freely in the working plane. For 2-D tracking, the purpose of the visual tracking controller is to control the hand movement in accordance with the images taken, until the object position in the image plane coincides with the expected position. In this case, the visual mapping model is 2-D, with 2-D motion parameters of the object (position, velocity and acceleration) in image feature space as inputs and 2-D hand movements in robot control space as output. An ANN is constructed with six inputs and two outputs and 30 nodes in the hidden layer, which is simpler compared with that used in simulations. The steps in the experiments are as follows.

1) Obtain the expected image offline. The mass position of the object in image plane is adopted as the image feature.
2) Train the ANN offline with the similar procedures in simulations. Six hundred groups of training samples are randomly collected in robot's working space. The ANN converges after 25 000 iterations.
3) Exercise real-time feedback control by using the trained ANN.

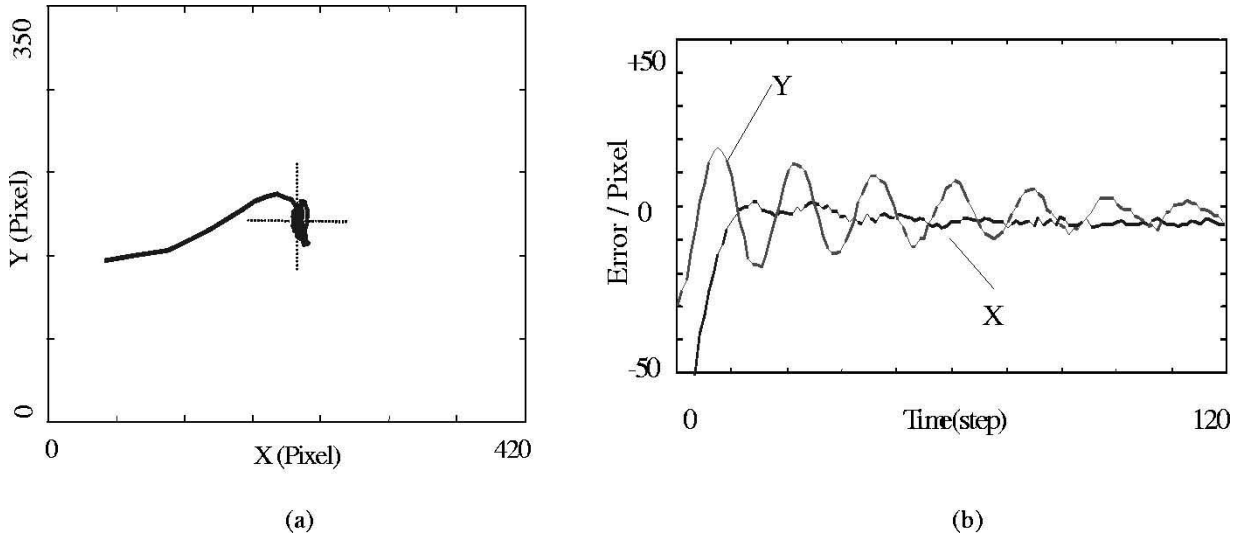

Fig. 13. (a) Projection trajectory of the object in the image plane (the cross-point is the expected position). (b) Error curve of the object's position in the image plane.

In the training-sample collection and the real-time control, the video sampling rate is 15 images/s. The image size in processing is always 420×350 pixels. All the images are first binarized with a properly predefined threshold to decrease the impacts of image noises, object shadow, and inconsistent lighting of the environments on object detection.

A. Low-Speed Visual Tracking

The expected image of the object for the experiment is shown in Fig. 9. The image of the initial object position is shown in Fig. 10. In the experiment, a round object moves in a certain direction with a speed unknown to robot control. Its area in the binary image plane is about 1800 pixels. The parameters for motion planner with the PI control law are empirically chosen as $c_1 = 0.24$ m/pixel $\cdot$ s and $c_2 = 0.8$ m/pixel $\cdot$ s. (Note that here, the error signals are measured in pixel, while in Simulations, the error signals are measured in metric. Thus, c_1 used here has different scale factor from that used in Simulations.) Fig. 11 demonstrates the visual tracking experimental results, where Fig. 11(a) is the position trajectory of the object in image plane when the camera is controlled moving in the robot control space, and Fig. 11(b) is the tracking error in the X and Y directions in the image plane, respectively. It is seen that the object projection is driven to the expected position and that visual tracking is basically achieved.

The feedforward signal $\boldsymbol{a}_{oi}$ for the image feature acceleration caused by the object's unknown motion is estimated iteratively by (9)–(12). The output of the feedforward controller is actually the sum of the output of the PI controller and the feedforward signal $\boldsymbol{a}^p_{fi}(k) + K_F \boldsymbol{a}_{oi}(k)$, where we choose $K_F = 0.3$ and $m = 6$ empirically in control. It is seen in Fig. 11 that the transient process is satisfactory, and the steady tracking errors are only ± 2 pixels in the X direction and ± 5 pixels in the Y direction in image plane, respectively.

B. Visual Tracking With the Changed Camera Pose

With a change of the pitch angle of the camera for about $+20$ degrees, the new expected image of the object is taken and shown in Fig. 12.

At the initial stage, the robot hand moves to an arbitrary position in its working space but with the object in the camera's field of view. Fig. 13 shows the trajectory of the image feature varying under both object motions and the controlled camera motions. Note that no new training process for the ANN is run for the changed hand-eye relations. The effective tracking process is still obtained though the dynamic tracking error in the y direction, which is about ± 10 pixels, is a little bit increased compared with that shown in Fig. 11. Here, a similar feedforward controller as in the Section VI-A is also used to compensate for object movement that is regarded as external disturbances to the motion planner.

VII. Conclusion

A nonlinear visual mapping model for uncalibrated coordination of eye-in-hand robotic system is proposed in this paper. This model is more powerful and general than the image Jacobian matrix model, thus providing more rooms for making full use of capacity of the neural network and taking advantage of a prior knowledge of system configuration via offline training. Moreover, it is advantageous over the image Jacobian matrix model in the sense that it inherently avoids tracking singularity problem and is straightforward to be extended to applications of high-dimensional tracking. Since the overall computational complexity of the coordination control is split into offline training and online planning, dynamic tracking is consequently easy and efficient to achieve. Thus, in methodology, this scheme is a good solution for proper tradeoff between offline modeling and online control.

Though the proposed scheme is successful in principle for dynamic coordination control of the uncalibrated robotic hand-eye system, the relationship between the ANN structure and the visual tracking system is not yet clear. It is a challenge to design an efficient ANN to approach the robotic visual mapping model quickly and accurately with relatively low computational load, which is the future work of this research.

Acknowledgment

The authors are very grateful to the associate editor and anonymous reviewers, whose comments were very helpful and critical in improving the quality of the paper.

References

[1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robotics Automat.*, vol. 12, pp. 651–670, Oct. 1996.

[2] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robotics Automat.*, vol. RA–3, pp. 404–417, Oct. 1987.

[3] M. Jaegersand, O. Fuentes, and R. Nelson, "Experimental evaluation of uncalibrated visual servoing for precision manipulation," *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 2874–2880, 1997.
[4] C. Schering and B. Kersting, "Uncalibrated hand-eye coordination with a redundant camera system," *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 1366–1372, 1998.
[5] B. Yoshimi and P. Allen, "Alignment using an uncalibrated camera system," *IEEE Trans. Robotics Automat.*, vol. 11, pp. 516–521, Aug. 1995.
[6] H. Sutanto, R. Sharma, and V. Varma, "Image based autodocking without calibration," *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 974–979, 1997.
[7] J. Su and Y. Li, "Real time estimation of eye-hand relationship and planar robotic tracking," in *Proc. Chinese World Congr. Intell. Contr. Intell. Automat.*, Xi'an, China, 1997, pp. 73–76.
[8] J. Hespanha *et al.*, "What can be done with an uncalibrated stereo system," *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 1366–1372, 1998.
[9] N. Papanikolopoulos and P. Khosla, "Adaptive robotic visual tracking: Theory and experiments," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 429–445, Mar. 1993.
[10] K. Hashimoto, T. Ebine, and H. Kimura, "Visual servoing with hand-eye manipulator-optimal control approach," *IEEE Trans. Robotics Automat.*, vol. 12, pp. 766–774, Oct. 1996.
[11] W. Miller, "Real-time application of neural networks for sensor-based control of robots with vision," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 825–831, July/Aug. 1989.
[12] H. Hashimoto, T. Kubota, M. Sato, and F. Hurashima, "Visual control of robotic manipulator based on neural networks," *IEEE Trans. Ind. Electron.*, vol. 39, pp. 490–496, Dec. 1992.
[13] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robotics Automat.*, vol. 8, pp. 313–326, June 1992.
[14] G. Hager, "A modular system for robust hand-eye coordination," *IEEE Trans. Robotics Automat.*, vol. 13, pp. 582–595, Aug. 1997.
[15] J. Feddema and C. Lee, "Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 1172–1183, Oct. 1990.
[16] E. Grosso, G. Metta, A. Oderra, and G. Sandini, "Robust visual servoing in 3-D reaching tasks," *IEEE Trans. Robotics Automat.*, vol. 12, pp. 732–742, Oct. 1996.
[17] A. Cretual and F. Chaumette, "Visual servoing based on image motion," *Int. J. Robotics Res.*, vol. 20, pp. 857–877, Nov. 2001.
[18] C. Colombo and B. Allotta, "Image-based robot task planning and control using a compact visual representation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 29, pp. 92–99, Feb. 1999.
[19] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robotics Automat.*, vol. 9, pp. 152–165, Apr. 1993.
[20] Y. Meng and H. Zhuang, "Self-calibration of camera-equipped robot manipulators," *Int. J. Robotics Res.*, vol. 20, pp. 909–921, Nov. 2001.
[21] H. Zhuang, L. Wang, and Z. Roth, "Simultaneous calibration of a robot and a hand-mounted camera," *Proc. IEEE Int. Conf. Robotics Automat.*, pp. 149–154, 1993.
[22] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics Automat.*, vol. RA–3, pp. 323–344, Aug. 1987.
[23] T. Drummond and R. Cipolla, "Real-time tracking of complex structures with online camera calibration," *Image Vision Comput.*, vol. 20, pp. 427–433, 2002.
[24] D. R. Hush and B. Horne, "Progress in supervised neural networks: What's new since Lippmann?," *IEEE Signal Processing Mag.*, vol. 10, pp. 8–39, 1993.

Incremental Learning With Balanced Update on Receptive Fields for Multi-Sensor Data Fusion

Jianbo Su, Jun Wang, and Yugeng Xi

***Abstract*—This paper addresses multi-sensor data fusion with incremental learning ability. A new cost function is proposed for the receptive field weighted regression (RFWR) algorithm based on the idea of back propagation (BP), so that the computation efficiency and the learning strategy of the modified RFWR are much more applicable for multi-sensor data fusion problem. Thus a new fusion structure and algorithm with incremental learning ability is constructed by adopting the modified RFWR algorithm together with the weighted average algorithm. Experiments of a two-camera unified positioning system are implemented successfully to test the proposed computation structure and algorithms.**

***Index Terms*—Back propagation, incremental learning, receptive field, sensor fusion.**

I. Introduction

Frequently in practice, a multi-sensor fusion system needs to be upgraded by integrating additional sensors into the system to adapt to more complex environments and tasks. Normally the structure and fusion algorithm of the system should totally be redesigned for the upgrade, even if most of the sensors in the system are retained without any changes [2]. This inefficiency can be overcome if the fusion system has incremental learning ability [11]. With this ability, the structure of the fusion system is easy to be upgraded and only the added sensors need to be trained before being included in the whole system.

Learning with spatially localized basis function [4], [16], [17] has been studied for many years in contrast to the learning with the global basis function [15]. A lot of applications have been accumulated such as robot control [6], chemical process modeling [7], nonlinear system estimation and control [8], image coding [18] and pattern recognition [9], [12], etc. Incremental learning ability from local receptive-field is proved to be extremely useful for approximating unknown functional relationships between input and output data streams [11]. Among these, Schaal and Atkeson proposed a Receptive Field Weighted Regression (RFWR) algorithm in [1]. This algorithm is related to constructive learning [10] and local function approximation based on the well-known radial basis function networks. But with some particular nonparametric regression techniques involved, RFWR is more efficient for incremental function approximation in the sense that it is not necessary to store the training data and discard receptive fields after using them. In addition, it can overcome some difficulties occurring normally in the incremental learning tasks, especially the bias-variance dilemma [13] and the negative interference problems.

However, direct application of RFWR in the multi-sensor data fusion system is not practical. Although some techniques from nonparametric statistics, such as leave-one-out local cross validation and the stochastic approximation, improve the effectiveness of learning in RFWR, they contribute much to the computational complexity of whole learning process. Moreover, RFWR is a receptive field based learning system. Learning in RFWR emphasizes only on adjustment in individual receptive field. Thus a multi-sensor data fusion system with this learning scheme may unexpectedly have inconsistent

Manuscript received January 16, 2002; revised June 21, 2002. This work was supported in part by the National Natural Science Foundation of China under Grant 69889501. This paper was recommended by Associate Editor I. Bloch.

The authors are with the Department of Automation, Shanghai Jiaotong University, Shanghai 200030, China (e-mail: jbsu@ sjtu.edu.cn).

Digital Object Identifier 10.1109/TSMCB.2002.806485