Top–Down Gaze Movement Control in Target Search Using Population Cell Coding of Visual Context

Jun Miao, *Member, IEEE*, Laiyun Qing, *Member, IEEE*, Baixian Zou, Lijuan Duan, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*—Visual context plays an important role in humans' top–down gaze movement control for target searching. Exploring the mental development mechanism in terms of incremental visual context encoding by population cells is an interesting issue. This paper presents a biologically inspired computational model. The visual contextual cues were used in this model for top–down eye-motion control on searching targets in images. We proposed a population cell coding mechanism for visual context encoding and decoding. The model was implemented in a neural network system. A developmental learning mechanism was simulated in this system by dynamically generating new coding neurons to incrementally encode visual context during training. The encoded context was decoded with population neurons in a top–down mode. This allowed the model to control the gaze motion to the centers of the targets. The model was developed with pursuing low encoding quantity and high target locating accuracy. Its performance has been evaluated by a set of experiments to search different facial objects in a human face image set. Theoretical analysis and experimental results show that the proposed visual context encoding algorithm without weight updating is fast, efficient and stable, and the population-cell coding generally performs better than single-cell coding and *k*-nearest-neighbor (*k*-NN)-based coding.**

***Index Terms*—Gaze movement control, neural encoding and decoding, population cell coding, target search, visual context.**

I. Introduction

TARGET search or object detection is an important ability of human vision system. Generally this process consists of two phases: 1) prediction of a target or an object's place; and 2) identification of the target or object at the predicted place. These two phases are consistent with the "dorsal-ventral" pathways and the well-known "where-what" models on visual processing within mammal and human brains [1]–[6]. Both the dorsal and the ventral pathways [7]–[12] start from visual cortex V1, and reach V2. After that the dorsal pathway goes to the branch from MT (V5) to posterior parietal cortex (PPC), and the ventral pathway takes the branch from V4 to inferior temporal (IT) cortex. The control of attention is believed to take place mostly in the dorsal pathway using the bottom–up low-level features, such as saliency or motion, to produce object location or "where" information. The ventral pathway is mainly associated with the identification of visual stimuli. It is responsible for providing "what" information, such as the top–down prior knowledge of targets' representation and objects' spatial features. There are physical connections between these two pathways. Both of them are projected to superior colliculus (SC) and related oculomotor nucleus via prefrontal eye field (FEF) to control the eye movement.

It is not easy either to model the biological mechanism or build a practical system for both aforementioned phases. They have been widely studied in the literature. By assuming the targets certainly exist in images, this paper focuses on the information coding for inferring targets' locations; i.e., how to predict the targets' locations with lower top–down memory to get higher locating accuracy?

In the models that simulate top–down attention, there are generally two kinds of cues used for gaze movement control in target search: the cues about targets such as color, shape, or scale [12]–[18], and the cues about the visual context that contains the target and the relevant objects or environmental features with their spatial relationship [19]–[27].

A lot of computational model with object detection methods [28]–[32] have been developed by using the first type of cues. These methods mainly used the object-centered matching techniques. These methods did not predict where the targets are but compared the object features with each image region to verify if that region is the location of the target to be searched. Especially for the classical object detection methods, an original image are usually rotated m times and rescaled n times and then an object detector moves pixel by pixel on the transformed images l times to compare each image window with the target features. Thus, the detector will spend a total of mnl times to locate targets in the original image. This technique generally considers each object is independent and neglects the relevance between the target and the relevant objects or environmental features.

Manuscript received February 02, 2010; revised May 17, 2010; accepted May 28, 2010. Date of publication June 28, 2010; date of current version September 10, 2010. This research was supported in part by the National Basic Research Program of China (2009CB320902), the Natural Science Foundation of China (60673091, 60702031, and 60970087), the Hi-Tech Research and Development Program of China (2006AA01Z122), the Beijing Natural Science Foundation (4072023 and 4102013), the Beijing Municipal Education Committee (KM200610005012), and the Beijing Municipal Foundation for Excellent Talents (20061D0501500211).

J. Miao is with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China (e-mail: jmiao@ict.ac.cn).

L. Qing is with the School of Information Science and Engineering, Graduate University of the Chinese Academy of Sciences, Beijing, 100049, China (e-mail: lyqing@gucas.ac.cn).

B. Zou is with the Department of Information Science and Technology, College of Arts and Science, Beijing Union University, Beijing, 100083, China (e-mail: zoubx@ygi.edu.cn).

L. Duan is with the College of Computer Science and Technology, Beijing University of Technology, Beijing, 100124, China (e-mail: ljduan@bjut.edu.cn).

W. Gao is with the Institute of Digital Media, Peking University, Beijing, 100871, China (e-mail: wgao@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAMD.2010.2053365

In a recent psychological research [33], the second kind of top–down cues, i.e., the visual context, was reported to play a really helpful role in humans' top–down gaze movement control for target searching. This is also proved by psychological experiments [34]–[40] through examining the response time (RT). These experiments show that the response time can be decreased dramatically when the relationship between the background and the location of an object in a trail (image) is known. Chun [35] stated this reduction on RTs was influenced by "contextual cueing." Henderson and his colleges have done a lot of research work focusing on the role of context in the object searching. They have examined many other indexes besides RTs, such as fixation location, saccade length, and the relationship among the sequenced fixation locations [36]. After carrying out two experiments, they found that fixation location can be predicted based on the combination of current location and context. In other words, the local feature of current fixation location and its peripheral areas can influence the next fixation position.

In literature, there is a small amount of research work have been done by using visual context on object searching. Torralba used global features or global context to predict a horizontally long narrow region where the target is more likely to be appeared. Since it does not provide an accurate estimation on the x coordinate [19], therefore Torralba suggested using an object detector to search the target in that horizontally long narrow predicted region for accurate localization, which was implemented in the literature [23], [24]. Kruppa, Santana, and Schiele [25] used an extended object template that contains local context to detect extended targets and infer the location of the target via the ratio between the size of the target and the size of the extend template. Bergboer, Postma, and Herik [26] introduced local-contextual information to verify the candidates provided by an object detector, in order to reduce the false detection rate. Miao *et al.* [27] proposed a visual perceiving and eyeball-motion controlling neural network to search target by reasoning with visual context encoded with a singe cell coding mechanism. This representation mechanism led to a relatively large encoding quantity for memorizing the prior knowledge about the target's spatial relationship contained in the visual context.

The single-cell coding means using one cell or one response to represent one object or control the movement. In contrast to it, the population-cell coding uses an ensemble of cells or responses to represent an object or synthesize a movement [41]. Single and population cell coding mechanisms have been an argumentative issue in understanding human brain and vision functions, which was discussed and debated in the special issue for binding problem [42]. Wang [43] addressed that the main problem of the single-cell coding is that it would not allow perceiving novel objects, which is an ability the perceptual system undoubtedly possesses.

Usually eye movement is affected by the bottom–up and the top–down saliency map and visual context, in which multiple salient regions or objects are contained. If the single-cell coding is reasonable for representing a region or an object, it may imply that the population-cell coding is more suitable for representing multiregions and multiobjects in the visual context. In this paper, we propose a developmental neural network system that encodes the top–down knowledge of the visual context and infers the location of the target using a population-cell coding mechanism. The proposed system possesses the following characteristics.

First, it is a biological-inspired neural network system with visual sensors, internal-representation system and eye movement controller. It consists of four layers of neurons: input layer, feature neuron layer, single/population coding neuron layer, and eye movement control neuron layer (see Fig. 3). The first layer is the input layer and it uses five overlapped neuron arrays to form five visual fields in different scales to approximate the primate's retina whose sensing neuron density of the central part is larger than the density of the surrounding area. The second layer employs a group of highly selective features. These features simulate the sparse coding basis functions to encode the images from five visual fields into the connection weights between the second layer and the third layer. The spatial relationship $(\Delta x, \Delta y)$ between the center of a visual field and the target is encoded into two connection weights between a coding neuron in the third layer and the two movement neurons in the fourth layer. The weights between the second layer and the third layer and the weights between the third layer and the fourth layer are inner encoding presentation for the visual context. The third layer uses a single neuron or a group of population neurons to represent an object or a visual field image, and control or synthesize the eye movement through connection weights from the third layer to the fourth layer. The fourth layer simulates a movement controller. It uses the responses of two orthogonal movement $(\Delta x, \Delta y)$ neurons to activate the gaze movements in horizontal and vertical direction, respectively. The $(\Delta x, \Delta y)$ representation is consistent with the anatomy structure of an eye. There are two pairs of muscles in an eye. The muscle contraction is in proportion to the responses of the superior colliculus or related oculomotor nucleus. These muscles are responsible for controlling the orthogonal movements, which enables eyeball rotation and gaze movements [10].

Second, it is a dynamic learning architecture. It has the developmental and incremental learning characteristics reflected in the interactions with the visual image environments. These characteristics are consistent to the idea of the self-organizing autonomous incremental learner (SAIL) or the autonomous mental development [44]. Our developmental learning algorithm is introduced in Section III-A. Its main strategy is that when the system cannot infer the target's position correctly based on the current visual field image and the patterns encoded in the connection weights between the second layer and the third layer, the system generates a new coding neuron in the third layer with its two groups of connection weights to encode the current visual context (the current visual field image and the spatial relationship from the center of the current visual field to the center of the target) in an incremental coding mode.

Third, it is a task-driven visual search system. It uses the modulated spatial relationship as top–down information to move the gaze to the potential places of the targets. The encoded spatial relationship is modulated based on the percentage of the responses from the coding neurons associated with different visual context patterns.

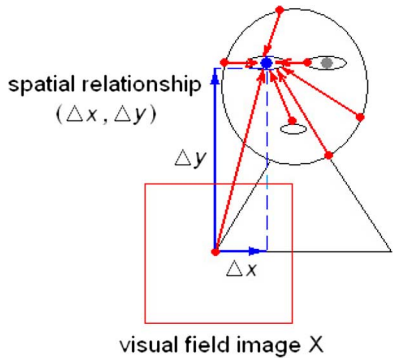


Fig. 1. Visual context $(\mathbf{X}, (\Delta x, \Delta y))$ in terms of the visual field image $\mathbf{X}$ and the spatial relationship $(\Delta x, \Delta y)$ between two object centers or between the center of the target and the center of the visual field image (the target in this scene is the left eye).

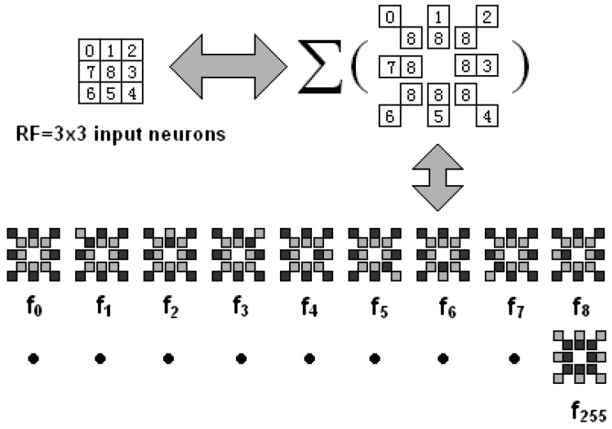


Fig. 2. Extend LBP features extracted by 256 feature neurons, each of which is computed by a sum of eight pairs of differences between surrounding pixels $(\text{labels} = 0 \sim 7)$ and the central pixel $(\text{label} = 8)$ in its receptive field $(\text{RF}) = 3 \times 3$ input neurons (pixels). They are illustrated in the 256 feature templates above, in which the gray box represents weight 1 while the black box represents weight -1.

To value the model's performance, we compared our system with a single-cell coding system and a k-NN-based coding system on encoding quantity and target locating accuracy by using a real-world image database. Our experimental results indicated that the population-cell coding mechanism generally performed better than other two systems in both encoding quantity and target locating accuracy.

This paper is organized as follows. Section II introduces highly selective coding features employed in the system. Section III describes the developmental learning structure using population-cell coding mechanism and its principle on encoding visual context and controlling gaze movement in target search. Learning properties of the population cell coding are discussed in Section IV. Experiments for three coding systems applied on a real-world image database are compared and analyzed in Section V. Conclusion and discussion are given in Section VI.

II. Features Employed for Encoding Visual Context

Visual context is related to two types of features: low-level features and the high-level features. The low-level features are responsible for representing the global and local images from visual fields. The high-level features are responsible for representing the spatial relationship which was described in the horizontal and the vertical distances $(\Delta x, \Delta y)$ between centers of two objects or between the center of the target and the center of the visual field as shown in Fig. 1.

Learning or encoding a context may produce a great amount of internal representation information. Employing features suitable for concise context representation is important for a practical system's efficiency. Physiological experiments done by Young and Yamane [45] show that monkeys only used a small number of neurons in their inferotemporal cortex (where the "what" information of objects is stored) to represent a human face image. This is the strategy of the sparse population coding occurred in primate animals' visual neural system. It can be introduced to employ the features that may decrease the encoding quantity as much as possible.

Interacting with the external environment and developing features to describe the observation is a characteristic of a developmental system. For example, in a recent proposed developmental model "where-what network 1" [12], an image set is learned via the in-place Hebbian learning to obtain a group of lobe features [46], [47]. These produced features are similar to ICA orientation filters. It is well known that independent components of natural scenes are edge filters [48], and are similar to sparse coding features [49]. In literature, there are many algorithms [50]–[53] have been designed to calculate these similar sparse coding features. In this paper, we focus on the developmental learning on the high-level knowledge on visual context, rather than the low-level features. Therefore, there is no learning process applied between the input layer and the feature neuron layer in our system. As long as the features are highly selective edge-like filters and are similar to the sparse coding features, we concerned that they are suitable to be used for the system.

Recently, a set of features called local binary patterns (LBP) [54] has become popular because of its high selectivity and fast computation characteristics. LBP is a binary code. Each binary code represents one of 256 patterns for an image block with 3×3 pixels. Originally, the LBP coding features contain 256 discrete codes used to represent 256 types of image blocks. However, these discrete codes cannot be used to compute the value of the connection weight between a feature neuron (in the second layer) and a coding neuron (in third layer) in our system. We used the Hebbian rule $\Delta w_{a,b} = \alpha R_a R_b$ to calculate the connection weights within our system. Here α is a learning rate. R_a and R_b are responses of two connected neurons, respectively. As illustrated in Fig. 2, to conquer this, we extended the LBP features to the new features with a continuous output R_{ij} by using the basis functions $\{f_j\}$ $(0 \leqslant j \leqslant 255)$ (see Fig. 2)

$$\begin{cases} R_{ij} = f_j(\mathbf{X}_i) = \mathbf{W}_{ij}^{\mathrm{T}} \mathbf{X}_i^{\mathrm{E}} = \sum_{l=0}^{7} w_{il,j}(x_{il} - x_{i8}) \\ w_{il,j} = (-1)^{b_l} \end{cases} \quad (1)$$

where the vector $\mathbf{X}_i = (x_{i0}\ x_{i2}\ \ldots\ x_{i8})^{\mathrm{T}}$ represents the ith image block or receptive field image composed of 3 $\times$ 3 pixels or 3 $\times$ 3 input neurons, and its extended form is $\mathbf{X}_i^{\mathrm{E}} = (x_{i0}x_{i8}x_{i1}x_{i8}\ldots x_{il}x_{i8}\ldots x_{i7}x_{i8})^{\mathrm{T}}$; $\mathbf{W}_{ij} =$

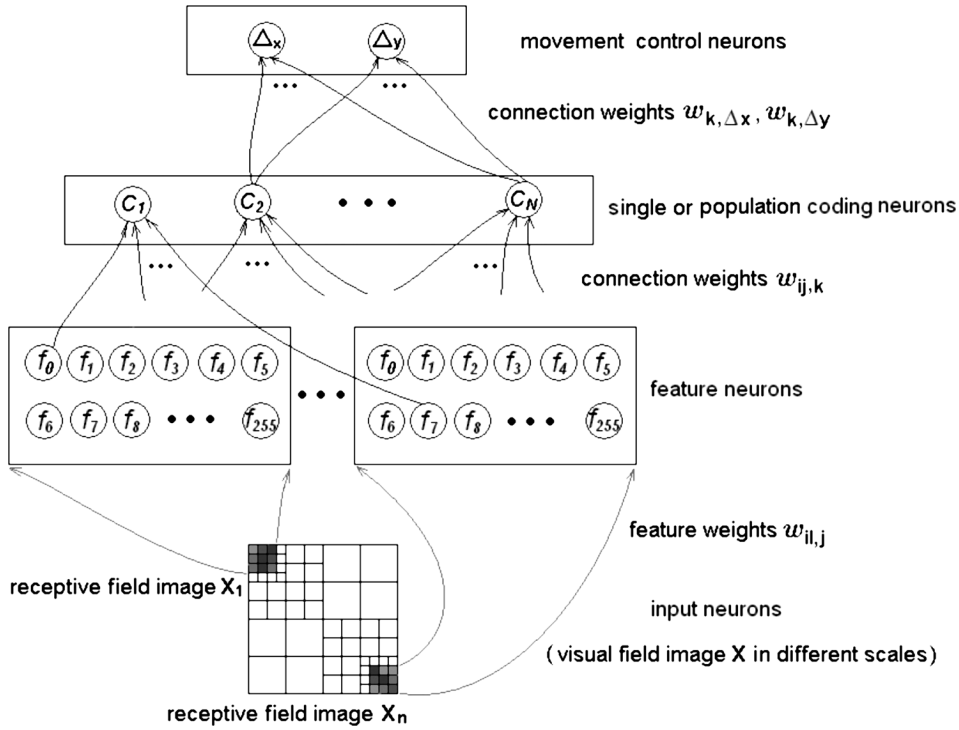


Fig. 3. Single- or population-cell coding structure for visual field image representation and gaze movement controlling.

$(w_{i0,j} - w_{i0,j}w_{i1,j} - w_{i1,j} \ldots w_{il,j} - w_{il,j} \ldots w_{i7,j} - w_{i7,j})^{\mathrm{T}}$ $(0 \leqslant l \leqslant 7)$ represents the jth basis function which consists of eight pairs of weights; the term R_{ij} represents the response of the jth feature extracted from the ith image block. The index j is a discrete number among $0 \sim 255$, which corresponds to a eight-bit binary code: $(\mathrm{b}_0\mathrm{b}_1 \ldots \mathrm{b}_l \ldots \mathrm{b}_7)_2$, where

$$b_l = \begin{cases} 0, & \text{if } (x_{il} - x_{i8}) < 0 \\ 1, & \text{otherwise} \end{cases}, \quad (l = 0 \sim 7). \tag{2}$$

In our coding system illustrated in Fig. 3, there are 256 feature neurons in the second layer. These neurons extract the extended LBP features $\{R_{ij} = f_j(\mathbf{X}_i)\}$ $(j = 0 \sim 255)$ for each receptive field image $\mathbf{X}_i$. Only the first m $(1 \leqslant m \leqslant 256)$ neurons with the largest responses $\{R_{ij'} = f_{j'}(\mathbf{X}_i)\}$ $(R_{ij'} \in \{R_{ij}\}, j' = 1 \sim m, j = 0 \sim 255)$ win the competition. To maximally decrease the coding quantity, m can be set to 1 for enough sparsity.

III. Visual Context Coding Architecture and Algorithms

A unified developmental neural coding structure is designed for the single and the population cell coding for visual context, which is illustrated in Fig. 3. The coding structure consists of two parts: the visual field image coding and the spatial relationship coding. The part of visual field image coding includes the first three layers: the input layer, the feature neuron layer, and the coding neuron layer. This part inputs images from a group of visual fields in different resolutions. Then it extracts features and encodes the current visual field image represented as the connection weights between the second layer and the third layer. The part of spatial relationship coding includes the last two layers: the coding neuron layer and the movement control neuron layer. The spatial relationship is encoded as the distance between two object centers or the distance between the center of the target and the center of the current visual field image and represented as the connection weights between the third layer and the fourth layer.

The corresponded visual context encoding and decoding algorithms for this neural coding architecture are introduced in the following sections.

A. Visual Context Encoding

In this paper, the visual context refers to the visual field image and the spatial relationship $(\Delta x, \Delta y)$ between the center of the visual field and the center of the target. To encode such context, the corresponding representation coefficients need to be calculated and stored. The details of the algorithm are described as in Table I.

The key part of the algorithm is dynamically generating coding neurons. The coding neurons are linked by the feature neurons in the second layer and two movement control neurons in the fourth layer with two groups of connection weights to represent the encoded context knowledge and experience. When the coding system can not search the target in the given precision $ER(s)$ depending on the encoded context, the system generates a new coding neuron in the third layer with its two groups of connection weights $\{w_{ij,k}\}$ and $\{(w_{k,\Delta x}, w_{k,\Delta y})\}$ to encode the current visual context (the current visual field image and the spatial relationship from the center of the current visual field to the center of the target) in an incremental coding mode. From this point of view, the proposed encoding algorithm has the developmental and incremental learning characteristics which are consistent to the idea of the SAIL and the autonomous mental development [44]. The encodings of the

TABLE I
ALGORITHM FOR VISUAL CONTEXT ENCODING

BEGIN LOOP1 Select a scale s from a set S for the current visual field;
BEGIN LOOP2 Select a starting gaze point (x, y) as the center of the visual field from an initial gaze point set G_{xy} distributed in the context area of the target;
1. Input an image from the current visual field, and then predict the moving distance $(\Delta\hat{x}, \Delta\hat{y})$ of the gaze towards the target center. The real distance should be $(\Delta x, \Delta y)$;
2. If the prediction error $ER=\sqrt{(\Delta\hat{x}-\Delta x)^2+(\Delta\hat{y}-\Delta y)^2}$ is larger than a maximum error limit $ER(s)$ for the scale s of the current visual field, move the center of the visual field to the new gaze point position $(x+\Delta\hat{x}, y+\Delta\hat{y})$; go to 1 until $ER \le ER(s)$ or the iteration number is larger than a maximum limit;
3. If $ER>ER(s)$, generate a new coding neuron (let its response $R_k=1$); encode the visual context by computing and storing the connection weights $\{w_{ij,k}\}$ (initialized to zeros) between the new coding neuron and the feature neurons (their responses $R_{ij}=f_j(\mathbf{X}_i)$) and the connection weights $(w_{k,\Delta x}, w_{k,\Delta y})$ (initialized to zeros) between the new coding neuron and two movement control neurons (let their responses $R_{\Delta x}=\Delta x$ and $R_{\Delta y}=\Delta y$) respectively using the Hebbian rule $\Delta w_{a,b}=\alpha R_a R_b$;
END LOOP2
END LOOP1

visual field images and the spatial relationship are formulized in the following two sections.

1) Encoding of Visual Field Images: The kth coding neuron in the third layer represents (or encodes) a visual field image pattern $\mathbf{X}^{(k)}$ with a group of connection weights $\{w_{ij,k}\}$ between the feature neurons (in the second layer) and itself. The ijth feature neuron extract the jth feature $\{R_{ij} = f_j(\mathbf{X}_i^{(k)})\}$ $(0 \leqslant j \leqslant 255)$ from the ith receptive field image $\mathbf{X}_i^{(k)}$ $(1 \leqslant i \leqslant \mathrm{n})$. All the receptive field images $\{\mathbf{X}_i^{(k)}\}$compose the visual field image $\mathbf{X}^{(k)}$, i.e., $\mathbf{X}^{(k)} = (\mathbf{X}_1^{(k)}\mathbf{X}_2^{(k)}\cdots\mathbf{X}_n^{(k)})$. The connection weights $\{w_{ij,k}\}$ are computed with Hebbian rule

$$\begin{cases} \Delta w_{a,b}(t) = \alpha R_a R_b \\ w_{a,b}(t+1) = w_{a,b}(t) + \Delta w_{a,b}(t) \end{cases} \tag{3}$$

where a is the learning rate; t is the iteration number; R_a and R_b are responses of two neurons which are connected by a synapse with a connection weight $w_{a,b}$. Thus, each weight $w_{ij,k}$ between the ijth feature neuron and the kth coding neuron is calculated as

$$\begin{cases} w_{ij,k}(0) = 0, \quad \Delta w_{ij,k}(0) = \alpha R_{ij} R_k = \alpha f_j\left(\mathbf{X}_i^{(k)}\right) R_k \\ w_{ij,k}(1) = w_{ij,k}(0) + \Delta w_{ij,k}(0) = \alpha f_j\left(\mathbf{X}_i^{(k)}\right) R_k \end{cases} \tag{4}$$

where α and R_k are the learning rate and the response of the kth coding neuron, respectively. Both they are set to be 1 for simplifying computation, and then the (4) is changed to

$$w_{ij,k} = f_j\left(\mathbf{X}_i^{(k)}\right). \tag{5}$$

Therefore, the visual images were encoded based on the (5). The lengths of all the weights $\{w_{ij,k}\}$ were normalized to one for unified similarity computation and comparison.

Our experiments described in Section V showed that updating weights $\{w_{ij,k}\}$ with one step or multisteps did not make much difference on system performance. This is because our system does not learn the internal representation through a fixed-number of neurons which the classical learning machines normally do, e.g., the Perceptron [59] or the multilayer perceptron (MLP) [60]. Our coding system dynamically generates a nonfixed number of population coding neurons with their connecting weights as internal representation. Our system is less sensitive to the weight adjustment compared with other learning algorithms. We will make a more detailed discussion and explanation in Section IV-B.

2) Encoding of Spatial Relationship: The spatial relationship $(\Delta x_k, \Delta y_k)$ between the center of the visual field and the center of the target is encoded in terms of two connection weights $(w_{k,\Delta x}, w_{k,\Delta y})$. They linked the kth coding neuron and the two movement control neurons and are learned by the Hebbian rule

$$\begin{cases} w_{k,\Delta x}(0)=0, \quad \Delta w_{k,\Delta x}(0)=\beta R_k R_{\Delta x}=\beta R_k \Delta x_k \\ w_{k,\Delta x}(1)=w_{k,\Delta x}(0)+\Delta w_{k,\Delta x}(0)=\beta R_k \Delta x_k \end{cases} \tag{6}$$

$$\begin{cases} w_{k,\Delta y}(0)=0, \quad \Delta w_{k,\Delta y}(0)=\beta R_k R_{\Delta y}=\beta R_k \Delta y_k \\ w_{k,\Delta y}(1)=w_{k,\Delta y}(0)+\Delta w_{k,\Delta y}(0)=\beta R_k \Delta y_k \end{cases} \tag{7}$$

where β and R_k are the learning rate and the response of the kth coding neuron, respectively. Similarly, both of them are set to 1 for simplifying computation, and then (6) and (7) are simplified to

$$\begin{cases} w_{k,\Delta x} = \Delta x_k \\ w_{k,\Delta y} = \Delta y_k \end{cases}. \tag{8}$$

B. Visual Context Decoding for Gaze Movement Control

Visual context decoding includes the responding of a single or population coding neuron(s), the decoding of visual field images and the decoding of the spatial relationship. Here, the spatial relationship decoding has direct relation to the control of the gaze movement for target search. They are formulized in following sections.

1) Response of a Single or Population Coding Neuron(s): When the coding system inputs a visual field image $\mathbf{Y}$ for test or perception, a single cell or population cells, in the third layer may respond(s) through competition among the total N coding neurons to represent a visual field image pattern. As described in Fig. 3, for the ith receptive field image $\mathbf{Y}_i$, the kth coding

neuron receives m responses $\{R_{ij'}\}$ $(1 \leqslant j' \leqslant m \leqslant 256)$ weighted by $\{w_{ij',k}\}$ from m feature neurons which extract features $\{R_{ij'} = f_{j'}(\mathbf{Y}_i)\}$ from $\mathbf{Y}_i$. Therefore, for the visual field image $\mathbf{Y}$, which is composed of the receptive field images $\{\mathbf{Y}_i\}$ $(1 \leqslant i \leqslant n)$, the response of the kth coding neuron in the third layer is

$$\begin{aligned} R_k &= C_k(\mathbf{Y}) = C_k(\mathbf{Y}_1\, \mathbf{Y}_2 \ldots \mathbf{Y}_n) \\ &= \sum_{i=1}^{n}\sum_{j'=1}^{m} w_{ij',k} R_{ij'} = \sum_{i=1}^{n}\sum_{j'=1}^{m} w_{ij',k} f_{j'}(\mathbf{Y}_i) \end{aligned} \quad (9)$$

where $w_{ij,k} \in \{w_{ij',k}\}$, $R_{ij'} \in \{R_{ij}\}$, $f_{j'}(\mathbf{Y}_i) \in \{f_j(\mathbf{Y}_i)\}$, $j' = 1 \sim m$ and $j = 0 \sim 255$. The weights $\{w_{ij',k}\}$ are obtained at the encoding or training stage discussed in Section III-A1. The $R_{ij'}$ is the response of the j'th feature neuron for the receptive field image $\mathbf{Y}_i$, belonging to the first m largest responses among the total feature responses $\{R_{ij}\}$ $(j = 0 \sim 255)$. Substituting (5) into (9), we get

$$\begin{aligned} R_k &= C_k(\mathbf{Y}) = \sum_{i=1}^{n}\sum_{j'=1}^{m} w_{ij',k} f_{j'}(\mathbf{Y}_i) \\ &= \sum_{i=1}^{n}\sum_{j'=1}^{m} f_{j'}\left(\mathbf{X}_i^{(k)}\right) f_{j'}(\mathbf{Y}_i). \end{aligned} \quad (9a)$$

Let $\mathbf{W}_{\mathbf{X}^{(k)}} = (w_{i=1j'=1,k} w_{i=1j'=2,k} \ldots w_{i=n,j'=m,k})^{\mathrm{T}}$, $\mathbf{f}_{\mathbf{X}^{(k)}} = (f_{j'=1}(\mathbf{X}_{i=1}^{(k)}) f_{j'=2}(\mathbf{X}_{i=1}^{(k)}) \ldots f_{j'=m}(\mathbf{X}_{i=n}^{(k)}))^{\mathrm{T}}$, and $\mathbf{f}_{\mathbf{Y}} = (f_{j'=1}(\mathbf{Y}_{i=1}) f_{j'=2}(\mathbf{Y}_{i=1}) \ldots f_{j'=m}(\mathbf{Y}_{i=n}))^{\mathrm{T}}$, then (9a) is changed to its inner product form between two groups of features

$$R_k = \mathbf{W}_{\mathbf{X}^{(k)}}^{\mathrm{T}} \mathbf{f}_{\mathbf{Y}} = \mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k)}}. \quad (9b)$$

Equation (9b) indicates the response of the kth coding neuron in the third layer, which is a similarity measure between the new image $\mathbf{Y}$ and the kth visual field image pattern $\mathbf{X}^{(k)}$ memorized in the coding system.

2) Decoding of Visual Field Images: In the third layer of the coding system, the first M largest responses $\{R_{k'}\}$ $(1 \leqslant k' \leqslant M)$ of M coding neurons among the total N coding neurons represent a visual field image. Therefore, the visual field image $\mathbf{Y}$ that is composed of n receptive field images $\{\mathbf{Y}_i\}$ $(1 \leqslant i \leqslant n)$, can be approximately reconstructed with the M encoded visual field image patterns $\{\mathbf{X}^{(k)} | 1 \leqslant k' \leqslant M\}$ which are weighted by the percentage of the first M largest responses of coding neurons $\{R_{k'} | 1 \leqslant k' \leqslant M\}$, or the m basis functions $\{\mathbf{W}_{ij'} | 1 \leqslant j' \leqslant m \leqslant 256\}$ modulated by the connection weights $\{w_{ij',k'} | 1 \leqslant j' \leqslant m \leqslant 256, 1 \leqslant k' \leqslant M\}$ from feature neurons to M coding neurons

$$\begin{cases} \mathbf{Y} \approx \hat{\mathbf{Y}} = \sum\limits_{k'=1}^{M} R_{k'}^{*} \hat{\mathbf{X}}^{(k')} \\ R_{k'}^{*} = \dfrac{R_{k'}}{\sum\limits_{k'=1}^{M} R_{k'}} = \dfrac{\mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}}{\sum\limits_{k'=1}^{M} \mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}}, (1 \leqslant m \leqslant 256, 1 \leqslant M \leqslant N) \\ \hat{\mathbf{X}}^{(k')} = \sum\limits_{i=1}^{n}\sum\limits_{j'=1}^{m} w_{ij',k'} \mathbf{W}_{ij'} \end{cases} \quad (10)$$

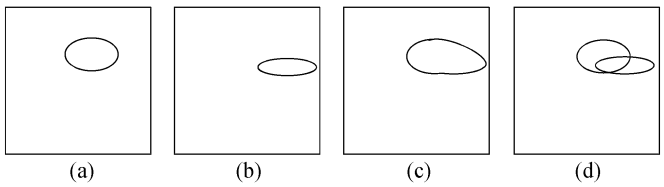


Fig. 4. Illustration of visual field image decoding or reconstruction. (a) The first encoded visual field image associated with the first coding neuron; (b) The second encoded visual field image associated with the second coding neuron; (c) A new visual field image; (d) Reconstructed visual field image using the first and the second encoded visual field images if adopting the population-cell coding mechanism (here is two-cell coding); Otherwise the reconstructed image will be Fig. 4(a) or (b) if adopting the single-cell coding mechanism.

where $R_{k'} \in \{R_k\}$, $w_{ij',k'} \in \{w_{ij',k}\}$, $\mathbf{W}_{ij'} \in \{\mathbf{W}_{ij}\}$, $k' = 1 \sim M, k = 1 \sim N, j' = 1 \sim m$ and $j = 0 \sim 255$; $\{\mathbf{W}_{ij'}\}$ are m basis functions introduced in Section II, which correspond to the first m largest responses of feature neurons in the second layer using these basis functions. When $M > 1$, it means that the system is using population cells in the third layer to represent the visual field image; when $M = 1$, it means that the system is using one single cell to represent the visual field image. For the case of single cell coding, (10) is changed to

$$\mathbf{Y} \approx \hat{\mathbf{Y}} = \hat{\mathbf{X}}^{(k'=1)} = \sum_{i=1}^{n}\sum_{j'=1}^{m} w_{ij',k'=1} \mathbf{W}_{ij'}. \quad (11)$$

Fig. 4 illustrates the visual field image decoding or reconstruction using single or population cell coding mechanism. It shows that when the coding system encountering a new object image that is different from the individually encoded objects, the population cell coding mechanism makes a better image understanding (or decoding). Of course, if the system perceives a new object image which is much like an encoded object in its memory and there is one coding neuron respond with the largest response that is much larger that of other neurons, the single cell coding mechanism is suitable.

3) Decoding of Spatial Relationship for Gaze Movement Control: Gaze movement control is directly responsible for visual object search. We implemented the gaze movement into a two-layer structure: single- or population-cell coding layer and the movement control layer (see Fig. 3). The movement control neurons were divided into two categories, respectively responsible for the moving distances along two axis, x and y. Their responses $(R_{\Delta x}, R_{\Delta y})$ represent the relative distance $(\Delta x, \Delta y)$ from the current gaze point (x, y) (or the center of the current visual field image) to the target center. For the current visual field image input, the first M coding neurons with the largest responses play a main role in activating the movement control neurons. When $M = 1$, the system uses the single-cell coding controlling mechanism, otherwise it uses the population-cell coding mechanism. The responses of gaze movement control neurons can be formulated as

$$\begin{cases} R_{\Delta x} = \sum\limits_{k'=1}^{M} w_{k',\Delta x} R_{k'}^{*} \\ R_{\Delta y} = \sum\limits_{k'=1}^{M} w_{k',\Delta y} R_{k'}^{*} \\ R_{k'}^{*} = \dfrac{R_{k'}}{\sum\limits_{k'=1}^{M} R_{k'}} \end{cases} \quad (12)$$

TABLE II
ALGORITHM FOR GAZE MOVEMENT CONTROL

BEGIN LOOP1 Select a starting gaze point (x_J, y_J) as the center of the visual field from a initial gaze point set G_{xy} randomly distributed on the image;

BEGIN LOOP2 Select a scale s_I from the set S for the current visual field in the order of from the maximum to the minimum;

Input an image from the current visual field, and output a relative position prediction in terms of gaze movement $(\Delta\hat{x}_I, \Delta\hat{y}_I)$ for the real relative position of the target center $(\Delta x, \Delta y)$;

END LOOP2

The position of the target center (x, y) starting from the initial gaze point (x_J, y_J) is predicted by

$$\hat{x}_J = x_J + \sum_I \Delta\hat{x}_I \quad \text{and} \quad \hat{y}_J = y_J + \sum_I \Delta\hat{y}_I;$$

END LOOP1

Computing the density $D(x, y)$ of the gaze point distribution $\{(\hat{x}_J, \hat{y}_J)\}$;

Select the position with the largest density as the finally predicted target position:

$$(\hat{x}, \hat{y}) = \arg\max\{D(x, y)\}.$$

where $R_{k'}$ is the k'th largest response of a coding neuron among the total N coding neurons; $R^*_{k'}$ is the ratio of a single response $R_{k'}$ to the sum of M responses. $R^*_{k'}$ is used for synthesizing gaze movement. $w_{k',\Delta x}$ and $w_{k',\Delta y}$ are the weights for the connection between the k'th coding neuron and the movement control neurons in x and y directions, respectively. At a learning (or encoding) stage, both $w_{k',\Delta x}$ and $w_{k',\Delta y}$ are calculated using (6)–(8). Substituting (8) and (9b) into (11), the synthesis of movement can be represented as

$$\begin{cases} R_{\Delta x} = \sum_{k'=1}^{M} R^*_{k'} \Delta x_{k'} \\ R_{\Delta y} = \sum_{k'=1}^{M} R^*_{k'} \Delta y_{k'} \\ R^*_{k'} = \dfrac{R_{k'}}{\sum_{k'=1}^{M} R_{k'}} = \dfrac{\mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}(k')}}{\sum_{k'=1}^{M} \mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}(k')}}, \quad (1 \leqslant M \leqslant N). \end{cases} \tag{13}$$

Formula (13) means that the gaze movement distances (decoded at the perception or test stage) are the sum of the weighted spatial relationship (encoded at the learning or training stage, which are weighted by the first M largest responses of coding neurons). Especially, when $M = 1$ (single cell coding), the responses $(R_{\Delta x}, R_{\Delta y})$ of two movement control neurons are activated by a single neuron who has encoded a historical spatial relationship $(\Delta x_{k'=1}, \Delta y_{k'=1})$ into connection weights $w_{k'=1,\Delta x}$ and $w_{k'=1,\Delta y}$, respectively. In this case, (13) is simplified to (14)

$$\begin{cases} R_{\Delta \mathrm{x}} = \Delta x_{k'=1} \\ R_{\Delta \mathrm{y}} = \Delta y_{k'=1}. \end{cases} \tag{14}$$

Fig. 5 illustrates the spatial relationship decoding (or gaze movement synthesis) using the single and the population cell coding mechanisms. It shows that when the coding system encounters an image containing a new spatial relationship different to the individually encoded relationship, the population-cell coding mechanism makes a better gaze movement synthesis or target position prediction. Conversely, when the system perceives a new spatial relationship which is very close to an encoded relationship in the memory, the single-cell coding mechanism will be more suitable.

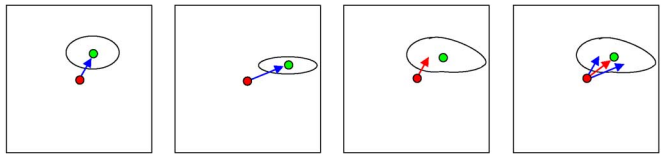


Fig. 5. Illustration of spatial relationship decoding or gaze movement synthesis to predict the position of a new target center. (a) The encoded spatial relationship $(\Delta x_1, \Delta y_1)$ associated with the first coding neuron. (b) The encoded spatial relationship $(\Delta x_2, \Delta y_2)$ associated with the second coding neuron. (c) Predicting a new target center with a gaze movement $(\Delta x_1, \Delta y_1)$ controlled by a single cell with the encoded spatial relationship $(\Delta x_1, \Delta y_1)$. (d) Predicting the new target center with a synthesized movement $w_1(\Delta x_1, \Delta y_1) + w_2(\Delta x_2, \Delta y_2)$ modulated by population cells (here are two coding neurons) with two weighted spatial relationships.

An entire algorithm for gaze movement control on target searching is given in Table II.

The algorithm uses a gradually searching strategy that is moving an initial gaze point to the center of target from the largest visual field to the smallest visual field by decoding the global and the local context. The visual field imaging and searching are illustrated in Fig. 6.

To avoid getting an instable searching result, the algorithm uses multiple search results to evaluate the position of the target center, which is illustrated in Fig. 7.

The case of searching multiple targets, using two targets as example, can be handled in two ways: 1) if the two targets have no close relationship, for example, the mouths from two different persons, we carry out two searches, respectively, for two targets; and 2) if the two targets have close spatial relationship with their local context, (for example, two associated objects, such as a river and a ship, a road and a car, and a running car and the car's driver; or one object and its subobjects, such as the driver and his head, the head and its left eye; or two subobjects included in a common object, such as the left eye and the mouth), we encoded the global context between the envi-

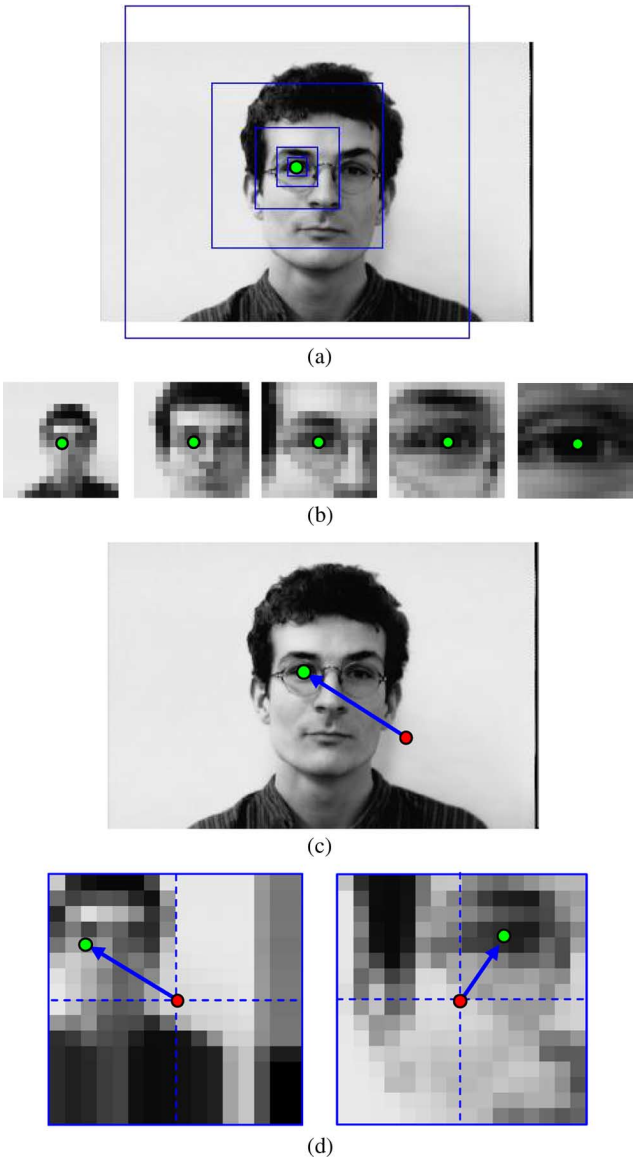


Fig. 6. Illustration of gradually visual context encoding or decoding. (a) Five visual fields centered at a gaze point (here is the left eye center). (b) Five visual field images (16×16 pixels, $\mathrm{scales} = 5, 4, 3, 2,$ and 1) sub-sampled from the original image (320×214 pixels) with $\mathrm{intervals} = 16, 8, 4, 2, 1$ pixel(s). (c) The spatial relationship between one given starting gaze point and the target center. (d) Encoding or decoding the visual context between current gaze points and the target center, gradually from largest visual field to smaller ones (here two scales of visual fields and the corresponding spatial relationships are shown).

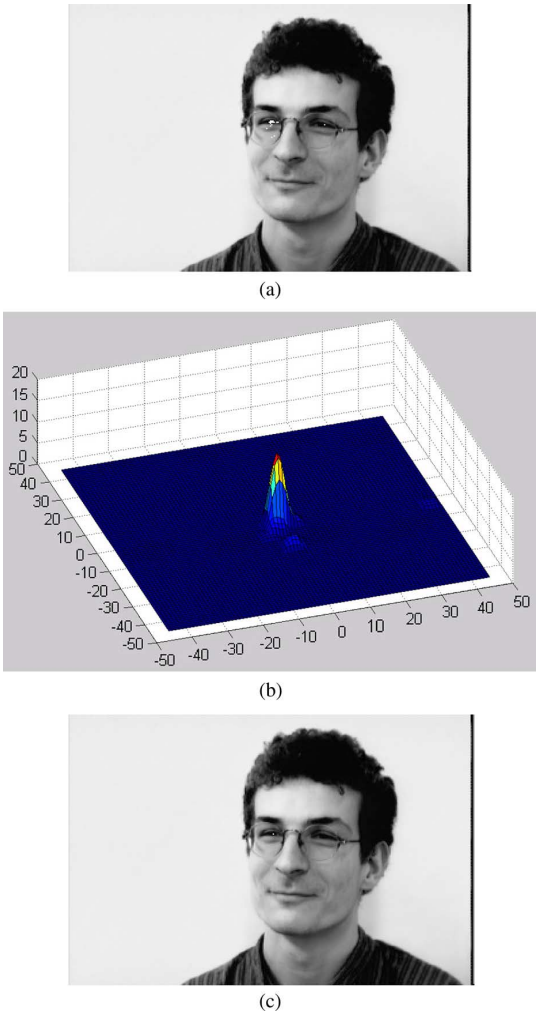


Fig. 7. Illustration of fine target localization through computing the maximum density of the final gaze points. (a) The distribution of final gaze points (white points) representing located target (left eye) centers, which start from a group of initial gaze points randomly distributed on the image. (b) The density of final gaze points or located target (left eye) centers. (c) Using the position with the highest gaze point density as the finally located target center (the white point).

ronmental points and the first target, and then encode the local context between the first target and the second target. By this way, the coding system could save quite a large amount of encoding quantity, especially when the number of the targets is very large. Figs. 8 and 9 illustrate that using global and local context together for multiple targets searching can save more encoding information than searching them separately.

IV. Learning Properties of the Population Cell Coding

In this section, we discuss our population coding system in two respects. The first is its learning properties underlying the visual context encoding and gaze movement controlling or synthesizing. The second is a theoretical analysis and comparison between our system without weight updating and the classical learning machines with weight updating in terms of efficiency and stability.

A. Modeling Context Representation as a Learning Problem

A learning problem can be proposed as: Given a group of visual contexts $\{(\mathbf{X}^{(k)}, (\Delta x_k, \Delta y_k)) | 1 \leqslant k \leqslant N\}$ and a visual field image $\mathbf{Y}$, how to estimate the unknown relative distances $(\Delta x, \Delta y)$ of the target center?

One solution is letting the half-unknown visual context $(\mathbf{Y}, (\Delta x, \Delta y))$ be represented or synthesized by the known visual contexts $\{(\mathbf{X}^{(k)}, (\Delta x_k, \Delta y_k)) | 1 \leqslant k \leqslant N\}$, i.e.

$$(\mathbf{Y}, (\Delta x, \Delta y)) = \sum_{k=1}^{N} c_k \cdot \left(\mathbf{X}^{(k)}, (\Delta x_k, \Delta y_k)\right). \qquad (15)$$

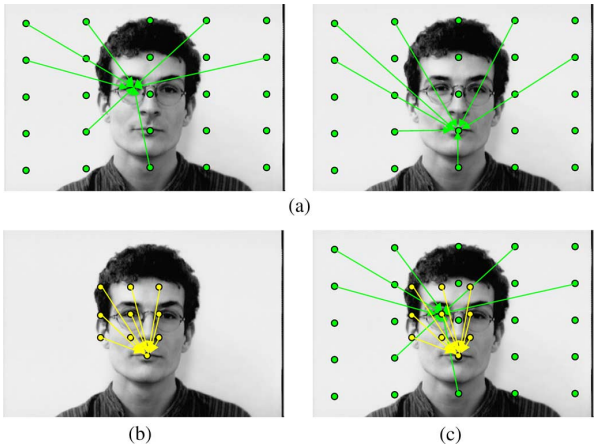


Fig. 8. Encoding and decoding the global and local context for searching multiple targets. (a) Encoding the global context between the environment and the targets (left eye or mouth). (b) Encoding the local context between the targets (the mouth and the region surrounding the left eye). (c) Encoding the global context firstly and local context secondly for target search in sequence.

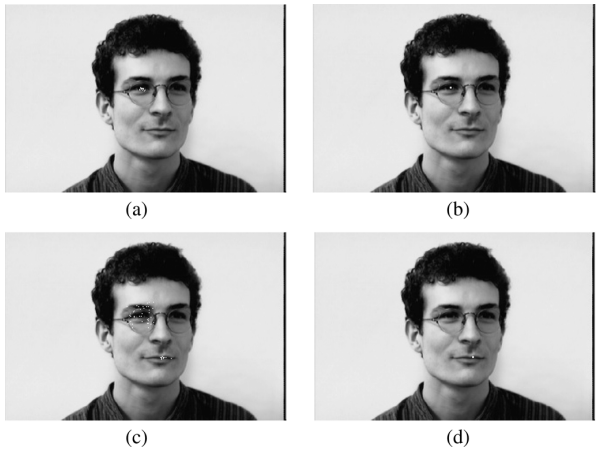


Fig. 9. Combining global context and local context for searching target centers. (a) Locating the left eye center using the global context from a group of initial gaze points randomly distributed on the image. (b) Fine left eye center localization (represented by the white point) after computing the maximum density of the final gaze points. (c) Locating the mouth center using the local context from a group of initial gaze points around the located left eye center. (d) Fine mouth center localization (represented by the white point) after the maximum density computation.

Then the leaning problem becomes an issue of how to determine the values of the coefficients $\{c_k\}$. To compute the coefficients $\{c_k\}$, we divided (15) into three parts

$$\begin{cases} \mathbf{Y} = \sum_{k=1}^{N} c_k \mathbf{X}^{(k)} \\ \Delta x = \sum_{k=1}^{N} c_k \Delta x_k \\ \Delta y = \sum_{k=1}^{N} c_k \Delta y_k. \end{cases} \tag{16}$$

The coefficients $\{c_k\}$ could be obtained by decomposing the known $\mathbf{Y}$ into a group of basis functions $\{\mathbf{X}^{(k)}\}$. Then, these coefficients are used to synthesize the unknown relative distances $(\Delta x, \Delta y)$ along with the known spatial relationships $\{(\Delta x_k, \Delta y_k)\}$.

Usually, the exact value of these coefficients $\{c_k\}$ can not be obtained in a simple and easy way. Therefore, we used an estimated visual context $(\hat{\mathbf{Y}}, (\Delta\hat{x}, \Delta\hat{y}))$ to approximate the real visual context $(\mathbf{Y}, (\Delta x, \Delta y))$ instead. Then (16) is transformed to

$$\begin{cases} \mathbf{Y} \approx \hat{\mathbf{Y}} = \sum_{k'=1}^{M} c_{k'} \mathbf{X}^{(k')} \\ \Delta x \approx \Delta\hat{x} = R_{\Delta x} = \sum_{k'=1}^{M} c_{k'} \Delta x_{k'} \\ \Delta y \approx \Delta\hat{y} = R_{\Delta y} = \sum_{k'=1}^{M} c_{k'} \Delta y_{k'} \end{cases} \tag{17}$$

where $c_{k'} \in \{c_k\}$, $\mathbf{X}^{(k')} \in \{\mathbf{X}^{(k)}\}$, $(\Delta x_{k'}, \Delta y_{k'}) \in \{(\Delta x_k, \Delta y_k)\}$, $k' = 1 \sim M$, $k = 1 \sim N$, and $M = 1 \sim N$.

Comparing (17) to (10) and (13), we can get the corresponding coefficients $\{c_{k'} | 1 \leqslant k' \leqslant M\}$

$$c_{k'} = \frac{\mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}}{\sum_{k'=1}^{M} \mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}}, \quad (1 \leqslant k' \leqslant M). \tag{18}$$

Thus, decoding the spatial relationship $(\Delta x, \Delta y)$ with the encoded visual contexts $\{(\mathbf{X}^{(k')}, (\Delta x_{k'}, \Delta y_{k'}) | 1 \leqslant k' \leqslant M\}$ to produce a gaze movement for target locating can be modeled with a regression function

$$\begin{cases} (\Delta x, \Delta y) \approx (\Delta\hat{x}, \Delta\hat{y}) = F_{rg}(\Delta x_{k'}, \Delta y_{k'}) \\ \quad = \sum_{k'=1}^{M} c_{k'} \cdot (\Delta x_{k'}, \Delta y_{k'}) \\ c_{k'} = \frac{\mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}}{\sum_{k'=1}^{M} \mathbf{f}_{\mathbf{Y}}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}^{(k')}}} \end{cases} \tag{19}$$

where the coefficient $c_{k'}$ is the percentage form of the similarity between the new visual field image $\mathbf{Y}$ and the k'th encoded visual field image pattern $\mathbf{X}^{(k')}$. Particularly, when using the single coding mechanism $(M = 1)$, (19) is simplified to

$$(\Delta x, \Delta y) \approx (\Delta\hat{x}, \Delta\hat{y}) = (\Delta x_{k'=1}, \Delta y_{k'=1}). \tag{20}$$

From (20), it can be learned that for the case of single-cell coding, the system produces a movement associated with a memorized visual field image pattern which is most similar to the new visual field image. If the coding system encoded enough visual field image patterns $\{\mathbf{X}^{(k)}\}$ $(1 \leqslant k \leqslant N)$ and associated spatial relationship $\{(\Delta x_k, \Delta y_k)\}$ $(1 \leqslant k \leqslant N)$, a new visual field image $\mathbf{Y}$ can be easily located in the neighbor area of an encoded $\mathbf{X}^{(k)}$ in the data space, as illustrated in Fig. 10(a). In this case, the single-cell coding is suitable and its associated distance prediction $(\Delta x_k, \Delta y_k)$ is accurate enough. However, the encoding quantity for coding system to memorize such visual context could be very large. In other words, the system needs a large amount of training data to obtain a good prediction performance. Therefore, it is not economic to implement a practical system in such a coding mechanism.

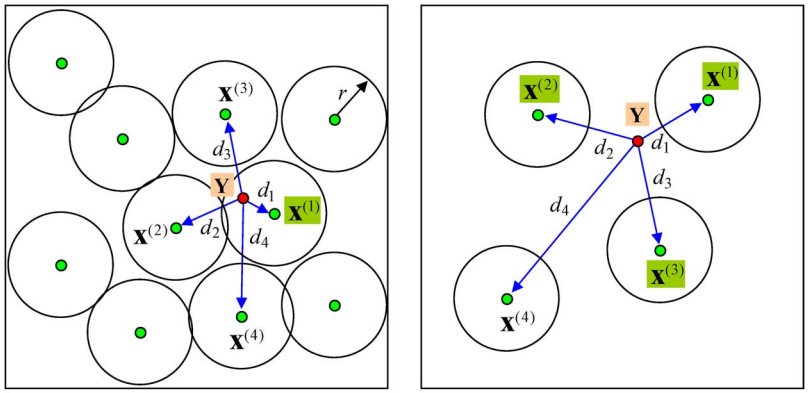


Fig. 10. Illustration of encoded visual field image patterns $\{\mathbf{X}^{(k)}\}$ with their neighbor areas (radius r), the test visual field image $\mathbf{Y}$ and their distances $\{d_i\}$ or the similarity measurements (e.g., $\{1/(1+d_i)\}$) in a data space. (a) Densely encoded samples suitable for single cell decoding for a new sample (e.g., using $\mathbf{X}^{(k'=1)}$ to represent $\mathbf{Y}$ for distances $d_1 < r \ll d_2 < d_3 < d_4$); (b) Sparsely encoded samples suitable for population cell decoding (e.g., using $\{\mathbf{X}^{(k')}|1 \leqslant k' \leqslant 3\}$ to represent $\mathbf{Y}$ for distances $r < d_1 < d_2 < d_3 \ll d_4$).

From (19), for the case of population-cell coding, the system produces a movement according to a group of encoded visual field image patterns to which the new visual field image is similar. If the sparse visual field image patterns are stored in the coding system, the possibility of a new visual field image to be located in the neighbor area of an encoded image pattern is very small, as illustrated in Fig. 10(b). In this case, the single-cell coding can not provide an accurate representation and prediction. Thus, the prediction should be compensated by other cells that are also similar to the new input. Therefore, population-cell coding is suitable here and the gaze movement is synthesized by a group of encoded movements $\{(\Delta x_{k'}, \Delta y_{k'})\}$ $(1 \leqslant k' \leqslant M)$ associated with the similar encoded image patterns $\{\mathbf{X}^{(k')}\}$ $(1 \leqslant k' \leqslant M)$.

In order to build a system with the possible best generalization, it is necessary to minimize the structural risk (or the generalization error) of the system according to the theory of statistical learning [55]. For our system with M population coding neurons involved in to represent the visual field image and synthesize the gaze movement, if T is the number of targets in a test set, the mean $\mathrm{mea(M)}$ and the standard deviation $\mathrm{std(M)}$ of the target locating errors are formulized, respectively [see (21) and (22) at the bottom of the page].

In literature [56], the expected prediction error (EPE) is defined as an evaluation of test error or generalization error that makes a bias-variance tradeoff. Based on it, a simplified comprehensive test error $ce(M)$ can be formulated as

$$ce(M) = \sqrt{\mathrm{mea}^2(\mathrm{M}) + \mathrm{std}^2(\mathrm{M})}. \tag{23}$$

Generally, it is difficult to get the best model by finding a system with the smallest test error directly. According to the principle of the structural risk minimization [55], the problem can be transformed to find a system by minimizing the system's structural risk $SR(M, N)$

$$SR(M, N) = ce_{tr}(M) + mc(N) \tag{24}$$

where $ce_{tr}(M)$ is the comprehensive error for the training set; $mc(N)$ is a function of N and represents the system's model complexity; N is the number of coding neurons in the third layer, representing the number of visual context patterns encoded in the system; M determines how many cells involved in to represent the visual field image and synthesize the gaze movement. Here the parameter M directly influences the target locating error for each search, and consequently influences the learning procedure that determines the final value of the number N, i.e., the number of total coding neurons generated after the training procedure.

$$\begin{aligned}\mathrm{mea(M)} &= \frac{1}{T}\sum_{t=1}^{T}\sqrt{\left(\Delta x^{(t)} - \Delta\hat{x}^{(t)}\right)^2 + \left(\Delta y^{(t)} - \Delta\hat{y}^{(t)}\right)^2} \\ &= \frac{1}{T}\sum_{t=1}^{T}\sqrt{\left(\Delta x^{(t)} - \sum_{k'=1}^{M} c_{k'}\Delta x_{k'}^{(t)}\right)^2 + \left(\Delta y^{(t)} - \sum_{k'=1}^{M} c_{k'}\Delta y_{k'}^{(t)}\right)^2}\end{aligned} \tag{21}$$

$$\mathrm{std(M)} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(\sqrt{\left(\Delta x^{(t)} - \Delta\hat{x}^{(t)}\right)^2 + \left(\Delta y^{(t)} - \Delta\hat{y}^{(t)}\right)^2} - \mathrm{mea(M)}\right)^2}. \tag{22}$$

TABLE III
ALGORITHM FOR DETERMINING THE NUMBER OF POPULATION CODING NEURONS

BEGIN LOOP P_i=1 to p ($0 \leq p < 1$) with a step Δp(Δp<0), where P_i is the factor of determining M_i, i.e., the number of population coding neurons with the first M_i largest responses;

1. Train the coding system with the given maximum error limit and the value of M_i which is controlled by the factor P_i, then get the system complexity N_i after training, where N_i is the number of layer-3 coding neurons generated in the system;
2. Sort all the N_i coding neurons' responses in a sequence from the maximum response *MaxR* to the minimum response; if the (M_i+1)-th ($1 \leq M_i \leq N_i$) neuron's response is the first one smaller than or equaling to P_i * *MaxR*, then P_i , M_i and N_i are recoded;

END LOOP

Get P among $\{P_i\}$, where P corresponds to the smallest N_i, and then get M from P.

Therefore, how to set M $(1 \leqslant M \leqslant N)$, the number of population neurons involved in visual context coding, becomes a key problem in selecting a model for possible best performance. Theoretically, M could be obtained by minimizing the structural risk

$$M = \arg\operatorname{Min}\{SR(M,N)\} = \arg\operatorname{Min}\{ce_{tr}(M) + mc(N)\}. \tag{25}$$

However, it is practically difficult to get the value of M in such a way. An alternative approximating method [55] is: Setting a maximum training error limit $\max\_ce_{tr}$; among all the systems whose training errors $\{ce_{\mathrm{tr}}(M_i)\}$ are smaller than $\max\_ce_{\mathrm{tr}}$, finding a system with the minimum model complexity $\min\_mc$ among all the model complexities $\{mc(N_i)\}$; with the comprehensive training error $ce_{\mathrm{tr}}(M^*)$ of the found system, getting the corresponding value of M by using

$$\begin{aligned} M &= \arg\operatorname{Min}\{mc(N_i) | ce_{\mathrm{tr}}(M_i) \leq \max\_ce_{\mathrm{tr}}\} \\ &= M^* = \arg\{\min\_mc | ce_{\mathrm{tr}}(M_i) = ce_{\mathrm{tr}}(M^*)\}. \end{aligned} \tag{26}$$

Fig. 10(b) shows that the distances $\{d_{(k')}\}$ ($k' = 1 \sim M$, here $M = 3$) between the new visual field image $\mathbf{Y}$ and M encoded visual field images $\{\mathbf{X}^{(k')}\}$ in the data space are far smaller than other distances $\{d_{(k')}\}$ ($k' = M+1 \sim N$), i.e., $d_1 < d_2 < \ldots < d_M \ll d_{M+1} < \ldots < d_N$. By transforming the distance measurement into a similarity measurement, e.g., $s_{(k')} = 1/(1 + d_{(k')})$, Fig. 10(b) shows that there are M encoded visual field images $\{\mathbf{X}^{(k')}\}$ ($k' = 1 \sim M$, here $M = 3$) are most similar to the new visual field image $\mathbf{Y}$, i.e., $s_1 > s_2 > \ldots > s_M \gg s_{M+1} > \ldots > s_N$. Our experimental results showed that M is not a constant parameter to be selected directly for the system's best performance. Instead, we used another factor $P = s_M/s_1$ to control M. The parameter P and M can be obtained by Table III.

Please note that the result of M will be 1 and N when $P = 1$ and $P = 0$, respectively.

B. System Stability and Insensitivity to Weight Updating

As described in Section III-A.1, the performance of our system using multistep learning (which updates the weights with more than one steps) is almost the same as the performance of the system updating its weights with only one step. We try to explain this phenomenon by a comparison between our coding system and a classical learning machine Perceptron [59]. Then, the analysis is briefly extended to the multilayer Perceptron [60].

With reference to Fig. 11(a), Perceptron is a two-layer neural network that performs supervised learning. Its first layer is an input layer with $n + 1$ input neurons, where the first n neurons input a signal that can be represented as a n-dimension vector $\mathbf{X} = (x_1 x_2 \ldots x_n)^{\mathrm{T}}$, and the $(n + 1)$th neuron inputs the constant 1. Its second layer is the output layer in which there is only one neuron. It updates the connection weight vector $\mathbf{W} = (w_1 w_2 \ldots w_n)^{\mathrm{T}}$ and the bias weight b between two layers till the learning is converged. For a linear responding neuron, the system's output function is

$$f(\mathbf{X}) = \mathbf{W}^{\mathrm{T}}\mathbf{X} + b. \tag{27}$$

Thus, Perceptron produces a hyperplane ($f(\mathbf{X}) = 0$) as illustrated by the solid line in Fig. 12(a), which segments the data space into two subspaces, A and B. If the network outputs a positive value ($f(\mathbf{X}) > 0$), the input is classified to class A, otherwise it is classified to class B, which is indicated by

$$f(\mathbf{X}) : 0 \Rightarrow \mathbf{X} \in \mathrm{A/B}. \tag{28}$$

Through a weight vector modification $\mathbf{W}_1 = \mathbf{W}_0 + \Delta\mathbf{W}$ with $\Delta\mathbf{W} = (\Delta w_1 \Delta w_2 \ldots \Delta w_n)^{\mathrm{T}}$, the system's output corresponds to a variation

$$\begin{cases} f_0(\mathbf{X}) = \mathbf{W}_0^{\mathrm{T}}\mathbf{X} + b \\ f_1(\mathbf{X}) = \mathbf{W}_1^{\mathrm{T}}\mathbf{X} + b = (\mathbf{W}_0 + \Delta\mathbf{W})^{\mathrm{T}}\mathbf{X} + b \\ \quad = f_0(\mathbf{X}) + (\Delta\mathbf{W})^{\mathrm{T}}\mathbf{X} \end{cases}. \tag{29}$$

Then the new hyperplane is

$$f_1(\mathbf{X}) = f_0(\mathbf{X}) + (\Delta\mathbf{W})^{\mathrm{T}}\mathbf{X} = 0 \tag{30}$$

which is shown by the dashed line in Fig. 12(a). For any $\mathbf{X}_0$ and $\mathbf{X}_1$ that are located on the hyperplanes $f_0(\mathbf{X}_0) = 0$ and $f_1(\mathbf{X}_1) = 0$, respectively, it can be deduced that the difference of two hyperplanes is

$$\Delta\mathbf{X} = \mathbf{X}_1 - \mathbf{X}_0 = \begin{cases} -b(\Delta\mathbf{W})^{-1} & \mathrm{if(b} \neq 0) \\ \Delta\mathbf{W}^* & \mathrm{otherwise} \end{cases} \tag{31}$$

where $\Delta\mathbf{W}^* \perp \Delta\mathbf{W}$ and $\|\Delta\mathbf{W}^*\| = \|\Delta\mathbf{W}\|$.

Fig. 11(b) illustrates how our cording system represents and discriminates two classes A and B. It uses two groups of population coding neurons in the second layer to represent classes A and B, respectively. The numbers of two groups of population

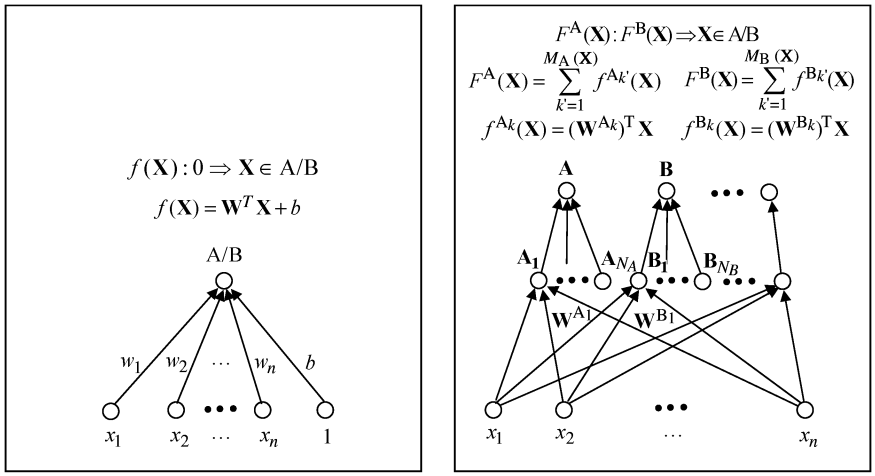


Fig. 11. Comparison of two-class (A/B) discrimination principles of two learning systems; (a) Perceptron and (b) our population coding system.

coding neurons are $N_{\mathbf{A}}$ and $N_{\mathbf{B}}$, respectively. The responses of two coding neurons for classes A and B are

$$\begin{cases} f^{\mathrm{A}_{k_1}}(\mathbf{X})=(\mathbf{W}^{\mathrm{A}_{k_1}})^{\mathrm{T}}\mathbf{X}, \\ f^{\mathrm{B}_{k_2}}(\mathbf{X})=(\mathbf{W}^{\mathrm{B}_{k_2}})^{\mathrm{T}}\mathbf{X}, \end{cases} (1\leqslant k_1\leqslant N_{\mathbf{A}},\quad 1\leqslant k_2\leqslant N_{\mathbf{B}}) \tag{32}$$

where $\mathbf{W}^{\mathrm{A}_{k_1}}$ and $\mathbf{W}^{\mathrm{B}_{k_2}}$ are two weight vectors representing two groups of connecting weights which are between the k_1th coding neuron for class A and the input neurons and between the k_2th coding neuron for class B and the input neurons.

In the third layer, the two class neurons A and B are used to compute the total responses of population coding neurons that represent two classes. An input $\mathbf{X}$ activates population coding neurons with the first $(M_{\mathbf{A}}(\mathbf{X})+M_{\mathbf{B}}(\mathbf{X}))$ largest responses $\{f^{\mathrm{A}_{k'_1}}|1\leqslant k'_1\leqslant M_{\mathbf{A}}(\mathbf{X})\leqslant N_{\mathbf{A}}\}$ and $\{f^{\mathrm{B}_{k'_1}}|1\leqslant k'_2\leqslant M_{\mathbf{B}}(\mathbf{X})\leqslant N_{\mathbf{B}}\}$, where $M_{\mathbf{A}}(\mathbf{X})$ and $M_{\mathbf{B}}(\mathbf{X})$ are numbers of two groups of population coding neurons activated by the input $\mathbf{X}$ for representing classes A and B, respectively. In other words, the connection weight vectors $\{\mathbf{W}^{\mathrm{A}_{k'_1}}|1\leqslant k'_1\leqslant M_{\mathbf{A}}(\mathbf{X})\}$ and $\{\mathbf{W}^{\mathrm{B}_{k'_2}}|1\leqslant k'_2\leqslant M_{\mathbf{B}}(\mathbf{X})\}$ between these coding neurons and the input layer are close to $\mathbf{X}$. As illustrated in Fig. 12(b), a new input $\mathbf{Y}$ activates one class-A coding neuron represented with $\mathbf{W}_0^{\mathrm{A}_1}$ and two class-B coding neurons represented with $\{\mathbf{W}_0^{\mathrm{B}_1},\mathbf{W}_0^{\mathrm{B}_2}\}$ within $\mathbf{Y}$'s neighborhood with radius R. Therefore, the responses of class neurons A and B in the third layer can be calculated by

$$\begin{cases} F^{\mathrm{A}}(\mathbf{X})=\sum\limits_{k'_1=1}^{M_{\mathbf{A}}(\mathbf{X})}f^{\mathrm{A}_{k'_1}}(\mathbf{X})=\sum\limits_{k'_1=1}^{M_{\mathbf{A}}(\mathbf{X})}(\mathbf{W}^{\mathrm{A}_{k'_1}})^{\mathrm{T}}\mathbf{X} \\ F^{\mathrm{B}}(\mathbf{X})=\sum\limits_{k'_2=1}^{M_{\mathbf{B}}(\mathbf{X})}f^{\mathrm{B}_{k'_2}}(\mathbf{X})=\sum\limits_{k'_2=1}^{M_{\mathbf{B}}(\mathbf{X})}(\mathbf{W}^{\mathrm{B}_{k'_2}})^{\mathrm{T}}\mathbf{X} \end{cases} \tag{33}$$

where $1\leqslant M_{\mathbf{A}}(\mathbf{X})\leqslant N_{\mathbf{A}}$, $1\leqslant M_{\mathbf{B}}(\mathbf{X})\leqslant N_{\mathbf{B}}$. Thus, in the local data space around the input $\mathbf{X}$, the local hyperplane is determined only by two groups of the encoded weight vectors $\{\mathbf{W}^{\mathrm{A}_{k'_1}}|1\leqslant k'_1\leqslant M_{\mathbf{A}}(\mathbf{X})\}$ and $\{\mathbf{W}^{\mathrm{B}_{k'_2}}|1\leqslant k'_2\leqslant M_{\mathbf{B}}(\mathbf{X})\}$. The local hyperplane is represented by

$$F^{\mathrm{A}}(\mathbf{X})=F^{\mathrm{B}}(\mathbf{X})\quad\text{or}\quad F^{\mathrm{A}}(\mathbf{X})-F^{\mathrm{B}}(\mathbf{X})=0. \tag{34}$$

The global hyperplane or hypersurface can be also represented by (34). It is illustrated by the solid line in Fig. 12(b), which is combined by multiple local hyperplanes and segments the data space into two subspaces A and B. The system discriminates two classes by a comparison of the responses of two class neurons

$$F^{\mathrm{A}}(\mathbf{X}):F^{\mathrm{B}}(\mathbf{X})\Rightarrow\mathbf{X}\in\mathrm{A/B} \tag{35}$$

which means if $F^{\mathrm{A}}(\mathbf{X})$ is larger than $F^{\mathrm{B}}(\mathbf{X})$, then $\mathbf{X}$ is classified to A, otherwise it is classified to B.

Similarly, if there is a weight vector modification for those relevant population coding neurons, e.g., $\{\Delta\mathbf{W}^{\mathrm{A}_{k'_1}}=(\Delta w_1^{\mathrm{A}_{k'_1}}\Delta w_2^{\mathrm{A}_{k'_1}}\ldots\Delta w_n^{\mathrm{A}_{k'_1}})^{\mathrm{T}}|1\leqslant k'_1\leqslant M_{\mathbf{A}}(\mathbf{X})\}$ and $\{\Delta\mathbf{W}^{\mathrm{B}_{k'_2}}=(\Delta w_1^{\mathrm{B}_{k'_2}}\Delta w_2^{\mathrm{B}_{k'_2}}\ldots\Delta w_n^{\mathrm{B}_{k'_2}})^{\mathrm{T}}|1\leqslant k'_2\leqslant M_{\mathbf{B}}(\mathbf{X})\}$, the responses of the coding neurons in second layer and the class neurons in the third layer for classes A and B are reflected with two groups of variations

$$\begin{cases} f_1^{\mathrm{A}_{k'_1}}(\mathbf{X})=\left(\mathbf{W}_1^{\mathrm{A}_{k'_1}}\right)^{\mathrm{T}}\mathbf{X} \\ \quad=\left(\mathbf{W}_0^{\mathrm{A}_{k'_1}}+\Delta\mathbf{W}^{\mathrm{A}_{k'_1}}\right)^{\mathrm{T}}\mathbf{X} \\ \quad=f_0^{\mathrm{A}_{k'_1}}(\mathbf{X})+\left(\Delta\mathbf{W}^{\mathrm{A}_{k'_1}}\right)^{\mathrm{T}}\mathbf{X} \\ f_1^{\mathrm{B}_{k'}}(\mathbf{X})=\left(\mathbf{W}_1^{\mathrm{B}_{k'_2}}\right)^{\mathrm{T}}\mathbf{X} \\ \quad=\left(\mathbf{W}_0^{\mathrm{B}_{k'_2}}+\Delta\mathbf{W}^{\mathrm{B}_{k'_2}}\right)^{\mathrm{T}}\mathbf{X} \\ \quad=f_0^{\mathrm{B}_{k'_2}}(\mathbf{X})+\left(\Delta\mathbf{W}^{\mathrm{B}_{k'_2}}\right)^{\mathrm{T}}\mathbf{X} \end{cases} \tag{36}$$

$$\begin{cases} F_1^{\mathrm{A}}(\mathrm{X}) = \sum\limits_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X})} f_1^{\mathrm{A}_{k_1'}}(\mathbf{X}) = \sum\limits_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X})} f_0^{\mathrm{A}_{k_1'}}(\mathbf{X}) \\ \quad + \sum\limits_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X})} \left(\Delta \mathbf{W}^{\mathrm{A}_{k_1'}}\right)^{\mathrm{T}} \mathbf{X} = F_0^{\mathrm{A}}(\mathrm{X}) \\ \quad + \sum\limits_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X})} \left(\Delta \mathbf{W}^{\mathrm{A}_{k_1'}}\right)^{\mathrm{T}} \mathbf{X} \\ F_1^{\mathrm{B}}(\mathrm{X}) = \sum\limits_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X})} f_1^{\mathrm{B}_{k_2'}}(\mathbf{X}) = \sum\limits_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X})} f_0^{\mathrm{B}_{k_2'}}(\mathbf{X}) \\ \quad + \sum\limits_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X})} \left(\Delta \mathbf{W}^{\mathrm{B}_{k_2'}}\right)^{\mathrm{T}} \mathbf{X} = F_0^{\mathrm{B}}(\mathrm{X}) \\ \quad + \sum\limits_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X})} \left(\Delta \mathbf{W}^{\mathrm{B}_{k_2'}}\right)^{\mathrm{T}} \mathbf{X}. \end{cases} \tag{37}$$

The new hyperplane or hypersurface produced by the population coding system is

$$\begin{aligned} F_1^{\mathrm{A}}(\mathbf{X}) = F_1^{\mathrm{B}}(\mathbf{X}) \quad & \text{or} \\ F_1^{\mathrm{A}}(\mathbf{X}) - F_1^{\mathrm{B}}(\mathbf{X}) &= F_0^{\mathrm{A}}(\mathbf{X}) - F_0^{\mathrm{B}}(\mathbf{X}) + (\delta \mathbf{W})^{\mathrm{T}} \mathbf{X} \\ &= 0 \end{aligned} \tag{38}$$

where $\delta \mathbf{W} = \sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X})} \Delta \mathbf{W}^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X})} \Delta \mathbf{W}^{\mathrm{B}_{k_2'}}$.

For any $\mathbf{X}_0$ that is located on the hyperplane $F_0^{\mathrm{A}}(\mathbf{X}) = F_0^{\mathrm{B}}(\mathbf{X})$, we have

$$\begin{aligned} F_0^{\mathrm{A}}(\mathbf{X}_0) - F_0^{\mathrm{B}}(\mathbf{X}_0) = & \sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \left(\mathbf{W}_0^{\mathrm{A}_{k_1'}}\right)^{\mathrm{T}} \mathbf{X}_0 \\ & - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \left(\mathbf{W}_0^{\mathrm{B}_{k_2'}}\right)^{\mathrm{T}} \mathbf{X}_0 \\ = & \left(\Delta \mathbf{W}_0^{\mathbf{BA}}\right)^{\mathrm{T}} \mathbf{X}_0 = 0 \end{aligned} \tag{39}$$

where $\Delta \mathbf{W}_0^{\mathbf{BA}} = \sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{B}_{k_2'}}$. Correspondingly, there is a $\Delta \mathbf{W}_1^{\mathbf{BA}} = \sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{B}_{k_2'}}$ after the weight vector modification. From (39), it can be deduced that $\mathbf{X}_0$ is orthogonal to $\Delta \mathbf{W}_0^{\mathbf{BA}}$, i.e., $\mathbf{X}_0 \perp \Delta \mathbf{W}_0^{\mathbf{BA}}$.

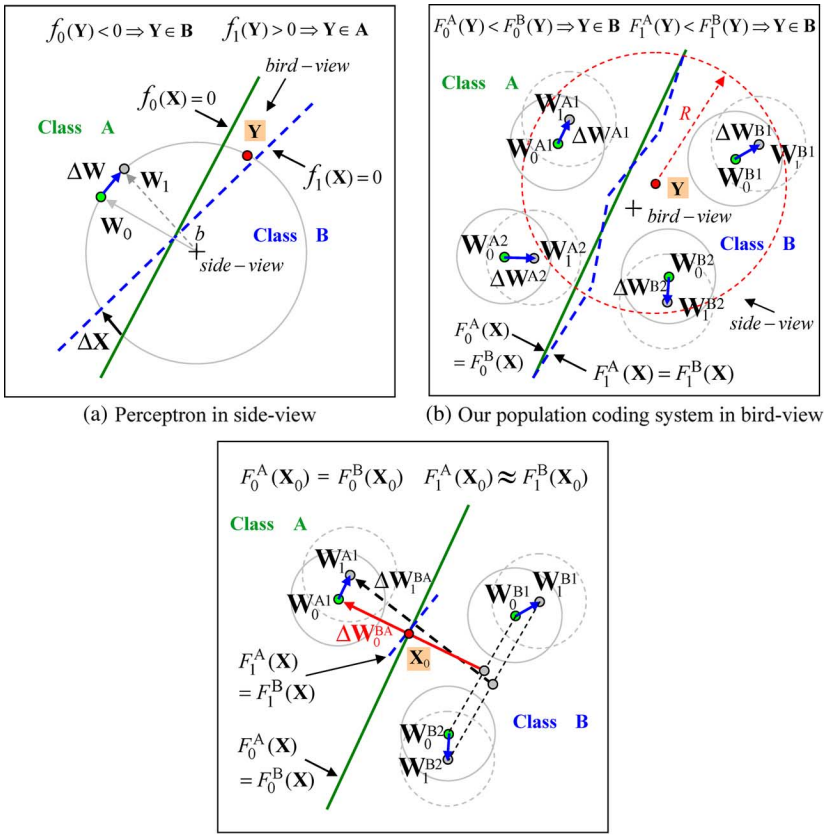
As illustrated in Fig. 12(c), $\Delta \mathbf{W}_0^{\mathbf{BA}}$ and $\Delta \mathbf{W}_1^{\mathbf{BA}}$ are two vectors across two local hyperplanes. They mean the differences of two pairs of central vectors $\{\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{A}_{k_1'}}, \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{B}_{k_1'}}\}$ and $\{\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{A}_{k_1'}}, \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{B}_{k_1'}}\}$, which represent classes A and B before and after the weight updating respectively. When the responses $\{f_0^{\mathrm{A}_{k_1'}}(\mathbf{X}_0), f_1^{\mathrm{A}_{k_1'}}(\mathbf{X}_0)\}$ and $\{f_0^{\mathrm{B}_{k_2'}}(\mathbf{X}_0), f_1^{\mathrm{B}_{k_2'}}(\mathbf{X}_0)\}$ of population coding neurons $\{\mathbf{A}_{k_1'}\}$ and $\{\mathbf{B}_{k_2'}\}$ $(1 \leqslant k_1' \leqslant M_{\mathbf{A}}(\mathbf{X}_0), 1 \leqslant k_2' \leqslant M_{\mathbf{B}}(\mathbf{X}_0))$ are enough large, or if the weight vectors $\{\mathbf{W}_0^{\mathrm{A}_{k_1'}}, \mathbf{W}_1^{\mathrm{A}_{k_1'}}\}$ and $\{\mathbf{W}_0^{\mathrm{B}_{k_2'}}, \mathbf{W}_1^{\mathrm{B}_{k_2'}}\}$ before and after the weight updating are enough close to the data point $\mathbf{X}_0$, the hyperplane spanned by the vectors $\Delta \mathbf{W}_0^{\mathbf{BA}}$ and $\Delta \mathbf{W}_1^{\mathbf{BA}}$ is approximately orthogonal to $\mathbf{X}_0$. In other words, $\mathbf{X}_0$ is approximately orthogonal to $\Delta \mathbf{W}_1^{\mathbf{BA}}$, i.e., $\mathbf{X}_0 \widetilde{\perp} \Delta \mathbf{W}_1^{\mathbf{BA}}$, or $(\Delta \mathbf{W}_1^{\mathbf{BA}})^{\mathrm{T}} \mathbf{X}_0 \approx 0$. Thus, using (37)–(39), we have

$$\begin{aligned} & F_1^{\mathrm{A}}(\mathbf{X}_0) - F_1^{\mathrm{B}}(\mathbf{X}_0) \\ & \quad = F_0^{\mathrm{A}}(\mathbf{X}_0) - F_0^{\mathrm{B}}(\mathbf{X}_0) + (\delta \mathbf{W})^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = (\delta \mathbf{W})^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = \left(\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \Delta \mathbf{W}^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \Delta \mathbf{W}^{\mathrm{B}_{k_2'}} \right)^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = \left(\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \left(\mathbf{W}_1^{\mathrm{A}_{k_1'}} - \mathbf{W}_0^{\mathrm{A}_{k_1'}} \right) \right. \\ & \qquad \left. - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \left(\mathbf{W}_1^{\mathrm{B}_{k_2'}} - \mathbf{W}_0^{\mathrm{B}_{k_2'}} \right) \right)^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = \left(\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_1^{\mathrm{B}_{k_2'}} \right)^{\mathrm{T}} \mathbf{X}_0 \\ & \qquad - \left(\sum_{k_1'=1}^{M_{\mathbf{A}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{A}_{k_1'}} - \sum_{k_2'=1}^{M_{\mathbf{B}}(\mathbf{X}_0)} \mathbf{W}_0^{\mathrm{B}_{k_2'}} \right)^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = \left(\Delta \mathbf{W}_1^{\mathbf{BA}}\right)^{\mathrm{T}} \mathbf{X}_0 - \left(\Delta \mathbf{W}_0^{\mathbf{BA}}\right)^{\mathrm{T}} \mathbf{X}_0 \\ & \quad = \left(\Delta \mathbf{W}_1^{\mathbf{BA}}\right)^{\mathrm{T}} \mathbf{X}_0 \\ & \quad \approx 0 \end{aligned} \tag{40}$$

which means the data point $\mathbf{X}_0$ located on the hypersurface $F_0^{\mathrm{A}}(\mathbf{X}) - F_0^{\mathrm{B}}(\mathbf{X}) = 0$ is also approximately located on the hypersurface $F_1^{\mathrm{A}}(\mathbf{X}) - F_1^{\mathrm{B}}(\mathbf{X}) = 0$. In other words, the two hypersurfaces produced by our population coding system before and after a weight updating is near to each other when the responses of those population coding neurons activated by the data point $\mathbf{X}_0$ are enough large.

As illustrated in Fig. 12(a) and 12(b), the global hyperplane produced by Perceptron is determined by the connection weights from the input layer to the neuron in the output layer. Our coding system transforms such global hyperplane into multiple local hyperplanes determined by the connection weights from the input layer to the population coding neurons in the second layer in Fig. 11(b). Local variation led by the modification of weights associated with each coding neuron in our system is limited and is not as large as the global variation led by the modification of weights associated with the neuron in the output layer of the Perceptron.

As for a more complex learning machine, such as the multilayer Perceptron (MLP) [60], which can produce a global hypersurface composed of multiple local hyperplanes. Each local hyperplane is controlled by the connecting weights between a neuron in the hidden layer and the input layer. And the global hypersurface is controlled by the connecting weights between a class neuron in the output layer and all the neurons in the hidden layer rather than sparse population coding neurons as our system does. This case is similar to that of Perceptron. Perceptron outputs a global hyperplane which is controlled by the connecting weights between a class neuron in the output layer and all the neurons in the input layer. In other words, due to the



(a) Perceptron in side-view (b) Our population coding system in bird-view

(c) Local hyper-plane variation of our population coding system to the weight modification

Fig. 12. Illustration of hyperplanes and their stabilities in Perceptron and our coding system in a data space, respectively, in a side-view and a bird-view. (a) Two hyperplanes ($f_0(\mathbf{X}) = \mathbf{W}_0^T\mathbf{X} + b = 0, f_1(\mathbf{X}) = \mathbf{W}_1^T\mathbf{X} + b = 0$) produced by Perceptron before and after a weight modification $\mathbf{W}_1 = \mathbf{W}_0 + \Delta\mathbf{W}$. A test input $\mathbf{Y}$ is sensitive to this larger variation for it is classified to class B ($f_0(\mathbf{Y}) < 0$) and then to class A ($f_1(\mathbf{Y}) > 0$). (b) Two hyperplanes ($F_0^{\mathrm{A}}(\mathbf{X}) = F_0^{\mathrm{B}}(\mathbf{X}), F_1^{\mathrm{A}}(\mathbf{X}) = F_1^{\mathrm{B}}(\mathbf{X})$) produced by our coding system before and after a group of weight modification $\{\mathbf{W}_1^{\mathrm{A}i} = \mathbf{W}_0^{\mathrm{A}i} + \Delta\mathbf{W}^{\mathbf{A}i}, \mathbf{W}_1^{\mathrm{B}j} = \mathbf{W}_0^{\mathrm{B}j} + \Delta\mathbf{W}^{\mathrm{B}j}\}$. A test input $\mathbf{Y}$ is less sensitive to the smaller variation for it is always classified to class B ($F_0^{\mathrm{A}}(\mathbf{Y}) < F_0^{\mathrm{B}}(\mathbf{Y}), F_1^{\mathrm{A}}(\mathbf{Y}) < F_1^{\mathrm{B}}(\mathbf{Y})$). Our coding system transforms the global hyperplane controlled by the connection weights associated with the output neuron in Perceptron into segmented local hyperplanes controlled by the connection weights associated with the population coding neurons. Local variation led by the weight adjustment associated with each coding neuron is limited and is not as large as the global variation led by that associated with the output neuron in Perceptron. (c) When the responses of the population coding neurons activated by the input $\mathbf{X}_0$ are enough large, or if the weight vectors $\mathbf{W}_1^{\mathrm{A}i}$ and $\{\mathbf{W}_0^{\mathrm{B}j}, \mathbf{W}_1^{\mathrm{B}j}\}$ associated with these population coding neurons are enough close to the data point $\mathbf{X}_0$, the local hyperplane variation of our population coding system to the weight modification is small and the hyperplane spanned by the class difference vectors $\Delta\mathbf{W}_0^{\mathbf{BA}}$ and $\Delta\mathbf{W}_1^{\mathbf{BA}}$ is approximately orthogonal to $\mathbf{X}_0$. The data point $\mathbf{X}_0$ located on the hypersurface $F_0^{\mathrm{A}}(\mathbf{X}) = F_0^{\mathrm{B}}(\mathbf{X})$ is also approximately located on the hypersurface $F_1^{\mathrm{A}}(\mathbf{X}) = F_1^{\mathrm{B}}(\mathbf{X})$

influence caused by the weight modification associated to all the hidden neurons, MLP is also sensitive to the weight updating. In addition, the number of the neurons in the "hidden layer" of our system is not fixed as that in the hidden layer of MLP. In MLP's classical form, if the input neurons are viewed as the first layer neurons, the number of neurons in the second layer or the hidden layers is set to a predetermined number. Each hidden-layer neuron with its connecting weights to the first layer represents a hyperplane in the data space. The number of neurons in the hidden layer determines the corresponding number of hyperplanes to be learned. The weight updating means rotating and translating all the existed hyperplanes to fit or separate the data. However, the complexities of problems are different for various data and tasks. It needs a flexible model with different complexities to fit the different tasks. Classical MLP fails to satisfy this requirement with its fixed number of the hidden-layer neurons or unchangeable model complexity. Therefore, the authors consider that a very important aspect to developmental learning is how to determine the number of neurons in hidden layer or the complexity of the model, especially for simulating the development of a baby's brain since he or she is born till grows to be an adult whose number of brain neurons is stable. To address this problem, we proposed the developmental system which dynamically generating new coding neurons in the "hidden layer," it memorizes or encodes the new visual context when it interacts with the environment and fails to predict the target location with its current visual context knowledge.

The above theoretical analysis and comparison try to explain the phenomenon that the performance of our system using multistep learning is very close to the performance of the system with only one step weight-updating. However, our system with one-step weight updating is faster and stable on visual context encoding.

V. Experiments on Coding for Gaze Movement Control in Target Search

We implemented two visual context coding systems, respectively, by using single-cell coding and population-cell coding

Fig. 13. Face database of the University of Bern (320 × 214 pixels).

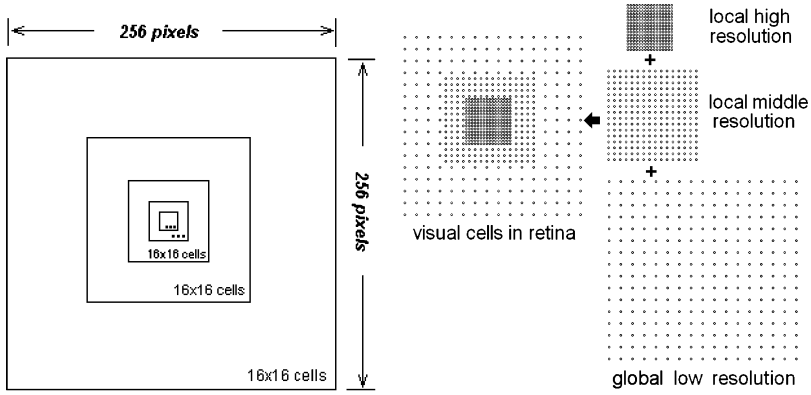


Fig. 14. Retina imaging simulation. (a) Input neurons composed of five overlapped visual fields with 16 × 16 cell arrays and different intervals [16, 8, 4, 2, and 1 pixel(s)] between two adjacent neurons. (b) An illustration of a simplified rectangle distribution of visual sensing cells in the primate retina (here, only overlapped input neuron arrays in three scales are shown).

mechanisms for target searching. In addition, a full-encoding system, k-NN-based coding system, is built for efficiency and performance comparison. In this system, the visual context is encoded at the learning stage regardless of the predicting results, based on all given starting gaze points uniformly distributed on images in all the scales. And the visual context is decoded at the test stage by using k-NN. In other words, the system encodes all the visual contexts it encountered at the encoding stage and uses the first k coding neurons with largest responses to represent a visual field image and synthesize a gaze movement at the decoding stage.

The three coding systems are compared based on searching two kinds of targets: the left eye centers and the mouth centers of humans. The head-shoulder image database from the University of Bern [57] has been used. In this database, totally there are 300 images from 30 people in ten different poses (ten images each person). The image size is 320 × 214 pixels. The average radius of the eyeballs of these 30 persons is 4.02 pixels. Fig. 13 lists the first ten images.

For each target center, two experiments were designed to compare the systems' performance. The first experiment (Exp. 1) used a training set consisting of 30 images (the frontal pose image of 30 persons) and a test set with 210 images (30 people in nine other different poses). The second experiment (Exp. 2) used a training set with 90 images (nine people in these 10 different poses), and a test set with 210 images (21 people in these 10 poses), respectively.

A. Structure of Coding Systems

All the coding systems have a group of visual fields in five scales (256 × 256, 128 × 128, 64 × 64, 32 × 32, and 16 × 16 pixels). These visual fields are used to input global or local images by sampling the training and the testing images (320 × 214 pixels). For each scale (or resolution), in the first layer of a coding system, there are five 16 × 16 input neuron arrays with different intervals (16, 8, 4, 2, and 1 pixels). These neurons simulate the distribution of the visual sensing neurons in primate's retinas. In a primate's retina, the density of sensing neurons is high in the central fovea and low in the surrounding area. Each input neuron samples a pixel or a small region [16 × 16, 8 × 8, 4 × 4, 2 × 2, and 1 × 1 pixel(s)] at the corresponding position of images. Illustrated in Fig. 14 there are totally $5 \times 16 \times 16 = 1280$ input neurons in the first layer of this coding structure.

Fig. 2 shows that there are 256 kinds of extended LBP features for each receptive field. In the second layer of the coding system, each feature neuron extracts an extended LBP feature from its receptive field of 3 × 3 input neurons. Each receptive field has 1/2 overlap with its neighboring receptive fields in five visual fields. Thus, there are totally $[16-(3-1)]^2 \times 256 \times 5 = 250\,880$ feature neurons. Among these feature neurons, at most $250\,880 \times (1/256) = 980$ neurons (the first m feature neurons with largest responses, $m = 1$ for sparsity, see Section II) contribute to activate the single or population coding neurons in the third layer.

The number of coding neurons in the third layer is dynamic, which is dependent on the natural categories of the visual context and different data and tasks.

The number of gaze movement control neurons in the fourth layer is two. The two control neurons output two values representing gaze shifts in x and y directions, respectively.

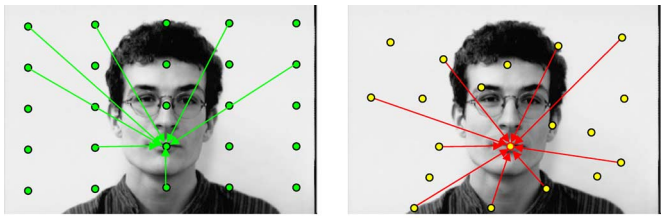


Fig. 15. Visual context learning and testing. (a) Encoding or learning visual context between the mouth center and a group of initial gaze points uniformly distributed on the image. (b) Decoding or testing for gaze movement control for mouth center search from a group of initial gaze points randomly distributed on the image.

B. Experiment on Encoding or Learning

In Section IV, we proposed a model selection algorithm that determining the number of the first M coding neurons. In order to verify the algorithm to be reasonable and feasible, we carried out an experiment on encoding the visual context and searching the mouth center.

Illustrated in Fig. 15, the visual context was encoded (or learned) with a group of initial gaze points uniformly distributed on the image, and decoded (or tested) with a group of initial gaze points randomly distributed on the image. The system was trained and tested on mouth center searching based on visual context encoding and decoding strategies.

Table IV listed the experimental results obtained by using the algorithm shown in Table I to construct the coding system where Exp. 1 used 30 images for training and 270 images for testing; Exp. 2 used 90 images for training and 210 images for testing; the parameter P is the similarity factor defined in Section IV for determining the value of M (the number of the population coding neurons with the first M largest responses); the averaged M means the average number of coding neurons participated in visual context encoding or decoding procedure; N is the number of total coding neurons generated in the third layer, which represented the system's model complexity; the system's generalization error (or test error) is evaluated by the comprehensive error $\sqrt{\mathrm{mea}^2 + \mathrm{std}^2}$ (unit: pixel) described in Section IV-A. It combines the mean (mea) and the standard deviation (std) of target locating errors. The model consistency was evaluated according to the comprehensive test errors generated with different model complexities by using these two datasets. The evaluation results are plotted in Fig. 16.

It can be seen that for Exp. 1 (30 training images versus 270 testing images), the smallest model complexity ($N = 2.875$ thousand coding neurons) corresponds to the best generalized performance $(\mathrm{comprehensive\ error} = 3.54\ \mathrm{pixels})$ when $\mathrm{P} = 0.8$. For Exp. 2 (90 training images verus 210 test images), the smallest model complexity ($N = 7.703$ thousand coding neurons) corresponds to the third best performance $(\mathrm{comprehensive\ error} = 3.34\ \mathrm{pixels})$. These two experiments indicated that the method which we proposed to select a system with good generalized performance is practical and reasonable.

C. Experiments on Gaze Movement Control for Target Search

We carried out a group of visual context coding experiments for searching two targets: the left eye center and the mouth center. These experiments were carried out in two ways: individual search and sequential search. The individual search means encoding and decoding the visual context for the left eye center and the mouth center independently. The sequential search means encoding the visual context from initial gaze points for the first target, and then encoding the visual context from the points around the first target to the second target. After encoding, these sequential contexts are decoded to search two targets one by one.

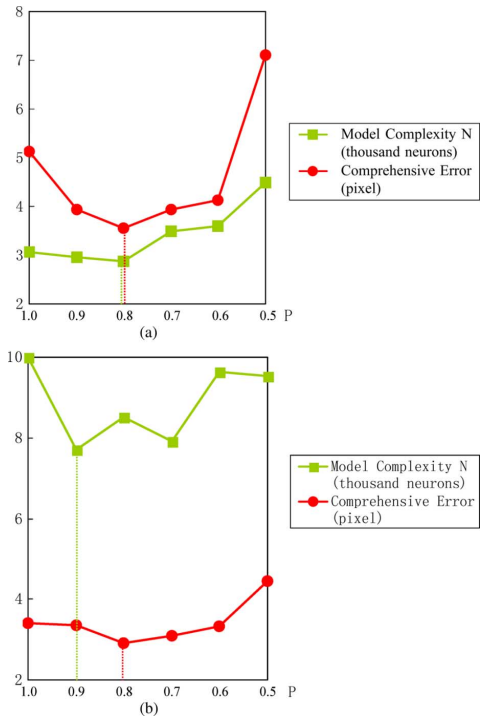


Fig. 16. Correspondence between system complexity and comprehensive test error. (a) Exp. 1: when $P = 0.8$, the smallest model complexity ($N = 2.875$ thousand coding neurons) corresponds to the best generalized performance $(\mathrm{comprehensive\ error} = 3.54\ \mathrm{pixels})$. (b) Exp. 2: when $P = 0.9$, the smallest model complexity ($N = 7.703$ thousand coding neurons) corresponds to a performance $(\mathrm{comprehensive\ error} = 3.34\ \mathrm{pixels})$ close to the best performance $(\mathrm{comprehensive\ error} = 2.91\ \mathrm{pixels})$ when $P = 0.8$.

From Table IV, we know that $P = 0.8$ and $P = 0.9$ make the two systems have the smallest complexities for Exp. 1 and Exp. 2, respectively. Their corresponding average values of M are 2.92 and 1.95 (coding neurons), which means that two groups of population cells are responsible for visual context coding. When $P = 1.0$ the average value of M is one, which means a single cell is responsible for coding. We compared the single and the population cell coding with a benchmark coding system, the k-NN-based coding system. Table V presented the comparison results with fields: the number of feature neurons in the second layer of our coding system, the number of coding neurons in the third layer, the number of connection weights between feature neurons and coding neurons, the mean and the standard deviation of locating errors and the comprehensive test error.

According to our experimental results, the k-NN-based coding system provided the best performance with $k = 3$. The experimental results listed in Table V can be summarized in following three aspects:

TABLE IV
MODEL SELECTION: COMPLEXITY VERSUS GENERALIZATION ERROR

Exp.1 (30 vs. 270)	learning parameters	P	1	0.9	0.8	0.7	0.6	0.5
		average M (neuron)	1	1.67	2.92	5.49	10.58	22.28
	model complexity	N (neuron)	3068	2956	2875	3484	3603	4493
	locating error (pixel)	mean (*mea*)	2.58	2.28	2.44	2.63	2.98	4.27
		standard deviation (*std*)	4.42	3.21	2.57	2.92	2.85	5.67
		comprehensive error $\sqrt{mea^2+std^2}$	5.12	3.94	3.54	3.93	4.12	7.10
Exp.2 (90 vs. 210)	learning parameters	P	1	0.9	0.8	0.7	0.6	0.5
		average M (neuron)	1	1.95	3.96	8.00	16.69	35.28
	model complexity	N (neuron)	9991	7703	8513	7921	9642	9532
	locating error (pixel)	mean (*mea*)	2.29	2.39	2.35	2.48	2.72	3.26
		standard deviation (*std*)	2.53	2.34	1.72	1.87	1.93	3.04
		comprehensive error $\sqrt{mea^2+std^2}$	3.41	3.34	2.91	3.11	3.34	4.46

Fig. 17. Two examples illustrate that the system trained with vertical frontal images can search targets (left eyes and mouths) in other poses to some extent. Located results are represented by two white points in each image.

1) With large samples, in the case of Exp. 2 (90 images are selected as training images), the population-cell coding system provide the similar target locating accuracy as the single-cell coding system and the k-NN-based coding system did. However, it required the lest encoding information. For example, the ratios of the average encoding quantity (0.35 million connection weights for the left eye center) required by the population-cell coding system to the encoding quantities (0.43 and 6.87 million connection weights for left eye center) required by the single-cell coding system and the k-NN-based coding system are about 77% and 5%, respectively;
2) With small samples, in the case of Exp. 1 (30 images are selected as training images), the locating accuracy for the left eye center by the population-cell coding system is 3.11 pixels which is 35.6% and 16.4% higher than the accuracies (4.83 and 3.72 pixels) provided by the single-cell coding system and the k-NN-based coding system, respectively. Meanwhile, the encoding quantity required by the population coding is 12% and 95% smaller than the single and k-NN coding systems;
3) Sequential search can save 16.4% encoding quantity in contrast to individual search with the similar locating accuracy.

In Exp. 1, although the coding system is trained with vertical frontal images for searching eye centers and mouth centers, it can handled the face images in different poses to some extent as two examples illustrated in Fig. 17.

VI. CONCLUSION AND DISCUSSION

In this paper, a population cell coding mechanism for visual context learning and gaze movement controlling are presented. The encoding algorithm proposed in our paper has the developmental and incremental learning characteristics. The fast learning properties of the encoding algorithm with only one-step weight-updating was theoretically proved and compared with the classical learning machines in terms of efficiency and stability. A practical method of model selection in terms of determining the number of the first M population coding neurons for good generalization performance is suggested according to the statistical learning theory. In order to apply it to practical object detection tasks, the issues of encoding quantity and locating accuracy were discussed. The main measures for solving the problems include population cell coding, making use of sequential context, and the computation of maximum density of final gaze points.

Our theoretical analysis and experimental results indicated that the population-cell coding system is generally more efficient than the single-cell coding system and the k-NN-based coding system in representing the visual context and controlling the gaze motion for target searching. The population-cell coding has demonstrated a significant advantage on the case of small samples over other two coding systems. It reached 35.6% and 16.4% higher target locating accuracies and required 12% and 95% lower coding quantities compared with the single-cell coding system and the k-NN-based coding system, respectively.

Because this paper intends mainly to discuss the efficiency of visual context encoding and its top–down control for gaze movement in target search, some respects are not included in the current system. Correspondingly, there are several limitations in our system. 1) Authors temporally did not make use of the bottom–up saliency and the top–down target cues to search and verify targets, so the system assumes that targets are existed in images. These two cues will be utilized in next version of the system. 2) the system assumes targets are in a strong-relevant context. We found that there are two types of visual context: strong-relevant context and weak-relevant context. The

TABLE V
EXPERIMENTS: PERFORMANCES OF THREE CODING SYSTEMS FOR MULTITARGET SEARCH

target	experiment	coding system	number of feature neurons in layer 2	number of coding neurons in layer 3	number of connection weights between feature neurons and coding neurons (million)	locating error (pixel)		
						Mean (*mea*)	standard deviation (*std*)	comprehensive error $\sqrt{mea^2+std^2}$
Left eye center	Exp.1 (30 vs. 270)	Single cell (P=1.0)	250,880	2314	0.43	2.15	4.33	4.83
		Population cell (P=0.9)	250,880	1906	0.35	1.93	2.44	3.11
		k-NN (k=3)	250,880	37379	6.87	2.10	3.07	3.72
	Exp.2 (90 vs. 210)	Single cell (P=1.0)	250,880	7340	5.02	1.64	1.47	2.20
		Population cell (P=0.8)	250,880	5405	1.02	1.89	1.22	2.25
		k-NN (k=3)	250,880	111801	20.7	1.86	1.01	2.12
Mouth center (individual search)	Exp.1 (30 vs. 270)	Single cell (P=1.0)	250,880	3068	0.57	2.58	4.42	5.12
		Population cell (P=0.8)	250,880	2875	0.53	2.44	2.57	2.95
	Exp.2 (90 vs. 210)	Single cell (P=1.0)	250,880	9991	1.88	2.29	2.53	3.41
		Population cell (P=0.9)	250,880	7703	1.46	2.39	2.34	3.34
Mouth center (sequential search)	Exp.1 (30 vs. 270)	Single cell (P=1.0)	250,880	2794	0.54	2.17	3.83	4.4
		Population cell (P=0.8)	250,880	2469	0.48	2.27	2.23	3.18
	Exp.2 (90 vs. 210)	Single cell (P=1.0)	250,880	7700	1.55	2.11	2.48	3.26
		Population cell (P=0.7)	250,880	6186	1.20	2.56	1.77	3.11

strong context exists between a target and other surrounding objects, which are interconnected in a relatively stable mode, such as between the target eye and the objects nose, mouth, and head, or between the target license plate and the objects of car lights, car windows, the car driver, and the car body. The weak context exists among objects that are interconnected in a relatively loosing mode, such as among hands, feet, and the head, or among humans, cars, roads, trees, and the sky. The current system for the weak-relevant context does not perform as well as for the strong-relevant context. The reasons are: a) there are no bottom–up saliency cues for reducing the candidate regions to be searched; and b) "it does not allow accurate estimation of the x coordinate" of the target by using globe futures or context [19]. To address these issues, the top-down context and target cues should be combined with the bottom–up saliency cues and applied in a temporal reasoning mechanism for locating the target accurately.

We still have much work to do for improving our proposed population-cell coding model. In a 2010 Intenational Joint Conference on Neural Networks (IJCNN 2010) panel session [58], the organizers Weng and Roy put forward an open problem: as the brain is "skull-closed," how does it fully autonomously develop its internal representation from one task to the next? For our system presented in this paper, this question becomes: for the visual context encoded for searching the eye center, how to transfer this internal representation to the task of searching the mouth center or other targets? The mouth can be viewed as an equal object that is similar to the eye, because both of them are contained in a face object. To search an eye, we need to reason with the encoded visual context that is from the scene to the human body, from the human body to the face and from the face to the eye. For searching a different target such as a mouth, the first two parts of the visual context can be shared. Therefore, the system only needs to encode the local context from the face or the eye to the mouth. For a more difficult task, such as to search a car, there is no much encoded context knowledge can be shared. However, there is still some weak-relevant context can be helpful on searching. For example, cars usually run on the roads or stop in a car park or beside houses. The car-relevant objects, such as roads, the car park and houses, have relations to humans, where humans often exist. So the encoded context from the scene to the human can be utilized. Of course, in this case, the top–down cues of car features are more important. How to share the encoded representations for different objects is a challenge. Fortunately, there is a great amount of research work have been carried out on transfer learning in the literature [61]–[63]. They can be utilized to study the autonomous mental development. We will explore these issues in our future research.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their constructive comments and suggestions which helped improve the quality of the paper. The authors would also like to thank the technical assistance from graduate students C. Chi and Z. Ma who provided the written material about response time in context studies and the code for drawing the Fig. 7(b), respectively. The authors would like to offer their thanks to Dr. Y. Fu for careful proofreading.

REFERENCES

[1] G. Schneider, "Contrasting visuomotor functions of the tectum and cortex in the golden hamster," *Psychol. Forschung*, vol. 31, no. 1, pp. 52–62, 1967.

[2] R. Held, D. Ingle, G. Schneider, and C. Trevarthen, "Locating and identifying: Two modes of visual processing," *Psychol. Forschung*, vol. 31, no. 1, pp. 42–43, 1967.

[3] L. Ungerleider and M. Mishkin, "Two cortical visual systems," in *Analysis of Visual Behavior*, D. J. Ingle, M. A. Goodale, and R. J. W. Mansfield, Eds. Cambridge, MA: MIT Press, 1982, pp. 549–586.

[4] L. Ungerleider and J. Haxby, "'What' and 'where' in the human brain," *Current Opinion Neurobiol.*, vol. 4, no. 2, pp. 157–165, 1994.

[5] What and Where Pathways [Online]. Available: http://www.scholarpedia.org/article/What_and_where_pathways

[6] B. Velichkovsky, M. Joos, J. Helmert, and S. Pannasch, "Two visual systems and their eye movements: Evidence from static and dynamic scene perception," in *Proc. 27th Conf. Cogn. Sci. Soc.*, Stresa, Italy, Jul. 21–23, 2005, pp. 2283–2288.

[7] N. Broadbent, L. Squire, and R. Clark, "Spatial memory, recognition memory, and the hippocampus," in *Proc. Nat. Acad. Sci. USA*, 2004, vol. 11, pp. 14515–14520.

[8] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2007.

[9] R. Mcpeek and E. Keller, "Saccade target selection in the Superior Colliculus during a visual search task," *J. Neurophysiol.*, vol. 88, no. 4, pp. 2019–2034, 2002.

[10] G. Shepherd, *Neurobiology*, 2nd ed. London, U.K.: Oxford Univ. Press, 1988.

[11] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, 2nd ed. Berlin, Germany: Springer-Verlag, 2007.

[12] Z. Ji, J. Weng, and D. Prokhorov, "Where-what network 1: "Where" and "what" assist each other through top-down connections," in *Proc. 7th IEEE Int. Conf. Develop. Learn.*, Monterey, CA, 2008, pp. 61–66.

[13] R. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, Jun. 2007.

[14] G. Zelinsky, W. Zhang, B. Yu, X. Chen, and D. Samaras, "The role of top-down and bottom-up processes in guiding eye movements during visual search," in *Proc. Adv. Neural Inform. Process. Syst.*, Vancouver, BC, Canada, 2006.

[15] M. Cerf, J. Harel, W. Einhaeuser, and C. Koch, "Predicting human gaze using low-level saliency combined with face detection," in *Proc. Adv. Neural Inform. Process. Syst.*, Vancouver, BC, Canada, 2007.

[16] R. Milanese, H. Wechsler, S. Gil, J. Bost, and T. Pun, "Integration of bottom-up and top-down cues for visual attention using non-linear relaxation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Hilton Head, SC, 1994, pp. 781–785.

[17] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artif. Intell.*, vol. 78, pp. 507–545, 1995.

[18] V. Navalpakkam, J. Rebesco, and L. Itti, "Modeling the influence of task on attention," *Vis. Res.*, vol. 45, no. 2, pp. 205–231, 2005.

[19] A. Torralba, "Contextual priming for object detection," *Int. J. Comput. Vis.*, vol. 53, no. 2, pp. 169–191, 2003.

[20] A. Oliva, A. Torralba, M. Castelhano, and J. Henderson, "Top down control of visual attention in object detection," in *Proc. IEEE Int. Conf. Image Process.*, Barcelona, Spain, 2003, vol. I, pp. 429–432.

[21] K. Murphy, A. Torralba, and W. Freeman, "Using the forest to see the trees: A graphical model relating features, objects, and scenes," in *Proc. Adv. Neural Inform. Process. Syst.*, Vancouver, BC, Canada, 2003.

[22] A. Torralba, A. Oliva, M. Castelhano, and J. Henderson, "Contextual guidance of eye movements and attention in real-world scenes: The role of global features on objects search," *Psychol. Rev.*, vol. 113, 2006.

[23] K. Ehinger, B. Hidalgo-Sotelo, A. Torralba, and A. Oliva, "Modelling search for people in 900 scenes: A combined source model of eye guidance," *Vis. Cogn.*, vol. 17, no. 6–7, pp. 945–978, 2009.

[24] L. Paletta and C. Greindl, "Context based object detection from video," in *Proc. Int. Conf. Comput. Vis. Syst.*, Graz, Austria, 2003, pp. 502–512.

[25] H. Kruppa, M. Santana, and B. Schiele, "Fast and robust face finding via local context," in *Proc. Joint IEEE Int. Workshop Vis. Surveillance Perform. Eval. Tracking Surveillance*, Nice, France, 2003.

[26] N. Bergboer, E. Postma, and H. van den Herik, "Context-based object detection in still images," *Image Vis. Comput.*, vol. 24, pp. 987–1000, 2006.

[27] J. Miao, X. Chen, W. Gao, and Y. Chen, "A visual perceiving and eyeball-motion controlling neural network for object searching and locating," in *Proc. Int. Joint. Conf. Neural Netw.*, Vancouver, BC, Canada, 2006, pp. 4395–4400.

[28] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. Comput. Vis. Pattern Recog.*, San Juan, Puerto Rico, 1997, vol. 3, pp. 130–136.

[29] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces and cars," in *Proc. Comput. Vis. Pattern Recog.*, Hilton Head, SC, 2000, vol. 1, pp. 746–751.

[30] P. Viola and M. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, 2004.

[31] X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in *Proc. Comput. Vis. Pattern Recog.*, Washington, DC, 2004.

[32] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1408–1423, Nov. 2004.

[33] G. Malcolm and J. Henderson, "Combining top-down processes to guide eye movements during real-world scene search," *J. Vis.*, vol. 10, no. 2, pp. 1–11, 2010.

[34] M. Chun and Y. Jiang, "Contextual cueing: Implicit learning and memory of visual context guides spatial attention," *Cogn. Psychol.*, vol. 36, pp. 28–71, 1998.

[35] M. Chun, "Contextual cueing of visual attention," *Trends Cogn. Sci.*, vol. 4, no. 5, pp. 170–178, 2000.

[36] J. Henderson, P. Weeks, Jr., and A. Hollingworth, "The effects of semantic consistency on eye movements during complex scene viewing," *J. Exp. Psychol.: Human Perception Perform.*, vol. 25, no. 1, pp. 210–228, 1999.

[37] M. Kunar, S. Flusberg, and J. Wolfe, "Contextual cueing by global features," *Perception Psychophys.*, vol. 68, no. 7, pp. 1204–1216, 2006.

[38] J. Brockmole, M. Castelhano, and J. Henderson, "Contextual cueing in naturalistic scenes: Global and local context," *J. Exp. Psychol.: Learn. Memory and Cogn.*, vol. 32, no. 4, pp. 699–706, 2006.

[39] J. Brockmole and J. Henderson, "Using real-world scenes as contextual cues for search," *Vis. Cogn.*, vol. 13, no. 1, pp. 99–108, 2006.

[40] K. Chua and M. Chun, "Implicit scene learning is viewpoint dependent," *Perception Psychophys.*, vol. 65, no. 1, pp. 72–80, 2003.

[41] M. Bear, B. Connors, and M. Paradiso, *Neuroscience: Exploring the Brain*, 2nd ed. New York: Lippincott Williams & Wilkins, 2001.

[42] "Special issue on binding problem," *Neuron*, vol. 24, no. 1, 1999.

[43] D. Wang, "The time dimension for scene analysis," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1401–1426, Jun. 2005.

[44] J. Weng and W. Hwang, "From neural networks to the brain: Autonomous mental development," *IEEE Comput. Intell. Mag.*, vol. 1, no. 3, pp. 15–31, Aug. 2006.

[45] M. Young and S. Yamane, "Sparse population coding of faces in the inferotemporal cortex," *Science*, vol. 256, no. 1, pp. 1327–1330, 1992.

[46] J. Weng and N. Zhang, "Optimal in-place learning and the lobe component analysis," in *Proc. Int. Joint Conf. Neural Netw.*, Vancouver, BC, Canada, 2006, pp. 3887–3894.

[47] J. Weng, T. Luwang, H. Lu, and X. Xue, "Multilayer in-place learning networks for modeling functional layers in the laminar cortex," *Neural Netw.*, vol. 21, pp. 150–159, 2008.

[48] A. Bell and T. Sejnowski, "The independent components of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.

[49] S. Hornillo-Mellado, R. Martin-Clemente, C. Puntonet, and J. Gorriz, "Connections between ICA and sparse coding revisited," *Lecture Notes Comput. Sci.*, vol. 3512, pp. 1035–1042, 2005.

[50] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vis. Res.*, vol. 37, pp. 3313–3325, 1997.

[51] A. Hyvarinen and P. Hoyer, "A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images," *Vis. Res.*, vol. 41, no. 18, pp. 2413–2423, 2002.

[52] D. Lee and H. Seung, "Learning the parts of objects with nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[53] J. Weng and M. Luciw, "Dually optimal neuronal layers: Lobe component analysis," *IEEE Trans. Autonom. Mental Develop.*, vol. 1, no. 1, pp. 68–85, May 2009.

[54] T. Ahonen, A. Hadid, and M. Pietikainen, "Face recognition with local binary patterns," in *Proc. 8th Eur. Conf. Comput. Vis.*, Prague, Czech Republic, 2004, vol. 3021, pp. 469–481.

[55] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Wiley, 1995.

[56] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer-Verlag, 2001.

[57] The Face Database of the University of Bern [Online]. Available: ftp://iamftp.unibe.ch/pub/Images/FaceImages/ 2008

[58] Between Bottom-Up and Top-Down What is "The Much In-Between"? Panel Session for IJCNN, 2010 [Online]. Available: http://www.cse.msu.edu/ei/IJCNN10panel
[59] F. Rosenblatt, "Perceptron simulation experiments," in *Proc. Inst. Radio Eng.*, 1960, vol. 48, pp. 301–309.
[60] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.
[61] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.
[62] R. Raina, A. Ng, and D. Koller, "Constructing informative priors using transfer learning," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, 2006.
[63] W. Dai, Y. Chen, G. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. Adv. Neural Inform. Process. Syst.*, Vancouver, BC, Canada, 2008, vol. 21.

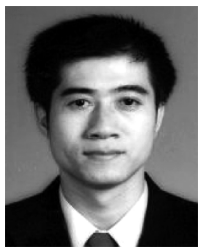
Jun Miao (S'00–M'04) received the B.Sc. and M.Sc. degrees in computer science from Beijing University of Technology, Beijing, China, in 1993 and 1999, respectively. He received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2005.

He is currently an Associated Professor at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His research interests include artificial intelligence, neural networks, neural information processing, image understanding, and biological vision. He has published more than 30 research articles in refereed journals and proceedings on face detection, visual neural networks, visual neural information coding, neural oscillation, image segmentation, visual perception, and cognition. His two main contributions are the technique of Human Face Gravity-Center Template for face detection and the model of Visual Perceiving and Eyeball-Motion Controlling Neural Network for visual search, respectively.

Dr. Miao is a member of the China Computer Federation and a member of the Chinese Society for Neuroscience. He is the recipient of Microsoft Fellowship Award in 2000, a recipient of 2003 Shanghai Science and Technology Progress Awards (the First Award), and a recipient of 2005 National Science and Technology Progress Awards of China (the Second Award).

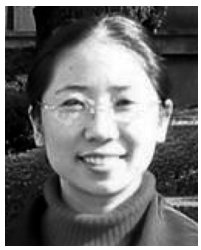
Laiyun Qing (S'03–M'09) received the B.Sc. and M.Sc. degrees in computer science from Northeastern University, Shenyang, China, in 1996 and 1999, respectively. She received the Ph.D. degree in computer science from Chinese Academy of Sciences, Beijing, in 2005.

She is currently an Associated Professor at the School of Information Science and Engineering, Graduate University of the Chinese Academy of Sciences, Beijing. Her research interests include pattern recognition, image processing, and statistical learning. Her current research focuses on visual perception and cognition.



Baixian Zou received the M.Sc. degree in applied mathematics from South East University, Nanjing, China, in 1996, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2004.

He is now a Lecturer at the Department of Information Science and Technology, College of Arts & Science of Beijing Union University, China. He is also a part-time Associate Professor at School of Mathematics and Computer Science, Fujian Normal University, Fujian, China. His research involves image processing and computer network, especially sparse coding for the natural image. He has published more than 15 research articles on related research subjects, including sparse coding, network traffic modeling, network anomaly detection, and MPLS traffic engineering.



Lijuan Duan (M'08) received the B.Sc. and M.Sc. degrees in computer science from Zhengzhou University of Technology, Zhengzhou, China, in 1995 and 1998, respectively. She received the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, in 2003.

She is currently an Associated Professor at the College of Computer Science and Technology, Beijing University of Technology, China. Her research interests include artificial intelligence, image processing and machine vision, and information security. She has published more than 40 research articles in refereed journals and proceedings on image retrieval, neural oscillation, image segmentation, visual perception, and cognition.

Wen Gao (M'88–SM'05–F'09) received the B.Sc. and M.Sc. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1985 and 1988, respectively. He received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently a Professor of Computer Science at Peking University, Beijing, China. Before joining Peking University, he was a Professor of Computer Science at Harbin Institute of Technology from 1991 to 1995, and a professor at the Institute of Computing Technology of Chinese Academy of Sciences. He has published extensively, including four books and over 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics.

Dr. Gao serves on the editorial board for several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, the *EURASIP Journal of Image Communications*, and the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also served on the advisory and technical committees of numerous professional organizations.