

Bio-inspired Perception Sensor (BIPS) Concept Analysis for Embedded Applications

Louise Sarrabezolles^{1,2}[0000-0003-3042-0586],
Antoine Manzanera²[0000-0001-5718-411X], and
Nicolas Hueber¹[0000-0003-4374-5160]

¹ French-German Research Institute of Saint-Louis, 68300 Saint-Louis, France

² ENSTA ParisTech, 91120 Palaiseau, France

Abstract. The Bio-inspired Perception Sensor (BIPS) component is a small and low power bio-inspired on-chip device which has been used in different computer vision applications (traffic analysis, driving assistance, object tracking). It caught the attention of the embedded vision community, since its specifications could help overcoming the time, size, weight and energy bottlenecks that still limits the current development of computer vision systems. For a long time, the lack of mathematical and algorithmic models of the component has prevented it to spread among the research community. But the recent formalization of BIPS basic functions and mechanisms has allowed to develop numerical models and simulators, in order to better evaluate the advantages and limitations of the concept. In this paper, we experimentally address the generalization capability of the BIPS concept, by evaluating it on the road lane detection application. This allows to illustrate how its parameters can be adapted for a specific vision task. This approach permits to automatically instantiate the main parameters, which stabilizes the system output and improves its performance. The obtained results reach the level of the *caltech-lanes* reference.

Keywords: Bio-inspired · Embedded Vision · Road Lane Detection.

1 Introduction

The detection and recognition applications in embedded computer vision systems for robots, autonomous vehicles, security cameras or drones, are limited by the constraints on their execution time, size, weight and power consumption. Building efficient, versatile and evolving algorithms in such context is one of today's challenges of Computer Vision. The current best algorithms according to the academic datasets and challenges (e.g. [8][5]) are based on deep learning methods, which imply huge learning databases and off-line training. Most of these algorithms are tested on multi-GPU environment: for example the KITTI challenge best algorithms were all tested on CPU/GPU platforms and yet, few of them are real-time [5]. There are several attempts to integrate deep networks on low power system-on-chip but they have to reduce the size of the network

and they are not processing the training phase [10]. As Ehsan et al. proposed for the UAVs application[4], we believe that a new strategy must be followed in the design of computer vision algorithms, based on fully hardware or hybrid solutions.

The bio-inspired computer vision community follows this strategy and proposed several interesting methods inspired by the human vision and leveraging the properties of electronic components. Among those, one particular component caught our attention: the Bio-inspired Perception Sensor (BIPS). This component is small and low power: 50 mm², 2 W [12]. It has been used and showed interesting results in different industrial computer vision applications such as traffic analysis, driving assistance and object tracking. For a long time, the concept of the BIPS has remained unknown to the community and the communication about its principles limited to some patents and a few general public articles [11][12]. Recently, through the study of Sarrabezolles et al. [14], a mathematical formalization has been established, allowing to evaluate the method on different practical applications.

In this paper, we present the results obtained using this method on the road lane detection application. The formalization and models recalled on Section 2 allow to adapt the method to road lane detection, as explained on Section 3. Using the *caltech-lanes* database [1], we were able to compare the performance of the method in this application and to illustrate the influence of its internal parameters on Section 4. The promising results consolidate the focusing of our research on this method and its possible improvements, as well as its potential adaptation to different computer vision applications, as discussed in Section 5.

2 The BIPS concept

The Bio-Inspired Perception Sensor (BIPS) component has been developed by P.Pirim [11] based on bio-inspired electronic designs. Its experimental development did not provide a true formalization of the concept behind, thus impeding its use for academic researches. The recent study by Sarrabezolles et al. [14] provides the basic functions underlying the BIPS concept, that can be decomposed into two parts: an extraction of local features, and an original region of interest detection and description process.

The extraction part is inspired from the retinal and first layer of the visual cortex processes [6]. Its current implementation computes on the fly different kinds of local features: the tonal features luminance, saturation and hue; the structural features gradient and curvature; and the temporal features temporal variation and optical flow. Each feature can be represented by an application F defined on the pixel set $\llbracket 0; W \rrbracket \times \llbracket 0; H \rrbracket$, where W and H are the dimensions of the input images. A more detailed description of the local features used in the BIPS is given in [14].

The detection part is addressed by an original method called “dynamic attractor” (DA) by its inventor P. Pirim. It consists of the joint computation of feature histograms and the sequential detection of the histograms principal mode(s).

The original component works on three features: one descriptive (i.e. tonal, structural or temporal), and two spatial ones (i.e. the coordinates X and Y), but the formalization shows that the number of features could be extended [14]. This process is inspired by the visual cortex processing model, which shows plasticity [13] and combines two information streams, namely the dorsal and the ventral streams, semantically related to the "what" and the "where", respectively [7]. The DA process allows to find a bounding in the feature space, which selects a set of pixels representative of an object. This process is iterative and can be summarized by the following equations:

Initialization $k = 0$

$$\forall m \in \llbracket 1, M \rrbracket, A_m^0 = \min(\Omega_m) \text{ and } B_m^0 = \max(\Omega_m). \quad (1)$$

Iteration for $k = 1$ to K_{steps}

Histogram computation

$\forall n \in \llbracket 1, M \rrbracket,$

$$H_n^k(f) = \mathbf{card} \left\{ p \in \mathbb{I} \times \mathbb{J} \left/ \begin{array}{l} val(p) = 1, F_n(p) = f \text{ and} \\ \forall m \in \llbracket 1, M \rrbracket, A_m^k \leq F_m(p) \leq B_m^k \end{array} \right. \right\}. \quad (2)$$

which gives $q_{\max} = \max_{f \in \Omega} (H(f))$ and $f_{\max} = \arg \max_{f \in \Omega} (H(f))$ the histogram maximums and associated features, and where Ω_m is the feature space of the m^{th}

feature; $\mathbb{I} = \llbracket 0; \mathbf{W} \rrbracket$ et $\mathbb{J} = \llbracket 0; \mathbf{H} \rrbracket$ are the input images dimensions; A_m^k and B_m^k are the boundaries applied to the m^{th} feature; and val is a binary validation map that comes either from thresholds on the input images or from another DA.

These histogram computations are computed on the fly and provide the detection outputs: $(q_{\max}^{n,k}, f_{\max}^{n,k})_{n \in \llbracket 1, M \rrbracket}$ the histogram modes; $(A_n^k, B_n^k)_{n \in \llbracket 1, M \rrbracket}$ the feature boundaries; and N_k the number of selected pixels.

Boundaries update

The boundaries update is the most important part of the process, allowing and defining the convergence of the "dynamic attractor" (DA). Two kinds of update are possible, the first one concentrates on a unique histogram maximum which is the mode, while the second may encompass several maxima. This update is done on one single feature at each step of the DA iteration (Fig. 1).

At each k , $\exists! n \in \llbracket 1, M \rrbracket$, such that $k = qM + n, q \in \mathbb{R}_+$,

Mode 1: Concentrate on a unique maximum

$$\begin{aligned} A_n^{k+1} &= \min \left\{ f \in \Omega_n, f \leq f_{\max}^{n,k} \left/ \forall f' \in [f; f_{\max}^{n,k}], H_n^k(f') > \tau_n^k \right. \right\}, \\ B_n^{k+1} &= \max \left\{ f \in \Omega_n, f \geq f_{\max}^{n,k} \left/ \forall f' \in [f_{\max}^{n,k}; f], H_n^k(f') > \tau_n^k \right. \right\}, \end{aligned} \quad (3)$$

Mode 2: Encompass several maxima

$$\begin{aligned} A_n^{k+1} &= \max \left\{ f \in \Omega_n \left/ \forall f' \in [\min(\Omega_n); f], H_n^k(f') \leq \tau_n^k \right. \right\}, \\ B_n^{k+1} &= \min \left\{ f \in \Omega_n \left/ f' \in]f; \max(\Omega_n)], H_n^k(f') \leq \tau_n^k \right. \right\}, \end{aligned} \quad (4)$$

where τ_n^k is the threshold and must be strictly inferior to $q_{\max}^{n,k}$ to ensure the convergence. And $\forall m \in \llbracket 1, M \rrbracket, m \neq n, A_m^{k+1} = A_m^k$ and $B_m^{k+1} = B_m^k$.

In order to limit the processing time, a criterion on the size of the sought objects is used to define a threshold for stopping the convergence of the DA if its selected pixels are too few to correspond to such objects.

The DAs can be processed in parallel in different regions of the feature space, or even on different feature spaces. However, their convergence can only be formally proven in the general case when they are processed sequentially.

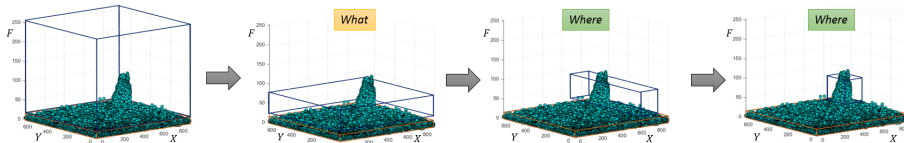


Fig. 1. Convergence steps of a “dynamic attractor” (DA) in the feature space. Example in the case of the detection of a tennis ball, using the hue and spatial coordinates.

This formalization brings forward several parameters that determine the detection process: the chosen feature space; the DA architecture; the size of the sought objects; the update mode; the detection threshold of each feature. Neither the choice of these parameters for a practical application nor their level of influence on the whole concept is made explicit by the inventor. Thus, in order to illustrate the behavior of the method and the possible influence of its parameters, we chose to apply it to a specific computer vision application.

3 Behavior analysis method

The road lane application is appropriate for the demonstration of the BIPS concept behavior, since the lanes are distinguished by their gradient angle and their position in the image. It is usually applied for autonomous driving, the existing solutions are based on edge detectors, Hough transform or RANSAC method [2][3][9]. For the BIPS concept, it allows to fix the architecture of the DAs, i.e. the choice of the input features with their order and the interaction mechanism between DAs through the binary validation map (Eq. 2), and to focus on the main parameters.

Each road lane can be described by its border lines which almost have (up to the perspective) the same gradient angle and are spatially close to each other. Thus, one lane can be detected by a DA working on the gradient angle feature and two spatial coordinates. The gradient angle is computed using the derivative Gaussian kernels. The chosen Gaussian kernel G has a standard deviation of 1.5 and a corresponding width of 9 pixels, to smooth the image and then to regularize the histograms. The polar transformation gives the gradient module and angle. A threshold is applied in order to validate only the pixels with a strong

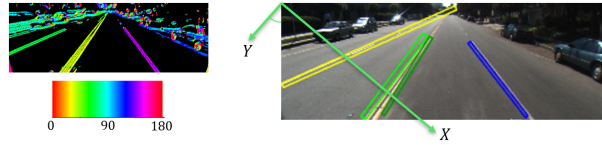


Fig. 2. On the left: the extracted gradient angle within $[0; 180]$, only the strong gradients are visualized. On the right: example of the rotated coordinates for the central road lane detection.

gradient. Then the two spatial coordinates can be rotated to obtain a narrower bounding of the lane. This choice implies a change in the DA formalization: the computation of the spatial features evolves during the iteration process. The rotation angle α_k is computed dynamically to better suit the direction of the detected lane. The chosen rotation put the axis Y along the lines and the axis X at 90° counterclockwise (Fig. 2). In that feature space, a line corresponds to one gradient angle and one position X , which can be detected using the mode 1 on these two features. Along the axis Y , the line will then be represented by a flat and low histogram, a strict threshold at zero is enough to detect this line.

A series of DAs as constructed here allows then to detect all the lines in the frame. However, in the context of road lane detection, we can use several assumptions to reduce the detected lines to only the road lanes.

1. The dashed lanes can be represented by one lane using the mode 2 on Y .
2. A lane is represented by two lines close to each other. The first detection can be used to obstruct the second using an extension of its borders.

The chosen architecture leaves several parameters free: the number of DAs n_{DA} ; the threshold on the size (in pixels) of the sought lines τ_N ; the threshold on the angle mode detection τ_{ang} and the threshold on the X mode detection τ_X . The two first parameters are somehow linked. In fact, the number of DAs needed corresponds to the number of objects sought, and the threshold on the size of the sought object determines this number. The first step of our experiment will be to find the relation between those two parameters. The mode detection thresholds τ_{ang} and τ_X can be chosen proportionally to the maximum of the computed histogram. The choice of a too low percentage brings noise to the selected mode, but the choice of a too high percentage may loose relevant information describing the lines. For the first experiments, we choose to set $\tau_{ang} = 30\%$ and $\tau_X = 50\%$.

4 Results on road lane detection

The experiments have been run on the *caltech-lanes* dataset. This dataset has been developed by the team of M.Aly [1]. It is composed of four sets of videos taken from a car. The clip contains several difficulties: curvatures, writing on the street, sun facing, strong shadows, passing cars, etc. Moreover, the ground truth concentrates only on visible road lanes from the car and provides a 3rd



Fig. 3. Challenging images from the caltech-lanes dataset and their ground truth.

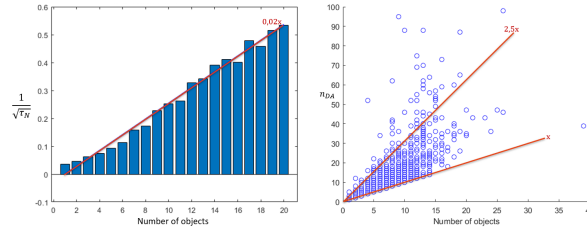


Fig. 4. On the left, linear regression made on the observation of $\frac{1}{\sqrt{\tau_N}}$ and the number of sought objects. On the right, observation of the necessary number of DAs compared to the number of sought objects.

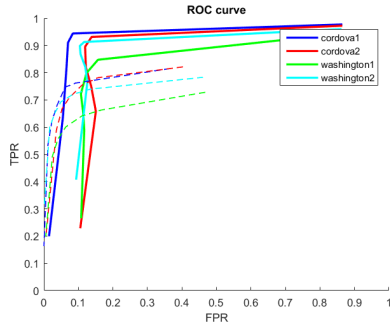
degree Bezier spline description of them, which may lead the BIPS detector to generate many false alarms (Fig. 3). To evaluate the output of the BIPS method, we compute the 1st degree spline s corresponding to the DA output and based on the two points: (X_{\max}, A_Y) and (X_{\max}, B_Y) . We use the same metric as Aly [1], i.e. the distance between the detected spline s and one ground truth spline s_{GT} .

$$\begin{aligned} d_{mean} &= \min \left(\underset{p \in s}{\text{mean}} (dist(p, s_{GT})) , \underset{p \in s_{GT}}{\text{mean}} (dist(p, s)) \right) \\ d_{median} &= \min \left(\underset{p \in s}{\text{median}} (dist(p, s_{GT})) , \underset{p \in s_{GT}}{\text{median}} (dist(p, s)) \right) \end{aligned} \quad (5)$$

The validation of the detection corresponds to $d_{mean} < 20$ and $d_{median} < 15$. Only one DA can validate the detection (the first one that converges). The others are considered as false positives.

Firstly, we run 100 DAs on each image of the dataset with a zero threshold on the size of the objects. The overall trend shows that the first engaged DAs are converging to the bigger objects whereas the following ones converge to smaller objects up to the noise level. The number of DAs necessary to detect all the sought object is then proportional to the number of objects. The study of this value on the four sets shows that the proportion is between 1 and 2.5 (Fig.4-right). In parallel, we can use the set of observation to estimate the relation between the number of objects and the threshold τ_N . The four image sets give the same relation founded by linear regression (Fig.4-left). The final relation between the number of DAs necessary and the threshold τ_N is:

$$n_{DA} = \frac{2.5}{0.02\sqrt{\tau_N}} \quad (6)$$



Clip	correct rate		false pos. rate	
	[Aly]	[BIPS]	[Aly]	[BIPS]
cordova1	91,62%	91,06%	5,66%	6,99%
cordova2	85,5%	89,55%	40,64%	11,95%
washington1	92,78%	79,89%	13,11%	12,35%
washington2	93,66%	89,86%	8,59%	10,42%

Fig. 5. On the left: ROC curves obtained with the four datasets using the variation of the threshold τ_N . The plain curves are computed with the DA number automation (Eq. 6); the dashed curves are computed with a fixed number of DAs. On the right: performance comparison with the *caltech-lanes* reference.

The ROC curve Fig.5 is established using different values of the threshold τ_N . The use of the established relation (Eq. 6) allows to increase the area under the curve for each video. The inflexion point of the curve is for $\tau_N = 10$ and $n_{DA} = 39$; using these values we obtain results that are at the level of the reference.

5 Discussion and conclusive remarks

In order to assess the potential and the generalization capability of the BIPS concept, we have chosen to apply it on the road lane detection application. In this specific context indeed, the architecture of the DA can be fixed allowing to focus this study to its main internal parameters. We have established the relation between the needed number of DAs and the size of the sought road lanes on the basis of the *caltech-lanes* dataset.

This relation both automates the parametrization and significantly improves the obtained results on the *caltech-lanes* database, which are at the level of the reference. It optimizes the number of DAs engaged, thus minimizing the computation time and resources. Furthermore it stabilizes the rate of false positives, that remains however high due to the ground truth. In fact, the centered two lanes are considered as only one lane, but are mostly detected by two DAs, and the curved lines cannot be entirely detected by this DA configuration. The method used by M.Aly exploits several pre- and post-processings, whereas this study relies entirely on the BIPS method.

Now both the quality and the efficiency should be improved by integrating a prediction mechanism inside the dynamical process. This should be the main focus of our future work. We will also investigate the possibility of modifying the DA architecture in order to address a wider range of applications.

Acknowledgement

We would like to thank P. Pirim for its help in the understanding of the BIPS component and its extended possibilities, and for sharing its valuable knowledge.

References

1. Aly, M.: Real time detection of lane markers in urban streets. In: 2008 IEEE Intelligent Vehicles Symposium. pp. 7–12 (Jun 2008)
2. Bar Hillel, A., Lerner, R., Levi, D., Raz, G.: Recent progress in road and lane detection: a survey. *Machine Vision and Applications* **25**, 727–745 (Apr 2014)
3. Borkar, A., Hayes, M., Smith, M.T.: Robust lane detection and tracking with RANSAC and Kalman filter. In: 2009 16th IEEE International Conference on Image Processing (ICIP). pp. 3261–3264 (Nov 2009)
4. Ehsan, S., McDonald-Maier, K.D.: On-board vision processing for small UAVs: time to rethink strategy. In: NASA/ESA Conference on Adaptive Hardware and Systems. pp. 75–81 (Jul 2009)
5. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361. IEEE Computer Society, Washington, DC, USA (Jun 2012)
6. Hubel, D.H.: *Eye, Brain, and Vision* (Scientific American Library, No 22). W. H. Freeman, 2nd edn. (May 1995)
7. Jiongjiong, W., Tiangang, Z., Maolin, Q., Antao, D., Kui, C., Zhanli, W., Cheng, Z., Ming, M., Yan, Z., Silu, F., Lin, C.: Relationship between ventral stream for object vision and dorsal stream for spatial vision: An fMRI+ERP study. *Human Brain Mapping* **8**(4), 170–181 (1999)
8. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollr, P.: Microsoft COCO: Common Objects in COntext. In: European Conference in Computer Vision (ECCV). pp. 740–755. Springer International Publishing (May 2014)
9. Nieto, M., Salgado, L., Jaureguizar, F., Arrospe, J.: Robust multiple lane road modeling based on perspective analysis. In: IEEE International Conference on Image Processing. pp. 2396–2399 (Oct 2008)
10. Ota, K., Dao, M.S., Mezaris, V., de Natale, F.G.B.: Deep Learning for Mobile Multimedia: A Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* **13**, 34:1–34:22 (Jun 2017)
11. Pirim, P.: *Processeur de perception bio-inspiré : une approche neuromorphique*. Techniques de l'ingénieur - Innovations en électronique et optoélectronique (May 2015)
12. Pirim, P.: Perceptive Invariance and Associative Memory Between Perception and Semantic Representation USER a Universal SEMantic Representation Implemented in a System on Chip (SoC). In: Biomimetic and Biohybrid Systems. pp. 275–287. Lecture Notes in Computer Science, Springer, Cham (Jul 2016)
13. Bach-y Rita, P., Tyler, M.E., Kaczmarek, K.A.: Seeing with the Brain. *International Journal of Human-Computer Interaction* **15**(2), 285–295 (Apr 2003)
14. Sarrabezolles, L., Manzanera, A., Hueber, N., Perrot, M., Raymond, P.: Dual field combination for unmanned video surveillance. In: SPIE Defense and Commercial Sensing, Real-Time Image and Video Processing. vol. 10223. International Society for Optics and Photonics, Anaheim, CA (May 2017)