

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

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

**Abstract**—In this paper, we propose an object learning system that incorporates sensory information from an automotive radar system and a video camera. The radar system provides 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 three-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. This paper 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 favorably with other learning algorithms. The result indicates that our learning system is the only one that is fit for incremental and online object learning in a real-time driving environment.

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

## I. INTRODUCTION

THE FIELD of intelligent vehicles has been rapidly growing over the last two decades [1]. Their examples include both fully autonomous driving vehicles [2]–[4] and advanced driver-assistance systems [5], [6], such as adaptive cruise control, lane departure warning, and the collision avoidance system. 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 under these driving conditions, recognition of the varied objects becomes one of the most critical tasks.

Vision and radar systems have complimentary properties for object validation. As one type of active sensor, a radar system has shown the good performance of object detection in driving environments. It provides fairly accurate measurements of the object distance and velocity and remains robust under various weather conditions. However, radars installed on a vehicle do not have enough lateral resolution to model object shapes, leading limitations when recognizing object types. On the contrary, video cameras called passive sensors are able to provide sufficient lateral resolution to analyze objects. The cues of shapes and, furthermore, the appearance, give more details for 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., [7]–[10]). Afterward, radar-vision approaches are more focused on the fusion at the target (e.g., vehicle and pedestrian) level. Grover *et al.* [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.* [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.* [13] treated video and radar sensors as two independent sources of target acquisition: Matched targets were validated by definition and did not require further processing, whereas unmatched radar targets were processed via motion and texture analysis for further validation. Alessandretti *et al.* [14] estimated regions of interest (ROIs) 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.* [15] and Bertozzi *et al.* [16] developed an optimized symmetry measure and new motion stereos, respectively, to detect and track other vehicles. Recently, Wu *et al.* [17] fused information from a stereo-camera and millimeter-wave radar to estimate the location, pose, and motion information of a threat vehicle within range up to 20 m.

However, the quantitative evaluation (e.g., average accuracy rate) of object recognition/detection is missing in most of the aforementioned work. In addition, the aforementioned fusion

Manuscript received November 4, 2009; revised July 28, 2010 and October 13, 2010; accepted October 26, 2010. Date of publication December 17, 2010; date of current version June 6, 2011. The Associate Editor for this paper was M. Da Lio.

Z. Ji was with Michigan State University, East Lansing, MI 48824 USA. He is now with the Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (e-mail: jizhengp@cse.msu.edu).

M. Luciw was with Michigan State University, East Lansing, MI 48824 USA. He is now with Dalle Molle Institute for Artificial Intelligence, 6928 Manno-Lugano, Switzerland (e-mail: luciwmat@gmail.com).

J. Weng is with the Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824 USA (e-mail: weng@cse.msu.edu).

S. Zeng is with the Research and Development Center, General Motors Inc., Warren, MI 48090 USA (e-mail: shuqing.zeng@gm.com).

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/TITS.2010.2094188

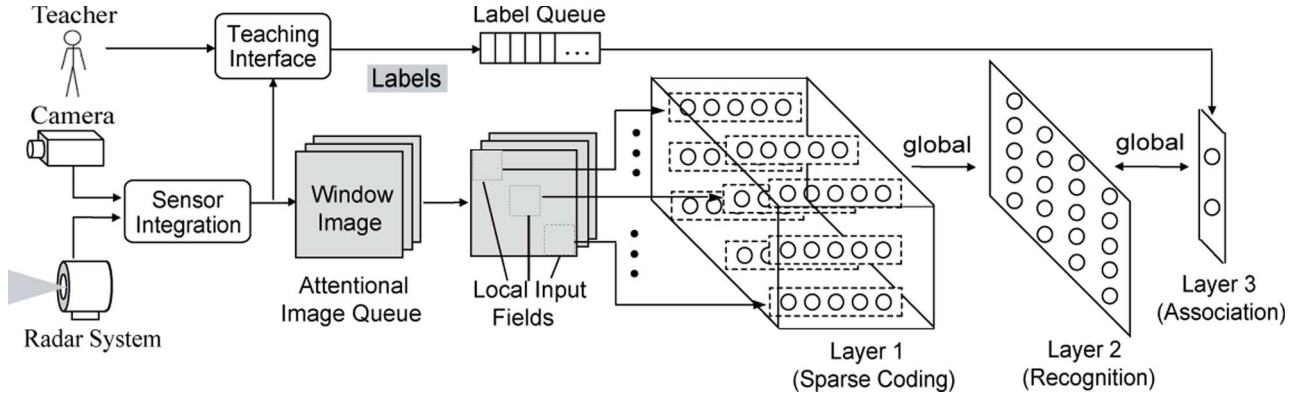


Fig. 1. Outline of the system architecture. The camera and the radar system work together to generate a set of attended window images, containing environment objects. A teacher communicates with the system through an interface to train the class labels of objects. A three-layer network provides the processing and learning of the extracted window images. The number of neurons in each layer is specified at a 3-D 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, which are associated with layer 3 with the teacher’s output tokens.

works mainly detected key objects (i.e., vehicles or pedestrians) using object-specific features, such as blobs, edges, symmetries, and motion. The object-specific (or 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 the advantage of radar-vision integration to achieve an efficient attention selection on candidate 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 three-way computations (i.e., bottom-up, top-down, and lateral) to code object samples in an overcomplete 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 here requires a combination of the following challenges, where no existing work as we know can meet them all: 1) a general radar-vision fusion framework with non-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; and 5) integration of supervised learning (via top-down propagation) and unsupervised learning (via bottom-up propagation) in any order suited for development.

All the preceding properties, coupled with a nurturing and challenging environment, as experienced through sensors and effectors, allow the automatic perceptual awareness to emerge in intelligent vehicles.

## II. ARCHITECTURES

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

TABLE I  
SENSOR SPECIFICATIONS OF THE RADAR SYSTEM

Key parameters	Specification
Refreshing rate	10 Hz
No. of targets	max. of 20 targets
Max. range	150m ± max(5%, 1.0m)
Field of view	180° (< 30m); 15° (> 30m)
Range rate	±56m/s ± 0.75m/s

TABLE II  
SENSOR SPECIFICATIONS OF THE VIDEO CAMERA

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

common language<sup>1</sup> between the system and the teachers or users. In this paper, 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 such as counting and prediction.

## III. COARSE ATTENTION SELECTION

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

As shown in Fig. 2 (right), a group of target points in 3-D world coordinates can be detected from the radar system, with

<sup>1</sup>The language can be as simple as a predefined set of tokens or as complex as human spoken languages.

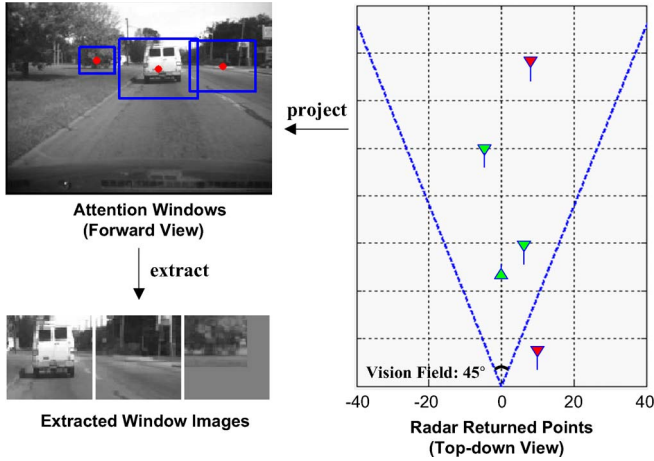


Fig. 2. Projection of valid radar points (green) onto the image plane, where window images are extracted for further recognition (best viewed in color).

a detection range up to 150 m. Each radar point is presented by a triangle that is associated with a bar, whose length and direction indicate the relative speed of an object. As a rudimentary but necessary attention selection mechanism, we discarded radar returns more than 80 m in distance ahead or more than 8 m to the right or left outside the vehicle path (e.g., the red triangle points in Fig. 2 (right) are omitted).

Based on the estimation of the maximum height (3.0 m) and maximum width (3.8 m) of environment targets, a rectangular target window (with a fixed size of  $3.0 \times 3.8 \text{ m}^2$ ) is generated to be centered at each valid radar point. All the target windows at each time  $t$  are then projected onto 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 larger, the attention window in the image is smaller, and *vice versa*.

For each attention window, the pixels are extracted as a single image, and most of the nontarget 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, whereas the rest of the image is ignored. This is called coarse attention selection: finding candidate areas purely based on the 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 good segmentation of the object itself but instead 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.

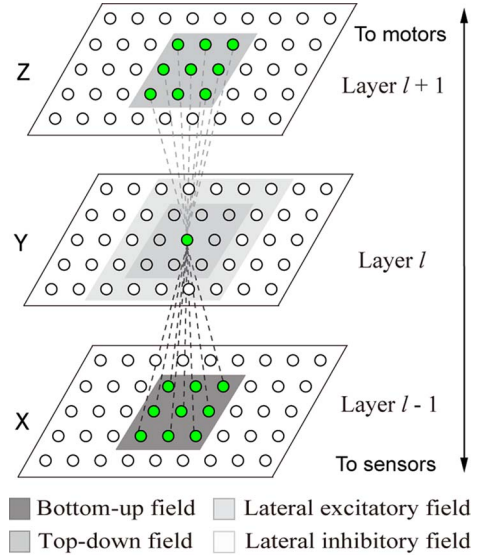


Fig. 3. General structure of the network connection. Neurons are placed (given a position) on different layers in an end-to-end hierarchy—from sensors to motors. Only the connections to a centered cell are shown, but all the other neurons in the feature layer have the same default connections.

#### IV. OBJECT LEARNING NETWORK

The attended window images are coded and learned through the proposed neural network (see Fig. 1) via three layers, up until 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 the following 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.

The object learning network is incrementally updated at discrete times  $t = 0, 1, 2, \dots$ , 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

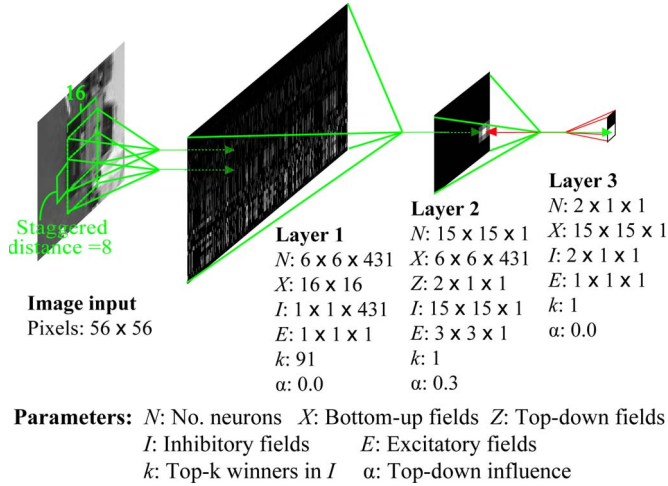


Fig. 4. Example of layer representations (i.e., responses) in the proposed neural network, including a specific set of resource parameters implemented (best viewed in color). Green and red directed lines show the bottom-up and top-down connections to the firing neurons, respectively. It is noted that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

regions [18]. Given  $t = 1, 2, \dots, \tau$  ( $\tau = 500\,000$  in this application), the network receives  $56 \times 56$ -pixel (which is the same as the attention window dimension) natural image patches, which were randomly selected from the 13 natural images.<sup>2</sup> Neurons are learned through the in-place learning algorithm described in Algorithm 2, however, without supervision on motors. After 500 000 updates of layer-1 neurons, their bottom-up features tends 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 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 multicell, goal to the learning, such as the minimization of mean-square error for a precollected (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.

**Algorithm 1:** Network processing procedure

- 1: **for**  $t = 1, 2, \dots, \tau$  **do**
- 2:     Grab a whitened natural image patch  $s(t)$ .
- 3:     **for**  $l = 1$  **do**
- 4:         Get the bottom-up fields  $\mathbf{x}(t)$  from  $s(t)$ . The top-down fields  $\mathbf{z}(t)$  are set to  $\mathbf{0}$ .

- 5:          $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)|L(t))$ , where  $L(t)$  presents the state of current layer  $l$ , including its bottom-up and top-down weighs, neural ages, etc.
- 6:     **end for**
- 7:     **end for**
- 8:     **for**  $t = \tau + 1, \tau + 2, \dots$  **do**
- 9:         Grab the attention window image  $s(t)$ .
- 10:         Impose the motor vector (labeled)  $\mathbf{m}(t)$  to layer 3.
- 11:         **for**  $1 \leq l \leq 3$  **do**
- 12:             **if**  $l = 1$  **then**
- 13:                 Get the bottom-up fields  $\mathbf{x}(t)$  from  $s(t)$ . The top-down fields  $\mathbf{z}(t)$  are set to  $\mathbf{0}$ .
- 14:             **else if**  $l = 2$  **then**
- 15:                 Get the bottom-up fields  $\mathbf{x}(t)$  from the previous layer representation (responses) and the top-down fields  $\mathbf{z}(t)$  from  $\mathbf{m}(t)$ .
- 16:             **else**
- 17:                 Get the bottom-up fields  $\mathbf{x}(t)$  from the previous layer representation (responses). The top-down fields  $\mathbf{z}(t)$  are set to  $\mathbf{0}$ .
- 18:             **end if**
- 19:              $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t)|L(t))$ .
- 20:         **end for**
- 21:     **end for**

In the succeeding sections, we will go through critical components of the neural network to achieve robust and efficient object recognition. Section V will address the statistical optimality of neurons’ weight adaptation in both spatial and temporal aspects. Section VI will explain how the sparse coding scheme is performed by layer 1 and why such a coding scheme is favorable compared with its original pixel representation. Section 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 object recognition.

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

- 1: **for**  $1 \leq i \leq N_l$  **do**
- 2:     Compute the preresponse of neuron  $i$  from bottom-up and top-down connections

$$\hat{y}_i^{(l)}(t + 1) = g_i \left( (1 - \alpha_l) \frac{\mathbf{w}_{\mathbf{b}_i}^{(l)}(t) \cdot \mathbf{x}_i^{(l)}(t)}{\|\mathbf{w}_{\mathbf{b}_i}^{(l)}(t)\| \|\mathbf{x}_i^{(l)}(t)\|} + \alpha_l \frac{\mathbf{w}_{\mathbf{t}_i}^{(l)}(t) \cdot \mathbf{z}_i^{(l)}(t)}{\|\mathbf{w}_{\mathbf{t}_i}^{(l)}(t)\| \|\mathbf{z}_i^{(l)}(t)\|} \right) \quad (1)$$

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

<sup>2</sup> Available at <http://www.cis.hut.fi/projects/ica/imageica/>.

piecewise linear approximation.  $\alpha_l$  is a layer-specific weight that controls the influence of the top-down part.

3: **end for**

4: Simulate lateral inhibition, and decide the winner, i.e.,

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

5: The cells in excitatory neighborhood  $E^{(l)}$  are also considered as winners and added to the winner set  $\mathcal{J}$ .

6: The responses  $y_j^{(l)}$  of winning neurons are copied from their preresponses  $\hat{y}_j^{(l)}$ .

7: 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

$$\mu(n_j) = \begin{cases} 0, & \text{if } n_j \leq t_1 \\ c(n_j - t_1)/(t_2 - t_1), & \text{if } t_1 < n_j \leq t_2 \\ c + (n_j - t_2)/r, & \text{if } t_2 < t \end{cases} \quad (2)$$

where parameters  $t_1 = 20$ ,  $t_2 = 200$ ,  $c = 2$ , and  $r = 2000$  in our implementation.

8: Determine the temporal plasticity of winning neurons, based on each age-dependent  $\mu(n_j)$ , i.e.,

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

9: Update the synaptic weights of winning neurons using its scheduled plasticity, i.e.,

$$\mathbf{w}_{\mathbf{b}_j}^{(l)}(t+1) = (1 - \Phi(n_j)) \mathbf{w}_{\mathbf{b}_j}^{(l)}(t) + \Phi(n_j) \mathbf{x}_j^{(l)}(t) y_j^{(l)}(t+1). \quad (3)$$

10: All other neurons keep their ages and weight unchanged.

## V. LEARNING OPTIMALITY

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

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

where  $R_i \cap R_j = \phi$ , if  $i \neq j$ . Each region is represented by a single unit feature vector  $\mathbf{w}_{\mathbf{b}_i}$ ,  $i = 1, 2, \dots, N$ , 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 overfit the space  $X$ .

From (3), a local estimator  $\mathbf{w}_{\mathbf{b}_i}$  can be expressed as

$$\Delta \mathbf{w}_{\mathbf{b}_i} = \Phi(n_i) [\mathbf{x}_i(t) y_i(t+1) - \mathbf{w}_{\mathbf{b}_i}(t)]. \quad (4)$$

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

$$\mathbf{x}_i(t) y_i(t+1) = \mathbf{w}_{\mathbf{b}_i}(t). \quad (5)$$

Considering a layer (e.g., layer 1 of the proposed network) in which the top-down connections are not available,<sup>3</sup> (5) can be rewritten as

$$\mathbf{x}_i(t) \frac{\mathbf{x}_i(t) \cdot \mathbf{w}_{\mathbf{b}_i}(t)}{\|\mathbf{w}_{\mathbf{b}_i}(t)\| \|\mathbf{x}_i(t)\|} = \mathbf{w}_{\mathbf{b}_i}(t) \quad (6)$$

such that

$$\mathbf{x}_i(t) \mathbf{x}_i^T(t) \mathbf{w}_{\mathbf{b}_i}(t) = \|\mathbf{w}_{\mathbf{b}_i}(t)\| \|\mathbf{x}_i(t)\| \mathbf{w}_{\mathbf{b}_i}(t). \quad (7)$$

Averaging both sides of (7) over  $\mathbf{x}_i(t)$ , which is conditional on  $\mathbf{w}_{\mathbf{b}_i}$  staying unchanged (i.e., converged), we have

$$\mathbf{C} \mathbf{w}_{\mathbf{b}_i} = \lambda \mathbf{w}_{\mathbf{b}_i} \quad (8)$$

where  $\mathbf{C}$  is the covariance matrix of inputs  $\mathbf{x}_i(t)$  over time  $t$ , and the scalar  $\lambda = \sum_t \|\mathbf{w}_{\mathbf{b}_i}(t)\| \|\mathbf{x}_i(t)\|$ . Equation (8) is the standard eigenvalue–eigenvector equation, which 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 an input covariance matrix. For this reason, the in-place neural learning becomes a principal component analyzer<sup>4</sup> [21], which is mathematically optimal to minimize the squared mapping/representational error, such that

$$\mathbf{w}_{\mathbf{b}_i}^* = \arg \min_{\mathbf{w}_{\mathbf{b}_i}} \sum_t \|(\mathbf{x}_i(t) \cdot \mathbf{w}_{\mathbf{b}_i}) \mathbf{w}_{\mathbf{b}_i} - \mathbf{x}_i(t)\|^2. \quad (9)$$

In addition, the multisectional function  $\mu(n)$  in (2) 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 linearly changes from  $t_1$  to  $t_2$ . In this section, neurons compete for different partitions by increasing their learning rates for faster convergence. Finally,  $n$  enters the third section, i.e., the long adaptation section, where  $\mu(n)$  increases at a rate of about  $1/r$ , meaning that the second weight  $(1 + \mu(n))/n$  in (2) approaches a constant  $1/r$  to trace a slowly changing distribution. This kind of plasticity scheduling is more suited for practical signals with unknown nonstationary statistics, where the distribution does follow the independent identically distributed assumption in the whole 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.

<sup>3</sup>The functional role of top-down connection will be specifically discussed in Section VII.

<sup>4</sup>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.

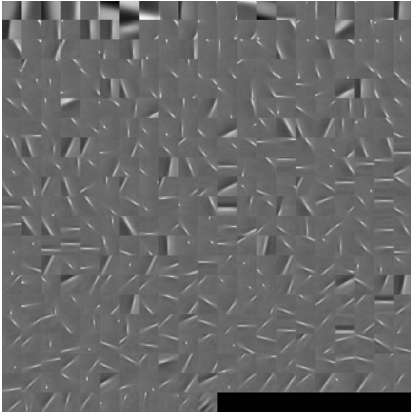


Fig. 5. Developed layer-1 features (431) in one neural column arranged in a 2-D grid. Each image patch shows a bottom-up weight ( $16 \times 16$  dimensions) of one neuron.

- 2) The recursive amnesic average formulation enables automatic determination of optimal step sizes in this incremental nonstationary 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 the network meant to run in online development, this low update complexity is very important.

## VI. SENSORY SPARSE CODING

In this section, we will discuss the important characteristics of the aforementioned dual optimalities in learning natural images, i.e., 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 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) is shown as image patches in Fig. 5. Many of the developed features resemble the orientation selective cells that were observed in the V1 area, as discussed in [27]. The mechanism of top- $k$  winning is used to control the sparseness of the coding. In the implemented network,  $k$  is set as 91 to allow about a quarter of the 431 components to be 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.

Because the object appearance in radar-attended window images could potentially vary quite a bit (the object invariance issue) and a “leaked-in” background may pose an 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) code

the object appearance from raw pixel space ( $56 \times 56$ ) to an overcomplete sparse<sup>5</sup> space ( $431 \times 36$ ). 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 in Section VIII, the sparse coding decomposes highly correlated redundant information in the pixel inputs and forms the sparse representations, where statistical dependence 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 preprocessing 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  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$  is a column vector, and  $\mathbf{D}$  is a diagonal matrix where the matrix element at row and column  $i$  is  $(1/\sqrt{\lambda_i})$ . ( $\lambda_i$  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

As described in Section II, the coded representation in layer 1 is fed forward to layer 2, which is associated with feedback 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 for the classification task(s). Fig. 6 shows 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 correctly partitioned 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. 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 better recognition.

<sup>5</sup>By overcomplete, 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.

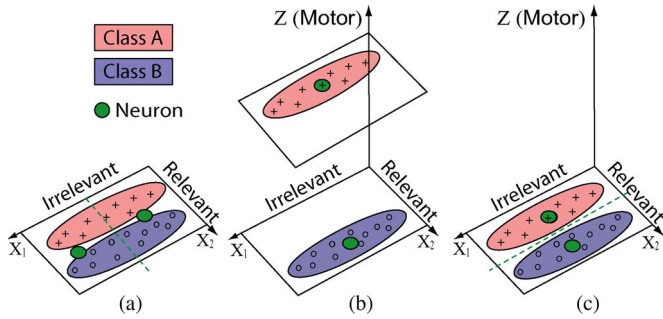


Fig. 6. Top-down connection role (best viewed in color). Here, bottom-up input samples contain two classes, which are indicated by samples “+” and “o,” respectively. To clearly see the effect, we assume that only two neurons are available in the local region. (a) Class mixed using only the bottom-up 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. (c) Class partitioned. In particular, 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 correctly partitioned according to the classes (see dashed line).

Consider the within-class variance  $w_X^2$  of input space  $X$

$$w_X^2 = \sum_{i=1}^n E \left\{ \|\mathbf{x} - \bar{\mathbf{x}}_i\|^2 | \mathbf{x} \in c_i \right\} p_i \quad (10)$$

and its total variance

$$\sigma_X^2 = E \left\{ \|\mathbf{x} - \bar{\mathbf{x}}\|^2 \right\} \quad (11)$$

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

$$r_X = \frac{w_X^2}{\sigma_X^2}. \quad (12)$$

From the aforementioned Neuronal Density Theorem, we know that the neurons will spread along the signal manifold to approximate the density of expanded input space  $X \times Z$ . Due to 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  $X$ , i.e.,

$$r_{X \times Z} = \frac{w_X^2 + w_Z^2}{\sigma_X^2 + \sigma_Z^2} < r_X. \quad (13)$$

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  $\sigma_Z^2$  is still large.

Overall, the preceding 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 purposes. Our data set is composed of ten different “environments,” i.e., 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). The proposed learning architecture is evaluated for a prototype of a two-class problem: vehicles and other objects, which can be extendable to learn any types of objects defined by external teachers. 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 Section 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, particularly 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 decorrelated, i.e., cross-object correlation is dramatically reduced. 2) Object information per class is maintained, i.e., within-object samples keep the high correlation.

### B. Top-Down Abstraction Effect

To evaluate the functional role of top-down abstraction discussed in Section VII, we first define the empirical “probability” of a neuron’s firing across classes

$$p_i = \frac{n(i)}{\sum_1^c n(i)}, \quad i \in 1, 2, \dots, c \quad (14)$$

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

As shown in Fig. 9 and discussed in Section 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.

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	...

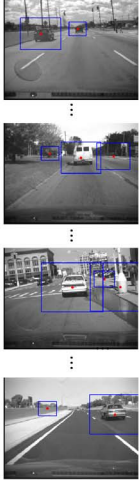


Fig. 7. Examples of radar data and corresponding images in the time sequence. It also shows some examples of different road environments in the experiment.

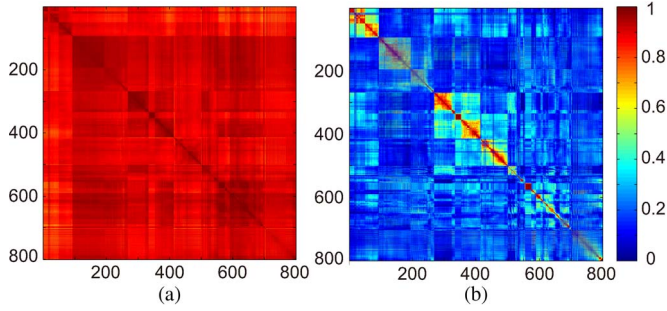


Fig. 8. Correlation matrix of (a) 800 window images in pixel space and (b) their corresponding sparse representations in layer-1 space (best viewed in color).

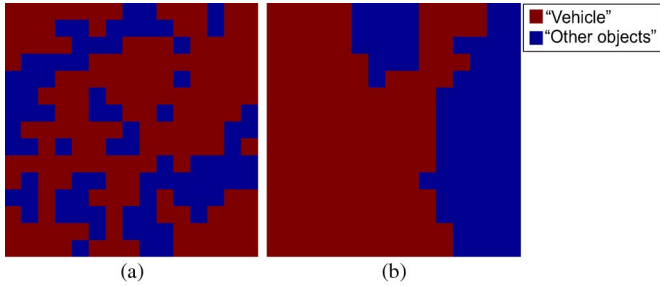


Fig. 9. Two-dimensional class map of  $15 \times 15$  neurons in layer 2 (best viewed in color) (a) without top-down connections and (b) with top-down connections. Each neuron is associated with one color, presenting a class with the largest empirical “probability”  $p_i$ .

C. Cross Validation

In this experiment, a tenfold cross validation is performed to evaluate the system performance. All the samples are shuffled and partitioned to ten folds/subsets, where nine folds are used for training, and the last fold is used for testing. This process is repeated ten 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% 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 with the performance without sparse coding in layer 1 [see Fig. 10(a)], we found that, on average, the recognition rate improves 16.81% for positive samples and 14.66% for

negative samples, respectively. Compared with 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

In the aspect of 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 ( $56 \times 56$ ) as vehicles and other objects: 1)  $K$ -nearest neighbor ( $K$ -NN), with  $K = 1$  and using an L1 distance metric for baseline performance; 2) incremental support vector machines (I-SVMs) [29]; 3) incremental hierarchical discriminant regression (IHDR) [30]; and 4) the proposed network described in this paper. We used a linear kernel for I-SVM, as suggested for high-dimensional problems [31]. We tried several settings for a radial basis function kernel; however, 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 Table III. K-NN performs fairly well but is prohibitively slow. IHDR utilizes 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 real-time training and testing. However, IHDR typically takes much memory. It allows sample merging of prototypes, but in such a case, it saved every training sample and thereby did not efficiently use memory. I-SVM performed the worst on



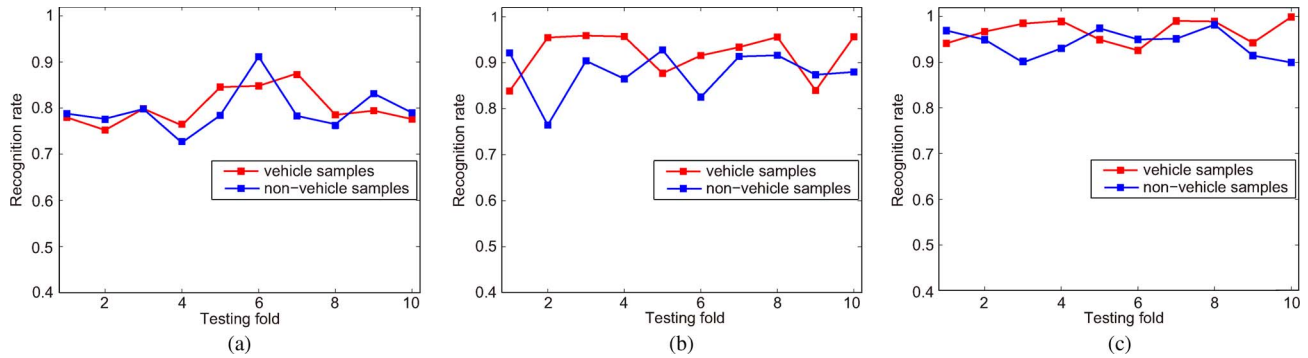


Fig. 10. Ten-fold cross validation (best viewed in color) (a) without sparse coding in layer 1, (b) without top-down connection from layer 3, and (c) of the proposed work.

TABLE III  
AVERAGE PERFORMANCE AND 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 ± 12.64%	74.43 ± 13.55%	<b>90.44 ± 8.33%</b>	n/a	891 ± 13.4ms
ISVM	71.54 ± 9.82%	73.23 ± 9.36%	69.32 ± 10.24%	161.2 ± 18.3ms	<b>2.4 ± 0.3ms</b>
IHDR	80.21 ± 6.14%	74.78 ± 10.24%	89.43 ± 5.38%	<b>4.2 ± 1.9ms</b>	6.4 ± 2.3ms
Proposed network	<b>87.01 ± 1.43%</b>	<b>89.32 ± 1.64%</b>	82.33 ± 6.54%	112 ± 8.2ms	42.3 ± 7.2ms

our high-dimensional data with amount of noise, but the testing speed is fastest since its decision making is 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  $15 \times 15$  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 are 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, particularly when information among input images is huge and highly redundant. The system only needs to handle information that is necessary for decision making.

An incremental online teaching interface is developed in C++ using a personal computer with 2.4-GHz Intel Core2 Duo central processing unit and 4-GB memory. The teacher could move through the collected images on 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 nonparallelized version, the speed is in real-time use. The average speed for training the entire system (not just the algorithm) is 12.54 sample/s, and the average speed for testing is 15.12 sample/s.

#### IX. CONCLUSION

In this paper, we have proposed and demonstrated a generic object learning system based on the automobile sensor fusion framework. Early attention selection is provided by an efficient integration of multiple sensory modalities (vision and radar). Extracted attended areas have been 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 and traffic signs) in various driving environments. Since the radar system is robust for various weather conditions, the sensor fusion framework can potentially extend to some severe weather conditions, such as rain or snow. Currently, it is assumed that each frame is independent of the next. Relaxing this assumption may lead to the exploration of 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 object in its environment.

#### REFERENCES

- [1] R. Bishop, “Intelligent vehicle R&D: A review and contrast of programs worldwide and emerging trends,” *Ann. Telecommun.*, vol. 60, no. 3/4, pp. 228–263, Mar./Apr. 2005.
- [2] C. Thorpe, M. H. Hebert, T. Kanade, and S. Shafer, “Vision and navigation for the Carnegie-Mellon Navlab,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 362–373, May 1988.
- [3] B. Ulmer, “Vita II—Active collision avoidance in real traffic,” in *Proc. IEEE Intell. Vehicles Symp.*, 1994, pp. 1–6.

[4] M. S. Darms, P. E. Rybski, C. Baker, and C. Urmson, "Obstacle detection and tracking for the urban challenge," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 475–485, Sep. 2009.

[5] H. Cheng, N. Zheng, X. Zhang, J. Qin, and H. van de Wetering, "Interactive road situation analysis for driver assistance and safety warning systems: Framework and algorithms," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 157–167, Mar. 2007.

[6] J. Piao and M. McDonald, "Advanced driver assistance systems from autonomous to cooperative approach," *Transp. Rev.*, vol. 28, no. 5, pp. 659–684, Sep. 2008.

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

[8] A. Gern, U. Franke, and P. Levi, "Advanced lane recognition—Fusing vision and radar," in *Proc. IEEE Intell. Vehicles Symp.*, 2000, pp. 45–51.

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

[10] U. Hofmann, A. Rieder, and E. D. Dickmanns, "Radar and vision data fusion for hybrid adaptive cruise control on highways," *Mach. Vis. Appl.*, vol. 14, no. 1, pp. 42–49, Apr. 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. Robot. Autom.*, 2001, pp. 14–15.

[12] T. Kato, Y. Ninomiya, and I. Masaki, "An obstacle detection method by fusion of radar and motion stereo," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 3, pp. 182–188, Sep. 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.*, 2004, pp. 819–824.

[14] G. Alessandretti, A. Broggi, and P. Cerri, "Vehicle and guard rail detection using radar and vision data fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 8, no. 1, pp. 95–105, Mar. 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.*, 2007, pp. 749–754.

[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.*, 2008, pp. 608–613.

[17] S. Wu, S. Decker, P. Chang, T. Camus, and J. Eledath, "Collision sensing by stereo vision and radar sensor fusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 606–614, Dec. 2009.

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

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

[20] E. Oja, "Principal components, minor components, and linear neural networks," *Neural Netw.*, vol. 5, no. 6, pp. 927–935, Nov./Dec. 1992.

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

[22] D. J. Field, "Relations between the statistics of natural images and the response properties of cortical cells," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 4, no. 12, pp. 2379–2394, Dec. 1987.

[23] N. Zhang and J. Weng, "Sparse representation from a winner-take-all neural network," in *Proc. Int. Joint Conf. Neural Netw.*, 2004, pp. 2209–2214.

[24] A. Hyvarinen, "Survey on independent component analysis," *Neural Comput. Surv.*, vol. 2, pp. 94–128, 1999.

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

[26] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *J. Physiol.*, vol. 148, pp. 574–591, 1959.

[27] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 13, 1996.

[28] J. Weng and M. Luciw, "Neuromorphic spatiotemporal processing," Michigan State Univ., East Lansing, MI, Tech. Rep. MSU-CSE-08-34, 2008.

[29] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Cambridge, MA, 2001, vol. 13, pp. 409–415.

[30] J. Weng and W. Hwang, "Incremental hierarchical discriminant regression," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 397–415, Mar. 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 Conf.*, 2005, pp. 1152–1163.



**Zhengping Ji** (M'09) received the B.S. degree in electrical engineering from Sichuan University, Chengdu, China, and the Ph.D. degree in computer science from Michigan State University, East Lansing.

After that, he was a Postdoctoral Fellow with the Center for the Neural Basis of Cognition, Carnegie Mellon University, Pittsburgh, PA. He is currently a Researcher with the Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM. His research interests include computer vision, mobile robotics, and autonomous mental development.

Dr. Ji is a member of the International Neural Network Society.



**Matthew Luciw** (M'10) received the M.S. and Ph.D. degrees in computer science from Michigan State University (MSU), East Lansing, in 2006 and 2010, respectively.

He was previously a member of the Embodied Intelligence Laboratory, MSU. He is currently a Researcher with Dalle Molle Institute for Artificial Intelligence IDSIA, Manno-Lugano, Switzerland. His research involves the study of biologically inspired algorithms for autonomous development of mental capabilities, particularly for visual attention

and recognition.

Dr. Luciw is a member of the IEEE Computational Intelligence Society.



**Juyang (John) Weng** (S'85–M'88–SM'05–F'09) received the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana.

He is a Faculty Member with the Cognitive Science Program and the Neuroscience Program with Michigan State University (MSU), East Lansing. Since the Cresceptron paradigm, he has further expanded his research interests in brain-mind inspired systems, including perception, cognition, behaviors, and motivation. He is currently a Professor with the Department of Computer Science and Engineering,

MSU. He is Editor-in-Chief of the *International Journal of Humanoid Robotics*.

Dr. Weng is a member of the Executive Board of the International Neural Network Society. He was a Principal Investigator for the National Science Foundation/Defense Advanced Research Projects Agency-funded Workshop on Development and Learning in 2000 [first International Conferences on Development and Learning (IDCL)], the Chairman of the Governing Board of the IDCL (2005–2007), the Chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004–2005), an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS MACHINE INTELLIGENCE, and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is an Associate Editor for the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT.



**Shuqing Zeng** (M'04) received the B.S. degree in electrical engineering from Zhejiang University, Hangzhou, China, the M.S. degree in computer science from Fudan University, Shanghai, China, and the Ph.D. degree in computer science from Michigan State University, East Lansing.

In 2004, he joined the Research and Development Center, General Motors Inc., Warren, MI, where he currently holds the position of Senior Research Scientist. His research interests include computer vision, sensor fusion, autonomous driving, and active-safety applications on vehicles.

Dr. Zeng is a member of Sigma Xi International Honor Society. He served as a Reviewer for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and as a Judge of the Intelligent Ground Vehicle Competition. He is a member of Tartan Racing team, which won first place at The Defense Advanced Research Projects Agency Urban Challenge.