BIOLOGICALLY-INSPIRED MODEL OF VISION-BASED INDEPENDENTLY MOVING OBJECTS DETECTION SYSTEM

Nikolay Chumerin[1], Marc M. Van Hulle[2]
Katholieke Universiteit Leuven, Belgium

Vision-based independent motion detection systems have attracted a lot of attention lately. Such sort of system could be used in on-board automotive assistance system to help driver prevent possible collisions with other independently moving objects (IMOs). In this paper we present a biologically inspired model of IMOs detection system. The proposed model, according to a widely accepted in neuroscience hypothesis, consists of two information-processing streams: "what" (crucial for objects recognition) and "where" (responsible for independent motion discrimination).

Introduction

Every year about 1.2 million people die in car accidents. Approximately every minute one person dies in car crash. More than 80% of accidents are collisions between moving objects. This sad statistics reveals why developing on-board drive assistance system has become an area of such active research recently.

One of the most important issues is the price of the final solution. It will be quite hard to sell a car at the price of an aircraft, even if it is extremely safe. That is why industry most interested in cheap vision-based systems.

The problem of independent motion detection from stereo video stream acquired by static cameras differs dramatically from the same problem in a case of moving cameras. In last case optical flow shows that everything is moving. Bumps and shakes during driving make IMOs detection even more complicated. That is why majority of classical optical flow based algorithms cannot detect independent motion with acceptable accuracy in non-static cameras case.

In this study we propose approach, which does not belong to computer vision mainstream and is based on separate processing and successive fusion of different information streams. The paper is organized as follows: in Section 1 we present outlook of the model, in Sections 2 and 3 describe two model's information streams, in Section 4 we explain training procedures, and in the last section we present results and conclude our study.

1. Model outline

The main reason for using two different streams was unsatisfactory quality of final classification based solely on the "where" stream. One of the possible explanations is in losing of relative position information (among positively classified pixels) in classification. Without additional tricks we cannot get rid of high noise ratio and "IMOs in the sky". On the hand, we cannot use only "what" stream alone, just because it works with static frames and does not deal with temporal information. It means that "what" stream is not able to distinguish between *moving* and *static* (with respect to environment) objects. And the last, but not the least, argument for two processing streams was in wide support of this idea in visual neuroscience (for review, see [1]).

[1] NC is supported by the Belgian Fund for Scientific Research – Flanders (G.0248.03).
[2] MMVH is supported by the Excellence Financing program of the K.U.Leuven (EF 2005), the Belgian Fund for Scientific Research – Flanders (G.0248.03, G.0234.04), the Flemish Regional Ministry of Education (Belgium) (GOA 2000/11), the Belgian Science Policy (IUAP P5/04), and the European Commission (NEST-2003-012963, IST-2002-016276, IST-2004-027017)).

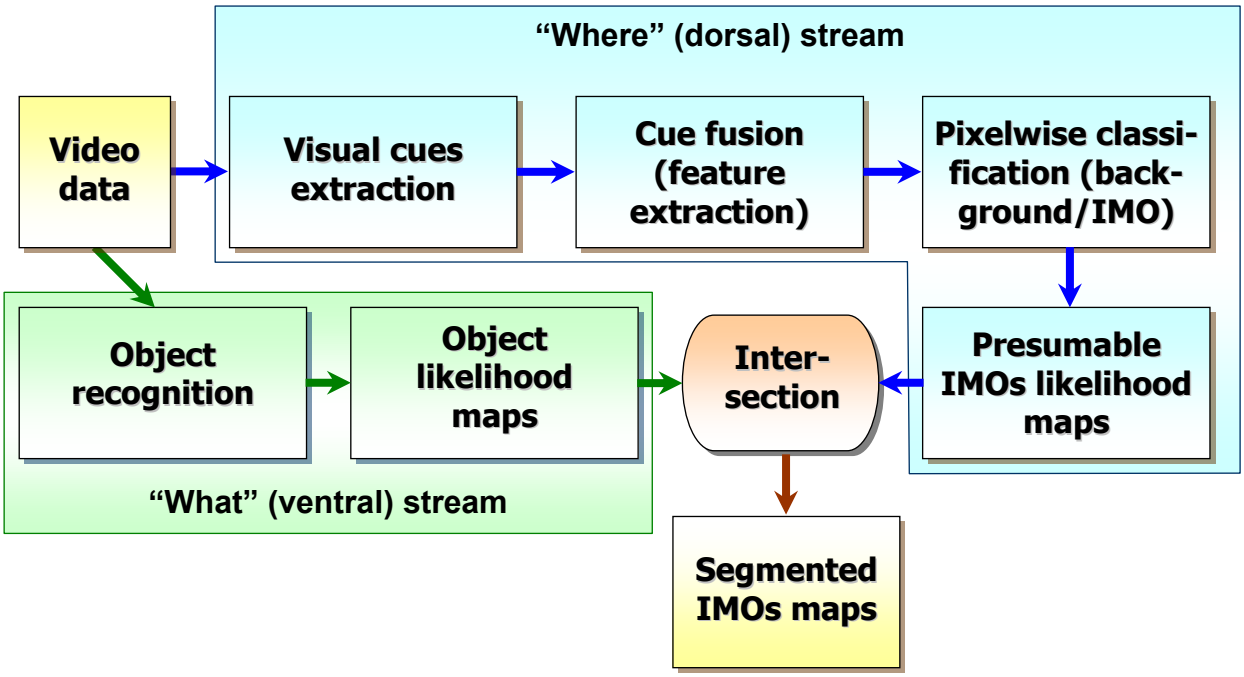

Fig. 1. Outline of proposed model.

2. "Where" stream

As it is shown on Fig. 1, to obtain Presumable IMOs likelihood maps in "where" stream we consequently carry out visual cues extraction, cue fusion and classification.

As visual cues we consider: *static and dynamic disparity* (stereo) [2], *motion in depth* [3], *population coding motion* (optical flow, two components) [4], *edges* (orientation) [5] and *relative coordinates* (two components). Unfortunately, calculation of all cues set is not possible for every pixel. To avoid sparseness of the data we had to set all unobtainable values to zero. In such a way for each pixel from original frame we obtain a vector of visual cues.

For cue fusion and classification we have tried a number of setups, but the best performance was shown by MLP in which these two steps are incorporated. In our experiments we tried different configurations for MLP, but all of them had 3 layers with (4–8) linear neurons in first layer, (8-16) nonlinear neurons in second layer and one linear output neuron. We trained MLP to classify every cues vector (corresponding to a pixel of entire image) into two classes: *background* or *IMO*. After training, MLP can be used for building likelihood (of being IMO) map for the entire frame.

3. "What" stream

For recognition of vehicles and other potentially dangerous objects (like bicycles, motorcycles and pedestrians) we have used the state of the art recognition paradigm – convolutional network LeNet, proposed by LeCun and colleagues [6]. Modifications of LeNet were successfully exploited for generic object recognition [7] and even for autonomous robot's obstacle avoidance system [8].

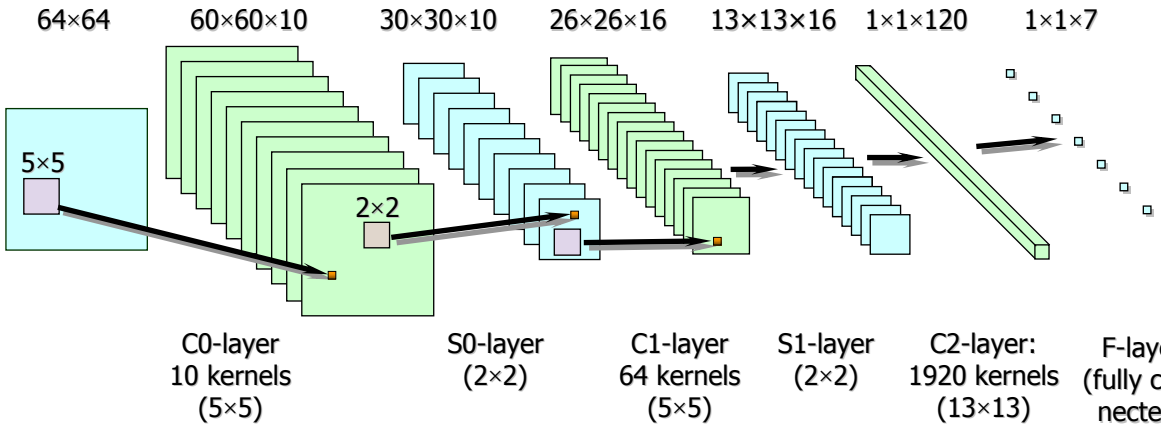

Fig. 2. LeNet – a feed forward convolutional neural network, used in "what" stream processing.

We have used CSCSCF configuration of LeNet (see Fig. 2) comprising six layers: three convolutional layers (C0–C2), two subsampling layers (S0–S1) and one fully connected layer (F). As an input LeNet get 64×64 grayscale image. Layer C0 convolute input with ten 5×5 kernels, adds (ten) corresponding biases and passes result to squashing function[3] to obtain ten 60×60 feature maps. In layer S0 each 60×60 map is subsampled to 30×30 map by non-overlapping 2×2 summation, multiplying by coefficient, adding bias and squashing. S0 layer has ten coefficient-bias couples (one couple for each feature map). Computations in C1 are the same as in C0 with only difference in connectivity: each C1's feature map is obtained as a sum of convolutions with a number of previous (S0's) maps (see Table 1).

S0 \ C1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	■				■					■			■		■	■
1	■					■				■	■			■		■
2	■	■				■	■						■	■		■
3		■				■	■				■	■			■	■
4		■	■				■	■				■				■
5			■				■	■			■	■				■
6			■	■				■	■			■		■		■
7				■				■	■				■	■		■
8				■	■				■	■	■				■	■
9					■				■	■			■		■	■

Table 1. S0-C1 connectivity matrix. In *i*-th column shaded cell corresponds to S0's feature map involved in C1's *i*-th feature map computation. Each shaded cell in the table corresponds to the independent kernel in LeNet. Number of kernels in C1: 6×3 + 9×4 + 1×10 = 64.

Layer S1 subsamples C1's feature maps in the same manner as S0 subsamples feature maps of C0. Last convolutional layer C2 has kernels sized 13×13 and 120 feature maps which are fully connected with all S1's 16 feature maps. It means that number of C2's kernels is 16×120 = 1920 and corresponding connectivity matrix should have all cells shaded. Output layer consists of 7 neurons, which are fully connected to C2's outputs. It means that each neuron in F (corresponding to a particular class *background*, *cars*, *motorbikes*, *trucks*, *buses*, *bicycles* and *pedestrians*) just squashes biased weighted sum of all C2's output.

LeNet scans input image (left frame) in two scales 320×256 and 640×512 with 64×64 sliding window and steps 8 (for 320×256) and 16 (for 640×512). For each position of the sliding window we add output of LeNet to corresponding (to window) range in 320×256 matrix, which after normalization is considered as probability map for considered class.

[3] $f(x) = A \cdot \tanh(S \cdot x)$, where parameters were chosen $A = 1.7159$ and $S = 2/3$ according to [6].

4. Training

For the training of the both streams we used two rectified stereo video sequences, consisting of 450 frames each. Using specially developed software we labeled left frames of sequences. These labels were used for cue fusion classifier training as well as for preparing training dataset for LeNet.

Due to memory limitations we used *small batches with increasing size* version of BFGS Quasi-Newton algorithm for cue fusion classifier training. Samples for each batch were randomly taken from all frames of all scenes. Training was stopped after reaching 0.06 (MSE) performance. For LeNet training we prepared a huge training dataset of 64×64 grayscale images (approximately 98K backgrounds, 49K cars, 5K motorbikes, 2.4K trucks, 1.2K bicycles, 150 buses, and 1.2K pedestrians). Images were taken not only from training sequences but mainly from publicly available recognition databases (LabelMe[4], VOC[5]). All kernels, coefficients and biases described in previous section are trainable parameters.

LeNet was trained with a stochastic version of the Levenberg-Marquardt algorithm with diagonal approximation of the Hessian [6]. Training was stopped after reaching misclassifications rate less then 1.5%.

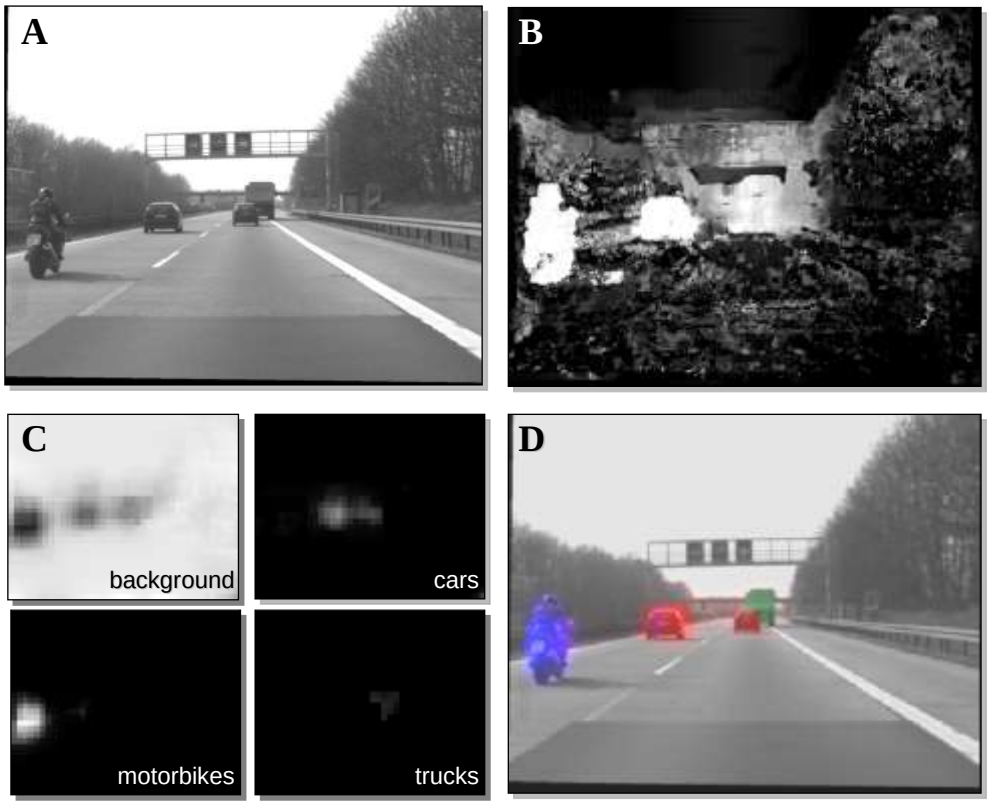

Fig. 3. Results for scene "motorway3": *A*, an original (left) frame of testing video sequence. *B*, output of the "where" stream: intensity of each pixel means probability of being part of the IMO. *C*, output the "what" stream: each subfigure contains probability map for one of the 7 classes: background, cars, motorbikes, trucks, buses, bicycles and pedestrians (where present here only four classes because rest maps contain values very close to zero). *D*, intersection of the IMOs map with the non-background map, superimposed over original image. Here we used different colors to present different classes.

[4] http://labelme.csail.mit.edu/
[5] http://www.pascal-network.org/challenges/VOC/

5. Results and conclusions

Presented model shows relative robustness in IMOs detection and classification (Fig. 3). Mixing two streams we increase reliability of result and automatically clean up it from undesirable noise. This is a crucial issue in the case when video stream is obtained from moving cameras.

Further improvement of the model's performance we see in combining two processing streams at earlier stages. Namely, using a common bank of Gabor-like (fixed/non-trainable) filters in visual cues extraction stage and in LeNet's C0 layer. This step will definitely reduce computations and it is also conformed by neuroscience: both streams originate from the primary visual cortex [1]. Another way to reduce computations is to reduce data amount to process. In dorsal stream it is possible to choose only most relevant cues for cue fusion. In ventral stream we can build object likelihood maps not for full frame, but only for regions containing motion information. The latter has obvious biological support: in real visual systems recognition is preferable to moving objects.

References

1. Ungerleider L.G., Pasternak T. Ventral and Dorsal Cortical Processing Streams. *The Visual Neurosciences*, 1st Edition. MIT Press, vol. 1, 2003, pp. 541–562.
2. Sanger T.D. Stereo disparity computation using Gabor filters. *Biological Cybernetics*, vol. 59, №6, 1988, pp. 405–418.
3. Solari F., Sabatini S.P., and Bisio G.M. Fast technique for phase-based disparity estimation with no explicit calculation of phase. *Electronic Letters*, 37(23): 1382-1383, November 2001.
4. Sabatini S.P., Cavalleri P., Solari F., and Bisio G.M. Physicalist computational structures for motion perception in mammal visual cortex. In *Proceedings of World Congress on Neuroinformatics* 2001, 24-29 September 2001, Vienna, Austria, pages 133-142.
5. Kolesnik M., Barlit A. Iterative Orientation Tuning in V1: A Simple Cell Circuit with Cross-Orientation Suppression. *Lecture Notes in Computer Science*, Springer, 2003, pp. 232–238.
6. LeCun Y., Bottou L., Bengio Y., Haffner P. Gradient-Based Learning Applied to Document Recognition. In *Proceedings of the IEEE*, vol. 86, №11, Nov. 1998, pp. 2278–2324.
7. LeCun Y., Huang F.J., Bottou L. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In *Proceedings of CVPR'04*, 2004.
8. LeCun Y., Muller U., Ben J., Cosatto E., Flepp B. Off-Road Obstacle Avoidance through End-to-End Learning. In *Advances in Neural Information Processing Systems*, vol. 18, 2006.