PRIFYSGOL ABERYSTWYTH UNIVERSITY

Aberystwyth University

Enhanced robotic hand-eye coordination inspired from human-like behavioral patterns

Chao, Fei; Zhu, Zuyuan ; Lin, Chih-Min; Hu, Huosheng ; Yang, Longzhi ; Shang, Changjing; Zhou, Changle

Published in:
IEEE Transactions on Cognitive and Developmental Systems

DOI:
10.1109/TCDS.2016.2620156

Publication date:
2016

Citation for published version (APA):
Chao, F., Zhu, Z., Lin, C.-M., Hu, H., Yang, L., Shang, C., & Zhou, C. (2016). Enhanced robotic hand-eye coordination inspired from human-like behavioral patterns. *IEEE Transactions on Cognitive and Developmental Systems*, *10*(2), 384-396. https://doi.org/10.1109/TCDS.2016.2620156

General rights
Copyright and moral rights for the publications made accessible in the Aberystwyth Research Portal (the Institutional Repository) are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the Aberystwyth Research Portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the Aberystwyth Research Portal

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

tel: +44 1970 62 2400
email: is@aber.ac.uk

Download date: 02. Mar. 2026

Enhanced Robotic Hand-eye Coordination inspired from Human-like Behavioral Patterns

Fei Chao, *Member, IEEE,* Zuyuan Zhu, Chih-Min Lin, *Fellow, IEEE,* Huosheng Hu, *Senior Member, IEEE,* Longzhi Yang, *Member, IEEE,* Changjing Shang, and Changle Zhou

***Abstract*—Robotic hand-eye coordination is recognized as an important skill to deal with complex real environments. Conventional robotic hand-eye coordination methods merely transfer stimulus signals from robotic visual space to hand actuator space. This paper introduces a reverse method: Build another channel that transfers stimulus signals from robotic hand space to visual space. Based on the reverse channel, a human-like behavior pattern: "Stop-to-Fixate", is imparted to the robot, thereby giving the robot an enhanced reaching ability. A visual processing system inspired by the human retina structure is used to compress visual information so as to reduce the robot's learning complexity. In addition, two constructive neural networks establish the two sensory delivery channels. The experimental results demonstrate that the robotic system gradually obtains a reaching ability. In particular, when the robotic hand touches an unseen object, the reverse channel successfully drives the visual system to notice the unseen object.**

***Index Terms*—Robotic hand-eye coordination, sensory motor mapping, human-like behavioral pattern, constructive neural network.**

I. Introduction

Scientists desire to create autonomous robots or agents that can work under complex unstructured and changing environments [1]. Compared with traditional robots, grounded on conventional artificial intelligence, research on "developmental robotics" focuses on building autonomous learning abilities by following human developmental procedures [2]–[4]. Recent studies also indicate that human-like behavioral patterns reduce robotic learning complexities and increase learning speed [5], [6]. Therefore, this work focuses on applying human-like behavioral patterns and developmental learning methods to enhance the robot's hand-eye coordination ability.

Robots with multiple-joint arms, or manipulators, are widely used within both industry and daily life [7]. Robotic hand-eye coordination, or reaching ability, regarded as a basic cooperation of robotic eyes and hands/arms, is implemented by two radically different methods: (1) the mathematical approach, which employs forward or inverse kinematics [8]; and (2) the learning approach [9], which uses mainly artificial neural networks. The mathematical approach is especially suitable for static or industrial environments (see [10]); however, the learning approach brings more self-adaptive properties to robots. Furthermore, because robotic hand-eye coordination involves the most basic application of robotic internal representation [11]–[13], developing human-like reaching and grasping skills is regarded as an important stage to establish higher robotic cognitive processes [14]–[17].

Current research mainly focuses on how to map visual stimuli to hand space. Usually, motor-motor coordination performs such a mapping mechanism [17]. In addition, much research suggests skills and knowledge be transferred from human beings to robots so as to reduce learning complexity [18]–[22]. Several other approaches have focused on applying artificial neural networks to build the robotic learning system [9], [23]. Meng et al. [24], [25] proposed an improved minimal resource allocation network model to build a robot mapping system to bring more developmental features into robots. Pierris and Kajic et al. used the Self-Organization Map algorithm to establish the robotic internal representation for robotic hand-eye coordination [26], [27]. On the other hand, Huelse et al. implemented a common spatial reference frame to coordinate visual input and arm postures and movements [28].

However, several important human reaching features (such as human automatic response actions) are not in the above research. In particular, if a human's hand touches an obstacle when performing a reaching movement, the human will cease hand movement and fixate on hand position [29]. This "Stop-to-Fixate" pattern may detect unexpected dangers and allow the human time to consider how to deal with the dangers. If a robot were endowed with this ability, its ability to survive would increase. Moreover, a human retinal structure brings many benefits, including fast saccade movements, simplified image processing, and so on [30]. However, these benefits are seldom involved in current robotic hand-eye coordination studies.

In contrast with the current research, this paper presents an enhanced robotic hand-eye coordination system. This system is able to transfer stimuli from eye to hand and from hand to eye. Thus, the human-like behavioral pattern, "Stop-to-Fixate", is established in the robot by using bi-directional stimulus delivery channels. In addition, a retina-like visual

F. Chao, C.-M. Lin, H. Hu, and C. Zhou are with the Cognitive Science Department, School of Information Science and Engineering, Fujian Province Key Lab of Machine Intelligence and Robotics, Xiamen University, China e-mail: (fchao@xmu.edu.cn). C.-M. Lin is also with the Department of Electrical Engineering, Yuan Ze University, Taiwan. H. Hu is also with the School of Computer Science and Electronic Engineering, University of Essex, UK.

Z. Zhu is with the School of Computer Science and Electronic Engineering, University of Essex, UK.

L. Yang is with the Department of Computer Science and Digital Technologies, Faculty of Engineering and Environment, Northumbria University.

C. Shang is with the Department of Computer Science, Aberystwyth University.

This work was supported by the Major State Basic Research Development Program of China (973 Program) (No. 2013CB329502), the Fundamental Research Funds for the Central Universities (No. 20720160126), and the National Natural Science Foundation of China (No. 61673322, 61673326, 61673328, 61402386, 61305061, 61203336, and 61273338).

mechanism and an incremental computational learning system are used to support the robotic hand-eye coordination ability. The enhanced robotic hand-eye coordination system extends the original idea [31] that works exclusively with the "Stop-to-Fixate" pattern and bi-directional delivery channels, thus allowing an enhanced robotic hand-eye coordination ability.

The remainder of this paper is organized as follows: Section II introduces the background and related work of robotic hand-eye coordination and constructive neural networks. Section III explains the system architecture and the implementation methods used in this architecture. Section IV describes the experimental results and discusses the results. Section V concludes the work and points out important future work.

II. Background and Related Work

A. Robotic and Human Hand-eye Coordination

How to establish the stimulus transformation from robotic visual space to robotic actuator space is the most principle part of robotic hand-eye coordination [17]. Usually, a robot firstly tries to detect any salient objects in images, which are captured by the robot's visual sensor. Then, the robot's hand-eye coordination system maps the visual stimulus information to the hand space. A corresponding place in the hand space is activated by the stimulus. After that, the deviation between the corresponding place and the current hand position is applied to drive the robotic arm/hand to move. Therefore, the robotic hand-eye coordination is converted to find the relationships that link the visual space to hand motor space.

Human reaching movements also involve the "eye-to-hand" pattern to guide hand actions. A human usually fixates on a target before his/her arm moves towards the target. However, humans own another behavioral pattern that leads humans to focus on their hands. Therefore, if the human's hand is disturbed by an obstacle when he/she performs a reaching movement, the human will stop the movement and turn his/her attention to what the hand is touching. At this moment, the human's eyes will fixate on his/her hand's current position. Then, the human will explore the obstacle. If the obstacle cannot draw the human's interest or attention, the human will continue the reaching movement.

Sann and Streri's findings support that humans own a type of intermodal object perception presented at birth [32]; this intermodal perception performs a bidirectional transfer between human's sight and touch. Therefore, a human's hand-eye coordination system contains a bi-directional reference frame that handles both "eye-to-hand" and "hand-to-eye" coordination. Human hand-eye coordination is established within the first two years after birth [11]. In addition, human infants apply spontaneous movements (also called "motor babbling" [33]) to gradually generate precise reaching functions. In particular, the bi-directional mappings are built simultaneously, rather than separately.

B. Human Saccade Movements

Unlike cameras, the human retina is not uniform; however, it consists of different sensor characteristics across its layout. The periphery of the retina is a region containing low accuracy, which are very sensitive to object changes or movements. The central region covers about 10 degrees of the visual field and the fovea, and this region provides the greatest acuity and color sensitivity [34]. Following the human retinal structure, a retina-like visual system is applied in this paper; in particular, the field density to be higher in the central area than the periphery. Each field in the periphery covers more image pixels; therefore, image information in the peripheral area is compressed.

In addition, Butko et al. suggested that the preference of human newborns to saccade towards faces is not innate, but can be learned very rapidly [35]. On the other hand, at age three months, an organized and robust set of hand-eye coordination behaviors begins to emerge [4]. The production of skilled reaching movements is a long-term achievement that takes two or three years to own. Therefore, this paper has made an assumption: Performing saccadic eye movements is an innate ability of our robot. With this assumption, our robot will focus on using the retina-like visual system and robot's hand spontaneous movements to produce human-like behavior patterns.

C. Constructive Neural Networks for Incremental Learning

The geometry of both a human's and a robot's body has complex kinematics and visual sensors that generate visual distortions, which cause the hand-eye mapping to be highly non-linear [36]. Because artificial neural networks contain excellent non-linear approximation ability, the robotic hand-eye mappings used various types of artificial neural networks [37]. For example, a double neural network structure mimicking the working loops between the basal ganglia and cerebra was adopted to create a robot system, which can handle reaching movements by using long and short movements [9].

Scientists from developmental robotics suggest using constructive neural networks to implement robotic information process systems [38]. During the training phase of such networks, the network structure is increasing. In particular, this increments allow for qualitative as well as quantitative growth. Such a growth phenomenon is very similar to the growth of human infant brain neurons when infants learn to represent their knowledge [38]. Therefore, constructive neural networks are more suitable for developmental robotics than other static topology networks, such as Back-Propagation or Radius-Based-Function networks. Furthermore, several types of constructive neural networks, e.g. [25], [39], have been used already to build robotic hand-eye coordination systems. However, the Locally Weighted Projection Regression (LWPR) network exhibits better non-linear approximation ability [39].

Recent research indicates that the LWPR network [40] is the most popular and powerful learning method for robots to learn to control their mechanisms [39]. The LWPR performs both input projection and incremental regression of local linear models simultaneously [39], [41]. Especially, the LWPR network is suitable for robotic incremental learning with redundant or sparse data in high dimensional spaces [42]. In addition, the LWPR networks are widely used to implement robotic internal representations, e.g. [3], [43], [44].

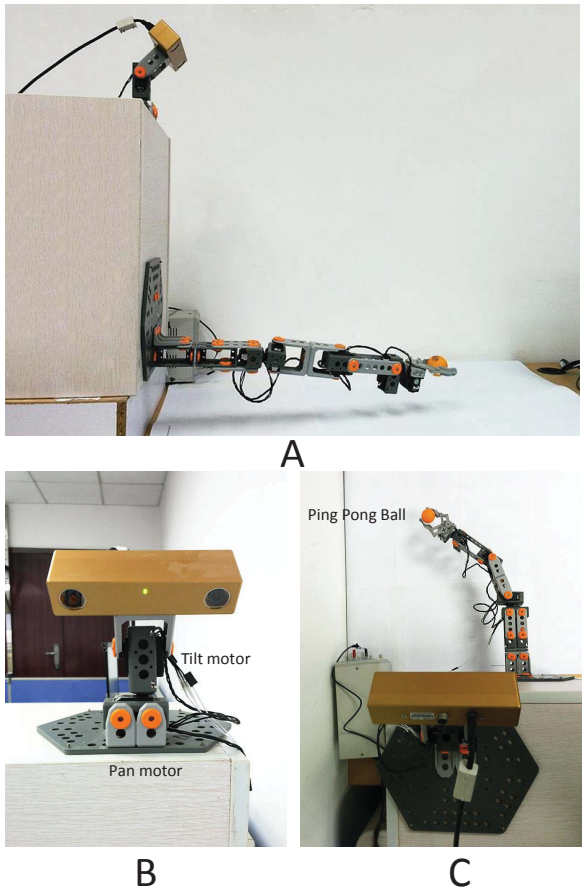


Fig. 1. The robot platform. Picture A illustrates the entire hand-eye coordination platform. Picture B shows the motorized camera system. Picture C shows the top view of the platform.

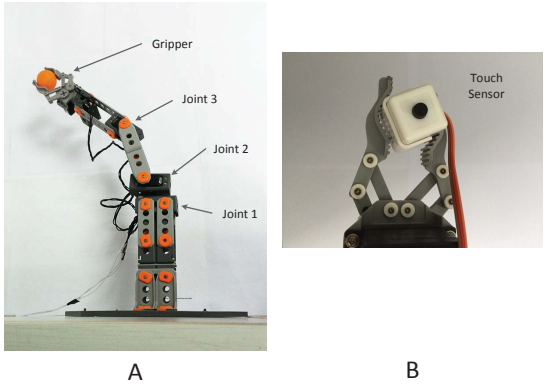


Fig. 2. The top-down view of the robot's arm. Picture A shows the positions of the three joints and the gripper. Picture B illustrates the touch sensor mounted in the gripper.

III. The Methods

A. The Human-like Robotic Hand-eye Coordination System

Figs. 1 and 2 show the experimental platform, which consists of a manipulator arm and a motorized camera system. The manipulator is installed on a vertical board (See Fig. 1-A); the camera is installed on top of the vertical board (Shown in Fig 1-B). The manipulator has five servo motors; however,

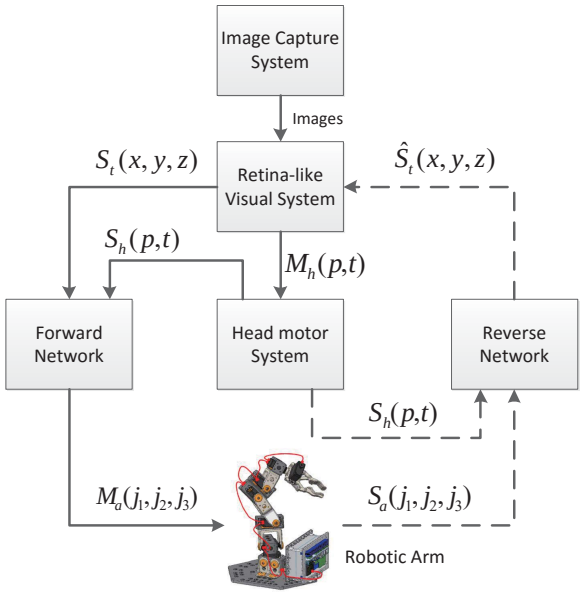


Fig. 3. The architecture of the human-like reaching system with two channels.

only three motors (labeled Joints 1, 2, and 3 in Fig. 2-A) are used in the experiments. Joint 1 rotates the whole arm. Joints 2 and 3 control the arm so that it moves in a plane. A gripper, holding an orange ping pong ball (See Fig. 1-C), is mounted on top of the arm. However, when the robot performs human-like reaching movements, the ping pong ball is replaced by a touch sensor (See Fig. 2-B). The sensor detects an object when the sensor moves close to the object.

The "Bumblebee 2" camera used in the platform provides three-dimensional space information; the camera also produces two-dimensional images. The Bumblebee 2 camera supplies 640×480 images at 48 frame-per-second (FPS) or 1024×768 images at 20 FPS. In this paper, the setup of 1024×768 images at 20 FPS is used.

Fig. 3 shows the human-like reaching architecture of the robotic system. The architecture contains two channels built by two LWPR networks: One channel transfers eye-to-hand stimuli; the other transfers reversely. The entire architecture consists of six sub-systems as follows: (1) image capture system; (2) retina-like visual system; (3) head motor system; (4) forward network; (5) reverse network; and (6) robotic arm.

The image capture and the retina-like visual systems imitate a human's visual system to process captured image information. The head motor system transforms visual location information to the head motor's joint values, which drives the robotic eye to gaze at desired positions. The forward network maps the visual information to the robotic arm's joint values. Inversely, the reverse network transforms the robotic arm joints values to the visual proprioceptive information. The robotic arm works as an actuator to simulate a human's arm.

For the eye-to-hand channel, (solid lines and arrows in Fig. 3), the robotic eye captures the target's position $S_t(x, y, z)$, where x, y and z represent the three-dimensional position in retina coordination, and perceives joint values $S_h(p, t)$ of the head motor, where S indicates preceptive values, p represents the joint value of the eye in the pan direction, and t represents

the eye's joint value in the tilt direction. Then, $S_t(x, y, z)$ and $S_h(p, t)$ are delivered into the forward network to predict the arm joint values $M_a(j_1, j_2, j_3)$ (j_1, j_2, and j_3 represent the labeled Joints 1, 2, and 3 in Fig. 2-A, respectively) that drives the robotic arm to the corresponding target position.

For the hand-to-eye channel, (dashed lines and arrows in Fig. 3), the arm's joint values $S_a(j_1, j_2, j_3)$ of the touched position and the current head's joint values $S_h(p, t)$ are input into the reverse network. The output of the reverse network is the object position $\widehat{S}_t(x, y, z)$ in the retina-like visual system. The hand-to-eye channel is activated only when the robotic arm touches an object. The details of each sub-system are specified in the following paragraphs.

1) Image Capture System: The image capture system captures images from the robot's workspace. A stereo vision camera, "Bumblebee 2", is used to build the image capture system. The camera produces a three-dimensional position for each pixel inside the captured images. In addition, the camera provides a balance between three-dimensional data quality, processing speed, and image size.

2) Retina-like Visual System: Fig. 4 shows the structure of the retina-like visual system. The visual system, inspired by our previous work [34], produces an initial processing on the images received from the image capture system. This retina-like visual system is widely used in many developmental robotic models [17], [45], [46].

To simplify the robot's implementation, the retina-like structure used in this paper contains the following three parts: (1) the central area, (2) the inner area, and (3) the outside area. The central area is located in the structure's center; the central area is not covered by retina cells (circles in Fig. 4). Moreover, the central area does not contain motor information. In contrast, both the inner and outside areas contain motor information to produce saccade movements. However, only the central area detects the three-dimensional positions, $S_t(x, y, z)$ of the target objects. The inner area consists of twelve retina cells (circles with smaller size in Fig. 4); each cell contains the head's motor value. The outside area also contains twelve retina cells (circles with larger size). Because the inner cell's size is smaller than that of the outside area cell, the saccadic movements generated by the inner area are more precise. Similarly, the cell size of the outside area is larger, the saccadic movements generated by the outside area are less precise.

The creation of the retina-like structure is based on the following equation:

$$d_i = d_o(\mu^i - 1) \tag{1}$$

where, d_i is the distance from the retina structure center to each retina cell's center; i denotes the indices of the inner and outside areas; $i \in [2, 3]$; $i = 2$ is the inner area, and $i = 3$ is the outside area. The parameters d_o and μ are used to set the size and position of each cell. In this paper, d_o and μ are empirically set at 586.52 and 1.1055, respectively.

For each retina cell's center angle β: each area contains twelve cells; thus, the angle difference between two adjacent cells is $30°$. Thus, β is defined by:

$$\beta_j = \frac{2\pi}{12} j = \frac{\pi}{6} j \tag{2}$$

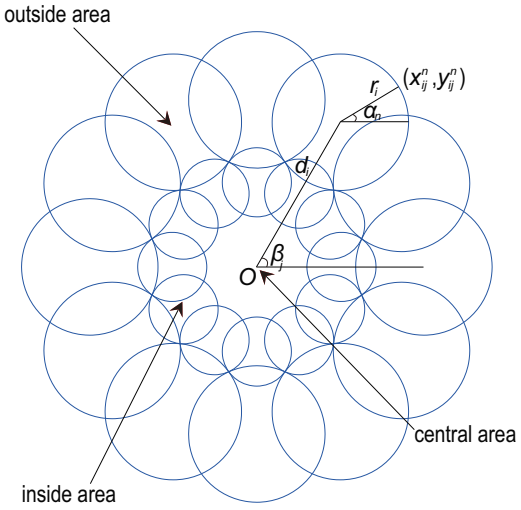


Fig. 4. The retina-like visual system with three areas: the central area, the inside area and the outside area. d is the distance from the retina structure center to each retina cell's center. β denotes each retina cell's center angle. α is the stepped size for drawing the structure.

where, j is the index of each cell. In each area, twelve cells are used in this paper; therefore, $j = [1 : 12]$. Each cell's radius is based on d_i, thus, the radius of each retina r_i is defined as:

$$r_i = d_i \pi c \tag{3}$$

where, c is the overlapping parameter, which defines the size of the overlapping area between two adjacent cells. c is empirically set at 0.13.

(x_{ij}^n, y_{ij}^n) represents the coordinates of the cell circle point. The following equations are applied to define each cell's boundary of the retina-like structure:

$$\begin{cases} x_{ij}^n = r_i \cos\alpha_n + d_i \cos\beta_j \\ y_{ij}^n = r_i \sin\alpha_n + d_i \sin\beta_j \end{cases} \tag{4}$$

where, n is the cell circle point index, α is the stepped size for drawing the structure, $\alpha \in [0, 2\pi]$ and $\Delta\alpha = \frac{\pi}{50}$.

3) Head Motor System: A motorized pan/tilt system builds the head motor system. The head motor system (1) perceives the pan/tilt motor positions, $S_h(p, t)$, and (2) drives the pan and tilt motors to move by following received motor values, $M_h(p, t)$. The pan/tilt system's motor encoders obtain p and t. Then, these motor values are retained in the retina-like visual system. In this work, the arm and head motors use pulse-width as position feedback and control value units. Therefore, the above two values are numerically equal. However, in order to distinguish perceptive (or sense) and motorial values, both $M_h(p, t)$ and $S_h(p, t)$ are used in this paper. $M_h(p, t)$ indicates only motor values, and $S_h(p, t)$ indicates only perceptive values.

4) Forward Network: The forward network maps the visual stimulus information to the robotic arm joint values. The network receives a target's three-dimensional coordinates, $S_t(x, y, z)$, and the head's joint values, $S_h(p, t)$. If the forward network has been fully trained, the network generates the robotic arm's desired joint values, $M_a(j_1, j_2, j_3)$. Then, the

robotic arm system uses the joint values to move the arm to the target position. A LWPR network is used to build the forward network.

5) Reverse Network: The reverse network applies the robotic head's positions and the robotic arm's positions to generate the robotic hand's projection in the visual space. The perceptive head joint values, $S_h(p,t)$ and the arm joint values, $S_a(j_1,j_2,j_3)$ are set as the network's input. The perceptive target's location within the visual coordinates, $\widehat{S}_t(x,y,z)$ is set as the network's output. In this case, when the robotic arm touches an object, if the reverse network has been fully trained, the robot perceives the object's position in visual space. Another LWPR network is used to build the reverse network.

6) Network Implementation: In this work, two LWPR networks are used to build the forward and reverse networks. The LWPR algorithm uses locally linear models, spanned by a small number of univariate regressions in selected directions in input space. A locally weighted variant of Partial Least Squares is employed for dimensionality reduction.

For each LWPR network, a weighting kernel is used to determine the locality of the input data. The kernel is defined in the following steps: For each data point, $\boldsymbol{x_i}$, the distance between each local unit's center, $\boldsymbol{c_k}$ and the data point is used to calculate the data point's weight value, $w_{k,i}$. If a Gaussian kernel is applied to the LWPR network, $w_{k,i}$ is denoted by:

$$w_{k,i} = \exp(-\frac{1}{2}(\boldsymbol{x_i} - \boldsymbol{c_k})^{\boldsymbol{T}} \boldsymbol{D_k}(\boldsymbol{x_i} - \boldsymbol{c_k})) \tag{5}$$

where, the positive distance metric, $\boldsymbol{D_k}$, determines the region's size and shape, which are updated during learning to match the training data.

The LWPR approximation is the sum of local linear models weighted by their respective Gaussian weights. The LWPR network integrates M local linear models to make predictions. When an input vector, $\boldsymbol{x_i}$, is given, each linear model calculates a prediction, y_k. The total output of the network is the weighted mean of all linear models as follows:

$$\hat{y} = \frac{\sum_{k=1}^{M} w_k y_k}{\sum_{k=1}^{M} w_k} \tag{6}$$

The training phase of the LWPR network training involves adjustments of projection direction, local regression parameters, and each local model's distance metrics $\boldsymbol{D_k}$. In particular, each local model is optimized individually based on the distance metrics. An incremental gradient descent algorithm is used to update the values of $\boldsymbol{D_k}$.

On the other hand, the LWPR network contains an incremental learning mechanism that allocates new locally linear models automatically as required. An outline of the algorithm is shown in Pseudo-code 1. In this pseudo-code, w_{gen} is a threshold, which is used to determine when to add a new local model to the network; and D_{def} is the initial distance metric in Eq. 5; and r denotes the initial number of projections in a local model. The root mean-squared error (RMSE) of a LWPR network is simply used to determine whether r should be increased during the training phase. If the error at the next projection is not larger than a certain percentage of the previous error:

$$\frac{error_{i+1}}{error_i} > \phi, \tag{7}$$

where ϕ is the ratio of the next error and previous error, and $\phi \in [0,1]$; the LWPR network will stop adding new projections to the local model.

Algorithm 1 The LWPR Training Procedure.

```
while training sample(x, y) is novel do
  for k = 1 to RF do
    calculate the activation from Eq. 5
    update the means of input and output
    update the local model
    update Distance Metrics
  end for
  if no linear model was activated by more than w_gen; then
    create a new RF with r = 2, c = x, D = D_def
  end if
end while
```

The LWPR network increases its local models and projections automatically during its training phase. If no existing local model elicits response (activation) greater than w_{gen}, a new local model is generated. Otherwise, no local model is generated. In this paper, w_{gen} is empirically set at 0.5. In order to achieve the best performance of the LWPR network, various values of w_{gen} have been tested. Based on the results, 0.5 is the best value for the LWPR networks used in this paper. On one hand, larger values increase the complexity of the network structure, so as to increase the training time. On the other hand, smaller values cause the network's poor performances. The initial number r of projections of each new model is the network's default value, 2.

7) The Robot's Arm: The robot's arm contains five joints; however, only three of them are used in this work. The three joints are sufficient to perform reaching movements in three-dimensional workspace. The arm receives motor values, $M_a(j_1,j_2,j_3)$ from the forward network and drives the arm to the corresponding point. In addition, the arm's inside encoders return the joints' current place, $S_a(j_1,j_2,j_3)$, to the reverse network.

B. The Robot's Learning System

The previous section introduced the basic human-like reaching system's architecture; however, the robot must learn its reaching ability from scratch. Therefore, the robot will generate a large number of spontaneous movements to gather sufficient training data for learning. The training data generation flowchart of the forward and reverse networks is illustrated in Fig. 5. The whole process contains five steps as numbered in the flowchart. The overall procedure is as follows: The retina-like visual system attempts to find any salient stimulus within the robotic workspace; if the stimulus is not caused by the robotic arm, the head system will fixate the salience; the current robotic head position and hand position are perceived

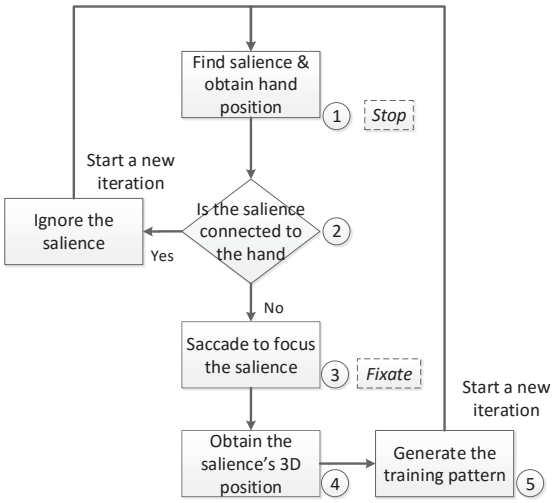


Fig. 5. The training data generation flowchart of the robot's learning system. The whole process contains five steps as labeled.

and retained for training the forward and reverse networks. The details of the five steps are as follows:

Step 1: Find any salient objects within captured images, and then, obtain the robotic arm's position. At this moment, only the robotic arm appears within the robot's view. Therefore, only the robot's arm causes salience. An orange ball is embedded in the robot's fingers; thus, the eye detects only the end of the robot's arm (fingertip). In the training phase, the robot must generate a number of spontaneous, random movements to generate enough training data. The fingertip's visual position, $\widehat{S}_t(x, y, z)$ and hand current joint values, $S_a(j_1, j_2, j_3)$, are obtained from the robot.

Step 2: Determine whether the salience is connected to the robot's arm. If the robot determines the salience is caused by its hand, rather than an object, the robot ignores the salience. Otherwise, if the robot does not recognize the salience as the arm, the robot continues the training process to move to the next step. Because only the robot's hand appears within the robot's view during the training phase, if the reverse network is not fully-trained, the robot will always regard its hand as salient.

The algorithm sends the hand's current position, $S_a(j_1, j_2, j_3)$, and the head's position, $S_h(p, t)$, to the reverse network. The output of the reverse network is the hand's proprioceptive position within the visual structure, $\widehat{S}_t(x, y, z)$. Then, the distance between $S_t(x, y, z)$ and $\widehat{S}_t(x, y, z)$ is used to determine whether the robot causes the salience.

$$\xi = \|S_t(x, y, z - \widehat{S}_t(x, y, z)\| \tag{8}$$

If ξ is less than a threshold, the hand causes salience, the robot ignores the salience, and the procedure returns to Step 1; on the other hand, if ξ is greater than a threshold, the salience is caused by the arm's new position that the robot cannot identify, and the procedure proceeds to Step 3. The threshold is empirically set at 20 in this paper.

Step 3: Use a saccadic function to focus the salience. When the head locates the salience, the eye will gaze on it. The saccadic ability is built before the robot starts to learn the hand-eye coordination. Therefore, in this work, the saccadic function is regarded as a robot's predefined (innate) ability. The use of saccades was established in our previous work (See [34] for further details).

Step 4: Obtain the salience's three-dimensional position. After the saccadic movement, the salience appears in the central area of the retina-like structure. The eyeball motor system detects the relative position of the salience based on the camera's current position. The central area provides the three-dimensional position that the inner and outside areas cannot provide. This structure (See Fig. 4), inspired by that of the human eye, reduces the visual system's complexity.

Step 5: Generate the training pattern. When all of the head's joint values, the salience's three-dimensional coordinates, and the arm's joint values are obtained. These information is retained in a temporal memory. Next, the robotic arm spontaneously moves to a new position and Steps 1 to 5 are repeated until the movement limit of the robot's arm is reached.

The collected training data sets in the temporal memory are used to train the two LWPR networks. Our current work applies an off-line training method to learn the collected data (See the experimental section for details).

C. Robot's Software and Hardware Control Structure

Fig. 6 shows the robot's hardware control structure, which contains two controllers: a control computer and a servo controller. The control computer holds the functions including the depth image capture, the retinal-like structure, the two constructive networks, and the network training data set. In addition, the control computer contains a USB socket to connect to the servo controller.

The joint motor control is contained in the servo controller, rather than in the control computer. From the control computer, the servo controller receives commands that are translated into a format that the arm and head's motors can process. In particular, the servo controller detects the joint values of the arm and the head and sends these values back to the control computer. The USB socket handles the communication between the control computer and the servo controller.

The head and arm's joint parameters are list in Table I. The servo motors use pulse-width (pw) as position unit; one unit approximately equal to $0.35°$. The Pan motor drives the camera to move horizontally, and the $Tilt$ motor moves vertically. In addition, because J_2 and J_3 are limited in proper working ranges, the arm cannot generate the joint redundant situation.

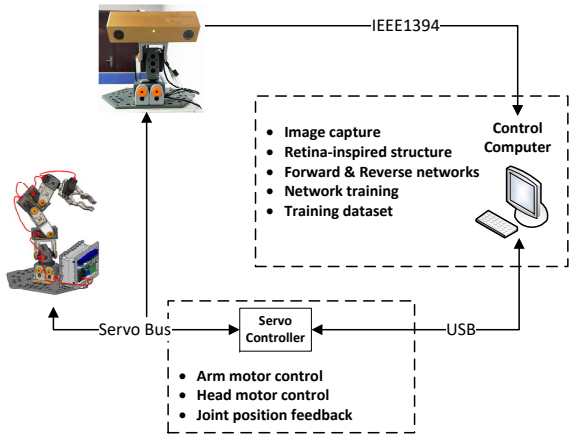


Fig. 6. The robot's software and hardware control structure. The robot's hardware control structure contains a control computer and a servo motor controller.

TABLE I
THE JOINT LIMITATIONS OF THE ARM AND PAN-TILT SYSTEM.

Joint	Min (pw)	Max (pw)
J_1	430	535
J_2	430	674
J_3	372	808
Pan	453	616
$Tilt$	651	698

IV. EXPERIMENTATIONS

A. Experimental Results and Analysis

As shown in Fig. 3, the enhanced robotic hand-eye coordination system consists of two networks, both of which are trained by the LWPR algorithm. Therefore, the experiments firstly generate the training data for the two networks. The experimental procedure is as follows: The robotic arm randomly performs spontaneous movements, and the related information is collected to train the forward and reverse networks.

TABLE II
THE LWPR NETWORK PARAMETERS USED IN THIS PAPER.

Parameters	Forward Network	Reverse Network
nIn	6	5
nInStore	6	6
nOut	3	4
n_data	4	2
diag_only	1	1
meta	0	0
meta_rate	250	250
penalty	1.0E-6	1.0E-6
init_alpha	250	250
norm_in	1	1
norm_out	1	1
init_D	50	50
init_M	7.071	7.071
w_gen	0.2	0.2
w_prune	1	1
init_lambda	0.999	0.999
final_lambda	0.99999	0.99999
tau_lambda	0.9999	0.9999
init_S2	1.0E-10	1.0E-10
add_threshold	0.5	0.5
kernel	LWPR_GAUSSIAN	LWPR_GAUSSIAN
update_D	1	1

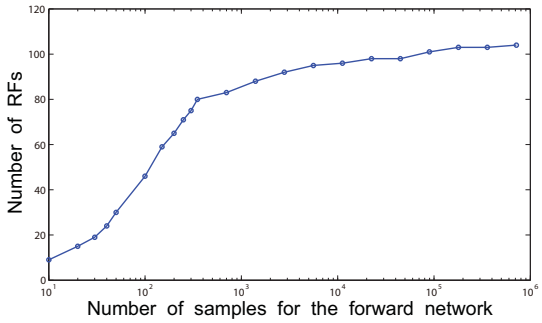


Fig. 7. The increasing curve of LWPR receptive fields in the forward network. The number of receptive fields grows rapidly during the first 300 iterations. The total account of the receptive fields is 108.

The C++ implementation of the LWPR algorithm, described in [47], is used in this paper. The LWPR model parameters are given in Table II. The input and output values of the forward and reverse networks are different. For the forward network, input is in the form of a five-value array whose elements contain the position of a target in gaze space, $S_t(x, y, z)$ and the position of the head $S_h(p, t)$; output is in the form of a three-value array that corresponds to the arm's joint position, $M_a(j_1, j_2, j_3)$.

For the reverse network, input is also in the form of a five-value array whose elements contain the arm's position, $S_a(j_1, j_2, j_3)$ and the position of the head, $S_h(p, t)$; output of the reverse network is in the form of a three-value array data corresponding to the salience's position, $\widehat{S}_t(x, y, z)$, within the robotic visual system.

In the experiment, the robot performs 350 spontaneous (random) movements in total; thus, 350 training samples are collected. 300 samples are used as the training set; the remaining 50 samples are used for testing.

Fig. 7 illustrates the increasing curve of LWPR's receptive fields (RFs) in the forward network. The x and y axes denote the number of iterations of the training set and the number of the network's receptive fields, respectively. In Fig. 7, the number of RFs grows rapidly during the first 300 iterations. However, after 300 iterations, the number of RFs increases slowly. The initial rapid increase in the number of RFs implies that (1) the training set of three hundred samples covers almost the entire data range, and (2) almost all of the locally linear models are formed by using those samples. When the number of iterations reaches $44,800$, the LWPR network becomes very saturated. The total account of the RFs is 108. In order to test whether the 300 samples cover the entire workspace, more samples (about 400) are used to train the forward network; however, the number of receptive fields converges to the same position with that of the network using 300 samples.

Fig. 8 shows the output error of the forward network. The output error is the Root Mean Squared Error (RMSE) between the robot's current hand position and the expected position. The output error of each training sample is obtained by:

$$e = \sqrt{\frac{\sum_{i=1}^{3}(j_i - \hat{j}_i)}{3}} \tag{9}$$

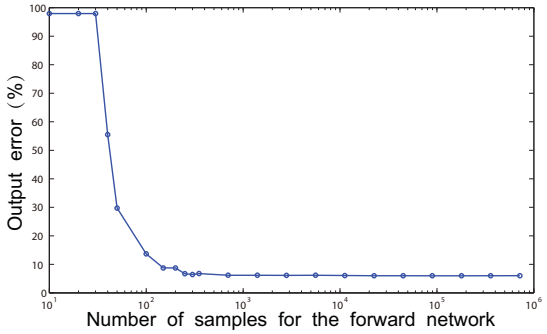


Fig. 8. The output error of forward network. The output error declines to a low level after around 600 iterations.

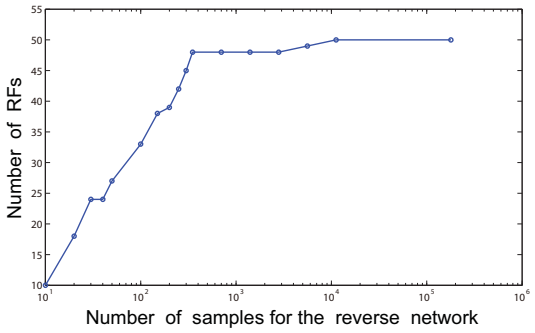


Fig. 9. The increase of LWPR receptive fields in the reverse network. The total account of the receptive fields is 50.

where, j_1, j_2, j_3 denote the expected three joint values of the robot arm; $\hat{j}_1, \hat{j}_2, \hat{j}_3$ denote the network's real output for the three joints. By the time the iterations of the forward LWPR network reach 600, the output error has declined from an initial level of almost 100% to a relatively low level ($< 10\%$), and remains at this level as the number of iterations increases. Regarding the LWPR algorithm, if the network's output error value is larger than 100%, the network's output will always be displayed as 100% as shown for the first 30 iterations in Fig. 8.

Fig. 9 illustrates the increasing curve of RFs in the reverse network. Compared with the number of RFs in the forward network (Fig. 7), the forward network contains more receptive fields. The forward network has more than 100 fields; in contrast, the reverse network has only 50 fields. The potential reason for this situation is that the reverse network only starts to function when stimuli are not in the central area of the retina-like structure; therefore, the retina-like structure does not require the reverse network to produce very accurate results. In this case, the computational cost of the reverse network is not heavy: only 50 receptive fields are sufficient for the reverse network. Therefore, the retina-like structure is able to reduce the training complexity of the reverse network.

Fig. 10 shows the evolution of the forward network's output error. Axes x and y in Fig. 10 denote the arm gripper's position in visual coordination (because the depth information for the retina-like structure is not used, the depth value is ignored in this figure). Axis z denotes the error values, expressed in $\%$. Each input of the forward network contains the three

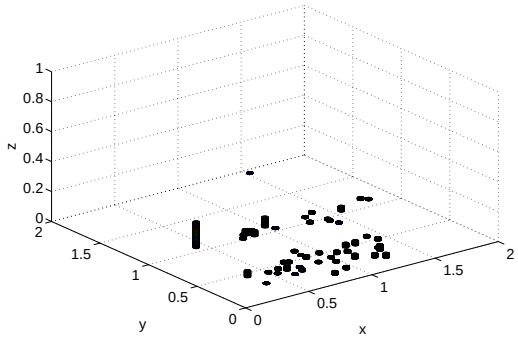


Fig. 11. The test result of the reverse network with 50 samples.

values, (x, y, z); to clearly demonstrate the error distribution, the figure uses only (x, y) values. Another reason to abandon the z value is that Joint, J_1, to a small range, so there are only a few number of different z values.

Fig. 10 contains nine stages (a to i). In each stage, the size of the training sample is twice that of the previous phase. By Stage c, the error is already at a very low level. In addition, almost all of the samples are very accurate; only several samples in the boundary area cause higher error. However, this situation cannot be regarded as a defect in the approach since the robotic arm has only a few chances to enter the boundary area; therefore, errors in the boundary cause many inaccurate reaching movements. Moreover, the error falls very quickly at the beginning of the iterations; for example, in Stage d, the value reaches a low level. However, the error is still larger than the expected error threshold; thus, the iterations does not cease until Stage i.

Fig. 11 shows the test results of the reverse network. The reverse network used in this figure went through $44,800$ iterations. For the 50 test samples, 96% samples' error values are less than 0.05; therefore, the network's performance is acceptable. Figs. 10 and 11 also imply that the reverse network requires only a few iterations to converge. Note that: the training and testing samples in Figs. 10 and 11 exhibit strip-like distributions. In faction, the random spontaneous movements almost cover the entire workspace. However, the pan/tilt system contains a hardware intervene problem; thus, several points in the workspace cannot be fixated by the robot's visual system. These un-fixated points are not involved in the training or testing samples.

Fig. 12 shows one complete human-like reaching movement. The first row of pictures is taken by a camera, mounted on the top of the entire workspace. The second row of pictures is the robot's captured images. The bottom row of pictures show the salient objects (indicated by a group of white points within a white square) and the visual central area (indicated by a white circle in each picture). First, an orange ball and a glass bottle are placed in the workspace. The robot can detect only the orange ball. A touch sensor is mounted on top of the arm to replace the orange ball used in the training phase. When the touch sensor detects an object, the robot's arm will

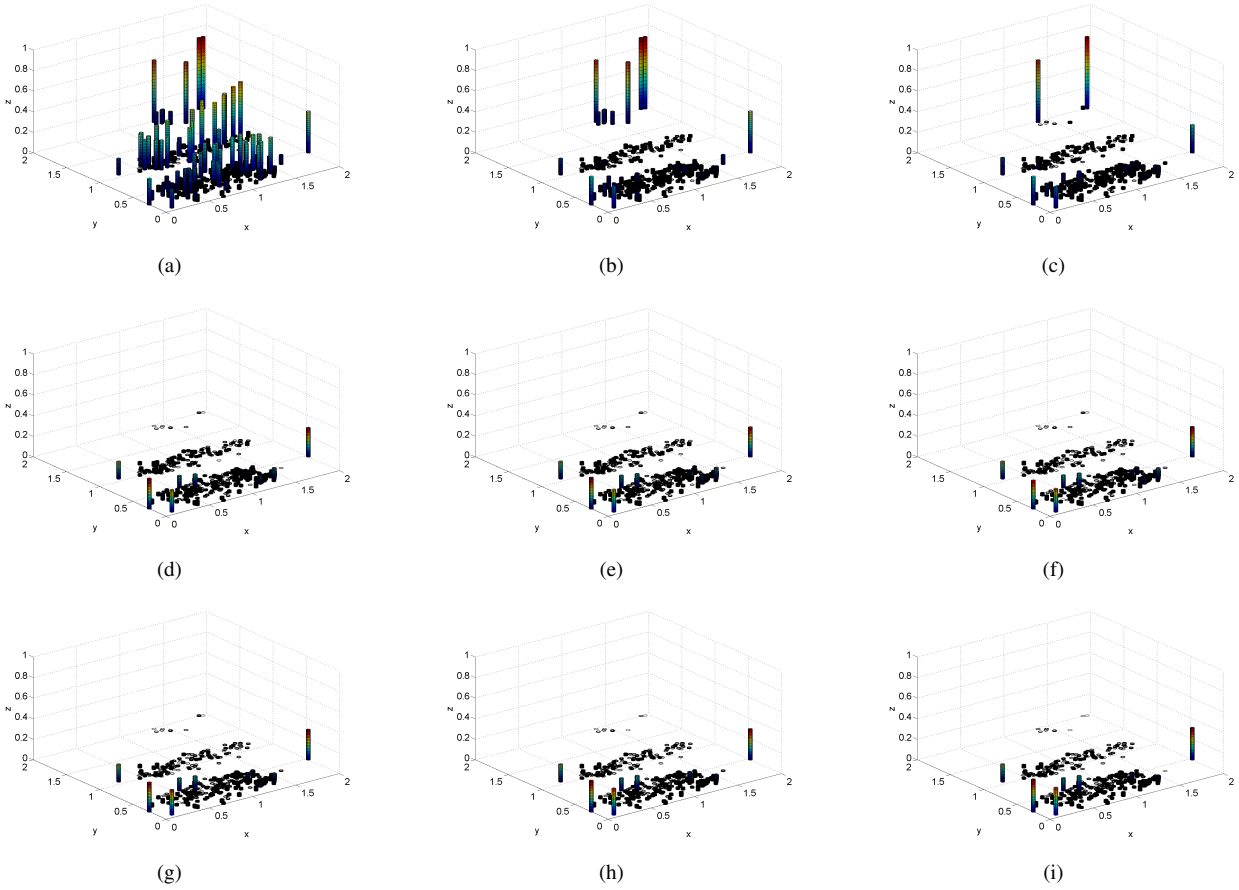


Fig. 10. The evolution progress of the forward network's output errors. The progress contains nine stages (a to i), each sub-figure shows the output error of each stage. In Stage d, the more than 90% training sample's error values are less than 5%.

stop moving. The reaching movement contains the following three steps:

Step 1: The robot detects a salience (caused by the orange ball); then, the eye fixates on the salience (the salience is within the central area); and the robot attempts to move its arm towards the salience.

Step 2: Because the glass bottle cannot be detected by the robot's eye, and the bottle is in the arm's way to the ball, the touch sensor detects the bottle so as to stop the arm. At this moment, the eye is still fixating on the ball (the salience is still within the central area).

Step 3: The reverse network is activated, and the eye does not notice the ball any longer. After a saccadic movement, the eye fixates upon the hand; meanwhile, the bottle also appears within the central area of the eye (the salience is outside the central area). Thus, the robot successfully produces a human-like reaching movement, "Stop-to-Fixate."

In this paper, more complex environments have not been considered yet. For example, the obstacle has various shapes, or is placed in different positions. Such complex environments require more hand-eye coordination movements, so as to perform a more intelligent obstacle-avoidance ability. To achieve this ability, the robot's retina-like visual system can take use of its central area, in which more visual information of the obstacle is obtained, to determine the shape and boundaries of the obstacle. In addition, retina-like visual system also use the visual system's inside and outside areas and the forward network to find a new movement trajectory to avoid the object.

Such more complex environment situation will deepen this research and exhibit our approach's capabilities against existing methods. Therefore, this situation leads our further important research.

B. Discussions and Comparison

The experimental results described above demonstrate how the robot learns to map from its visual stimuli to its hand and from its hand back to its visual space, as well as how the human-like reaching movements are successfully generated. The experimental observations show that the proposed approach is successful in generating a robot's reaching ability. It is necessary to compare our system with existing methods. However, it is very difficult to compare the existing work by evaluating each model's performance indicators; since each robotic hand-eye coordination model consists of different Degrees-of-Freedom, joint accuracy, or network parameters. However, four qualitative facts derived from the key principles of developmental robotics are used to build the comparison, so as to emphasize the main contributions of this paper. The four facts – 1) Hand-eye transformation method; 2) Human

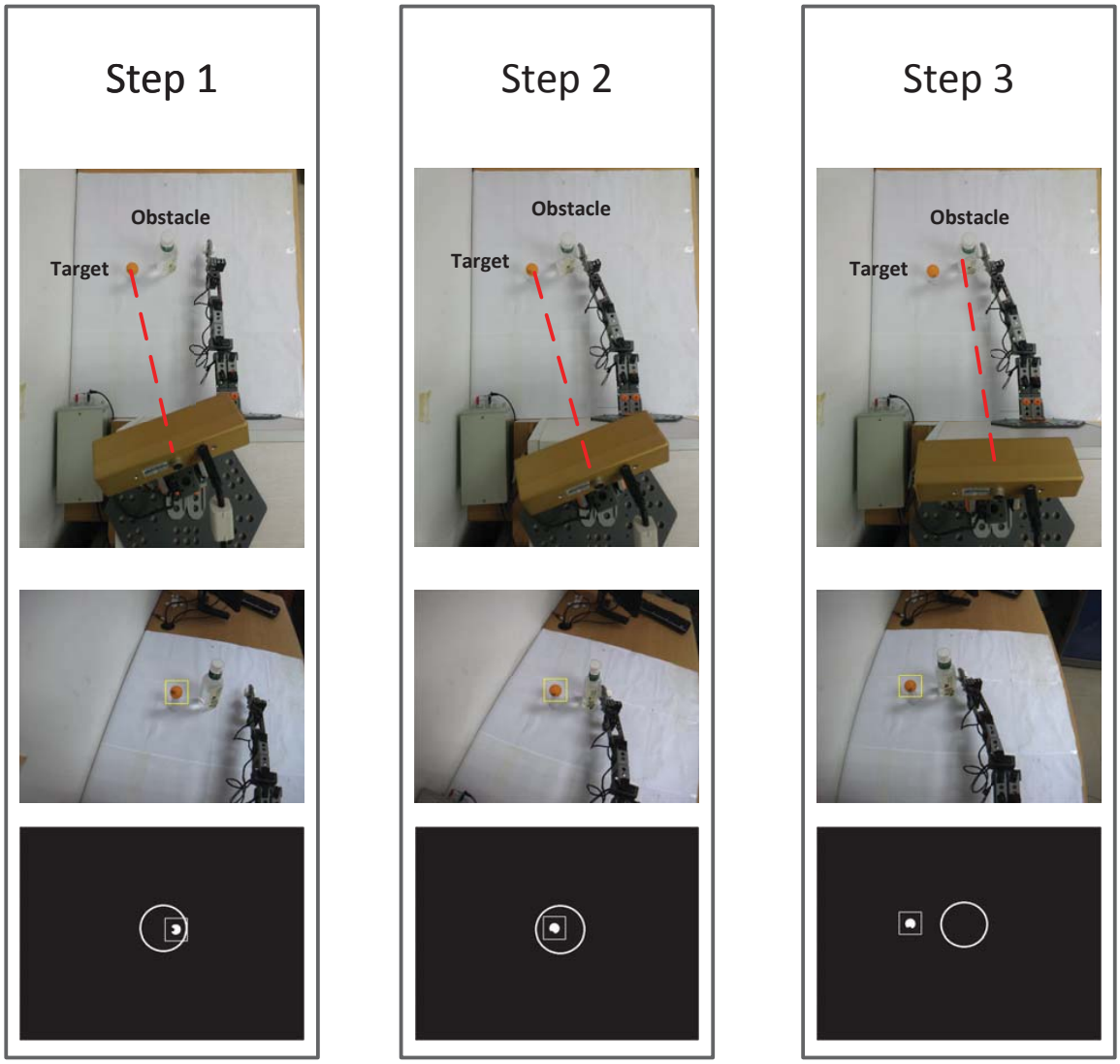


Fig. 12. The human-like reaching behaviorial pattern behaved by the robot. The first row of pictures show the top view of the entire workspace. The second row of pictures are captured by the robot. The bottom row of pictures show the salient objects and the visual central area. The the three columns illustrate one reaching movement's three steps.

behavioral pattern; 3) Autonomous object-exploring behavior; and 4) Incremental learning – are discussed as follows.

Almost all the existing research on robotic hand-eye coordination fails to mention how to build the mechanism for mapping hand sensorimotor space back to the visual space. Existing research focuses on applying various solutions to solve the non-linear mapping problem for the hand-eye coordination. Our previous work [17] developed a two-dimensional mapping mechanism, which can be expanded to have a certain level of reverse mapping ability. Similarity, Huelse et al.'s research also developed a common spatial reference frame, they claimed the frame supports transformation from the reaching space to the gaze space [28]. However, these work merely focused on two-dimensional reaching and might be very difficult to apply to three-dimensional hand-eye coordination.

In addition, human behavioral pattern is able to bring higher flexibility and better strain capacity to robots, while existing research [48]–[50] only follows robot hand's trajectories. Humans, recognized as the most evolved species, possess a powerful interaction ability with the natural world. Even several small automatic response movements, sometimes ignored by humans themselves, are generated in a long term evolution. Such automatic response movements protect humans from danger. Therefore, our purpose is to impart human behavioral patterns to a robot so as to enhance the robot's ability to survive within complex daily life. Our robot's "Stop-to-Fixate" pattern therefore is to mimic the human automatic response movement; and the robot's retina-like visual system reduces the computational cost of image processing.

Moreover, in this paper, if a robot does not possess a reverse network, the robot cannot find obstacles. In this case, the robot may lose opportunities to explore obstacle's properties, such as textures and appearances. Thus, our work is also useful to enable robots to develop an autonomous exploring ability. In addition, if a visual memory is developed into the robotic system, the robot will also own an obstacle-avoid capability.

TABLE III
SUMMARY OF THE COMPARISON WITH THE EXISTING APPROACHES

Options:	**Existing approaches:**	**Our approach:**
Hand-eye transformation method	almost all existing research applied only one channel, except [28]	two channels with bidirectional mapping directions
Human behavioral pattern	robot's vision merely follows robot hand's trajectories [48]–[50]	follows robot's and trajectories, performs the "Stop-to-Fixate" pattern and contains the human retina-like visual mechanism
Autonomous object-exploring behavior	few current research contains this feature	can find obstacles during its movements, and will further guide to develop the object-exploring behavior
Incremental learning	a small number of models exhibit this feature in learning system [39], [44], [51]	both the forward and reverse neural networks are incremental

Furthermore, only a small number of robotic hand-eye coordination models exhibit incremental learning mechanism in their neural network controllers [39], [44], [51]. In contrast, both the forward and reverse neural networks are incremental in this work. These four facts are belonged to the key principles of developmental robotics; several existing work contains only one or two facts within each model [4]. In contrast, our work contains more developmental principles within one system. These comparisons are summarized as in Table III.

V. CONCLUSIONS

In this paper, a developmental approach was successfully created for a robot to establish sensory-motor relationships between the robot's visual view and the robot's hand, so as to obtain the robot's hand-eye coordination with automatic response actions. Different from existing work, in our approach, a reverse transformation from the robot actuators space to the robot visual space was built. A retina-like visual system was used to obtain saccadic ability. Two LWPR networks were applied to implement the forward and reverse networks. When the training phase was completed, the robot was able to perform reaching movements; meanwhile, the human-like behavioral pattern "Stop-to-Fixate" was also established.

There is still room to improve the present work.

- In particular, the robot's arm is constrained to a predefined workspace; the joints redundancy problem that may affect the LWPR's learning results is simplified in the current work. Therefore, further efforts are required to solve the joint redundancy situation.
- In addition, the present work merely applies two LWPR networks to generate the human-like behavioral pattern; however, more human developmental learning approaches, e.g. "Lift-Constrain, Act, and Saturate" method [44], [52] can be incorporated into the work to contain higher human-like intelligence.
- Furthermore, this paper's work focuses on implementing the reverse network to bring a human-like behavioral pattern when the robot detected an obstacle; however, several important issues like how to deal with the obstacle with different shapes or how to avoid the obstacle are not considered. Therefore, further research on using the human-like behavior patterns to explore and avoid obstacles are required.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their invaluable comments and suggestions, which greatly helped to improve the presentation of this paper.

REFERENCES

[1] B. Doroodgar, Y. Liu, and G. Nejat, "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2719–2732, Dec 2014.

[2] M. Asada, K. Hosoda, Y. Kuniyoshi, T. Ishiguro, H. Inui, Y. Yoshikawa, M. Ogino, and C. Yoshida, "Cognitive developmental robotics: A survey," *IEEE Transactions on Autonomous Mental Development*, vol. 1, no. 1, pp. 12–34, 2009.

[3] A. Jennings and R. Ordonez, "Unbounded motion optimization by developmental learning," *IEEE Transactions on Cybernetics*, vol. 43, no. 4, pp. 1178–1188, Aug 2013.

[4] A. Cangelosi and M. Schlesinger, *Developmental Robotics: From Babies to Robots*, ser. Intelligent Robotics and Autonomous Agents. The MIT Press, 2015.

[5] F. Chao, Z. Wang, C. Shang, Q. Meng, M. Jiang, C. Zhou, and Q. Shen, "A developmental approach to robotic pointing via human-robot interaction," *Information Sciences*, vol. 283, pp. 288–303, 2014.

[6] X. Zhang, C. Zhou, M. Jiang, and F. Chao, "An approach to robot hand-eye coordination inspired by human infant development," *ROBOT*, vol. 36, no. 2, pp. 185–193, 2014.

[7] F. Chao, F. Chen, Y. Shen, W. He, Y. Sun, Z. Wang, C. Zhou, and M. Jiang, "Robotic free writing of Chinese characters via human robot interactions," *International Journal of Humanoid Robotics*, vol. 11, no. 1, pp. 1 450 007–1–26, March 2014.

[8] T. Zhou and B. E. Shi, "Learning visuomotor transformations and end effector appearance by local visual consistency," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 8, no. 1, pp. 60–69, March 2016.

[9] F. Chao, X. Zhang, H. Lin, C. Zhou, and M. Jiang, "Learning robotic hand-eye coordination through a developmental constraint driven approach," *International Journal of Automation and Computing*, vol. 10, no. 5, pp. 414–424, 2013.

[10] L. Jin and Y. Zhang, "G2-type srmpc scheme for synchronous manipulation of two redundant robot arms," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 153–164, Feb 2015.

[11] M. Lee, J. Law, and M. Hülse, "A developmental framework for cumulative learning robots," in *Computational and Robotic Models of the Hierarchical Organization of Behavior*, G. Baldassarre and M. Mirolli, Eds. Springer Berlin Heidelberg, 2013, pp. 177–212.

[12] A. Laflaquiere, A. Terekhov, B. Gas, and J. O'Regan, "Learning an internal representation of the end-effector configuration space," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 1230–1235.

[13] M. Antonelli, A. Gibaldi, F. Beuth, A. J. Duran, A. Canessa, M. Chessa, F. Solari, A. P. Del Pobil, F. Hamker, E. Chinellato *et al.*, "A hierarchical system for a distributed representation of the peripersonal space of a humanoid robot," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 4, pp. 259–273, 2014.

[14] D. M. Wolpert, J. Diedrichsen, and J. R. Flanagan, "Principles of sensorimotor learning," *Nature Reviews Neuroscience*, vol. 12, pp. 739–751, 2011.

[15] A. Gibaldi, S. P. Sabatini, S. Argentieri, and Z. Ji, "Emerging spatial competences: From machine perception to sensorimotor intelligence," *Robotics and Autonomous Systems*, vol. 71, pp. 1–2, 2015.

[16] P. Savastano and S. Nolfi, "A robotic model of reaching and grasping development," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 4, pp. 326–336, Dec 2013.

[17] F. Chao, M. H. Lee, C. Zhou, and M. Jiang, "An infant development-inspired approach to robot hand-eye coordination," *International Journal of Advanced Robotic Systems*, vol. 11, no. 15, 2014.

[18] H. Zhou, H. Hu, H. Liu, and J. Tang, "Classification of upper limb motion trajectories using shape features," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 42, no. 6, pp. 970–982, 2012.

[19] Z. Ju, G. Ouyang, M. Wilamowska-Korsak, and H. Liu, "Surface emg based hand manipulation identification via nonlinear feature extraction and classification," *IEEE Sensors Journal*, vol. 13, no. 9, pp. 3302–3311, Sept 2013.

[20] Z. Ju and H. Liu, "Human hand motion analysis with multisensory information," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 456–466, April 2014.

[21] Q. Meng, I. Tholley, and P. H. Chung, "Robots learn to dance through interaction with humans," *Neural Computing and Applications*, vol. 24, no. 1, pp. 117–124, 2014.

[22] J. Liu, Y. Luo, and Z. Ju, "An interactive astronaut-robot system with gesture control," *Computational intelligence and neuroscience*, vol. 2016, pp. 7 845 102:1–7 845 102:11, 2016. [Online]. Available: http://dx.doi.org/10.1155/2016/7845102

[23] M. Antonelli, A. J. Duran, E. Chinellato, and A. P. del Pobil, "Learning the visualcoculomotor transformation: Effects on saccade control and space representation," *Robotics and Autonomous Systems*, vol. 71, pp. 13–22, 2015.

[24] Q. Meng and M. Lee, "Error-driven active learning in growing radial basis function networks for early robot learning," *Neurocomputing*, vol. 71, no. 7, pp. 1449–1461, 2008.

[25] Q. Meng, M. Lee, and C. Hinde, "Robot competence development by constructive learning," in *Advances in machine learning and data analysis*. Springer, 2010, pp. 15–26.

[26] G. Pierris and T. Dahl, "A developmental perspective on humanoid skill learning using a hierarchical SOM-based encoding," in *2014 International Joint Conference on Neural Networks (IJCNN)*, July 2014, pp. 708–715.

[27] I. Kajić, G. Schillaci, S. Bodiroža, and V. V. Hafner, "Learning hand-eye coordination for a humanoid robot using SOMs," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, ser. HRI '14. New York, NY, USA: ACM, 2014, pp. 192–193. [Online]. Available: http://doi.acm.org/10.1145/2559636.2559816

[28] M. Hülse, S. McBride, J. Law, and M. Lee, "Integration of active vision and reaching from a developmental robotics perspective," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 4, pp. 355–367, 2010.

[29] L. Boucher, V. Stuphorn, G. D. Logan, J. D. Schall, and T. J. Palmeri, "Stopping eye and hand movements: Are the processes independent?" *Perception & Psychophysics*, vol. 69, no. 5, pp. 785–801, 2007.

[30] V. J. Traver and A. Bernardino, "A review of log-polar imaging for visual perception in robotics," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 378–398, 2010.

[31] Z. Wang, F. Chao, H. Lin, M. Jiang, and C. Zhou, "A human-like learning approach to developmental robotic reaching," in *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2013, pp. 581–586.

[32] C. Sann and A. Streri, "Perception of object shape and texture in human newborns: Evidence from cross-modal transfer tasks," *Developmental Science*, vol. 10, no. 3, pp. 399–410, 2007.

[33] J. Law, P. Shaw, K. Earland, M. Sheldon, and M. H. Lee, "A psychology based approach for longitudinal development in cognitive robotics," *Frontiers in Neurorobotics*, vol. 8, no. 1, 2014.

[34] F. Chao, M. Lee, and J. Lee, "A developmental algorithm for ocular–motor coordination," *Robotics and Autonomous Systems*, vol. 58, no. 3, pp. 239–248, 2010.

[35] N. J. Butko, I. Fasel, and J. R. Movellan, "Learning about humans during the first 6 minutes of life," in *International Conference on Development and Learning, Indiana*, 2006.

[36] Z. Liu, C. Chen, Y. Zhang, and C. Chen, "Adaptive neural control for dual-arm coordination of humanoid robot with unknown nonlinearities in output mechanism," *IEEE Transactions on Cybernetics*, vol. 45, no. 3, pp. 521–532, 2015.

[37] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 3, pp. 620–629, 2016.

[38] T. R. Shultz, "A constructive neural-network approach to modeling psychological development," *Cognitive Development*, vol. 27, no. 4, pp. 383–400, 2012.

[39] O. Sigaud, C. Salaün, and V. Padois, "On-line regression algorithms for learning mechanical models of robots: A survey," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1115–1129, 2011.

[40] S. Vijayakumar and S. Schaal, "Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space," in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, 2000, pp. 1079–1086.

[41] S. Schaal, C. Atkeson, and S. Vijayakumar, "Scalable techniques from nonparametric statistics for real time robot learning," *Applied Intelligence*, vol. 17, no. 1, pp. 49–60, 2002.

[42] D. Korkinof and Y. Demiris, "Online quantum mixture regression for trajectory learning by demonstration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2013, pp. 3222–3229.

[43] L. Jamone, M. Brandao, L. Natale, K. Hashimoto, G. Sandini, and A. Takanishi, "Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching," *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 556–567, 2014.

[44] J. Law, P. Shaw, M. Lee, and M. Sheldon, "From saccades to grasping: A model of coordinated reaching through simulated development on a humanoid robot," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 2, pp. 93–109, June 2014.

[45] P. Shaw, J. Law, and M. Lee, "An evaluation of environmental constraints for biologically constrained development of gaze control on an icub robot," *Paladyn*, vol. 3, no. 3, pp. 147–155, 2012.

[46] J. Law, P. Shaw, and M. Lee, "A biologically constrained architecture for developmental learning of eye-head gaze control on a humanoid robot," *Autonomous Robots*, vol. 35, no. 1, pp. 77–92, 2013.

[47] "Locally Weighted Projection Regression Implementation," 2012. [Online]. Available: http://wcms.inf.ed.ac.uk/ipab/slmc/research/software-lwpr

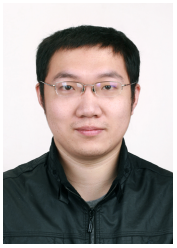
[48] F. Nori, L. Natale, G. Sandini, and G. Metta, "Autonomous learning of 3D reaching in a humanoid robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1142–1147.

[49] S. Ivaldi, M. Fumagalli, F. Nori, M. Baglietto, G. Metta, and G. Sandini, "Approximate optimal control for reaching and trajectory planning in a humanoid robot," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2010, pp. 1290–1296.

[50] M. Antonelli, B. J. Grzyb, V. Castelló, and A. P. Del Pobil, "Plastic representation of the reachable space for a humanoid robot," in *From Animals to Animats 12*. Springer, 2012, pp. 167–176.

[51] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*. ACM, 2007, pp. 255–262.

[52] M. Lee, Q. Meng, and F. Chao, "Staged competence learning in developmental robotics," *Adaptive Behavior*, vol. 15, no. 3, pp. 241–255, 2007.



Fei Chao (M2011) received the B.Sc. degree in mechanical engineering from Fuzhou University, Fuzhou, China, in 2004, the M.Sc. (Hons.) degree in computer science from the University of Wales, Aberystwyth, U.K., in 2005, and the Ph.D. degree in robotics from Aberystwyth University, Wales, U.K., in 2009. He was a Research Associate under the supervision of Prof. M. H. Lee with Aberystwyth University from 2009 to 2010. He is currently an Associate Professor with the Cognitive Science Department, Xiamen University, China. He has authored over 30 peer-reviewed journal and conference papers. His current research interests include developmental robotics, machine learning, and optimization algorithms.

Dr. Chao is a member of the China Computer Federation. He is also the Vice Chair of the IEEE Computer Intelligence Society Xiamen Chapter.

Zuyuan Zhu was born in Hubei, China, in 1990. He received the B.E. degree in automation from the Harbin Engineering University, Harbin, China, in 2013, and the M.Eng. degree in Computer Technology from the Xiamen University, Xiamen, China, in 2016. He is currently working towards the Ph.D. degree under the supervision of Prof. Huosheng Hu at the Human Centred Robotics Group in the University of Essex, Colchester, UK. His research interests include Robotic Assembly and Developmental Robotics.

Chih-Min Lin (M1987-SM1999-F2010) was born in Taiwan in 1959. He received the B.S. and M.S. degrees from the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from the Institute of Electronics Engineering, National Chiao Tung University, in 1986. He was the Honor Research Fellow with the University of Auckland, Auckland, New Zealand, from 1997 to 1998. He is currently the Vice President of Yuan Ze University, and a Chair Professor of the College of Electrical and Communication Engineering with Yuan Ze University, Taoyuan, Taiwan, and a "Changjiang Scholar Chair Professor" with Xiamen University, Xiamen, China. He has authored 164 journal papers and 165 conference papers. His current research interests include fuzzy neural network, cerebellar model articulation controller, intelligent control systems, and signal processing.

Professor Lin is a Fellow of the Institution of Engineering and Technology. He serves as an Associate Editor of the *IEEE Transactions on Cybernetics*, the *IEEE Transactions on Fuzzy Systems*, the *International Journal of Fuzzy Systems*, and the *International Journal of Machine Learning and Cybernetics*.

Huosheng Hu (M1994-SM2001) received the M.Sc. degree in industrial automation from Central South University, Changsha, China, in 1982, and the Ph.D. degree in robotics from the University of Oxford, Oxford, U.K., in 1993. He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., leading the robotics research. He has authored over 450 papers in the areas of robotics, humanrobot interaction, and pervasive computing.

Professor Hu is a Founding Member of the IEEE Robotics and Automation Society on Networked Robots, and a fellow of the Institution of Engineering and Technology and Institute of Measurement and Control. He has been the Program Chair and a member of the Advisory or Organizing Committee for many international conferences. He currently serves as an Editor-in-Chief of the *International Journal of Automation and Computing*, *Digital Communications and Networks*, and the *Journal of Robotics*, and an Executive Editor of the *International Journal of Mechatronics and Automation*.

Longzhi Yang (M2012) received the B.Sc. degree from Nanjing University of Science and Technology, Nanjing, China, the M.Sc. degree from Coventry University, Coventry, U.K., and Ph.D. degree from Aberystwyth University, U.K., all in Computer Science in 2003, 2006, and 2011 respectively. He is currently a senior lecturer at Northumbria University, Newcastle, U.K.. His research interests include fuzzy inference systems, machine learning, optimisation algorithms, intelligent control systems and the application of such techniques in real-world uncertain environments. He received the Best Student Paper Award at the 2010 IEEE International Conference on Fuzzy Systems.

Changjing Shang received a Ph.D. in computing and electrical engineering from Heriot-Watt University, UK. She is a University Research Fellow with the Department of Computer Science, Institute of Mathematics, Physics and Computer Science at Aberystwyth University, UK. Prior to joining Aberystwyth, she worked for Heriot-Watt, Loughborough and Glasgow Universities.

Her research interests include pattern recognition, data mining and analysis, space robotics, and image modelling and classification.

Changle Zhou received the Ph.D. degree from Beijing University, Beijing, China, in 1990. He is currently a Professor with the Cognitive Science Department, Xiamen University, Xiamen, China. His research interests lie in the areas of artificial intelligence, especially machine consciousness, brain-machine interface and robot dancing. His philosophical works lie in ancient oriental thoughts of Chinese, such as ZEN , TAO, YI etc., viewed from science.