Incremental Online Object Learning in a Vehicular Radar-Vision Fusion Framework

Zhengping Ji, *Member, IEEE,* Matthew Luciw, *Member, IEEE,* Juyang Weng, *Fellow, IEEE* and Shuqing Zeng, *Member, IEEE*

***Abstract*—In this report, we propose an object learning system that incorporates sensory information from an automotive radar system and a video camera. The radar system provides a coarse attention for the focus of visual analysis on relatively small areas within the image plane. The attended visual areas are coded and learned by a 3-layer neural network utilizing what is called in-place learning: each neuron is responsible for the learning of its own processing characteristics within the connected network environment, through inhibitory and excitatory connections with other neurons. The modeled bottom-up, lateral, and top-down connections in the network enable sensory sparse coding, unsupervised learning and supervised learning to occur concurrently. The presented work is applied to learn two types of encountered objects in multiple outdoor driving settings. Cross validation results show that the overall recognition accuracy is above 95% for the radar-attended window images. In comparison with the uncoded representation and purely unsupervised learning (without top-down connection), the proposed network improves the overall recognition rate by 15.93% and 6.35%, respectively. The proposed system is also compared with other learning algorithms favorably. The result indicates that our learning system is the only one fit for the incremental and online object learning in the real-time driving environment.**

***Index Terms*—Intelligent vehicle system, sensor fusion, object learning, biologically inspired neural network, sparse coding.**

I. Introduction

The field of intelligent vehicles has been rapidly growing in the last two decades [1]. Examples include both fully autonomous driving vehicles [2] [3] [4] and Advanced Safety Driver Assistance Systems (ASDAS) [5] [6], such as adaptive cruise control (ACC), lane departure warning (LDW) and collision avoidance system, etc. The success of intelligent vehicle systems depends on a rich understanding of the complex road environment, which contains many signals and cues that visually convey information, such as traffic lights, road signs, other vehicles, and pedestrians, to name a few. To take correct and intelligent actions in the driving conditions, recognition of the varied objects becomes one of the most critical tasks.

Vision and radar systems have complimentary properties for object detection and validation. As one type of active sensors, a radar system has shown good performance detecting objects in driving environments. It provides fairly accurate measurements of the object's distance and velocity, and remains robust under various weather conditions. However, present radars installed on a vehicle do not have enough lateral resolution to model object's shape, leading to a limitation of recognizing object types. On the contrary, video cameras, are able to provide sufficient lateral resolution to analyze objects. The cues of shapes, furthermore the appearance, give more details of the characteristics of different objects.

The fusion of radar and vision information has been widely discussed and utilized in intelligent vehicle systems. Early fusion framework analyzed radar positions in a vision-based lane recognition system to achieve better lane estimation (e.g. Jochem and Langer 1996 [7], Gern et al. 2000 [8] and Hofmann et al. 2000, 2003 [9] [10]). Afterwards, radar-vision approaches are more focused on the fusion for detecting target (e.g., vehicle, pedestrian, etc.) level. Grover et al. 2001 [11] extracted low level blob features in a single radar map and a single night-vision image. The fusion was performed in polar coordinates to determine vehicle localization based on angular positions. Kato et al. 2002 [12] fused radar tracks and motion stereos together to identify the distance and vertical boundaries of objects in an urban road environment. Sole et al. 2004 [13] treated video and radar sensors as two independent sources of target acquisition: the matched targets were validated and did not require further processing while unmatched radar targets were processed via motion and texture analysis for further validation. Alessandretti et al. 2007 [14] estimated regions of interest (ROI) from radar returns, where vertical symmetries were used to search vehicles in the attended small areas. Using the similar mechanism of ROI provided by radars, Kadow et al. 2007 [15] and Bertozzi et al. 2008 [16] developed an optimized symmetry measure and new motion stereos, respectively, to detect and track other vehicles. Recently, Wu et al. 2009 [17] fused information from a stereo-camera and millimeter-wave radar to estimate the location, pose and motion information of a threat vehicle within 20 meters range.

However, the quantitative evaluation (e.g., average recognition rate) of object recognition/detection is missing in most of the work above. In addition, aforementioned fusion researches mainly detected key objects (i.e., vehicles or pedestrians) using object-specific features, such as blobs, edges, symmetries and motion, etc. The object-specific (also called task-specific) perceptual approach is not suited to provide perceptual awareness in complex environments with various objects of interest.

In the proposed work, we take advantage of radar-vision integration to achieve an efficient attention selection on candi-

Zhengping Ji is with the Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM, 87545 and the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824. E-mail: jizhengp@cse.msu.edu.

Matthew Luciw and Juyang Weng are with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 E-mail: {luciw, weng}@cse.msu.edu.

Shuqing Zeng is with the Research and Development Center, General Motor Inc., Warren, MI 48090. Email: shuqing.zeng@gm.com

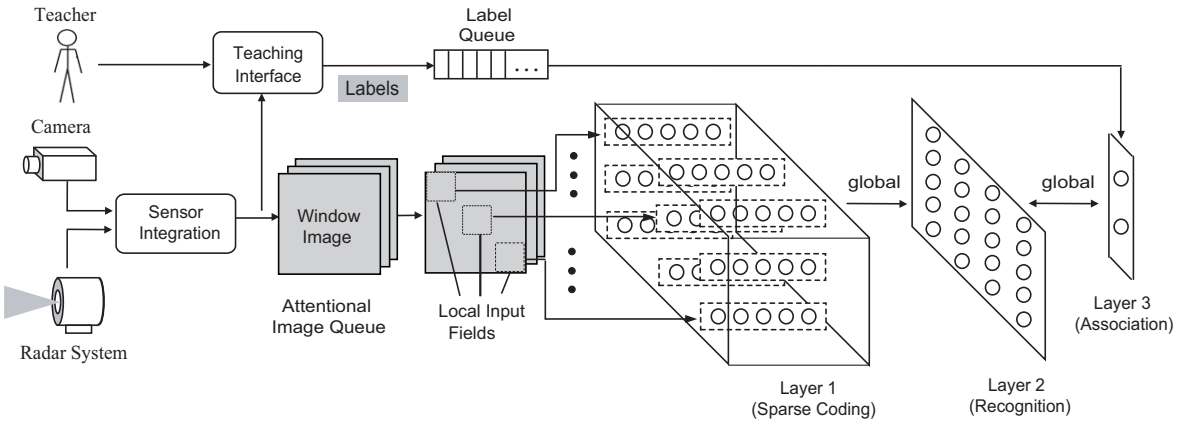


Fig. 1: An outline of the system architecture.

date targets, and employ a generic object learning network to identify object classes without using the low-level and mid-level object-specific features. A cortex-inspired neural network integrates 3-way computations (i.e., bottom-up, top-down and lateral) to code object samples in an over-complete space and learn the distribution of coded "key" object patterns for favorable recognition performance. Its in-place learning mechanism provides the incremental learning optimality, and comparatively low operational complexity even for a very large network.

A successful implementation requires a combination of the following challenges (to the best of our knowledge, no existing study meets them all): (1) General radar-vision fusion framework not constrained for a task-specific learning; (2) Visual sensory sparse coding via statistical independence of developed features; (3) Incremental object learning adaptive to the changing of environments and objects; (4) Online real-time speed due to low computation complexity; (5) Integration of supervised learning (via top-down propagation) and unsupervised learning (via bottom-up propagation) in any order suited for development. All the properties above, coupled with a nurturing and challenging environment, as experienced through sensors and effectors, allow the automatic perceptual awareness to emerge as an important research area in intelligent vehicles.

II. Architecture

An outline of the system architecture is shown in Fig.1. The eventual goal is to enable a vehicle-based agent to develop the ability of perceptual awareness, with applications including autonomous driving and advanced driver assistance. Perceptual awareness is a conceptual and symbolic understanding of the sensed environment, where the concepts are defined by a common language[1] between the system and the teachers or users. In this report, a teacher points out sensory examples of particular conceptual object classes (e.g., vehicle, pedestrian, traffic lights, and other objects that are potential driving hazards). The system learns to associate a symbolic token with the sensed class members, even those that have not been exactly sensed before, but instead share some common characteristics (e.g., a van can be recognized as a vehicle by the presence of a license plate, wheels and tail lights). More complicated perceptual awareness beyond recognition involves abilities like counting and prediction.

In Fig.1, the camera and the radar system work together to generate a set of attended window images, containing environment objects. A teacher communicates with the system through an interface to train the class labels of objects. A 3-layer network provides the processing and learning of the extracted window images. The number of neurons in each layer is specified at a 3D grid (see Fig. 4 for the set of parameters). Layer 1 encodes the local input fields of each window image using self-developed orientation-selective features. Neurons in layer 2 learn the sparse-coded object representations, associated by layer 3 with teacher's output tokens.

III. Coarse Attention Selection

Two kinds of external (outward looking) sensors are used in the proposed architecture. One is a system of radars, composed of one long-range radar and four short-range radars, utilized to find attended targets (with possible false alarms) in the environment. The other senses the vision modality. Information from this sensor is used to develop the ability to recognize objects and identify false alarms. Tables I & II specify the sensor parameters of radar and vision modalities, respectively.

TABLE I: Sensor specifications of the radar system

Key parameters	Specification
Refreshing rate	10 Hz
No. of targets	max. of 20 targets
Max. range	$150\text{m} \pm \mathbf{max}(5\%, 1.0\text{m})$
Field of view	180^{o} ($\leq 30\text{m}$); 15^{o} ($> 30\text{m}$)
Range rate	$\pm 56\text{m/s} \pm 0.75\text{m/s}$

As shown in Fig. 2 (right), a group of target points in 3D world coordinates can be detected from the radar system, with a detection range of up to 150 meters. Each radar point is presented by a triangle, associated with a bar, whose length

[1]The language can be as simple as a pre-defined set of tokens or as complex as human spoken languages.

TABLE II: Sensor specifications of the video camera

Key parameters	Specification
Refreshing rate	15 Hz
Field of view	45°
Resolution	320 × 240

and direction indicate the relative speed of an object. As a rudimentary but necessary attention selection mechanism, we discarded radar returns of more than 80 meters in distance ahead or more than 8 meters to the right or left outside the vehicle path (e.g., the red triangle points in Fig. 2 (right) are omitted).

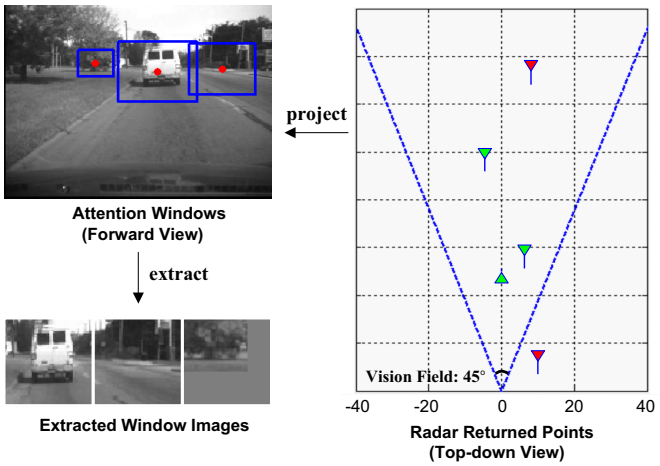


Fig. 2: A projection of valid radar points (green) onto the image plane, where window images are extracted for further recognition.

Based on the estimation of maximum height (3.0 meters) and maximum width (3.8 meters) of environment targets, a rectangular target window (with fixed size 3.0×3.8 m^2) is generated to be centered at each valid radar point. All the target windows at each time t are then projected into the corresponding image via perspective mapping transformation. The transformation is performed by the calibration data that contain the intrinsic and extrinsic parameters of each camera. For example, if the radar-returned object distance (to the host vehicle) is large, the attention window in the image is small and vice versa.

For each attention window, the pixels are extracted as a single image and most of the non-target or background pixels (e.g., the part of sky, road and side grass in Fig. 2 (upper left)) have been filtered out. Each image is normalized in size, in this case to 56 rows and 56 columns as shown in Fig. 2 (bottom left). To avoid stretching small images, if the attention window could fit, it was placed in the upper left corner of the size-normalized image, and the other pixels are set to be uniform gray.

There may be more than one object in each window image, but for the purpose of object identification, the image is assigned with only one label. The labeled radar windows create a set of selected areas while the rest of the image is ignored. This is called coarse attention selection — finding candidate areas

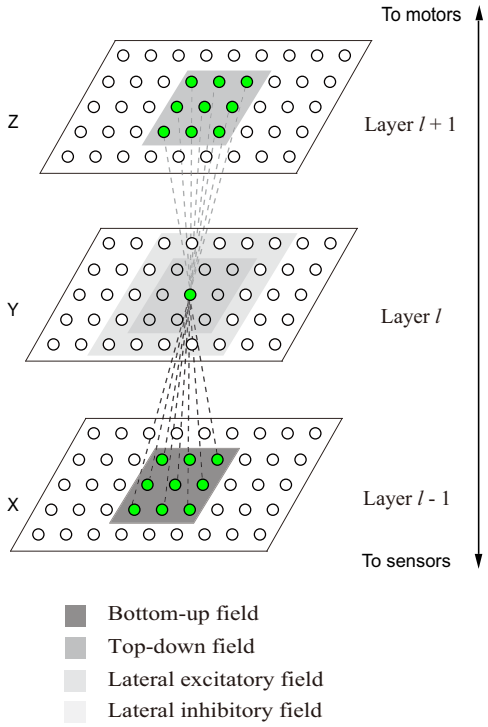


Fig. 3: General structure of the network connection. Neurons are placed (given a position) on different layers in an end-to-end hierarchy – from sensors to motors. Only the connections to a centered cell are shown, but all the other neurons in the same feature layer have the same default connections.

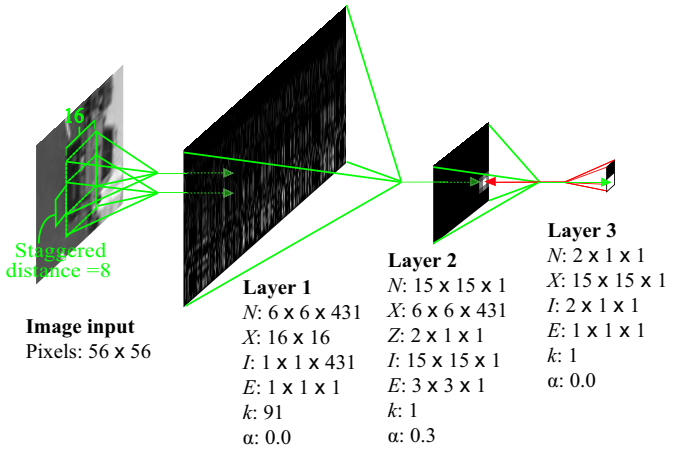
purely based on physical characteristics of radar returns. The attended window images may still contain some information unrelated to the object, such as "leaked-in" background behind the object. However, our object learning scheme does not require the good segmentation of the object itself, but instead it depends on the discriminant statistical distributions of the scenes in each window image. The proposed system can learn to detect and recognize multiple objects within the image captured by the video camera, as long as a radar point is returned for each one.

IV. Object Learning Network

The attended window images are coded and learned through the proposed neural network (see Fig. 1) via 3 layers, till the motor output, where each neuron in the motor layer corresponds to one object class. Fig. 3 shows the general structure of the network connection with three consecutive layers. Every neuron at layer l is connected with four types of connection weights:

1) Bottom-up weight vector $\mathbf{w_b}^{(l)}$ that links connections from its bottom-up field in the previous level.
2) Top-down weight vector $\mathbf{w_t}^{(l)}$ that links connections from its top-down field in the next level.
3) Lateral weight vector $\mathbf{w_h}^{(l)}$ that links inhibitory connections from neurons in the same layer (larger range).
4) Lateral weight vector $\mathbf{w_e}^{(l)}$ that links excitatory connections from neurons in the same layer (smaller range).

Note that each linked weight pair (i, j) shares the same value, i.e., $\mathbf{w_t}_{i,j}^{(l-1)} = \mathbf{w_b}_{j,i}^{(l)}$. Moreover, this work does not use explicit lateral connections, but instead uses an approximate method: the top-k winners (i.e., k largest responses) along with their excitatory neighbors update and fire. The suppressed neurons are considered laterally inhibited and the winning neurons are considered laterally excited.



Parameters: *N*: No. neurons *X*: Bottom-up fields *Z*: Top-down fields *I*: Inhibitory fields *E*: Excitatory fields *k*: Top-k winners in *I* α: Top-down influence

Fig. 4: An example of layer representations (i.e., responses) in the proposed neural network, including a specific set of resource parameters implemented. Green and red directed lines show the bottom-up and top-down connections to the firing neurons, respectively. It is noted that the bottom-up fields of layer 1 neurons are 16×16 local areas over the entire 56×56 image plane, with a stagger distance per 8 pixels, and the top-down fields are not available in layer 1 and layer 3. In addition, neural representations in layer 1 are reshaped to 36×431 for visualization purpose.

The object learning network is incrementally updated at discrete times, $t = 0, 1, 2, ...$, taking inputs sequentially from sensors and effectors, computing responses of all neurons, and producing internal and external actions through experience. Fig. 4 shows an example of network computation, layer by layer, as well as key parameters used in the network implementation.

As described in Algorithm 1, layer 1 of the proposed network develops earlier than other layers, which is inspired from the biological fact that early cortical regions in the brain (e.g., primary visual cortex) would develop earlier than the later cortical regions [18]. Given $t = 1, 2, ...500000$, the network receives 56×56-pixel (same as attention window dimension) natural image patches, which were randomly selected from the thirteen natural images[2]. Neurons are learned through the in-place learning algorithm described in Algorithm 2, however, without supervision on motors. After 500000 updates of layer 1 neurons, their bottom-up features tend to converge. Then the network perceives radar-attended images and all the layers are developed through the same in-place learning procedure in Algorithm 2, whereas supervised signals from a teacher are given in the motor layer 3.

The network performs an open-ended online learning while internal features "emerge" through interaction with its extracellular environment. All the network neurons share the same learning mechanism and each learns on its own, as a self-contained entity using its own internal mechanisms. In-place learning, representing a new and deeper computational understanding of synaptic adaptation, is rooted in the genomic equivalence principle [19]. It implies that there cannot be a "global", or multi-cell, goal to the learning, such as the minimization of mean-square error for a pre-collected (batch) set of inputs and outputs. Instead, every neuron is fully responsible for its own development and online adaptation while interacting with its extracellular environment.

In the following sections, we will go through critical components of the neural network in order to achieve the robust and efficient object recognition. Sec. V will address the statistical optimality of neurons' weight adaption in both spatial and temporal aspects. Sec. VI will explain how the sparse coding scheme is performed by layer 1 and why such a coding scheme is favorable compared to its original pixel representation. Sec. VII will describe the abstraction role of top-down connections to form the bridge representation in layer 2, along with its perspective to reducing within-object variance, and thereby, facilitating the object recognition.

V. Learning Optimality

In this section, we will discuss the learning optimality of the in-place learning algorithm described above. Given the limited resource of N neurons, the in-place learning divides the bottom-up space X into N mutually non-overlapping regions, such that

$$X = R_1 \cup R_2 \cup ... \cup R_N$$

where $R_i \cap R_j = \phi$, if $i \neq j$. Each region is represented by a single unit feature vector $\mathbf{w_b}_i$, $i = 1, 2, ..., N$, and all the vectors are not necessarily orthogonal. The in-place learning decomposes a complex global problem of approximation and representation into multiple, simpler and local ones so that lower order statistics (means) are sufficient. The proper choice of N is important for the local estimation of X. If N is too small, the estimation becomes inaccurate. On the other hand, if N is too large, it is possible to over-fit the space X.

From Eq. 14, a local estimator $\mathbf{w_b}_i$ can be expressed as:

$$\Delta \mathbf{w_b}_i = \Phi(n_i)[\mathbf{x}_i(t) y_i(t+1) - \mathbf{w_b}_i(t)] \quad (1)$$

When $\Delta \mathbf{w_b}_i = 0$, meaning that the learning weight $\mathbf{w_b}_i$ converges, we have

$$\mathbf{x}_i(t) y_i(t+1) = \mathbf{w_b}_i(t) \quad (2)$$

Consider a layer (e.g., layer 1 of the proposed network) in which the top-down connections are not available[3], Eq. 2 can be re-written as below:

$$\mathbf{x}_i(t) \frac{\mathbf{x}_i(t) \cdot \mathbf{w_b}_i(t)}{\|\mathbf{w_b}_i(t)\| \|\mathbf{x}_i(t)\|} = \mathbf{w_b}_i(t) \quad (3)$$

[2]Available at http://www.cis.hut.fi/projects/ica/imageica/

[3]The functional role of top-down connection will be specifically discussed in Sec. VII

such that

$$\mathbf{x}_i(t)\mathbf{x}_i^T(t)\mathbf{w}_{\mathbf{b}i}(t) = \|\mathbf{w}_{\mathbf{b}i}(t)\|\|\mathbf{x}_i(t)\|\mathbf{w}_{\mathbf{b}i}(t) \quad (4)$$

Averaging both sides of Eq. 4 over $\mathbf{x}_i(t)$, conditional on $\mathbf{w}_{\mathbf{b}i}$ staying unchanged (i.e., converged), we have

$$\mathbf{C}\ \mathbf{w}_{\mathbf{b}i} = \lambda\ \mathbf{w}_{\mathbf{b}i} \quad (5)$$

where $\mathbf{C}$ is the covariance matrix of inputs $\mathbf{x}_i(t)$ over time t and the scalar $\lambda = \sum_t \|\mathbf{w}_{\mathbf{b}i}(t)\|\|\mathbf{x}_i(t)\|$. Eq. 5 is the standard eigenvalue-eigenvector equation. It means that if a weight $\mathbf{w}_{\mathbf{b}i}$ converges in a local region of the bottom-up space X, the weight vector becomes one of the eigenvectors given input covariance matrix. For this reason, the in-place neural learning becomes a principal component analyzer (PCA)[4] [21], which is mathematically optimal to minimize the squared mapping/representational error, such that

$$\mathbf{w}_{\mathbf{b}i}^* = \arg\min_{\mathbf{w}_{\mathbf{b}i}} \sum_t \|(\mathbf{x}_i(t)\cdot\mathbf{w}_{\mathbf{b}i})\mathbf{w}_{\mathbf{b}i} - \mathbf{x}_i(t)\|^2. \quad (6)$$

In addition, the multi-sectional function $\mu(n)$ in Eq. (13) performs straight average $\mu(n) = 0$ for small n to reduce the error coefficient for earlier estimates. Then, $\mu(n)$ enters the rising section and changes from t_1 to t_2 linearly. In this section, neurons compete for the different partitions by increasing their learning rates for faster convergence. Finally, n enters the third section – the long adaptation section, where $\mu(n)$ increases at a rate about $1/r$, meaning that the second weight $(1+\mu(n))/n$ in Eq. (13) approaches a constant $1/r$ to trace a slowly changing distribution. This kind of plasticity scheduling is more suited for practical signals with unknown non-stationary statistics, where the distribution does follow *i.i.d* assumption in all the temporal phase.

In summary, the in-place learning scheme balances dual optimalities for both limited computational resource (spatial) and limited learning experience at any time (temporal):

1) Given the spatial resource distribution tuned by neural computations, the developed features (weights) minimize the representational error.
2) The recursive amnesic average formulation enables automatic determination of optimal step sizes in this incremental non-stationary problem.

Because the in-place learning does not require explicit search in high-dimensional parameter space nor compute the second order statistics, it also presents high learning efficiency. Given each n-dimensional input $\mathbf{x}(t)$, the system complexity for updating m neurons is $O(mn)$. It is not even a function of the number of inputs t, due to the nature of incremental learning. For a network meant to run in online development, this low update complexity is very important.

VI. Sensory Sparse Coding

In this section, we will discuss important characteristics of above dual optimalities in learning natural images – a mixture of super-gaussian sources [22]. As discussed in [23], when the input is the super-gaussian mixture, the spatial optimality of minimizing representation error in the in-place learning can function as an Independent Component Analysis (ICA) algorithm [24], and its temporal optimality performs with surprising efficiency [25]. Such independent components would help separate the non-Gaussian source signals into additive subcomponents with mutual statistical independence.

An example of developed independent components (i.e., bottom-up weights of our layer 1) are shown as image patches in Fig. 5. Many of the developed features resemble the orientation selective cells that were observed in V1 area, as discussed in [26], [27]. The mechanism of top-k winning is used to control the sparseness of the coding. In the implemented network, k is set to 91 to allow about a quarter of 431 components active for one bottom-up field in a window image. Although the developed features appear like Gabor filters, the inside independent statistics of these developed features are not available in any formula-defined Gabor functions.

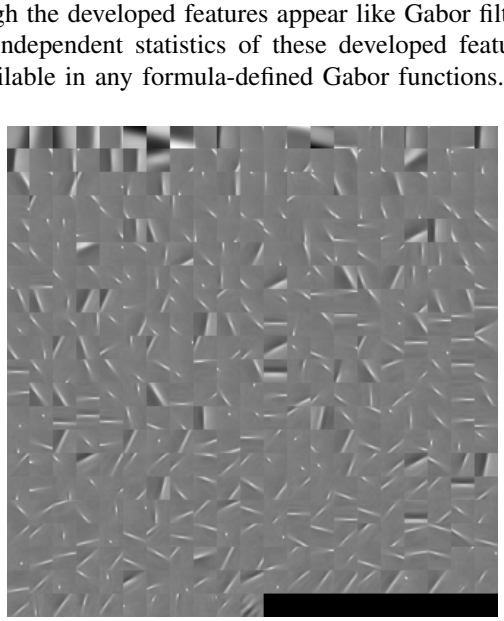


Fig. 5: Developed layer 1 features (431) in one neural column, arranged in a 2D grid. Each image patch shows a bottom-up weight (16×16 dimensions) of one neuron.

Because the object appearance in radar-attended window images could potentially vary quite a bit (the object invariance issue), and "leaked-in" background may pose some amount of noise, it is computationally inefficient to present and recognize objects using millions of pixels. The developed independent features in layer 1 (considered as independent causes) are able to code the object appearance from raw pixel space (56×56) to an over-complete, sparse[5] space (431×36). Such a sparse coding leads to lower mutual information among coded representations than pixel appearance, where the redundancy of input is transformed into the redundancy of firing pattern of cells [27]. This allows object learning to become a compositional problem, i.e., a view of a novel object is decomposed as a composite of a unique set of independent events. As shown in the experiment, Sec. VIII, the sparse coding decomposes high-correlated, redundant information in

[4]Although not shown here, Oja et al. [20] has proven that it is the first principal component that the neuron will find, and the norm of the weight vector tends to 1.

[5]By over-complete, it means that the number of code elements is greater than the dimensionality of the input space. By sparse, it means that only a few neurons will fire for a given input.

the pixel inputs and forms the representations where statistical dependency is reduced and "key" object information for later recognition is preserved.

It is worth mentioning that as natural images hold the vast inequities in variance along different directions of the input space, we should "sphere" the data by equalizing the variance in all directions [22]. This pre-processing is called whitening. The whitened sample vector $\mathbf{s}$ is computed from the original sample $\mathbf{s}'$ as $\mathbf{s} = \mathbf{W}\mathbf{s}'$, where $\mathbf{W} = \mathbf{V}\mathbf{D}$ is the whitening matrix. $\mathbf{V}$ is the matrix where each principal component $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$ is a column vector, and $\mathbf{D}$ is a diagonal matrix where the matrix element at row and column i is $\frac{1}{\sqrt{\lambda_i}}$ (λ_i is the eigenvalue of $\mathbf{v}_i$). Whitening is very beneficial to uncover the true correlations within the natural images, since it avoids the derived features to be dominated by the larger components.

VII. Top-down Abstraction

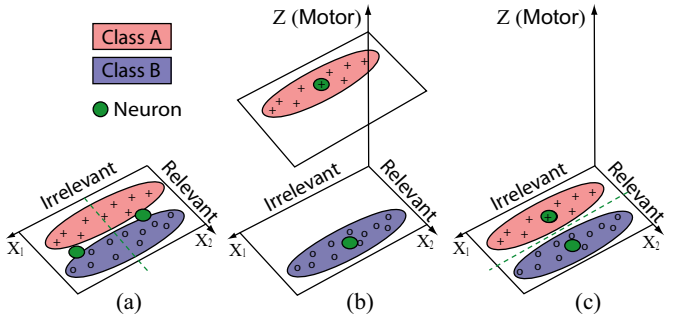


Fig. 6: Illustration of the top-down connection role. Here, bottom-up input samples contain two classes, indicated by samples "+" and "o", respectively. To see the effect clearly, we assume only two neurons are available in the local region. (a) Class mixed using only the bottom-up inputs. The two neurons spread along the direction of larger variance (irrelevant direction). The dashed line is the decision boundary based on the winner of the two neurons, which is a failure partition case. (b) Top-down connections boost the variance of relevant subspace in the neural input, and thus recruit more neurons along the relevant direction [28]. (c) Class partitioned. Especially during the testing phase, although the top-down connections become unavailable and the winner of the two neurons uses only the bottom-up input subspace X, the samples are partitioned correctly according to the classes (see dashed line).

As described in Sec. II, the coded representation in layer 1 is feed-forward to layer 2, which is associated with feed-back, top-down connections from supervised signals in layer 3. The top-down connections coordinate the neural competition and representations through two abstraction roles.

1) The top-down connections provide a new subspace where the relevant information (the information that is important to distinguish motor outputs) will have a higher variance than the irrelevant subspace. Since higher variance subspace will recruit more neurons due to the Neuronal Density Theorem [28], the representation acuity becomes higher in the relevant subspace, and more suited to the task(s) that were trained. Fig. 6 illustrates this top-down connection role. As shown in Fig. 6(c), the neurons largely spread along the relevant direction and are *invariant* to irrelevant information. The classes are partitioned correctly in the subspace (partitioned at the intersection with the dashed line) after the top-down connection, but before that, the classes in Fig. 6(a) are mixed in the bottom-up subspace X.
2) Via the top-down connections, neurons form topographic cortical areas according to the abstract classes, called topographic class grouping (TCG). That is, based on the availability of neurons, the features represented for the same motor class are grouped together to reduce the *relative within-class variance* and lead to the better recognition ability.
Consider the within-class variance w_X^2 of input space X

$$w_X^2 = \sum_{i=1}^{n} E\{\|\mathbf{x} - \bar{\mathbf{x}}_i\|^2 \mid \mathbf{x} \in c_i\}\ p_i \tag{7}$$

and its total variance

$$\sigma_X^2 = E\{\|\mathbf{x} - \bar{\mathbf{x}}\|^2\} \tag{8}$$

where $\bar{\mathbf{x}}_i$ is the mean of inputs in each class and $\bar{\mathbf{x}}$ is the mean of all the inputs. p_i denotes the probability of a sample belonging to the class c_i. Thus, the relative within-class variance of input space X can be written as

$$r_X = \frac{w_X^2}{\sigma_X^2} \tag{9}$$

From the Neuronal Density Theorem above, we know that the neurons will spread along the signal manifold to approximate the density of expanded input space $X \times Z$. Based on the top-down propagation from the motor classes, we have $w_Z^2/\sigma_Z^2 < w_X^2/\sigma_X^2$, such that the expanded input space $X \times Z$ has smaller relative within-class variance than that in X.

$$r_{X\times Z} = \frac{w_X^2 + w_Z^2}{\sigma_X^2 + \sigma_Z^2} < r_X. \tag{10}$$

Note that if top-down space Z consists of one label for each class, the within-class variance of Z is zero: $w_Z^2 = 0$ but the grand variance σ_Z^2 is still large.

Overall, above two abstraction properties work together to transform the meaningless (iconic) inputs into the internal representation with abstract class meanings.

VIII. Experimental Results

In this section, we will conduct multiple experiments based on the described system architecture and its learning advantages. An equipped vehicle is used to capture real-world images and radar sequences for training and testing purpose. Our dataset is composed from 10 different "environments" – stretches of roads at different looking places and times. Fig. 7 shows a few examples of corresponding radar and image data in different environment scenarios. In each environment, multiple sequences were extracted. Each sequence contains some similar but not identical images (e.g., different scales, illumination and view point variation, etc.). The proposed learning architecture is evaluated for a prototype of two-class problem: vehicles and other objects, which can be extendable to learn any types of objects defined by external teachers.

FrameSeqID	FrameNum	FrameTime	ID1	ID2	...	LongDist1	LongDist2	...	LateralDist1	LateralDist2	...	Confidence1	Confidence2	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
L0815_01	1081	108518	133	104	...	26.8	126.9	...	-3.2	0.3	...	15	9	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
L0815_04	915	91865	143	242	...	30.2	11.5	...	-0.2	10.2	...	15	15	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
L0815_05	466	46821	101	34	...	76.8	10.4	...	5	-0.1	...	15	15	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
L0815_08	836	83940	139	157	...	21.5	69.3	...	2.9	-5.2	...	15	15	...

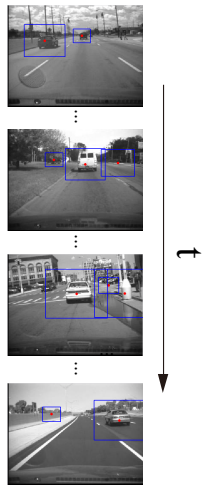


Fig. 7: Examples of radar data and corresponding images in the time sequence. It also shows some examples of different road environments in the experiment.

There are 1763 samples in the vehicle class and 812 samples in the other object class. Each large image from the camera is 240 rows and 320 columns. Each radar window is size-normalized to 56 by 56 and intensity-normalized to $\{0\ 1\}$.

A. *Sparse coding effect*

To verify the functional role of sparse coding discussed in Sec. VI, we captured 800 radar-attended window images from our driving sequences and presented them in an object-by-object order. Each object possibly appears in several window images with sequential variations. The correlation matrix of 800 window images is plotted in Fig. 8 (a), indicating the high statistical dependence among the samples, especially, across different objects. Each image is then coded for a sparse representation in layer 1. The correlation matrix of generated sparse representations is plotted in Fig. 8 (b). It shows the advantage in two aspects: (1) object samples are de-correlated, i.e., cross-object correlation is dramatically reduced; (2) object information is maintained, i.e., within-object samples keep the high correlation.

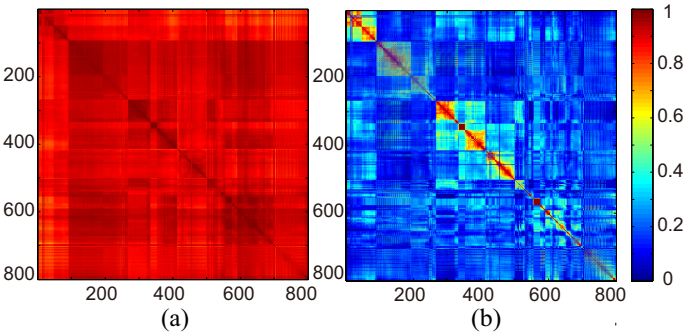


Fig. 8: Correlation matrix of (a) 800 window images in pixel space and (b) their corresponding sparse representations in layer 1 space.

B. *Top-down abstraction effect*

To evaluate the functional role of top-down abstraction discussed in Sec. VII, we first define the empirical "probability" of a neuron's firing across classes:

$$p_i = \frac{n(i)}{\sum_1^c n(i)} \qquad i \in 1, 2, ..., c \tag{11}$$

where $n(i)$ is the winning age of a neuron fired on a motor class i.

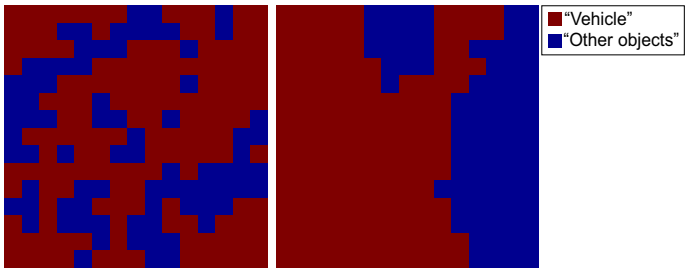


Fig. 9: 2D class map of 15×15 neurons in layer 2: (a) without top-down connections and (b) with top-down connections. Each neuron is associated with one color, presenting a class with the largest empirical "probability" p_i.

As shown in Fig. 9 and discussed in Sec. VII, neurons tend to distribute along the classes (i.e., "relevant information"). When the number of available neurons are larger than the number of classes, the neurons representing the same class are grouped together, leading to the lower within-class variance, i.e., simpler class boundaries. Through the mechanism of top-down abstraction, the network is able to develop both effective and efficient internal neural distributions.

C. *Cross validation*

In this experiment, a ten-fold cross validation is performed to evaluate the system performance. All the samples are shuffled and partitioned to 10 folds/subsets, where 9 folds are used for training and the last fold is used for testing. This process is repeated 10 times, leaving one fold for evaluation each time. The cross validation result is shown in Fig. 10 (c). The average recognition rate of the vehicle samples is 96.87%,

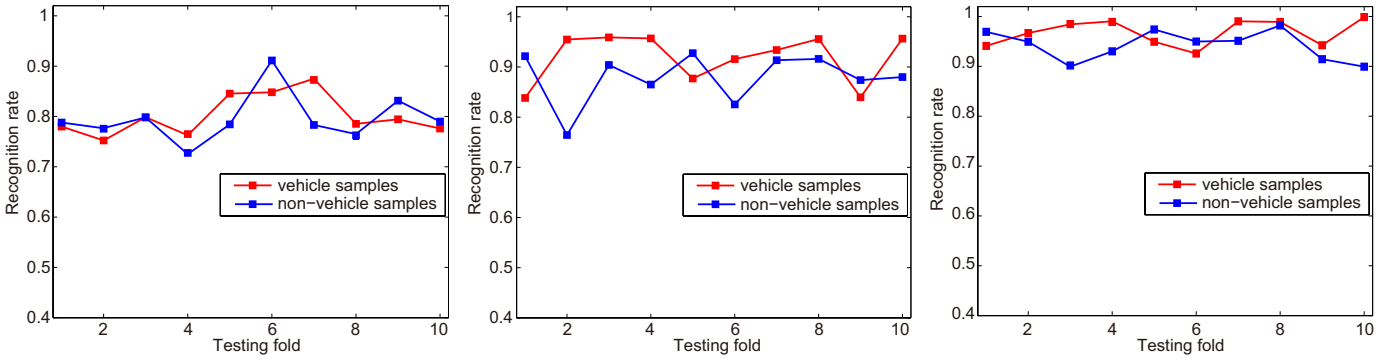


Fig. 10: 10-fold cross validation (a) without sparse coding in layer 1, (b) without top-down connection from layer 3 and (c) of the proposed work.

and 94.01% of the other object samples, where the average false positive and false negative rates are 2.94% and 6.72%, respectively. Compared to the performance without sparse coding in layer 1 (see Fig. 10 (a)), we found that, in average, the recognition rate improves 16.81% for positive samples and 14.66% for negative samples, respectively. Compared to the performance without top-down supervision from layer 3 (see Fig. 10 (b)), the recognition rate improves 5.83% for positive samples and 7.12% for negative samples, respectively.

D. Performance comparison

For an open-ended visual perceptual development, an incremental (learning one image perception per time), online (cannot turn the system off to change or adjust), real-time (fast learning and performing speed), and extendable (the number of classes can increase) architecture is expected. We compare the following incremental learning methods in MATLAB to classify the extracted window images (56×56) as vehicles and other objects: (1) K-Nearest Neighbor (K-NN), with K=1, and using a L1 distance metric for baseline performance; (2) Incremental Support Vector Machines (I-SVM) [29]; (3) Incremental Hierarchical Discriminant Regression (IHDR) [30] and (4) the proposed network described in this report. We used a linear kernel for I-SVM, as is suggested for high-dimensional problems [31]. We did try several settings for a radial basis function (RBF) kernel, but the system training becomes extremely slow and the performance improvement is not obvious.

Instead of randomly selecting samples in cross validation, we used a "true disjoint" test, where the time-organized samples are broken into ten sequential folds. Each fold is used for testing per time. In this case, the problem is more difficult, since sequences of vehicles or objects in the testing fold may have never been seen. This truly tests generalization.

The results are summarized in Tables III. K-NN performs fairly well, but is prohibitively slow. IHDR combines the advantage of K-NN with an automatically developed tree structure, which organizes and clusters the data well. It is extremely useful for the fast retrieval due to its logarithmic complexity. IHDR performs the recognition better than K-NN, and is much faster for the real-time training and testing. However, IHDR typically takes a lot of memory. It allows sample merging of prototypes, but in such case it saved every training sample, and thereby did not use memory efficiently. I-SVM performed the worst on our high-dimensional data with amount of noise, but the testing speed is fastest, since its decision making only based on a small number of support vectors. A major problem with I-SVM is lack of extendibility. By only saving support vectors to make the best two-class decision boundary, it throws out information that may be useful in distinguishing other classes added later.

Overall, the proposed network is able to perform the recognition better than all other methods using only 15×15 layer 2 neurons with a top-down supervision parameter $\alpha = 0.3$. It is also fairly fast, and efficient in terms of memory. The proposed work does not fail in any criteria, although it is not always the "best" in each category. The proposed work also has its major advantages in extendibility. New tasks, more specifically, new object classes can be added later without changing the existing learning structure of the network.

E. Incremental and online learning

The proposed neural network is incrementally updated by one piece of training data at a time, and the data is discarded as soon as it has been "seen". The incremental learning enables the recognition system to learn while performing online. This is very important for the intelligent vehicle systems, especially when information among input images is huge and highly redundant. The system only needs to handle information necessary for the decision making.

An incremental online teaching interface is developed in C++ using a PC with 2.4 GHz Intel Core2 Duo CPU and 4GB memory. The teacher could move through the collected images in the order of their sequence, provide a label to each radar window, train the agent with current labels, or test the agent's developed knowledge. Even in this non-parallelized version, the speed is in real-time use. The average speed for training the entire system (not just the algorithm) is 12.54 samples/s and the average speed for testing is 15.12 samples/s.

IX. Conclusion

In this report, we proposed and demonstrated a generic object learning system based on the automobile sensor fusion

TABLE III: Average performance & comparison of learning methods over "true disjoint" test

Learning method	Overall accuracy	"Vehicle" accuracy	"Other objects" accuracy	Training time per sample	Testing time per sample
K-NN	$78.45 \pm 12.64\%$	$74.43 \pm 13.55\%$	**90.44 ± 8.33%**	n/a	891 ± 13.4ms
ISVM	$71.54 \pm 9.82\%$	$73.23 \pm 9.36\%$	$69.32 \pm 10.24\%$	161.2 ± 18.3ms	**2.4± 0.3**ms
IHDR	$80.21 \pm 6.14\%$	$74.78 \pm 10.24\%$	$89.43 \pm 5.38\%$	**4.2± 1.9**ms	6.4 ± 2.3ms
Proposed network	**87.01± 1.43%**	**89.32± 1.64%**	$82.33 \pm 6.54\%$	112 ± 8.2ms	42.3 ± 7.2ms

framework. Early attention selection is provided by an efficient integration of multiple sensory modalities (vision and radar). Extracted attended areas are sparsely coded by the neural network using its layer 1 features that were developed from the statistics of natural images. Layer 2 of the network further learns in reaction to the coupled sparse representation and external class representations, where each cell in the network is a local class-abstracted density estimator. The proposed system architecture allows incremental and online learning, which is feasible for real-time use of any vehicle robot that can sense visual information, radar information, and a teacher's input.

For future work, we would like to test the system performance on the other critical objects (e.g., pedestrians, traffic signs, etc.) in various driving environments. Since the radar system is robust for various weather conditions, the sensor fusion framework can potentially be extended to some severe weather conditions, such as in heavy rain or snow. Currently, it is assumed that each frame is independent from the next. Relaxing this assumption may lead the way to explore temporal information of images, which should benefit the effectiveness of the learning system. We hope that these improvements will eventually lead to a vehicle-based agent that can learn to be aware of any type of objects in its environment.

REFERENCES

[1] Richard Bishop. Intelligent vehicle R&D: A review and contrast of programs worldwide and emerging trends. *Annals of Telecommunications*, 60(3-4):228–263, 2005.

[2] C. Thorpe, M. H. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the Carnegie-Mellon Navlab. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(3):362–373, 1988.

[3] B. Ulmer. Vita II - active collision avoidance in real traffic. In *Proc. IEEE Intell. Vehicles Symp.*, pages 1–6, 1994.

[4] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson. Obstacle detection and tracking for the urban challenge. *IEEE Trans. Intelligent Transportation Systems*, 10(3):475–485, 2009.

[5] H. Cheng, N. Zheng, X Zhang, J. Qin, and H. Wetering. Interactive road situation analysis for driver assistance and safety warning systems: Framework and algorithms. *IEEE Trans. Intelligent Transportation Systems*, 8(1):157–167, 2007.

[6] J. Piao and M. Mcdonald. Advanced driver assistance systems from autonomous to cooperative approach. *Transp. Rev.*, 28:659–684, 2008.

[7] T. Jochem and D. Langer. Fusing radar and vision for detecting, classifying and avoiding roadway obstacles. In *Proc. IEEE Intell. Vehicles Symp.*, pages 333–338, 1996.

[8] A. Gern, U. Franke, and P. Levi. Advanced lane recognition - fusing vision and radar. In *Proc. IEEE Intell. Vehicles Symp.*, pages 45–51, 2000.

[9] U. Hofmann, A. Rieder, and E.D. Dickmanns. Ems-vision: Application to hybrid adaptive cruise control. In *Proc. IEEE Intell. Vehicles Symp.*, pages 468–473, 2000.

[10] U. Hofmann, A. Rieder, and E.D. Dickmanns. Radar and vision data fusion for hybrid adaptive cruise control on highways. In *Ini'l Journal of Machine Vision and Applications*, volume 14, pages 42–49, 2003.

[11] R. Grover, G. Brooker, and H. F. Durrant-Whyt. A low level fusion of millimeter wave radar and night-vision imaging for enhanced characterization of a cluttered environment. In *Proc. Australian Conf. on Robotics and Automation*, pages 14–15, 2001.

[12] T. Kato, Y. Ninomiya, and I. Masaki. An obstacle detection method by fusion of radar and motion stereo. *IEEE Trans. Intelligent Transportation Systems*, 3(3):182–188, 2002.

[13] A. Sole, O. Mano, G. Stein, H. Kumon, Y. Tamatsu, and A. Shashua. Solid or not solid: Vision for radar target validation. In *Proc. IEEE Intell. Vehicles Symp.*, pages 819–824, 2004.

[14] G. Alessandretti, A. Broggi, and P. Cerri. Vehicle and guard rail detection using radar and vision data fusion. *IEEE Trans. Intelligent Transportation Systems*, 8(1):95–105, 2007.

[15] U. Kadow, G. Schneider, and A. Vukotich. Radar-vision based vehicle recognition with evolutionary optimized and boosted features. In *Proc. IEEE Intell. Vehicles Symp.*, pages 749–754, 2007.

[16] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. C. Antonello, and M. Miglietta. Obstacle detection and classification fusing radar and vision. In *Proc. IEEE Intell. Vehicles Symp.*, pages 608–613, 2008.

[17] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath. Collision sensing by stereo vision and radar sensor fusion. *IEEE Trans. Intelligent Transportation Systems*, 10(4):606–614, 2009.

[18] E. R. Kandel, J. H. Schwartz, and T. M. Jessell, editors. *Principles of Neural Science*. McGraw-Hill, New York, 4th edition, 2000.

[19] D. E. Sadava, H. C. Heller, G. H. Orians, W. K. Purves, and D. M. Hillis. *Life, the science of biology*. Freeman, New York, 8th edition, 2006.

[20] E. Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5:927–935, 1992.

[21] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

[22] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, pages 2379–2394, 1987.

[23] N. Zhang and J. Weng. Sparse representation from a winner-take-all neural network. In *Proc. Ini'l Joint Conf. on Neural Networks*, 2004.

[24] A. Hyvarinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.

[25] J. Weng and M. Luciw. Dually optimal neuronal layers: Lobe component analysis. *IEEE Trans. on Autonomous Mental Development*, 1(1):68–85, 2009.

[26] D. H. Hubel and T. N. Wiesel. Receptive feilds of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[27] B. A. Olshaushen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 13 1996.

[28] J. Weng and M. Luciw. Neuromorphic spatiotemporal processing. Technical report, MSU-CSE-08-34, 2008.

[29] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In *Advances in Neural Information Processing Systems*, volume 13, pages 409–415, Cambridge, MA, 2001.

[30] J. Weng and W. Hwang. Incremental hierarchical discriminant regression. *IEEE Trans. Neural Networks*, 18(2):397–415, 2007.

[31] B. L. Milenova, J. S. Yarmus, and M. M. Campos. Svm in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. In *Proc. 31st VLDB Conference*, 2005.

Algorithm 2 In-place learning procedure: $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) \mid L(t))$

for $1 \le i \le N_l$ **do**
  Compute pre-response of neuron i from bottom-up and top-down connections:

$$\hat{y}_i^{(l)}(t+1) = g_i\Big((1-\alpha_l)\frac{\mathbf{w_b}_i^{(l)}(t)\cdot\mathbf{x}_i^{(l)}(t)}{\|\mathbf{w_b}_i^{(l)}(t)\|\|\mathbf{x}_i^{(l)}(t)\|} + \alpha_l\frac{\mathbf{w_t}_i^{(l)}(t)\cdot\mathbf{z}_i^{(l)}(t)}{\|\mathbf{w_t}_i^{(l)}(t)\|\|\mathbf{z}_i^{(l)}(t)\|}\Big) \quad (12)$$

  where $\mathbf{x}_i^{(l)}(t)$ and $\mathbf{z}_i^{(l)}(t)$ are bottom-up and top-down input fields of neuron i. g_i is a sigmoid function with piecewise linear approximation. α_l is a layer-specific weight that controls the influence of top-down part.
end for
Simulate lateral inhibition and decide the winner:

$$j = \arg\max_{i\in I^{(l)}} \hat{y}_i^{(l)}(t+1)$$

The cells in excitatory neighborhood $E^{(l)}$ are also considered as winners and added to the winner set $\mathcal{J}$.
The responses $y_j^{(l)}$ of winning neurons are copied from their pre-responses $\hat{y}_j^{(l)}$.
Update the number of hits (cell age) n_j for the winning neurons: $n_j \leftarrow n_j + 1$. Compute $\mu(n_j)$ by the amnesic function:

$$\mu(n_j) = \begin{cases} 0 & \text{if } n_j \le t_1, \\ c(n_j - t_1)/(t_2 - t_1) & \text{if } t_1 < n_j \le t_2, \\ c + (n_j - t_2)/r & \text{if } t_2 < t, \end{cases} \quad (13)$$

  where parameters $t_1 = 20$, $t_2 = 200$, $c = 2$, $r = 2000$ in our implementation.
Determine the temporal plasticity of winning neurons, based on each age-dependent $\mu(n_j)$:

$$\Phi(n_j) = (1 + \mu(n_j))/n_j,$$

Update the synaptic weights of winning neurons using its scheduled plasticity:

$$\mathbf{w_b}_j^{(l)}(t+1) = (1-\Phi(n_j))\mathbf{w_b}_j^{(l)}(t) + \Phi(n_j)\mathbf{x}_j^{(l)}(t)y_j^{(l)}(t+1) \quad (14)$$

All other neurons keep their ages and weight unchanged.

Algorithm 1 Network processing procedure

for $t = 1, 2, ...500000$ **do**
  Grab a whitened natural image patch $\mathbf{s}(t)$.
  for $l = 1$ **do**
    Get the bottom-up fields $\mathbf{x}(t)$ from $\mathbf{s}(t)$. The top-down fields $\mathbf{z}(t)$ are set to $\mathbf{0}$.
    $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) \mid L(t))$, where $L(t)$ presents the state of current layer l, including its bottom-up and top-down weighs, neural ages, etc.
  end for
end for
for $t = 500001, 500002, ...$ **do**
  Grab the attention window image $\mathbf{s}(t)$.
  Impose the motor vector (labeled) $\mathbf{m}(t)$ to layer 3.
  for $1 \le l \le 3$ **do**
    if $l = 1$ **then**
      Get the bottom-up fields $\mathbf{x}(t)$ from $\mathbf{s}(t)$. The top-down fields $\mathbf{z}(t)$ are set to $\mathbf{0}$.
    else if $l = 2$ **then**
      Get the bottom-up fields $\mathbf{x}(t)$ from the previous layer representation (responses) and the top-down fields $\mathbf{z}(t)$ from $\mathbf{m}(t)$.
    else
      Get the bottom-up fields $\mathbf{x}(t)$ from the previous layer representation (responses). The top-down fields $\mathbf{z}(t)$ are set to $\mathbf{0}$.
    end if
    $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) \mid L(t))$.
  end for
end for

Zhengping Ji received his B.S. degree in electrical engineering from Sichuan University, and his Ph.D. degree in computer science from Michigan State University. He is now a research staff at Los Alamos National Laboratory. Before that, he was a postdoctoral fellow at Center for the Neural Basis of Cognition, Carnegie Mellon University. His research interests include computer vision, mobile robotics and autonomous mental development. He is a member of International Neural Network Society and a member of the IEEE.

Matthew Luciw received his MS degree in 2006 and PhD degree in May, 2010, both from Michigan State University (MSU) and both in computer science. He is currently working as a researcher at the Dalle Molle Institute for Artificial Intelligence (IDSIA), Manno-Lugano, Switzerland. He was previously a member of the Embodied Intelligence Laboratory at MSU. His research involves the study of biologically-inspired algorithms for autonomous development of mental capabilities, especially for visual attention and recognition. He is a member of the IEEE Computational Intelligence Society.

Juyang (John) Weng received his BS degree from Fudan University, and MS and PhD degrees from University of Illinois, Urbana-Champaign, all in Computer Science. He is now a professor at the Department of Computer Science and Engineering, Michigan State University, East Lansing, Michigan. He is also a faculty member of the Cognitive Science Program and the Neuroscience Program at Michigan State University. Since Cresceptron (ICCV 1993, IJCV 1997) — the first published work that reports recognizing and segmenting general objects from their visual appearances in natural complex backgrounds, he has further expanded his research interests in brain-mind inspired systems, especially a computationally efficient and unified framework for the autonomous development of a variety of mental capabilities by active robots and animals, including perception, cognition, behaviors, motivation, and thinking. He has published research articles on related subjects, including task muddiness, intelligence metrics, mental architectures, vision, audition, touch, attention, recognition, autonomous navigation, reaching, manipulation, and language acquisition. He is an editor-in-chief of International Journal of Humanoid Robotics and an associate editor of the new IEEE Transactions on Autonomous Mental Development, and a member of the Executive Board of the International Neural Network Society. He was a PI (with Ida Stockman) and a program chairman for the NSF/DARPA funded Workshop on Development and Learning 2000 (1st ICDL), the Chairman of the Governing Board of the International Conferences on Development and Learning (ICDL) (2005-2007, http://cogsci.ucsd.edu/ triesch/icdl/), the chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004-2005), a program chairman of 2nd ICDL, a general chairman of 7th ICDL (2008) and 8th ICDL (2009), an associate editor of IEEE Trans. on Pattern Recognition and Machine Intelligence, an associate editor of IEEE Trans. on Image Processing. He and his coworkers developed SAIL and Dav robots as research platforms for research on autonomous mental development. He is a fellow of IEEE.

Shuqing Zeng received his B.S. degree in electrical engineering from Zhejiang University, his M.S. degree in computer science from Fudan University, and his PhD degree in computer science from the Michigan State University. He joined the R&D center of General Motors Corporate in 2004 and currently holds the position of senior research scientist. He is a member of IEEE and Sigma Xi International Honor Society. He is a member of Tartan Racing team who won the first place of The Defense Advanced Research Projects Agency (DARPA) Urban Challenge. He served as a reviewer to IEEE Transactions on Pattern Analysis and Machine Intelligence and as a judge to Intelligent Ground Vehicle Competition (IGVC). His research interests include computer vision, sensor fusion, autonomous driving, and active-safety applications on vehicle.