Radar-Vision Fusion for Object Classification

Zhengping Ji and Danil Prokhorov
Technical Research Department
Toyota Technical Center - TEMA
Ann Arbor, MI 48105 USA
dvprokhorov@gmail.com

ABSTRACT: We propose an object classification system that incorporates information from a video camera and a long-range radar system. Our system operates in two steps. The first step is attention selection, in which the radar guides a selection of a small number of candidate images for analysis by the camera. In the second step, a multiple layer in-place learning network (MILN) is used to distinguish images of different objects. Though it is more flexible in terms of variety of classification tasks, the system currently demonstrates its high accuracy in comparison with others on real-world data of a two-class recognition problem.

KEYWORDS: radar, camera, attention selection, MILN

1 Introduction

Future automotive vehicles will have many more sensors and systems dedicated to various driver support, semi- and fully-autonomous functions. As one type of active sensors, the radar system has shown a reasonable performance of target detection in relatively simple environments (e.g., highways). It provides reasonably accurate measurements of object distance and velocity in various weather conditions. However, typical vehicle radars do not have enough lateral resolution to model the object shapes, which leads to a limitation for the extraction of object features. On the contrary, typical video cameras, called passive sensors, provide sufficient lateral resolution to locate the boundaries of an object. The cues of shapes and appearances may provide sufficient characteristics for classification of different objects. Considering the complementary properties of the two sensors, we can combine them in a single system for improved performance.

The fusion of data from radar and vision has been widely discussed for driver-assistance tasks (see, e.g., [2], [6], [3], [7], [8]). Instead of a generic object classification architecture, most of work is dedicated to detect specifically vehicles or pedestrians. In the proposed system, we took the advantage of radar-vision integration to achieve an efficient attention selection on candidate targets and employ a general-purpose classification and regression network, Multi-layer In-place Learning Network (MILN) proposed in [5]. Its in-place learning mechanism holds comparatively low operational complexity even for very large networks. Our proposed system is extendable to different driving environments and different classification tasks.

In what follows, we first describe the problem statement and present the operation architecture of the proposed system. In Sec. 3, the learning algorithms is discussed. The experimental results and conclusions are reported in Sec. 4 and Sec. 5, respectively.

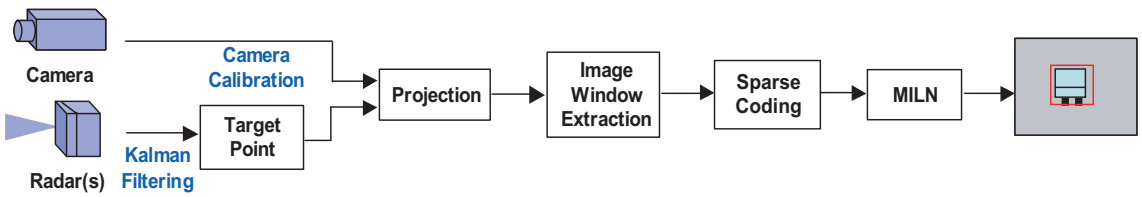

Figure 1: Operation architecture of the proposed system.

2 Problem statement

Our specific problem is to learn to recognize objects in the path of a moving vehicle. The system should be designed to detect and classify specific objects in the driving environment, i.e., separate objects into two classes: "vehicles" and "non-vehicles".

The operation architecture of the proposed system is shown in Fig.1. For this system, two external (outward looking) sensors are necessary. One is a video camera, sensing the vision modality for the development of object recognition. The other is a long-range radar for finding salient regions (possible objects of interest) within the captured image.

2.1 Radar and camera integration

As shown in Fig. 2, a group of target points (red) in 3D world coordinates can be detected from the long-range radar sensor, which scans in the horizontal field of 15^{o}, with the detection range up to 150 meters.

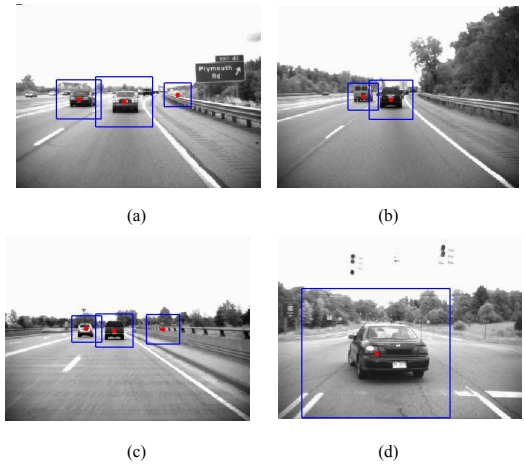

Figure 2: Examples of images containing radar returns (red) which are used to generate attention windows (blue borders). This figure shows examples of the different road environments in our dataset.

The radar-centered coordinates are projected into the image reference system using the perspective mapping transformation. The transformation is performed using the calibration data that contain the intrinsic and extrinsic parameters of each camera. An attention/radar window is created within the image, taking into account expected maximum height and width of the vehicles.

For each radar window, the associated pixels are extracted as a single image. Each image is normalized in size to 56 rows and 56 columns of pixels, and the pixel intensities are normalized from 0 to 1. To avoid stretching small images into the fixed-sized window, each small image is placed

in the upper left corner of the size-normalized image, and the other pixels are set to intensities of 0.5. Sometimes more than one object may be captured in each radar window, but for the purpose of classification, the radar window is assigned with only one label, either "vehicle" or "non-vehicle". We are not concerned about accurate labeling of multiple radar returns in the 56×56 window because it is quite rare that objects of different labels end up in the same window due to design specifics of our system.

By our creating a *set* of attention windows for each image, the output of the required mapping is greatly simplified. It allows us to learn detection and recognition of multiple objects within the same captured image, as long as there is a radar point returned from each object.

Our sensor integration leads to two advantages: (a) instead of manually segmenting salient objects in an image, the radar system automatically provides areas of attention for the objects, obviating the need for potentially time-consuming human interpretations; (b) in each image from the camera, the desired output is simplified from a number of labels for objects in the original image to the individual label for each of the radar windows.

3 Multilayer in-place learning networks

The 56×56 images are fed into our classifier – Multilayer In-place Learning Network (MILN). In in-place learning, each neuron is not allowed to utilize partial derivatives (as in the popular backpropagation) or covariance matrices. Each neuron learns on its own, i.e., using only the data it receives from its inputs.

The MILN may have multiple layers. In this application, the images are considered to be on the bottom (Layer 0) and the class labels on the top (Layer L, where $L = 3$). Each MILN neuron in hidden layer l ($0 < l < L$) has two types of input connections:

- bottom-up (excitatory) weight vector $\mathbf{w_b}$ that links input lines from the previous layer $l - 1$ to this neuron,
- top-down (excitatory or inhibitory) weight $\mathbf{w_t}$ that links the output from the neurons in the next layer $l + 1$ to this neuron,

where the supervision is achieved by the top-down projection from the decision makers. Note that the bottom-up weights of neurons in the layer $l + 1$ act as the top-down weights of neurons in the layer l. Each linked weight pair (i, j) shares the same value, i.e., $\mathbf{w_t}_i^l = \mathbf{w_b}_j^{l+1}$.

Algorithm 1 is the MILN learning algorithm $\mathbf{z}(t) = \text{MILN}(\mathbf{s}(t), \mathbf{m}(t))$, where $\mathbf{s}(t)$ is the input vector (vectorized 56×56 image) at time t, and $\mathbf{m}(t)$ is the corresponding target vector (supervision signal). The vector $\mathbf{z}(t)$ contains responses of neurons in every layer. Note that Algorithm 1 may learn continuously ($n \to \infty$ in step 2), if required.

4 Experiments and results

We used a properly equipped test vehicle to capture real-world image and radar sequences for training and testing. Our preliminary dataset consists of 400 images – stretches of roads in different places of city or highway environment, under different viewpoint variations, illuminations and scales (see Figure 2 for a few examples of different environments). The images were captured with 0.5-second sampling. There were 499 samples in the vehicle class and 264 samples in the other object class. For all tests, each original image captured by the camera was 240 rows and 320 columns. (Each radar window was size-normalized to 56 by 56 image and intensity-normalized to $\{0, 1\}$.)

Algorithm 1 MILN

For $l = 1, ...L-1$, set the output of the layer l at time $t = 0$ to be $\mathbf{z}_l = \mathbf{0}$, where $\mathbf{0}$ denotes a zero vector.
for $t = 1, 2, ..., n$ **do**
  $\mathbf{y}(t) = \mathbf{s}(t)$;
  for $l = 1..., L-1$ **do**
    Set top-down input: $\mathbf{e}(t) = \mathbf{m}(t)$ if $l = L-1$, and $\mathbf{e}(t) = \mathbf{0}$ otherwise.
    for $i = 1, 2, ..., c$ **do**
      Compute pre-response of neuron i from bottom-up and top-down input connections as:

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

$$+\alpha_l\frac{\mathbf{w}_{\mathbf{t}_i}^l(t)\cdot\mathbf{e}(t)}{\|\mathbf{w}_{\mathbf{t}_i}^l(t)\|\|\mathbf{e}(t)\|})$$

      where $\alpha_l = 0.3$, and g_i is the standard neural network sigmoidal function.
      Simulating lateral inhibition, decide the winner: $j = \arg\max_{1\le i\le c}\{\hat{\mathbf{z}}_l(t)\}$.
      If $l = 1$, the 3×3 neighboring cells are also considered as winners and added to the winner set $\mathbf{J}$ for the subsequent updating of the weights $\mathbf{w_b}$, otherwise $\mathbf{J} = \{j\}$.
      The winner set $\mathbf{J}$ may still contain neurons with zero pre-responses $\hat{\mathbf{z}}_l$. Define a sub-set $\mathbf{J}' \subseteq \mathbf{J}$, such that the response $z_{l,j} = \hat{z}_{l,j}$ if $\hat{z}_{l,j} \neq \mathbf{0}$, $\forall j \in \mathbf{J}'$.
      Update the number of hits (cell age) n_j ($j \in \mathbf{J}'$): $n_j \leftarrow n_j + 1$, and compute $\mu(n_j)$ by the amnesic function:

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

      where plasticity parameters $t_1 = 20, t_2 = 200, c = 2, r = 2000$ in our implementation.
      Update the winner neuron(s) $j \in \mathbf{J}'$:

$$\mathbf{w}_{\mathbf{b}_j}^l(t) = w_1\mathbf{w}_{\mathbf{b}_j}^l(t-1) + w_2 z_{l,j}\mathbf{y}(t),$$

      where the scheduled plasticity is determined by its two age-dependent weights w_1,w_2:

$$w_1 = \frac{n_j - 1 - \mu(n_j)}{n_j}, w_2 = \frac{1 + \mu(n_j)}{n_j},$$

      with $w_1 + w_2 \equiv 1$.
      All other neurons keep their ages and weight unchanged: For all $1 \le i \le c$, $i \notin \mathbf{J}'$, $\mathbf{w}_{\mathbf{b}_i}^l(t) = \mathbf{w}_{\mathbf{b}_i}^l(t-1)$.
    end for
    $\mathbf{y}(t) = \mathbf{z}_l(t)$;
  end for
end for

Our learning method incrementally updates the network weights using one piece of training data at a time. After the network initialization with $10 \times 10 = 100$ samples $\mathbf{w}_{\mathbf{b}_i^1} = s(i), \forall i = 1, 2, ..., 100$, the rest of data is used for the incremental learning by the network. The network can be trained quickly: approximately 25% of samples are sufficient to achieve no less than 80% recognition rate on all the data samples if each training sample is presented only once. Even quicker and more accurate training results from relearning already presented samples.

4.1 Performance Evaluation

Four different algorithms were compared for the purpose of their performance evaluation in the proposed sensor fusion system, where an efficient (memory controlled), real-time (incremental and fast), autonomous (without turning the system off to change or adjust), and extendable (the number of classes can increase) architecture is sought. We tested the following classification methods: k-Nearest Neighbor (NN), with $k = 1$ and $L1$ distance metric for baseline performance, incremental Support Vector Machine (I-SVM)[1][1], Incremental Hierarchical Discriminant Regression (IHDR) [11] and our method MILN. We used a linear kernel for I-SVM as suggested for high-dimensional problems [4]. (We experimented with several settings for an RBF kernel but did not observe as good a performance as with the linear kernel.)

Inputs to all systems were from the non-transformed – appearance or "pixel" – space with input dimension of $56 \times 56 = 3136$. The results of 10-fold cross validation are summarized in Table 1.

Table 1: Average performance & comparison of learning methods for pixel inputs

Learning Method	Overall Accuracy	"Vehicle" Accuracy	"Other Objects" Accuracy	Training Time Per Sample	Test time Per Sample	Final # Storage Elements
NN	$93.9 \pm 1.8\%$	$94.3 \pm 1.4\%$	$93.4 \pm 2.0\%$	N/A	432 ± 20 ms	621
I-SVM	$94.4 \pm 2.4\%$	$97.1 \pm 1.0\%$	$92.1 \pm 6.1\%$	134 ± 10ms	**2.2** $\pm$ **0.1** ms	**44.5** $\pm$ **2.3**
IHDR	**95.8** $\pm$ **1.0**%	$96.4 \pm 0.7\%$	**95.6** $\pm$ **2.8**%	**2.7** $\pm$ **0.4**ms	4.7 ± 0.6 ms	689
MILN	$94.6 \pm 2.3\%$	**97.1** $\pm$ **1.6**%	$91.2 \pm 5.3\%$	17.1 ± 2.0ms	8.8 ± 0.6ms	100

NN demonstrated the worst performance, and it is also prohibitively slow. IHDR combines the advantage of NN with an automatically developed overlaying tree structure that organizes and clusters the data. It is useful for extremely fast retrievals. IHDR performs better and faster than NN, and it can be used in real time. However, IHDR typically takes a lot of memory. It allows sample merging, but in this case it saved every training sample, not using memory efficiently. I-SVM performed well with both types of input, and it uses the least memory, in terms of the number of support vectors automatically determined by the data, but training time is the worst. Another potential problem with I-SVM is its lack of extendability to situations when the same data may be later expanded from the original two to more than two classes. As general purpose regressors, IHDR and MILN are readily extendable.

Overall, MILN is very competitive for any comparison category, although it is not always the "best" in all categories, as currently implemented. NN is too slow in testing, and I-SVM is too slow in training and not extendable without full retraining. IHDR uses too much memory and does not represent information efficiently and selectively.

[1]Software was obtained from http://www.biology.ucsd.edu/∼gert/svm/incremental/

5 Conclusion

Our results show promise for development of object recognition and understanding systems based on the described framework, in which the radar-directed attention mechanism reduces complexity of the analysis needed for each image frame. The proposed system architecture based on the MILN allows incremental learning, which is feasible for real-time use not only for automotive vehicles but also for any developmental robot that can sense its environment from multiple sensors.

References

[1] 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.

[2] C. Coue, T. Fraichard, P. Bessiere, and E. Mazer. Multi-sensor data fusion using bayesian programming: An automotive application. In *International Conference on Intelligent Robots and Systems*, Lausanne Switzerland, 2002.

[3] R. Grover, G. Brooker, and H. F. Durrant-Whyte. A low level fusion of millimeter wave radar and night-vision imaging for enhanced characterization of a cluttered environment. In *Proceedings 2001 Australian Conference on Robotics and Automation*, Sydney.

[4] C. Hsu, C. Chang, and C. Lin. A practical guide to support vector classification. 2003.

[5] T. Luwang J. Weng, H. Lu and X. Xue. A multilayer in-place learning network for development of general invariances. *International Journal of Humanoid Robotics*, 4(2), 2007.

[6] T. Jochem and D. Langer. Fusing radar and vision for detecting, classifying and avoiding roadway obstacles. In *Procedings IEEE Symposium on Intelligent Vehicles*, Tokyo.

[7] J. Laneurit, C. Blanc, R. Chapuis, and L. Trassoudaine. Multisensorial data fusion for global vehicle and obstacles absolute positioning. In *Proceddings of IEEE Intelligent Vehicles Symposium*, Columbus.

[8] Shunji Miyahara et al. Target tracking by a single camera based on range-window algorithm and pattern matching. In *SAE 2006 World Congress and Exhibition*, Detroit.

[9] B.A. Olshausen and D.J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14:481–487, 2004.

[10] B. A. Olshausen and D. J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by VI? *Vision Research*, 37:3311–3325, 1997.

[11] J. Weng and W.S. Hwang. Incremental hierarchical discriminant regression. *IEEE Trans. on Neural Networks*, 2006.

[12] J. Weng and N. Zhang. Optimal in-place learning and the lobe component analysis. In *Proc. World Congress on Computational Intelligence*, Vancouver, Canada, July 16-21 2006.