LA-UR- 10-07067

Approved for public release; distribution is unlimited.

Title:	Incremental Online Object learning in a Vehicular Radar-Vision Fusion Framework
Author(s):	Zhengping Ji Matthew Luciw Juyang Weng Shuqing Zeng
Intended for:	IEEE Transactions on Intelligent Transportation Systems

Los Alamos
NATIONAL LABORATORY
EST. 1943

Form 836 (7/06)

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 paper, 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, where every neuron is responsible for the learning of its own signal processing characteristics within its 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 the overall recognition accuracy 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 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 to fit all the challenging criteria for the development of an incremental and online object learning system.**

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

I. Introduction

Ever since the pioneering projects of driverless car in 1980's (e.g., the Euro EUREKA Prometheus Project and the USA Autonomous Land Vehicle [1]), many systems for autonomous driving have been created (e.g., [2] [3] [4]) and many more are under development. Yet, the constraints of autonomous driving did not require local perceptual awareness besides a classification of traversable and non-traversable areas. Skilled driving assistance systems, furthermore, require a rich understanding of the complex road environment, which contains many signals and cues that visually convey information, such as traffic lights, road signs, and many different types of objects, including other vehicles, pedestrians, and trash cans, to name a few. The skilled driving poses high requirements for the awareness of driving conditions, which possibly have a significant number of different objects. In order to take correct and intelligent actions in the driving conditions, recognition of the varied objects is one of the most critical tasks.

Vision and radar systems have complimentary properties in driving assistance systems. As one type of active sensors, a radar system has shown the good performance of target detection in driving environments. It provides fairly accurate measurements of 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 to a limitation of 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, 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 driving assistance systems. Early fusion framework utilized radar positions in a vision-based lane recognition system to achieve the better lane estimations (e.g. Jochem and Langer 1996 [5] and Gern et al. 2000 [6]). A more common approach was later described by Grover et al. 2001 [7] to perform the radar-vision fusion at target (e.g., key object) level, where a single radar map and a single night-vision image (using blob features) were fused in polar coordinates to determine vehicle localizations. Hofmann et al. 2003 [8] conducted a radar-based obstacle detection and an optical lane recognition interactively to identify the relevant vehicle for the controller of the host vehicle. Miyahara et al. 2006 [9] presented a range-window algorithm to generate regions of interest (ROI) from radar returns, where edge-based pattern matching was used for tracking of the object appeared in the attention window. Using the similar mechanism of ROI provided by radars, Kadow et al. 2007 [10] and Bertozzi et al. 2008 [11] developed an optimized symmetry measurement and motion stereos respectively to detect and track other vehicles. 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 (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

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

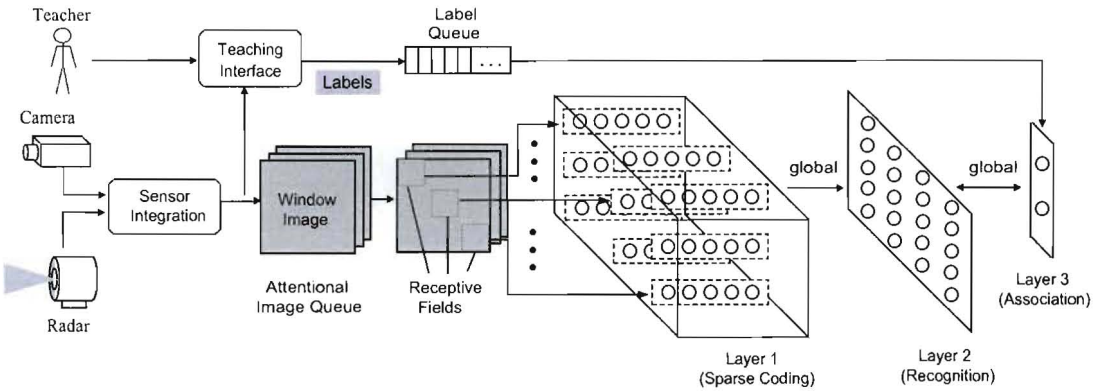

Fig. 1: System architecture of the vehicle-based agent. The camera and the radar work together to generate a set of attended window images, containing nearby objects. A teacher communicates with the system through an interface to train the class labels of objects. A 3-layer network provides the processing and learning of the extracted window images. The number of neurons in each layer is specified at a 3D grid (r rows $\times c$ columns $\times d$ depths). 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.

neural network integrates 3-way computations (i.e., bottom-up, top-down and lateral) to code object samples to an over-complete space and learn the distribution of coded "key" object patterns in favorable recognition performance. Its in-place learning mechanism provides the incremental learning optimality and comparatively low operational complexity even for large networks.

A successful implementation here requires a combination of the following challenges, where no existing work as we know can meet all: (1) General radar-vision fusion framework not constrained for a task-specific learning. (2) Visual sensory sparse coding via developed features with statistical independence. (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 in driving assistance systems.

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, for applications including intelligent driving assistance and autonomous driving. 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 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), where 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.

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

III. Coarse Attention Selection

Two external (outward looking) sensors are used in the proposed system. The first is the radar modality, utilized to find attended regions (possible nearby objects) within the image. The second senses the vision modality. Information from this sensor is used to develop the capability of object recognition. Table I & II specify the sensor parameters of radar and vision modalities, respectively.

TABLE I: Sensor Specifications of Radar System

Key parameters	Specification
Refreshing rate	10 Hz
No. of targets	max. of 20 targets
Distance	$2 \sim 150\text{m} \pm \mathbf{max}(5\%, 1.0\text{m})$
Angle	$15^{o} \pm \mathbf{max}(0.3^{o}, \text{range of } 0.1\text{m})$
Speed	$\pm 56\text{m/s} \pm 0.75\text{m/s}$

TABLE II: Sensor Specifications of Vision System

Key parameters	Specification
Refreshing rate	15 Hz
View of fields	45^{o}
Resolution	320 x 240

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 up to 150 meters. Each radar point is presented by a triangle, 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 meters in distance ahead

or more than 8 meters to the right or left outside the vehicle path (e.g., red triangle points in Fig. 2 (right) are omitted).

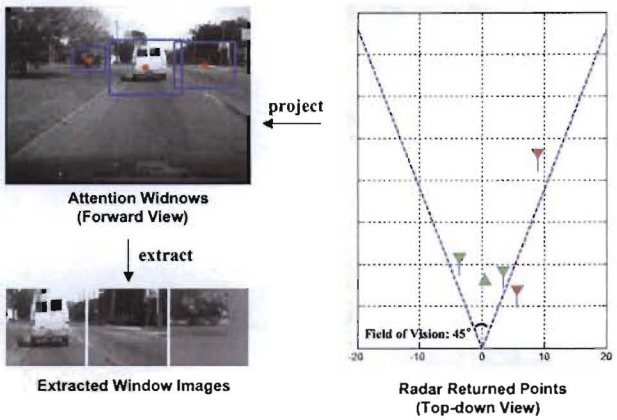

Fig. 2: A projection of effective radar points (green) onto the image plane, where window images are extracted for further recognition.

Based on the estimation of maximum height (3.0 meters) and maximum width (3.8 meters) of environment objects, a target window can be localized based on the radar-returned point (centered at the window). Each target window is projected into the image reference system, using a perspective mapping transformation (see Fig. 2 (upper left)). 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 lager, the attention window in the image is smaller and vice versa.

For each attention window, the pixels are extracted as a single image where most of the non-object pixels 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 part of the image becomes ignored. This is called coarse attention selection — finding candidate areas purely based on physical characteristics (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 depends on the discriminant statistical distributions of the scenes in each radar window. The proposed system can thereby 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 processed and learned through the proposed neural network (see Fig. 1) via 3 layers, till the motor output, where each neuron in the motor layer

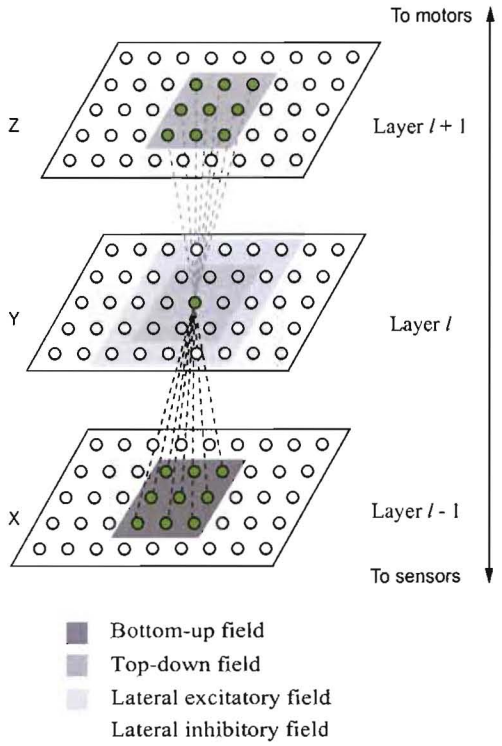

Fig. 3: General structure of the network connection. Neurons are placed (given a position) on different layers in an end-to-end hierarchy – from sensors to motors. Only the connections to a centered cell are shown, but all the other neurons in the feature layer have the same default connections.

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., $\mathbf{w_t}_i^{(l-1)} = \mathbf{w_b}_j^{(l)}$. Moreover, this work does not use explicit lateral connections, but instead uses an approximate method in which 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, $t = 0, 1, 2, ...$, 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.

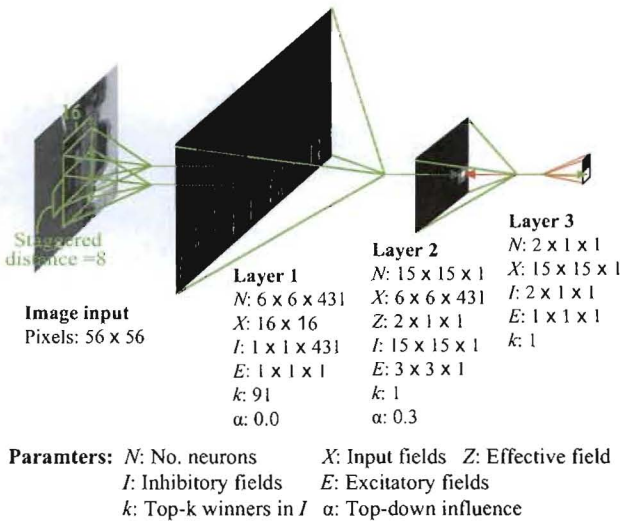

Fig. 4: An example of layer representations (i.e., responses) in the proposed neural network, including a specific set of resource parameters implemented. Green and red directed lines show the bottom-up and top-down connections to the firing neurons respectively. The top-down fields are not applicable in layer 1 and layer 2, and neural representation in layer 1 is reshaped to 36×431 for visualization purpose.

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 [12]. 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 the matured development of layer 1 features (i.e., the layer 1 bottom-up weights converge), 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 "emerged" 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 [13]. It implies that there can not 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.

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

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 0.
    $(\mathbf{y}(t+1), L(t+1)) = \text{In-place}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{z}(t) \mid L(t))$
  end for
end for
for $t = 500001, 500002, ...$ **do**
  Grab the attention window image $\mathbf{s}(t)$.
  Impose the motor vector (class labels) $\mathbf{m}(t)$ to layer 3.
  for $1 \leq l \leq 3$ **do**
    if $l = 1$ **then**
      Get the bottom-up fields $\mathbf{x}(t)$ from $\mathbf{s}(t)$.
    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 the next layer representation (responses).
    else
      Get the bottom-up fields $\mathbf{x}(t)$ from the previous layer representation (responses).
    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

In the following sections, we will go through the critical components or properties of the neural network to achieve robust and efficient object recognition. Sec. V will address statistical optimalities 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

$$X = R_1 \cup R_2 \cup ... \cup R_N \tag{4}$$

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

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

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

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

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 neuron-specific sigmoidal function or its piecewise linear approximation. α_l is a layer-specific weight that controls the maximum influence of top-down versus the bottom-up part.
3: **end for**
4: Simulating lateral inhibition, decide the winner: $j = \arg\max_{i\in J^{(l)}} \hat{y}_i^{(l)}(t+1)$;
5: The cells belonging to excitatory neighborhood $E^{(l)}$ are also considered as winners and added to the winner set $\mathcal{J}$.
6: The responses $\mathbf{y}^{(l)}$ are computed from the pre-responses. Only the winning neuron(s) have nonzero responses and are copied from $\hat{\mathbf{y}}^{(l)}$.
7: Update the number of hits (cell age) n_j for the winning neuron(s): $n_j \leftarrow n_j + 1$. Compute $\mu(n_j)$ by the amnesic function, Compute $\mu(n_j)$ by the amnesic function:

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

where plasticity parameters $t_1 = 20$, $t_2 = 200$, $c = 2$, $r = 2000$ in our implementation.
8: Update winning neuron(s) using its temporally scheduled plasticity:

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

where the scheduled plasticity is determined by its age-dependent weight:

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

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

On the other hand, if N is too large, it is possible to over-fit the space X.

From Eq. 3, a local estimator $\mathbf{w}_{\mathbf{b}i}$ can be expressed as:

$$\Delta\mathbf{w}_{\mathbf{b}i} = \Phi(n_i)[\mathbf{x}_i(t)y_i(t+1) - \mathbf{w}_{\mathbf{b}i}(t)] \quad (5)$$

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

$$\mathbf{x}_i(t)y_i(t+1) = \mathbf{w}_{\mathbf{b}i}(t) \quad (6)$$

Consider a layer (e.g., layer 1 of the proposed network) in which the top-down connections are not available[3], Eq. 7 can be re-written as below:

$$\mathbf{x}_i(t)\frac{\mathbf{x}_i(t)\cdot\mathbf{w}_{\mathbf{b}i}(t)}{\|\mathbf{w}_{\mathbf{b}i}(t)\|\|\mathbf{x}_i(t)\|} = \mathbf{w}_{\mathbf{b}i}(t). \quad (7)$$

such that

$$\mathbf{x}_i(t)\mathbf{x}_i^T(t)\mathbf{w}_{\mathbf{b}i}(t) = \|\mathbf{w}_{\mathbf{b}i}(t)\|\|\mathbf{x}_i(t)\|\mathbf{w}_{\mathbf{b}i}(t) \quad (8)$$

Averaging the both sides of Eq. 8 over $\mathbf{x}_i(t)$, conditional on $\mathbf{w}_{\mathbf{b}i}$ staying unchanged (i.e., converged), we have

$$\mathbf{C}\ \mathbf{w}_{\mathbf{b}i} = \lambda\ \mathbf{w}_{\mathbf{b}i} \quad (9)$$

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. 9 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 for the input covariance matrix. For this reason, the in-place neural learning becomes a principal component analyzer (PCA)[4] [15], which is mathematically optimal to minimize the squared mapping/representational error, such that

$$\mathbf{w}_{\mathbf{b}i}^* = \arg\min_{\mathbf{w}_{\mathbf{b}i}} \sum_t \|(\mathbf{x}_i(t)\cdot\mathbf{w}_{\mathbf{b}i})\mathbf{w}_{\mathbf{b}i} - \mathbf{x}_i(t)\|^2. \quad (10)$$

In addition, the multi-sectional function $\mu(n)$ in Eq. (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. It 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 the second weight $(1+\mu(n))/n$ in Eq. (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 *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 in the aspect of both limited computational resources (spatial) and limited learning experience at any time (temporal), such that

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

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

[4]Although not shown here, Oja et al. [14] has proven that it is the first principal component that the neuron will find, and the norm of the weight vector tends to 1.

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 important characteristics of above dual optimalities in learning natural images, i.e., a mixture of super-gaussian sources [16]. As discussed in [17], when the input is a super-gaussian mixture, the spatial optimality of minimizing representation error in the in-place learning can function as an Independent Component Analysis (ICA) algorithm [18], and its temporal optimality performed surprising efficiency [19]. Such independent components would help separate the non-Gaussian source signals into additive subcomponents supposing the 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 [20], [21]. 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 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 the formula defined Gabor functions.

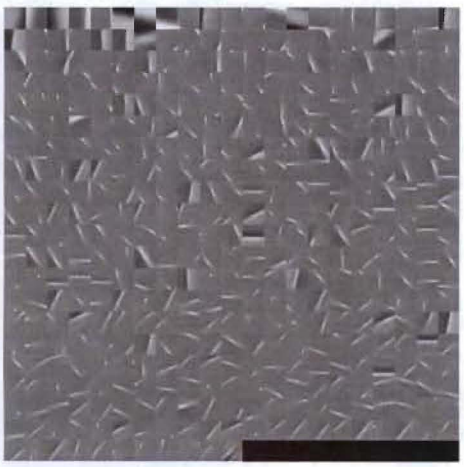

Fig. 5: Developed layer 1 features (431) in one neural column, arranged in a 2D grid. Each image patch shows a bottom-up weight (16×16 dimensions) of one neuron.

Because object appearance in radar-attended window images could potentially vary quite a bit (the object invariance issue), and "leaked-in" background may pose amount of noises, 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 [22] [23]. The redundancy of the input is transformed into the redundancy of the firing pattern of cells. This allows object learning and recall (associative learning) to become a compositional problem (i.e., an view of a novel object is decomposed as a composite of a unique set of independent events). As shown in Sec. VIII, the sparse coding is able to reduce redundant, high-correlated information in the pixel inputs and form the representations such that statistical dependency among them is reduced, while "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 [16]. This pre-processing is called whitening. The whitened sample vector $\mathbf{s}$ is computed from the original sample $\mathbf{s}_o$ as $\mathbf{s} = \mathbf{W}\mathbf{s}_o$, 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 $\frac{1}{\sqrt{\lambda_i}}$, and λ_i is the eigenvalue of v_i. Whitening is very beneficial to uncover the true correlations within the natural images, since it avoids derived features to be dominated by the larger components.

VII. Top-down Abstraction

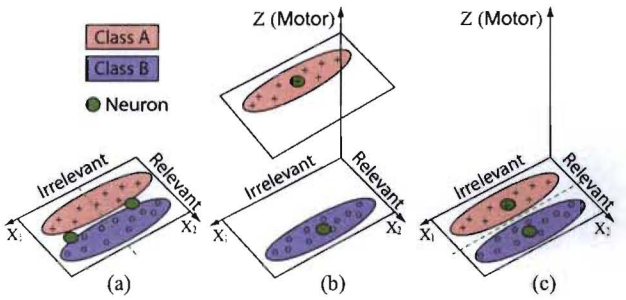

Fig. 6: Illustration of top-down connection roles. Top-down connections boost the variance of relevant subspace in the neural input, resulting in more neurons being recruited along relevant information. The bottom-up input samples contain two classes, indicated by samples "+" and "o" respectively. To see the effect clearly, 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 recruiting neurons along relevant directions. (c) Class partitioned. Especially during the testing phase, although the top-down connections become unavailable and the winner of the two neurons use only the bottom-up input subspace X, the samples are partitioned correctly according to the classes (see dashed line).

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 functional properties of abstraction as below.

1) The top-down connections provide a new subspace where the relevant information (the information that is important to distinguish between motor outputs) will

[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.

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	79.2	...	-0.2	-4.5	...	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	...

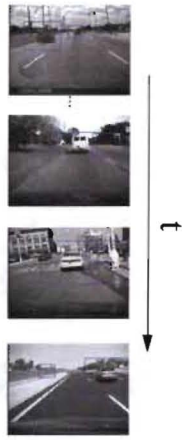

Fig. 7: Examples of radar data and corresponding images in time sequence. It also shows some examples of different road environments in the tested dataset.

have a higher variance than the irrelevant subspace. Since higher variance subspace will recruit more neurons due to the Neuronal Density Theorem [24], the representation acuity becomes higher in the relevant subspace, and the representation becomes more suited to the task(s) that were trained.

Fig. 6 illustrates this top-down connection roles. As the top-down connections correspond to relevant information, the variance in top-down signals boosts the total variance of the relevant subspace. This enhanced variance recruits the locally available neurons to spread along the relevant subspace. As shown in Fig. 6(c), the neurons 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 top-down connection, but before that, the classes in Fig. 6(a) are mixed in the bottom-up subspace X.

2) Neurons form topographic cortical areas according to 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*, leading to the better recognition ability. Consider the within-class variance w_X^2 of the input space X

$$w_X^2 = \sum_{i=1}^{n} E[\|\mathbf{x}_i - \bar{\mathbf{x}}\|^2 \mid \mathbf{x} \in c_i]^\top p_i \qquad (11)$$

and its total variance

$$\sigma_X^2 = E\|\mathbf{x} - \bar{\mathbf{x}}\|^2 \qquad (12)$$

where $\bar{\mathbf{x}}$ is the input mean of X. c_i denotes the class i and p_i denotes the probability of a sample belonging to class i.

The relative within-class variance for the input space X can be written as

$$r_X = \frac{w_X^2}{\sigma_X^2} \qquad (13)$$

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 $X \times Z$. Thanks to the top-down propagation from the motor classes, $w_Z^2/\sigma_Z^2 < w_X^2/\sigma_X^2$, such that the expanded input space $X \times Z$ has smaller relative within-class variance than that in X.

$$r_{X\times Z} = \frac{w_X^2 + w_Z^2}{\sigma_X^2 + \sigma_Z^2} < r_X. \qquad (14)$$

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

We used an equipped vehicle 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. From each environment, multiple sequences were extracted. Each sequence contains some similar but not identical images (different scales, illumination and view point variation etc.). The proposed learning architecture is evaluated in a prototype of 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. For all tests, 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, e.g., different scales, illumination and view point variation etc. The correlation matrix of window images is shown in Fig. 8 (a), indicating the high statistical dependence among the samples, especially, across different objects. Each image is further coded for a sparse representation in layer 1. The correlation matrix of generated sparse representations is shown in Fig. 8 (b). It takes the advantage in two aspects: (1) object samples are decorrelated by the coding process, i.e., cross-object correlation is dramatically reduced; (2) object information is maintained, i.e., within-object samples keep the high correlation.

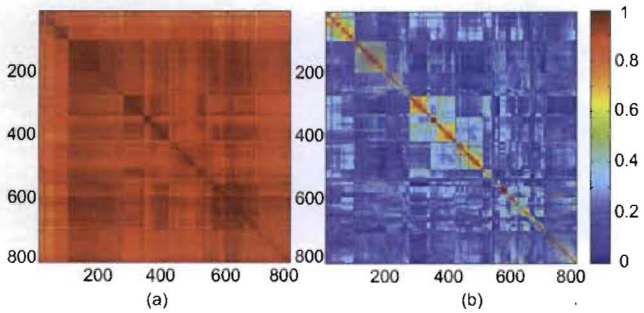

Fig. 8: Correlation matrix of (a) sampled 800 window images in pixel space and (b) their corresponding sparse representations in layer 1 space.

B. Top-down abstraction effect

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

$$p_i = \frac{n(i)}{\sum_1^c n(i)} \qquad i \in 1, 2, ..., c \tag{15}$$

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

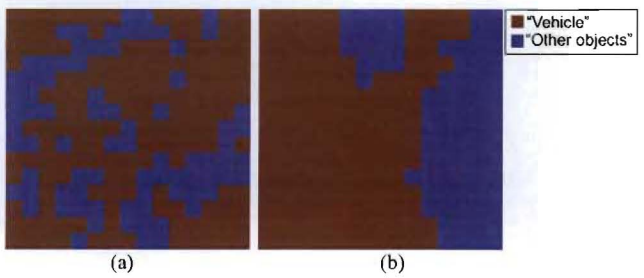

Fig. 9: 2D class maps of 15×15 neurons in layer 2. Each neuron is associated with one color, presenting a class with the largest empirical "probability" p_i.

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 lager 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%, 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 improved 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 improved 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 (56×56) 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) [25]; (3) Incremental Hierarchical Discriminant Regression (IHDR) [26] and (4) the proposed network described in this paper. We used a linear kernel for I-SVM, as is suggested for high-dimensional problems [27]. We did try several settings for a RBF kernel, but the system training becomes extremely slow and the performance improvement is not obvious (by 1%-3% only).

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. Nearest neighbor performs fairly well, but is prohibitively slow. IHDR combines the advantage of K-NN with an automatically developed overlapping tree structure, which organizes and clusters the data. It is useful for extremely fast retrievals due to logarithmic complexity. IHDR performs the recognition better than K-NN, and also is much faster for 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, thereby did not use memory efficiently. I-SVM performed the worst on the high dimensional data with

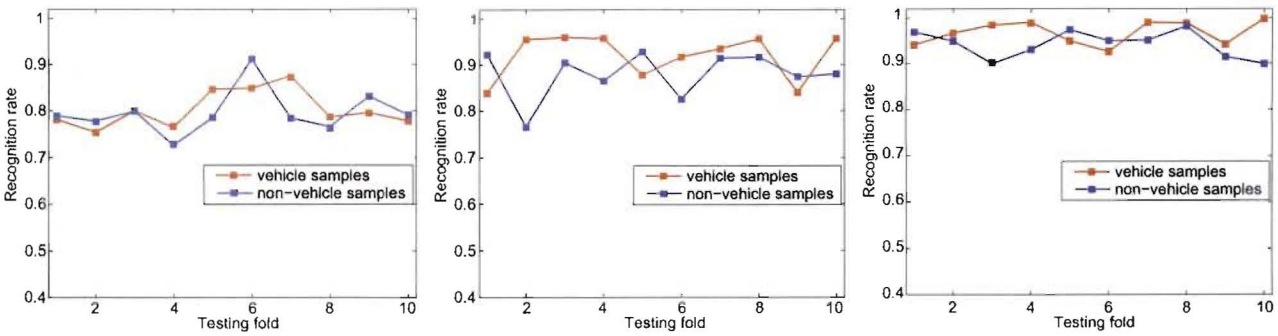

Fig. 10: 10-fold cross validation (a) without sparse coding in layer 1, (b)without top-down connection from layer 3 and (c) of the proposed work.

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

amount of noises, but the testing speed is fastest, since its decision making only based on the small number of support vectors. A major problem with I-SVM is lack of extendability – 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 that could be added later.

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. Overall, the proposed work does not fail in any criteria, although it is not always the "best" in any one category. the proposed work also has its major advantages in its extendability. New tasks, more specifically, new object classes can 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 entails the recognition system to learn while performing onboard for a vehicle. This is very important for the driving assistance and autonomous driving systems, especially as information in input images is huge and highly redundant. The system only needs 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 current 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 paper, we 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 are sparsely coded by the neural network using its layer 1 features developed from the statistics of natural images. Layer 2 of the network further learns in reaction to the coupled sparse (object) representation and external motor 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 in the driving environments, e.g, pedestrians, traffic signs, etc. Since the radar system is robust for various weather conditions, the sensor fusion frame work can potentially extend to some severe weather conditions, such as in rains or snows. Currently, it is assumed that each frame is independent from the next (which is certainly not usual the case). Relaxing this assumption may lead us to ways for the temporal information of images, which should provide a promising method to upgrade the efficiency of the learning system. We hope that these improvements will eventually lead to a vehicle based robot that can learn to be aware of any type of object in its environment.

References

[1] http://en.wikipedia.org/wiki/driverless_car.

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

[3] B. Ulmer. Vita ii - active collision avoidance in real traffic. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 1–6, Paris, October 1994.

[4] DARPA. DARPA urban challenge 2007: Rules. Technical report, DARPA, October 2007.

[5] T. Jochem and D. Langer. Fusing radar and vision for detecting, classifying and avoiding roadway obstacles. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 333–338, Tokyo, 1996.

[6] A. Gern, U. Franke, and P. Levi. Advanced lane recognition - fusing vision and radar. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 45–51, Dearborn, Michigan, October 2000.

[7] 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 *Proceedings of Australian Conference on Robotics and Automation*, pages 14–15, Sydney, November 2001.

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

[9] S. Miyahara, J. Sielagoski, A. Koulinitch, and F. Ibrahim. Target tracking by a single camera based on range-window algorithm and pattern matching. In *SAE 2006 World Congress and Exhibition*, Detroit, November 2006.

[10] U. Kadow, G. Schneider, and A. Vukotich. Radar-vision based vehicle recognition with evolutionary optimized and boosted features. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 749–754, Istanbul, Turkey, June 2007.

[11] M. Bertozzi, L. Bombini, P. Cerri, P. Medici, P. C. Antonello, and M. Miglietta. Obstacle detection and classification fusing radar and vision. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pages 608–613, Eindhoven, Netherlands, June 2008.

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

[13] 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.

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

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

[16] 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.

[17] N. Zhang and J. Weng. Sparse representation from a winner-take-all neural network. In *Proceedings of International Joint Confernce on Neural Networks*, Budapest, Hungary, 2004.

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

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

[20] 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.

[21] B. A. Olshaushen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy used by v1? *Vision Research*, 37(23):3311–3325, 1997.

[22] P. Foldiak. Forming sparse representations by local anti-hebbian learning. *Biological Cybernetics*, 64:165–170, 1990.

[23] 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.

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

[25] 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.

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

[27] 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 *Proceedings of 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 2005.

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 associate 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 B.S., M.S. and PhD degrees from Michigan State University, all in computer science. His research involves the study of biologically inspired algorithms for autonomous development of mental capabilities — especially for visual attention and recognition. He is a student member of the Society for Neuroscience and of the IEEE Computational Intelligence Society.

Juyang Weng received his B.S. degree from Fudan University, and his M.S. and Ph.D. 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. He is an editor-in-chief of International Journal of Humanoid Robotics and and an associate editor of the new IEEE Transactions on Autonomous Mental Development. He was a member of the Executive Board of the International Neural Network Society (2006-2008), the chairman of the Autonomous Mental Development Technical Committee of the IEEE Computational Intelligence Society (2004-2005), the Chairman of the Governing Board of the International Conferences on Development and Learning (ICDL) (2005-2007, http://cogsci.ucsd.edu/ triesch/icdl/), a general chairman of 7th ICDL (2008), the general chairman of 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 autonomous development. He is a fellow of the 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.