Full paper

Basic Behavior Acquisition with Multisensor Integration for a Robot System

Chenggang Liu * **and Jianbo Su**

Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, P. R. China

Received 24 June 2008; accepted 7 February 2009

Abstract

A robot system generally has several degrees of freedom of motions as well as different kinds of sensors. Basic behaviors are planned based on these sensors' feedback. We take advantage of the Jacobian matrix to describe the differential relations between the sensor feedback and the motor motions. Hence, the relation between two sensors could be formulated by the two respective Jacobian matrices of both sensors to motors. Multisensor integration can, thus, be employed for better performance of the robot system. Experiments of basic behavior acquisition like gazing and posture control of a robot head are conducted. The performances of the two basic behaviors before and after multisensor integration are compared, which demonstrate the performance and robustness of the proposed multisensor integration method.

Keywords

Visuo-inertial integration, behavior acquisition, multi-agent-based system, sensor-based robotics, gaze control

1. Introduction

Cognitive processes at high-level abstractions rely on a hierarchy of lower-level behaviors. Low-level autonomous behaviors can be constructed from basic sensorimotor behaviors [1, 2]. A basic sensorimotor behavior can be a reflex, which is a direct motor response to sensor feedback. Basic behavior designs are essential to a behavior-based robot. In order to acquire basic behaviors, most designers make assumptions about how autonomous robots perceive the world and take advantage of the prior knowledge of the sensorimotor maps. The basic behaviors are provided by a set of basic modules or layers defined as the fundamentals of a more complex system [3, 4]. This strategy reduces the chances of generating diverse, adaptive and unexpected behaviors from the dynamic interaction between the robot

* To whom correspondence should be addressed. E-mail: frankliu@sjtu.edu.cn

DOI:10.1163/016918609X12496339824450

and the environment [5]. If basic behaviors are determined without examination, it is most probable that the behavior control of the system might be inefficient or even completely wrong. An alternative to overcome these limitations is to acquire basic behaviors by the robot itself [6].

Many attempts have been made to achieve this goal, such as by reinforcement learning [7–10], by artificial neural networks [11, 12], by genetic algorithms [13, 14], etc. Owing to the simplicity of the plant and the enormous knowledge about the biological oculomotor control system, the head–eye control system is the best-studied sensorimotor control system [15, 16]. There have been some attempts to acquire basic behaviors, such as visual tracking and gaze control, from the interaction of the robot with the environment. In Ref. [17], the proposed robot first performs random movements, so that the relation between the actuator apparatuses and the visual sensors is modeled. The robot can then perform visual tracking. Marjanović *et al.* [18] use the self-supervised learning method to obtain the map between the eye, the head and the end-effector. Thus, the humanoid robot Cog can perform fundamental visuomotor coordination tasks, such as gazing at an object while reaching it. In Ref. [19], the system first learns a forward model that predicts how image features move in the visual field when the gaze is shifted. It can then perform gaze control by searching for the action that makes the target feature projected in the center area of the image plane. Although above systems can achieve high performance, gaze control driven only on the basis of visual information is easily exposed to failure when the head–eye system has to face unknown perturbations of the body [20]. In humans, the vestibular system senses the positions of the head and the body in space. The information provided by the vestibular system is fused at a very early processing stage with vision. It also plays an important role in gaze control [21, 22].

Gaze control integrated non-visual information has received little attention so far. In Ref. [23], the proposed binocular system is controlled by integrating visual and inertial information. The visuo-inertial integration is implemented by considering the geometry of the binocular system and the knowledge of actual gaze configuration (the gaze distance and direction). Shibata *et al.* [24] deal with the problem of visuo-inertial integration by reproducing a computational model of the biological reflexes system. Panerai *et al.* [20] use an adaptive neural network to learn to map visual and inertial signals to motor commands. Thus, the proposed system can adaptively generate stabilized gaze.

Apart from gaze control, the control of head posture is also important when the head system is used for the human–robot interaction. Humans use many kinds of body language, such as nodding, shaking one's head, etc. Both the vestibular system and the visual system play an essential role in the control of head posture [25]. There is work on the posture control of articulated mobile robots, but little on the integration of the posture control with visual information.

It is realized that many basic behaviors are inherently the kinds of direct motor responses to some sensor feedback. The sensors can be real sensors (hardware) or virtual sensors (software or an algorithm that gives the position of the attention point). More complex behaviors can be constructed either by sequencing or by integrating acquired basic behaviors. During assisted learning, for example, the robot should be able to make inquiries about objects to a nearby person, by means of gazing at the person, turning its head, gazing at an object and/or asking a question. This complex behavior can be constructed by sequencing the control of head posture and gaze control. Gaze control can be acquired based on visual sensors, while the control of head posture can be acquired based on inertial sensors. For many basic behaviors, the differential relations between the sensor feedback and the robot's motions can be linearly described. In the field of image-based visual servoing, the image Jacobian model is used to linearly describe the differential relation between the visual sensor feedback and the robot's motion [26, 27]. We also take advantage of the Jacobian matrix to describe the differential relations between the sensor feedback and the motor motions. Thus, basic behaviors, such as gaze control and the control of head posture, can be generated from the corresponding sensors. When there are multiple basic behaviors in the system, it is necessary to integrate all of them to generate coordinative behaviors. For a multisensor robot system, it is also expected to integrate different sensors so as to use their complementary features for better performance. We describe a strategy to take advantage of the Jacobian matrices to integrate these basic behaviors. Furthermore, we propose a multisensor integration method for better performance by reusing these Jacobian matrices. A system identification method is used to estimate the Jacobian matrices online and, thus, makes the system free from the innate knowledge regarding the sensors and the motors. In order to realize the flexibility of the proposed method, a multi-agent-based implementation is also proposed.

The rest of the paper is organized as follows. In Section 2, we propose our method to acquire basic behaviors. Section 3 proposes the multisensor integration method and Section 4 proposes the multi-agent-based implementation of the control system. Experiments are provided in Section 5 to demonstrate the validity and the performance of the proposed method. Conclusions and future work are provided in Section 6.

2. Behavior Acquisition

2.1. Basic Behavior Control

Suppose the robot system has n d.o.f. The joint variables of the n d.o.f. are denoted by $\mathbf{q} = [q_1, q_2, \ldots, q_n]^{\mathrm{T}}$. The control input $\mathbf{u}_{\mathrm{c}}(t)$ to the system at the time moment t is the desired increment of $\mathbf{q}$. Suppose the sensor s_i has l outputs. The current state and the desired state of the sensor s_i are denoted by $\mathbf{y}_{\mathrm{s}_i} = [y_1, y_2, \ldots, y_l]^{\mathrm{T}}$ and $\mathbf{r}_{\mathrm{s}_i} = [r_1, r_2, \ldots, r_l]^{\mathrm{T}}$, respectively. In order to acquire a basic behavior based on the sensor s_i, the problem is to determine the control input $\mathbf{u}_{\mathrm{c}}$ based on the current and the desired state of the sensor s_i.

The mappings between the sensors and the motors are currently considered to be continuous, differentiable and time-invariant. Thus, the relation between the sensor s_i and the joint variables is:

$$y_i = f_i(\mathbf{q}) \quad i = 1, \ldots, l. \tag{1}$$

Thus, the differential relation is:

$$\mathrm{d}y_i = \mathbf{a}_i^{\mathrm{T}}\,\mathrm{d}\mathbf{q} \quad i = 1, \ldots, l, \tag{2}$$

where $\mathbf{a}_i = [\frac{\partial y_i}{\partial q_1}, \frac{\partial y_i}{\partial q_2}, \ldots, \frac{\partial y_i}{\partial q_n}]^{\mathrm{T}}$. So:

$$\mathbf{y}_{\mathrm{s}_i} = \mathbf{f}(\mathbf{q}), \tag{3}$$

and

$$\mathrm{d}\mathbf{y}_{\mathrm{s}_i} = \mathbf{J}_{\mathrm{s}_i}\,\mathrm{d}\mathbf{q}, \tag{4}$$

where $\mathbf{J}_{\mathrm{s}_i} = [\frac{\partial \mathbf{f}}{\partial \mathbf{q}}] = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_l]^{\mathrm{T}}$ is a Jacobian matrix that is of all partial derivatives of the vector function $\mathbf{f}$. It relates the sensor s_i's feedback with the motor motions. We take advantage of the Jacobian matrix to describe the sensorimotor maps between the sensors and motors. The Jacobian matrix could be estimated by an on-line estimation method. In order to generate a basic behavior from the sensor s_i, the resolved-rate motion control [27] can be used. The proportional control law is given by:

$$\mathbf{u}_{\mathrm{c}} = \mathbf{K}_{\mathrm{s}_i}\mathbf{J}_{\mathrm{s}_i}^{+}(\mathbf{r}_{\mathrm{s}_i} - \mathbf{y}_{\mathrm{s}_i}), \tag{5}$$

where $\mathbf{K}_{\mathrm{s}_i}$ is the constant gain matrix of appropriate dimension and $\mathbf{J}_{\mathrm{s}_i}^{+}$ is the pseudo-inverse of $\mathbf{J}_{\mathrm{s}_i}$.

2.2. Multi-Behavior Integration

If there are multiple sensors in the robot system, there will be multiple behaviors based on these sensors. The controllers of these behaviors will drive the motors simultaneously, thus it is necessary to integrated all of these behaviors for a coordinative behavior. Suppose there are k sensors in the system, we solve the equations for $\mathrm{d}\mathbf{q}$ of all these sensors simultaneously as follows [28].

Combining (4) of all k sensors, we have:

$$\begin{bmatrix} \mathrm{d}\mathbf{y}_{\mathrm{s}_1} \\ \mathrm{d}\mathbf{y}_{\mathrm{s}_2} \\ \vdots \\ \mathrm{d}\mathbf{y}_{\mathrm{s}_k} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{\mathrm{s}_1} \\ \mathbf{J}_{\mathrm{s}_2} \\ \vdots \\ \mathbf{J}_{\mathrm{s}_k} \end{bmatrix} \mathrm{d}\mathbf{q}. \tag{6}$$

The least-squares solution of $\mathrm{d}\mathbf{q}$ is given by:

$$\mathrm{d}\mathbf{q} = \begin{bmatrix} \mathbf{J}_{\mathrm{s}_1} \\ \mathbf{J}_{\mathrm{s}_2} \\ \vdots \\ \mathbf{J}_{\mathrm{s}_k} \end{bmatrix}^{+} \begin{bmatrix} \mathrm{d}\mathbf{y}_{\mathrm{s}_1} \\ \mathrm{d}\mathbf{y}_{\mathrm{s}_2} \\ \vdots \\ \mathrm{d}\mathbf{y}_{\mathrm{s}_k} \end{bmatrix}. \tag{7}$$

Using the simple proportional feedback control, the control input to the system after integration can be determined as:

$$\mathbf{u}_\mathrm{c}(t) = \mathbf{K}_\mathrm{c} \begin{bmatrix} \mathbf{J}_{\mathrm{s}_1} \\ \mathbf{J}_{\mathrm{s}_2} \\ \vdots \\ \mathbf{J}_{\mathrm{s}_k} \end{bmatrix}^+ \begin{bmatrix} \mathbf{r}_{\mathrm{s}_1} - \mathbf{y}_{\mathrm{s}_1} \\ \mathbf{r}_{\mathrm{s}_2} - \mathbf{y}_{\mathrm{s}_2} \\ \vdots \\ \mathbf{r}_{\mathrm{s}_k} - \mathbf{y}_{\mathrm{s}_k} \end{bmatrix}, \tag{8}$$

where $\mathbf{K}_\mathrm{c}$ is the constant gain matrix of appropriate dimension. Let $\mathbf{J}_\mathrm{c} = [\mathbf{J}_{\mathrm{s}_1}^\mathrm{T}, \mathbf{J}_{\mathrm{s}_2}^\mathrm{T}, \ldots, \mathbf{J}_{\mathrm{s}_k}^\mathrm{T}]^\mathrm{T}$, $\Delta_i = \mathbf{r}_{\mathrm{s}_i} - \mathbf{y}_{\mathrm{s}_i}$ and $\Delta = [\Delta_1^\mathrm{T}, \Delta_2^\mathrm{T}, \ldots, \Delta_k^\mathrm{T}]^\mathrm{T}$, then (8) can be rewritten more concisely as:

$$\mathbf{u}_\mathrm{c} = \mathbf{K}_\mathrm{c}\mathbf{J}_\mathrm{c}^+ \Delta. \tag{9}$$

As (7) gives the least-square solution of d$\mathbf{q}$, such a control law will try to minimize the deviations from the desired states and the current states of all these sensors. Compared with conventional integration methods, such as averaging, voting, etc., this method is less arbitrary and more flexible.

2.3. Online Estimation of the Jacobian Matrix

In order to enable the robot to acquire basic behaviors online, an online estimation method of the Jacobian matrix is proposed. The recursive least-squares (RLS) estimation method is widely used for system identification [29]. RLS can be interpreted as a Kalman filter for the process:

$$\theta(k+1) = \theta(k) \tag{10}$$

$$\mathbf{v}(k) = \varphi(k)^\mathrm{T}\theta(k) + \nu(k), \tag{11}$$

where $\theta(k)$ is the state to be estimated, $\mathbf{v}(k)$ is the measurement, $\varphi(k)^\mathrm{T}$ is the measurement matrix and $\nu(k)$ is the noise. In order to use RLS to estimate the Jacobian matrix between sensors and motors, let:

$$\theta(k) = [\mathbf{a}_1^\mathrm{T}, \mathbf{a}_2^\mathrm{T}, \ldots]^\mathrm{T} \tag{12}$$

$$\mathbf{v}(k) = \mathbf{y}(k) - \mathbf{y}(k-1) \tag{13}$$

$$\Delta\mathbf{q}(k) = \mathbf{q}(k) - \mathbf{q}(k-1) \tag{14}$$

$$\varphi(k) = \begin{bmatrix} \Delta\mathbf{q}(k) & & 0 \\ & \ddots & \\ 0 & & \Delta\mathbf{q}(k) \end{bmatrix}, \tag{15}$$

where $\mathbf{q}$ are the joint variables, $\mathbf{y}$ is the current sensor state and $\mathbf{a}_i^\mathrm{T}$ is the ith row of the Jacobian matrix $\mathbf{J}$. Then $\mathbf{J}$ can be estimated at every time step recursively as:

$$\mathbf{K}(k+1) = \mathbf{P}(k)\varphi(k+1)(\mathbf{I} + \varphi^\mathrm{T}(k+1)\mathbf{P}(k)\varphi(k+1))^{-1} \tag{16}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + \mathbf{K}(k+1)[\mathbf{v}(k+1) - \varphi^\mathrm{T}(k+1)\hat{\theta}(k)] \tag{17}$$

$$\mathbf{P}(k+1) = (\mathbf{I} - \mathbf{K}(k+1)\varphi^\mathrm{T}(k+1))\mathbf{P}(k), \tag{18}$$

where $\mathbf{K}$ is the gain matrix and $\mathbf{P}$ is the covariant matrix. Such a RLS estimator has been proven to be an effective method for online estimation of the image Jacobian matrix [30]. The initial condition $\mathbf{P}(0)$ is chosen such that it is positive and proportional to the parameter prior covariance. Online estimation of the Jacobian matrices can make the system free from the knowledge regarding the sensors and the motors. Moreover, it can make the final system more robust against the parameter changes of sensors and motors.

3. Multisensor Integration

If there are multiple sensors in the system, it is expected to integrate different sensors in order to utilize their complementary features for better performance. For the sensor s_i, its current state at time moment t can be written as:

$$\mathbf{y}_{\mathrm{s}_i}(t) = \mathbf{y}_{\mathrm{s}_i}(kT_{\mathrm{s}_i}) + \int_{kT_{\mathrm{s}_i}}^{t} \mathrm{d}\mathbf{y}_{\mathrm{s}_i}, \tag{19}$$

where T_{s_i} denotes the sample period of the sensor s_i and kT_{s_i} is the recent time when the sensor s_i outputs the last data. If there is another sensor s_j, similar to (4), we have:

$$\mathrm{d}\mathbf{y}_{\mathrm{s}_j} = \mathbf{J}_{\mathrm{s}_j}\,\mathrm{d}\mathbf{q}, \tag{20}$$

in which $\mathbf{J}_{\mathrm{s}_j}$ is a $m \times n$ matrix. Hence, if $m = n$ and $\mathbf{J}_{\mathrm{s}_j}$ is non-singular, $\mathbf{J}_{\mathrm{s}_j}^{-1}$ exists. In this case:

$$\mathrm{d}\mathbf{q} = \mathbf{J}_{\mathrm{s}_j}^{-1}\,\mathrm{d}\mathbf{y}_{\mathrm{s}_j}. \tag{21}$$

From (4), (19) and (21), we have:

$$\begin{aligned}\hat{\mathbf{y}}_{\mathrm{s}_i|\mathrm{s}_j}(t) &= \mathbf{y}_{\mathrm{s}_i}(kT_{\mathrm{s}_i}) + \int_{kT_{\mathrm{s}_i}}^{t} \mathbf{J}_{\mathrm{s}_i}\mathbf{J}_{\mathrm{s}_j}^{-1}\,\mathrm{d}\mathbf{y}_{\mathrm{s}_j} \\ &= \mathbf{y}_{\mathrm{s}_i}(kT_{\mathrm{s}_i}) + \mathbf{J}_{\mathrm{s}_i}\mathbf{J}_{\mathrm{s}_j}^{-1}(\mathbf{y}_{\mathrm{s}_j}(t) - \mathbf{y}_{\mathrm{s}_j}(kT_{\mathrm{s}_i})),\end{aligned} \tag{22}$$

where $\hat{\mathbf{y}}_{\mathrm{s}_i|\mathrm{s}_j}(t)$ denotes the estimation of the sensor s_i's current state with the help of the sensor s_j.

If $m \neq n$, $\mathbf{J}_{\mathrm{s}_j}^{-1}$ does not exist. In this case, assume that $\mathbf{J}_{\mathrm{s}_j}$ is of full rank (i.e., $\mathrm{rank}(\mathbf{J}_{\mathrm{s}_j}) = \min(m, n)$). We can have a least-squares solution of $\mathrm{d}\mathbf{q}$. The general solution is:

$$\mathrm{d}\mathbf{q} = \mathbf{J}_{\mathrm{s}_j}^{+}\,\mathrm{d}\mathbf{y}_{\mathrm{s}_j} + \mathbf{Q}\mathbf{z}, \tag{23}$$

where $\mathbf{J}_{\mathrm{s}_j}^{+}$ is the pseudo-inverse of $\mathbf{J}_{\mathrm{s}_j}$, $\mathbf{Q}$ is the orthogonal projection operator on the null space of $\mathbf{J}_{\mathrm{s}_j}$ and $\mathbf{z}$ is an arbitrary vector of appropriate dimension.

If $m > n$, $\mathbf{Q} = 0$ and the pseudo-inverse of $\mathbf{J}_{\mathrm{s}_j}$ could be:

$$\mathbf{J}_{\mathrm{s}_j}^{+} = (\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}\mathbf{J}_{\mathrm{s}_j})^{-1}\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}. \tag{24}$$

Therefore:

$$\mathrm{d}\mathbf{q} = (\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}\mathbf{J}_{\mathrm{s}_j})^{-1}\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}\,\mathrm{d}\mathbf{y}_{\mathrm{s}_j}. \tag{25}$$

In this case, the system is over-constrained, which means the sensor s_j has more feedback than the robot's degrees of freedom and there are enough features to uniquely determine the robot motions. From (4), (19) and (25), we have:

$$\hat{\mathbf{y}}_{\mathrm{s}_i|\mathrm{s}_j}(t) = \mathbf{y}_{\mathrm{s}_i}(kT_{\mathrm{s}_i}) + \mathbf{J}_{\mathrm{s}_i}(\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}\mathbf{J}_{\mathrm{s}_j})^{-1}\mathbf{J}_{\mathrm{s}_j}^{\mathrm{T}}(\mathbf{y}_{\mathrm{s}_j}(t) - \mathbf{y}_{\mathrm{s}_j}(kT_{\mathrm{s}_i})). \tag{26}$$

If $m < n$, $\mathbf{Q} \neq \mathbf{0}$ in general and all vectors of the form $\mathbf{Qz}$ lie in the null space of $\mathbf{J}_{\mathrm{s}_j}$ and correspond to the unobservable components of the robot motions. In this case, the system is under-constrained, which means some components of the robot motions cannot be observed only by the sensor s_j. Unless we introduce another sensor feedback, $\mathrm{d}\mathbf{q}$ cannot be estimated. From (22) and (26), the current state estimation of the sensor s_i by the sensor s_j can be rewritten more concisely as:

$$\hat{\mathbf{y}}_{\mathrm{s}_i|\mathrm{s}_j}(t) = \mathbf{y}_{\mathrm{s}_i}(kT_{\mathrm{s}_i}) + \mathbf{J}_{\mathrm{s}_i}\mathbf{J}_{\mathrm{s}_j}^{+}(\mathbf{y}_{\mathrm{s}_j}(t) - \mathbf{y}_{\mathrm{s}_j}(kT_{\mathrm{s}_i})), \tag{27}$$

where $m \geqslant n$ and $\mathbf{J}_{\mathrm{s}_j}$ is of full rank or non-singular. Therefore, a sensor's current state can be obtained by its reading or its estimates by other sensors.

This kind of integration has the following advantages. (i) We do not correlate two sensors by the relationship between them directly, but by their respective relations with the robot self-motions. Thus, the Jacobian matrices available could be re-used to plan the robot motion. This strategy could hence reduce the computational complexity, especially when there are a large amount of sensors needed to be integrated. (ii) The state of a sensor that is computationally intensive could be estimated by the states of other sensors that are computationally cheap. (iii) The system could be more robust against failures. When a sensor fails, its state could still be estimated from those of the others. (iv) $\hat{\mathbf{y}}_{\mathrm{s}_i}$ has a higher sample rate than $\mathbf{y}_{\mathrm{s}_i}$. This can improve the control performance of the system in most cases.

4. Multi-Agent-Based Implementation of the Control System

The multi-agent-based strategy could be adopted to implement the proposed method of higher flexibility. The agents are modules of independent functions. We propose two kinds of agents, which are the sensor agent and the control agent. The implementation of the sensor agent is illustrated in Fig. 1. Each sensor in the robot system has a sensor agent. The inputs to the sensor agent i are the current state, $\mathbf{y}_{\mathrm{s}_i}$, the desired state, $\mathbf{r}_{\mathrm{s}_i}$, of the sensor s_i, the joint variables, $\mathbf{q}$, and the current states of other sensors. The output of the sensor agent i is the desired state change, Δ_i, of the sensor s_i. The Jacobian matrix in the sensor agent i is estimated each time when there is a new feedback from the sensor s_i. Multisensor integration is performed when any sensor state changes. The control agent calculates the control input to the system according to (9). The inputs to the control agent are the desired state changes from all sensor agents. The output of the control agent is the control input to the system, $\mathbf{u}_{\mathrm{c}}$.

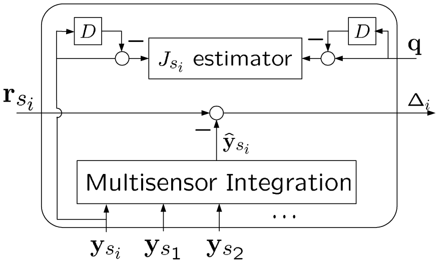


Figure 1. Implementation of the sensor agent.

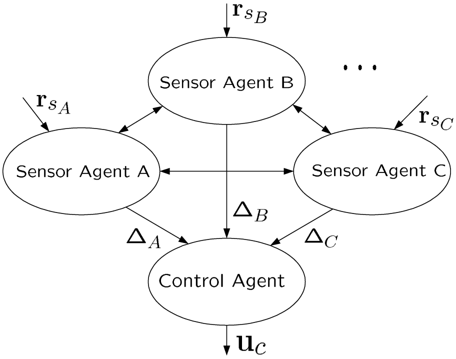


Figure 2. The multi-agent-based implementation of the control system.

The final multi-agent-based implementation of the control system is shown in Fig. 2. All agents are fully connected with the others. Each agent sends and receives additional data if necessary. If the desired state of a sensor agent is not explicitly defined, the sensor will not drive the motors. This is realized by excluding the sensor from (8). A specific behavior can be activated or deactivated by explicitly defining the desired state of the corresponding sensor or not. New behaviors can be acquired by adding new sensors to the system. More complex behaviors can be constructed either by sequencing or by integrating available basic behaviors.

5. Experiments

Without loss of generality, the robot head shown in Fig. 3 is taken as an example to test the proposed method. The artificial vestibular system of the robot head is simplified and is assembled with a low cost two-axis tilt sensor and an electronic compass module. They are fixed on the top of the robot head and used to sense the roll, pitch, and yaw of the head motions as shown in Fig. 3b. The visual system consists of two color CCD cameras with PAL TV output. Video signals are grabbed and processed by a workstation. These two sensory systems are integrated within a binocular architecture. Two stepper motors are used to drive the pan axis and the tilt axis (see Fig. 3d).

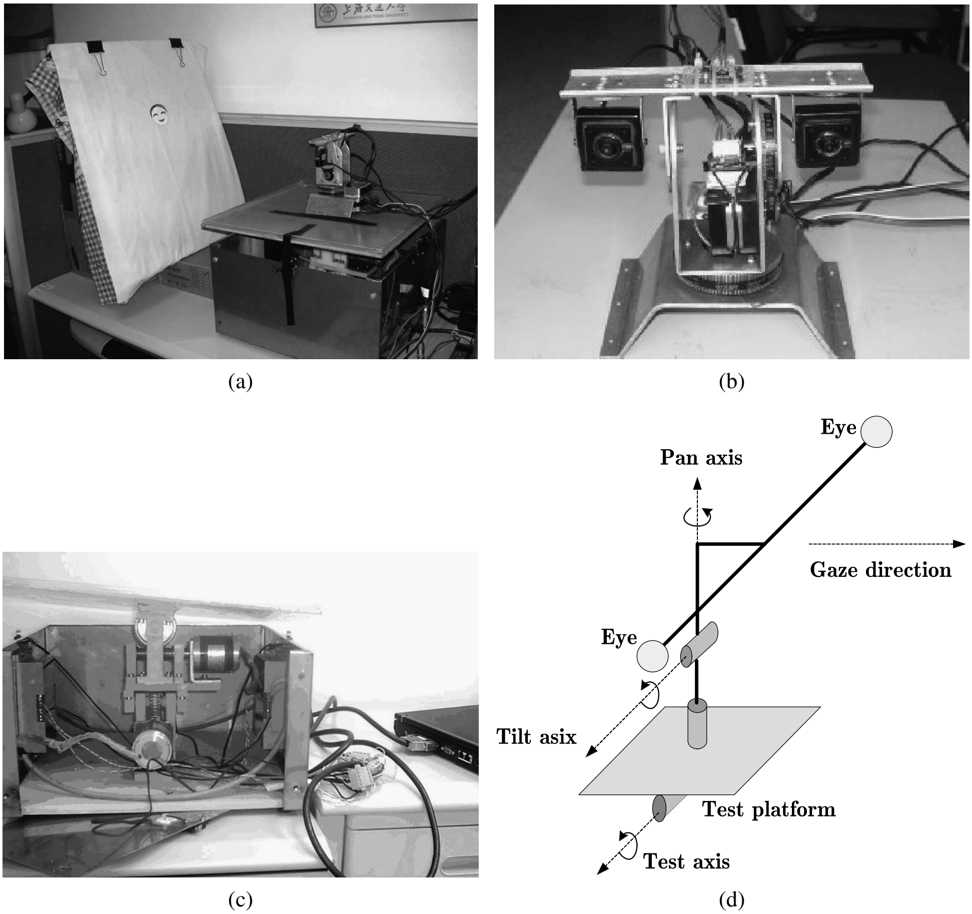

(a)
(b)
(c)
(d)

Figure 3. Experimental environment, robot head and coordinate systems. (a) Photo of the experiment environment. (b) Photo of the robot head. (c) Photo of the test platform. (d) The coordinate systems for the robot head.

In the experiments, we use three kinds of sensors and implement three sensor agents, i.e., the attention-detecting agent, the retinal slip agent and the vestibular agent. The current state and the desired state of the attention-detecting sensor are denoted by $\mathbf{y}_\mathrm{a}$ and $\mathbf{r}_\mathrm{a}$, respectively. They are the current and the desired positions of the attention point in current image. In experiments we choose the center of a cartoon human face as the attention point. The cartoon human face is detected by a boosted cascade of simple features method [31, 32]. It is performed every 500 ms by the workstation. Generally, retinal slip can denote both a position error and the velocity of an image on the retina. We explicitly use it as a position error. In our experiments, 10 feature points with large eigenvalues are selected in the 50 × 50 image whose center is the most recently detected attention point. Then they are tracked based on the sparse iterative method version of Lucas–Kanade optical flow in pyramids [33]. The retinal slip is the mean position error of all feature points.

The current state and the desired state of the retinal slip sensor are denoted by $\mathbf{y}_\mathrm{r}$ and $\mathbf{r}_\mathrm{r}$, respectively. The calculation of the retinal slip is performed every 50 ms by the workstation. The vestibular sensor's current state is the output of the tilt sensor and the electronic compass module, which is read every 33 ms. Its current state and desired state are denoted by $\mathbf{y}_\mathrm{v}$ and $\mathbf{r}_\mathrm{v}$, respectively.

Denote the head posture by $\mathbf{P}_\mathrm{h}(t)$, the posture of the head's base by $\mathbf{P}_\mathrm{b}(t)$ and the position of the attention point by $\mathbf{p}(t)$, then:

$$\mathbf{y}_\mathrm{a} = \mathbf{f}_\mathrm{a}(\mathbf{p}(t), \mathbf{P}_\mathrm{h}(t)) \tag{28}$$

$$\mathbf{y}_\mathrm{r} = \mathbf{f}_\mathrm{r}(\mathbf{p}(t), \mathbf{P}_\mathrm{h}(t)) \tag{29}$$

$$\mathbf{y}_\mathrm{v} = \mathbf{f}_\mathrm{v}(\mathbf{P}_\mathrm{h}(t)) \tag{30}$$

$$\mathbf{P}_\mathrm{h}(t) = \mathbf{g}(\mathbf{q}, \mathbf{P}_\mathrm{b}(t)), \tag{31}$$

where $\mathbf{q}$ are the joint variables. Thus, the differential relations are:

$$\dot{\mathbf{y}}_\mathrm{a} = \mathbf{J}_\mathrm{a}\dot{\mathbf{q}} + \delta_\mathrm{a} \tag{32}$$

$$\dot{\mathbf{y}}_\mathrm{r} = \mathbf{J}_\mathrm{r}\dot{\mathbf{q}} + \delta_\mathrm{r} \tag{33}$$

$$\dot{\mathbf{y}}_\mathrm{v} = \mathbf{J}_\mathrm{v}\dot{\mathbf{q}} + \delta_\mathrm{v}, \tag{34}$$

where $\mathbf{J}_\mathrm{a}$, $\mathbf{J}_\mathrm{r}$ and $\mathbf{J}_\mathrm{v}$ are the Jacobian matrices of the attention-detecting agent, the retinal slip agent and the vestibular agent, respectively. The coordinate systems for the head are shown in Fig. 3d. The pan axis and the tilt axis are not parallel, so $\mathbf{J}_\mathrm{v}$ is a 3×2 full rank matrix. In general, $\mathbf{J}_\mathrm{a}$ and $\mathbf{J}_\mathrm{r}$ are both 2×2 matrices and non-singular. Thus, all above matrices satisfy the condition of (27) and they could be used in the proposed multisensor integration way.

In the following experiments, the head's base is fixed on the top of a test platform (Fig. 3c). The test platform rotates back and forth on the test axis, which generates the external perturbation of the head. Only the left camera is used and a 320×240 image is acquired. Owing to the existence of delay in the control loop and unmodeled dynamics, large feedback gains may cause system instability. Therefore, the gain matrix $\mathbf{K}_\mathrm{c}$ in the control agent is assigned to a 2×2 diagonal matrix with all diagonal elements of 0.1. Without loss of generality, only tilt rotation movement of the head is considered for simplicity hereafter.

5.1. Jacobian Matrix Online Estimation

In the first experiment, both the test platform and the cartoon human face are static, while the head moves voluntarily. The voluntary movement is generated randomly with the highest frequency less than 1.5 Hz and the amplitude less than 150 steps as shown in Fig. 4. During the voluntary movement of the head, each sensor feedback is recorded and the Jacobian matrix in each sensor agent is estimated online. The initial condition, $\mathbf{P}(0)$, in each sensor agent is an identity matrix. For simplicity of expression, only one element of the mentioned matrix is shown in the following figures. As shown in Fig. 5, every *a priori* estimate error covariances matrix, $\mathbf{P}(k)$ in each sensor agent, tends to be zero and every Jacobian matrix is nearly invariable after 20 s.

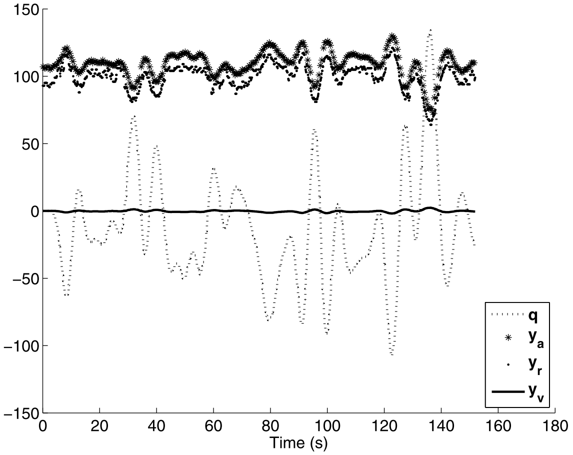


Figure 4. The voluntary movement of the head and each sensor feedback. $\mathbf{y}_a$ and $\mathbf{y}_r$ are both in pixels, $\mathbf{y}_v$ is in degrees, and $\mathbf{q}$ is in steps.

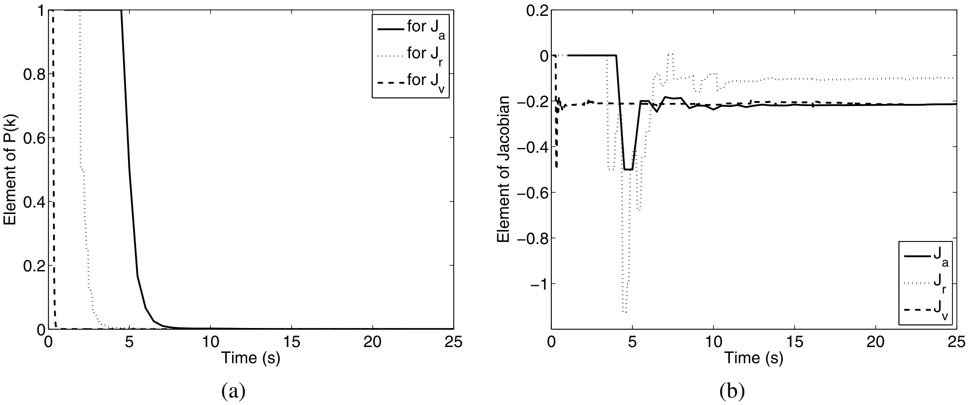


Figure 5. Jacobian matrix online estimation. (a) $\mathbf{P}(k)$ during Jacobian matrix estimation. (b) Jacobian matrices during estimation.

5.2. Gaze Control with Multisensor Integration

After the Jacobian matrix is nearly invariable in the attention-detecting agent, let $\mathbf{r}_a = [160, 120]^T$. When the test platform or the cartoon human face is moving, gaze control behavior can be observed. When the attention point is fixed in front of the head, let the test platform generate the motions of the head's base as shown in Fig. 6. The gaze control errors are recorded and shown in Figs 7–9. In order to compare the gaze control performance before and after multisensor integration, the gaze control errors' standard deviations, as a measure of spread, are compared. Before multisensor integration, the gaze control errors' standard deviation is about 8.526 pixels. After the multisensor integration, it drops to 4.348 and 3.147 pixels, respectively.

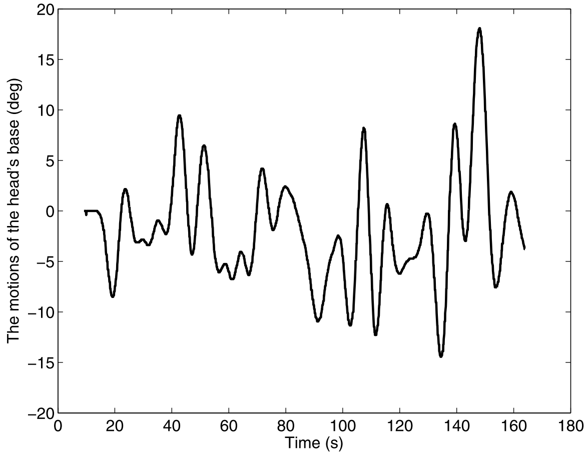


Figure 6. Motions of the head's base caused by the test platform.

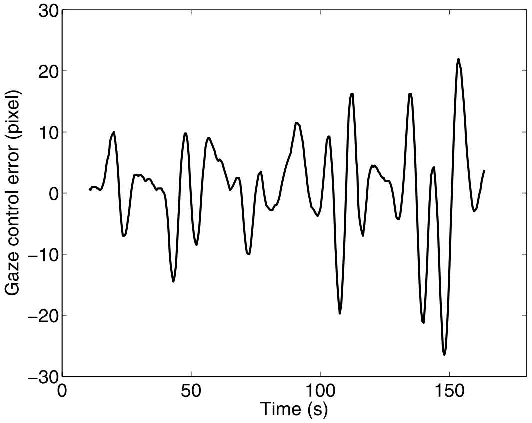


Figure 7. Gaze control before multisensor integration.

5.3. Head Posture Control with Multisensor Integration

After the Jacobian matrix is nearly invariable in the vestibular agent, let $\mathbf{r}_\mathrm{v} = [0, 0, 0]^\mathrm{T}$. When the test platform generates the same motions of the head's base as shown in Fig. 6, the posture control behavior of the head can be observed. The posture control errors are recorded and shown in Figs 10–12. The posture control errors' standard deviations, as a measure of spread, before and after multisensor integration are compared. Before multisensor integration, the head posture control errors' standard deviation is about 0.318°. After the vestibular sensor is integrated with the retinal slip sensor and the attention-detecting sensor, it drops to 0.217° and 0.210°, respectively.

The three experiments above show that the proposed method can generate gaze control and the control of head posture online by setting the corresponding sen-

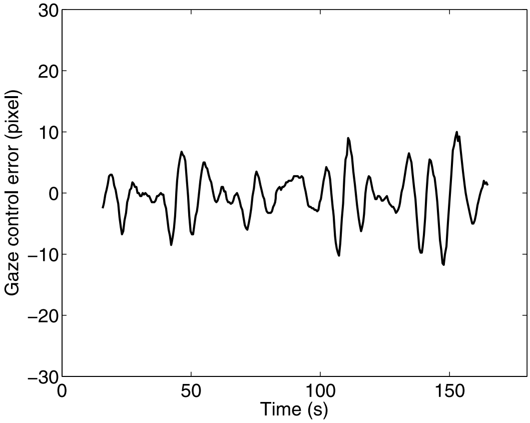


Figure 8. Gaze control after the integration of the attention-detecting sensor with the retinal slip sensor.

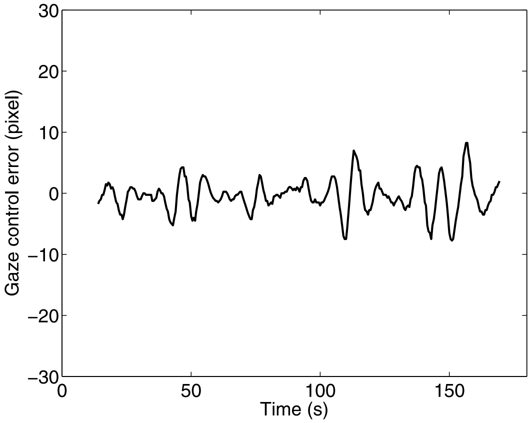


Figure 9. Gaze control after the integration of all sensors.

sor's desired state. They also demonstrate that the proposed multisensor integration method improves the control performance of the two behaviors.

5.4. More Benefits of the Multisensor Integration

Finally, we will further explore the advantages of the proposed multisensor integration in the robustness against some sensor failures and its computational simplification. The attention point detection is fragile to luminance change and image distortion. The attention-detecting sensor may fail to detect the attention point occasionally as we have seen during our experiments. In this experiment, we use a piece of white paper to cover a small part of the cartoon face occasionally. It will make the attention-detecting sensor fail to detect the attention point more frequently. The gaze control errors are compared before and after integration. Figures 13 and 14 show that the gaze control errors are reduced after the attention-detecting

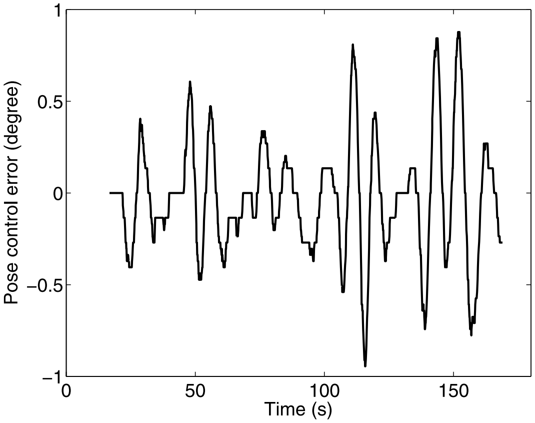


Figure 10. Posture control before multisensor integration.

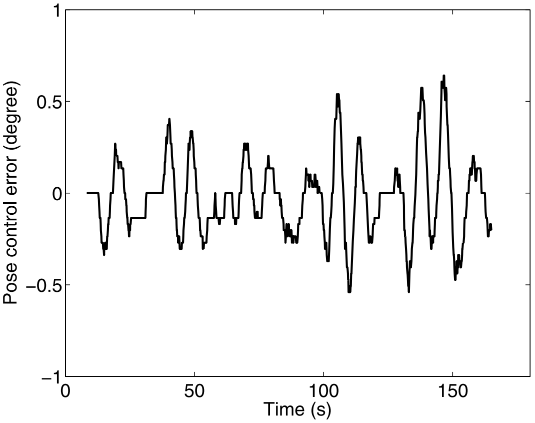


Figure 11. Posture control after the integration of the vestibular sensor with the retinal slip sensor.

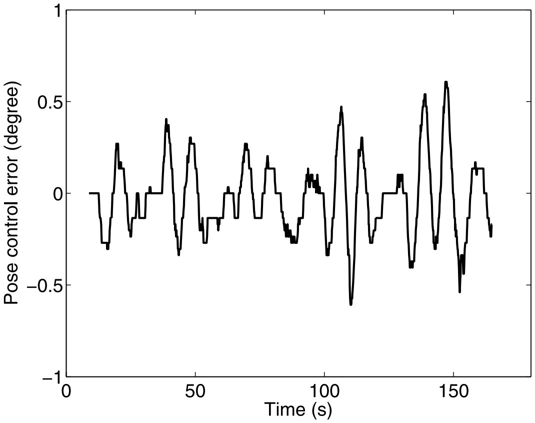


Figure 12. Posture control after the integration of all sensors.

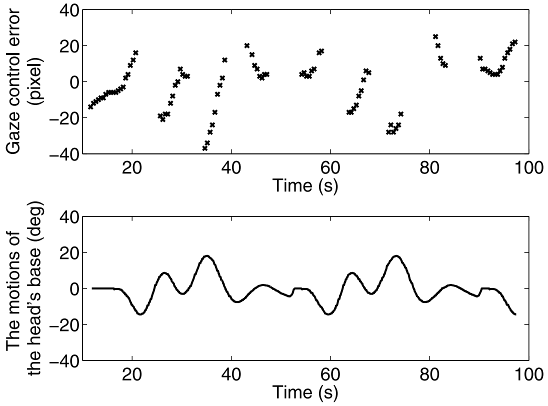


Figure 13. Gaze control before multisensor integration.

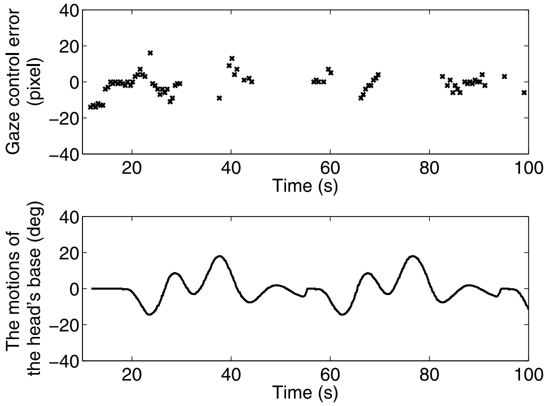


Figure 14. Gaze control after multisensor integration.

sensor integrated with the other two sensors. As only a small part of the cartoon face is covered, most feature points can still be tracked by the retinal slip sensor. The vestibular sensor can always track the motions of the head. Thus, when the attention-detecting sensor fails to detect the attention point, the position of the attention point can still be estimated from the current states of the retinal slip sensor and the vestibular sensor. This results in smaller gaze control errors. In this experiment, the attention-detecting sensor is computationally more intensive than the others. If we reduce the attention-detecting frequency deliberately, the system could achieve the same performance as before by the proposed multisensor method, but using less computation.

6. Conclusions and Future Work

Our research goal is to investigate a mechanism to acquire basic behaviors and multisensor integration for a more complex humanoid robot. This paper takes a first step towards this goal by exploring gaze control and the control of head posture. We demonstrate that Jacobian matrices cannot only be used to describe the sensorimotor correlation and then generate basic behaviors, but they can also be used to integrate acquired behaviors. Moreover, they can also correlate different sensors' feedback and then be used to realize multisensor integration. We propose the multi-agent-based implementation to realize the flexibility of the proposed method itself. A system identification method, RLS, is used to estimate the Jacobian matrix online. Online estimation makes the system free from advance calibration between the sensors and the motors, and makes the system robust against some sensor and motor failures. Multisensor integration improves the performance of gaze control and the control of head posture. It can also improve the robustness against some sensor failures and be used to reduce the overall computation for a given task.

Sensory latency and control delay normally exists in the proposed system, which will make the estimation of the Jacobian matrix, the system control and the multisensor integration more complex. How to cope with the delay and latency will be our further work. As in the conventional image Jacobian matrix approach, the convergence and stability of the proposed method is also under investigation.

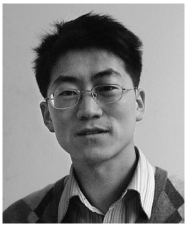
References

1. R. A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robotics Automat.* **2**, 14–23 (1986).
2. M. Mataric, Integration of representation into goal-driven behavior-based robots, *IEEE Trans. Robotics Automat.* **8**, 304–312 (1992).
3. R. C. Arkin, *Behavior-Based Robotics*. MIT Press, Cambridge, MA (1998).
4. H. Liu, A fuzzy qualitative framework for connecting robot qualitative and quantitative representations, *IEEE Trans. Fuzzy Syst.* **16**, 1522–1530 (2008).
5. S. Bovet and R. Pfeifer, Emergence of delayed reward learning from sensorimotor coordination, in: *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Edmonton, pp. 2272–2277 (2005).
6. M. Cambron and R. Peters II, Determination of sensory motor coordination parameters for a robot *via* teleoperation, in: *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*, Tucson, AZ, vol. 5, pp. 3252–3257 (2001).
7. P. Maes and R. A. Brooks, Learning to coordinate behaviors, in: *Proc. Nat. Conf. on Artificial Intelligence*, Boston, MA, pp. 796–802 (1990).
8. S. Mahadevan and J. Connell, Automatic programming of behavior-based robots using reinforcement learning, *Artif. Intell.* **55**, 311–365 (1992).
9. M. Asada, S. Noda, S. Tawaratsumida and K. Hosoda, Purposive behavior acquisition for a real robot by vision-based reinforcement learning, *Mach. Learn.* **23**, 279–303 (1996).
10. M. Asada, E. Uchibe and K. Hosoda, Cooperative behavior acquisition for mobile robots in dynamically changing real worlds *via* vision-based reinforcement learning and development, *Artif. Intell.* **110**, 275–292 (1999).
11. I. Suzuki, H. Yokoi and Y. Kakazu, Emergence of adaptive behavior by chaotic neural networks, in: *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation*, Kobe, vol. 1, pp. 151–156 (2003).

12. S. Bovet, Emergence of insect navigation strategies from homogeneous sensorimotor coupling, in: *Proc. 9th Int. Conf. on Intelligent Autonomous Systems*, Tokyo, pp. 525–533 (2006).
13. N. Chaumont, R. Egli and C. Adami, Evolving virtual creatures and catapults, *Artif. Life* **13**, 139–157 (2007).
14. C. Adami, Digital genetics: unravelling the genetic basis of evolution, *Nature Rev. Genet.* **7**, 109–118 (2006).
15. C. M. Brown, Gaze controls with interactions and delays, in: *Proc. Workshop on Image Understanding*, San Francisco, CA, pp. 200–218 (1989).
16. B. W. Peterson and F. J. Richmond, *Control of Head Movement*. Oxford University Press, Oxford (1988).
17. L. A. Olsson, C. L. Nehaniv and D. Polani, From unknown sensors and actuators to actions grounded in sensorimotor perceptions, *Connect. Sci.* **18**, 121–144 (2006).
18. M. Marjanović, B. Scassellati and M. Williamson, Self-taught visually-guided pointing for a humanoid robot, in: *Proc. 4th Int. Conf. on Simulation of Adaptive Behavior*, Cape Cod, MA, pp. 35–44 (1996).
19. P.-E. Forssén, Learning saccadic gaze control *via* motion prediction, in: *Proc. 4th Can. Conf. on Computer and Robot Vision*, Montréal, pp. 44–51 (2007).
20. F. Panerai, G. Metta and G. Sandini, Learning VOR-like stabilization reflexes in robots, in: *Proc. 8th Eur. Symp. on Artificial Neural Networks*, Bruges, pp. 95–102 (2000).
21. G. Grossman, R. Leigh, E. Bruce, W. Huebner and D. Lanska, Performance of the human vestibuloocular reflex during locomotion, *J. Neurophysiol.* **62**, 264–272 (1989).
22. M. C. Schubert and L. B. Minor, Vestibulo-ocular physiology underlying vestibular hypofunction, *Physical Ther.* **84**, 373–385 (2004).
23. F. Panerai, G. Metta and G. Sandini, Visuo-inertial stabilization in spacevariant binocular systems robotics and autonomous systems, *Robotics Autonomous Syst.* **30**, 195–214 (2000).
24. T. Shibata and S. Schaal, Biomimetic gaze stabilization based on feedback-error-learning with nonparametric regression networks, *Neural Network* **14**, 201–216 (2001).
25. T. Nagata, Y. Fukuoka, A. Ishida and H. Minamitani, Analysis of role of vision in human upright posture control, *Ann. Rep. Res. Reactor Inst. Kyoto Univ.* **2**, 1155–1158 (2001).
26. B. Espiau, F. Chaumette and P. Rives, A new approach to visual servoing in robotics, *IEEE Trans. Robotics Automat.* **8**, 313–326 (1992).
27. S. Hutchinson, G. Hager and P. Corke, A tutorial on visual servo control, *IEEE Trans. Robotics Automat.* **12**, 651–670 (1996).
28. T. Suehiro, H. Onda and K. Kitagaki, New method for integration of multi-sensor and multi-actuator robot system, in: *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Maui, HI, vol. 3, pp. 1299–1304 (2001).
29. K. Johan and B. Wittenmark, *Computer-Controlled Systems — Theory and Design*. Tsinghua University Press, Beijing (1997).
30. J. Qian and J. Su, Online estimation of image Jacobian matrix by Kalman–Bucy filter for uncalibrated stereo vision feedback, in: *Proc. IEEE Int. Conf. on Robotics and Automation*, Piscataway, NJ, vol. 1, pp. 562–567 (2002).
31. P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, in: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, HI, vol. 1, pp. I-511–I-518 (2001).
32. R. Lienhart and J. Maydt, An extended set of haar-like features for rapid object detection, in: *Proc. IEEE Int. Conf. on Image Processing*, Rochester, NY, vol. 1, pp. I-900–I-903 (2002).

33. B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in: *Proc. 7th Int. Joint Conf. on Artificial Intelligence*, Vancouver, pp. 674–679 (1981).

About the Authors



Chenggang Liu received the BS degree in Automation from Harbin Engineering University, Harbin, P. R. China, in 2000, and the MS degree in Control Theory and Control Engineering from Shanghai Jiao Tong University (SJTU), Shanghai, P. R. China, in 2003. He is currently a PhD candidate of Control Theory and Control Engineering at SJTU. His research interests include behavior-based robots, machine learning and biped walking control.

Jianbo Su received the BS degree in Control Theory and Control Engineering from Shanghai Jiao Tong University (SJTU), Shanghai, P. R. China in 1989, the MS degree in Pattern Recognition and Intelligent Systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, P. R. China, in 1992, and the PhD degree in Control Theory and Control Engineering from Southeast University, Nanjing, P. R. China, in 1995. From 1995 to 1997, he was a Postdoctoral Research Fellow in Robotics in the Chinese Academy of Sciences. He is currently a Professor in the Department of Automation, SJTU. His research interests include sensor-based robotics, multisensor data fusion, internet robotics, learning in robotics, computer vision and pattern recognition.