How Can a Robot Evaluate its own Behavior? A Neural Model for Self-Assessment
Adrien Jauffret, Caroline Grand, Nicolas Cuperlier, Philippe Gaussier, Philippe Tarroux

To cite this version:

HAL Id: hal-00828374
https://hal.archives-ouvertes.fr/hal-00828374
Submitted on 5 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
How Can a Robot Evaluate its own Behavior?  
A Neural Model for Self-Assessment

Adrien Jauffret, Caroline Grand, Nicolas Cuperlier, Philippe Gaussier and Philippe Tarroux  
Neurocybertic team  
ETIS laboratory, UMR 8051  
95000 Cergy Cedex, France.  
LIMSI laboratory, CNRS(UPR3251)  
91403 Orsay Cedex, France.  
adrien.jauffret@ensea.fr; caroline.grand@ensea.fr  
cuperlier@ensea.fr; gaussier@ensea.fr; philippe.tarroux@limsi.fr

Abstract—Allowing a robot to autonomously navigate wide and unknown environments not only requires a set of robust strategies to cope with miscellaneous situations, but also needs mechanisms of self-assessment for guiding learning and monitoring strategies. Monitoring strategies requires feedbacks on the behavior’s quality, from a given fitness system, to take correct decisions.

In this work, we focus on how violations of expectations of such fitness system can be detected. Following an incremental and bio-nimetic approach, we first present two different sensorimotor strategies our robot can use to navigate: a Place Cells based strategy and a road following strategy. Then, we present a neural architecture that may be able to evaluate both navigation strategies. This model is based on an online novelty detection algorithm using a neural predictor. This neural predictor learns contingencies between sensations and actions, giving the expected sensation based from the previous perception. Prediction error, coming from surprising events, provides a direct measure of the quality of the underlying sensorimotor contingencies involved.

We propose that this model might be a key structure toward self-assessment. We made several experiments that can account for such properties for both strategies.

I. INTRODUCTION

In this work, one of our goal is to provide biologically plausible models for robotic navigation. Following the concepts of bio-inspired robotic and a constructivist approach we present integrated robotic control architectures resulting from a close feedback loop between experiments on animals and robots. This leads to a better understanding of the mechanisms by which the brain processes spatial information.

In previous works, we developed a model of visual Place Cells that allows the robot to exhibit simple and robust behaviors using only few Place/Action associations [2], [3]. Since this strategy has been successfully tested in small environment, we met issues while trying to navigate larger and more complex ones (see Section II-A). We propose to add a second simple, efficient and biologically plausible road following strategy in order to overcome issues we met with the first one (see Section II-B). Finally, we propose a generic neural architecture able to evaluate both sensorimotor navigation strategies (see Section III). We wish this model could allow the system to regulate its strategies depending on their relevance and thus improve its autonomy.

Autonomy, in the field of robotics, is still an open and poorly defined problem, for which concepts remain to be invented. By autonomous, we mean a system able to develop its knowledge and to evaluate itself to decide whether its learning or behaviors are relevant or not according to the context. This means that not only does the robot need to code its knowledge but also the limits of its knowledge. This becomes all the more important in integrated robotic systems which have to make decisions based on observations drawn from a multitude of modalities [18]. Adding self-assessment capabilities to robots should be an interesting solution, mainly in social robotics where humans play a role in the cognitive development of the robot. It could allow the system to ask human for details about its task when needed as in the collaborative control system of Fong et al. [20]. The robot could also communicate its inability to improve its learning. Moreover, it should be able to detect problems by considering aspects of novelty in its predictions. The problem of self-assessment is then sensibly close to the class of novelty detection problems. Novelty detection is a commonly used technique to detect that an input differs in some respect from previous inputs. It is a useful ability for animals to recognize an unexpected perception that could be a potential predator or a possible victim [2]. It reduces the large amount of information received by the animal so that it can focus on unusual stimuli. A variety of novelty filters have been proposed where most of them work by learning a representation of a training set (containing only normal data), then trying to underline data that differ significantly from this training set. In the litterature, one can find different classes of methods such as statistical outlier detection, novelty detection with supervised neural networks, techniques based on self-organising map and gated dipole methods.

The standard approach to the problem of outlier detection [22], [23] is to estimate the unknown distribution $\mu$ of a set of $n$ independant random variables in order to be able to detect that a new input $X$ does not belong to the support
of $\mu$. In the same way, extreme value theory [24] focus on distributions of data that have abnormal values in the tails of the distribution that generates the data.

The first known adaptive novelty filter is that of Kohonen and Oja [25]. It proposes a pattern matching algorithm where new inputs are compared with the best-matching learned pattern, meaning that non-zero output is only seen for novel stimuli. But such mechanism is certainly not biologically plausible.

Supervised neural networks methods propose also novelty detection solutions by recognizing inputs that the classifier cannot categorize reliably. Such methods estimates kernel densities to compute novelty detection in the bayesian formalism [26], [27].

An other solution is given by gated dipole fields, first proposed by Grossberg [28], [29], then used to compare stimuli and model animal’s attention to novelty [32]. But systems based on gated dipoles cannot generalise to stimuli that do not have a dipole to represent them and cannot scale with the size of the dataset.

Self-organising networks also provide solutions to detect novelty using unsupervised learning [30] and particularly the so-called Adaptive Resonance Theory (ART) [35] network that uses a fixed vigilence threshold to add new nodes whenever none of the current categories represents the data. In a sense, the process of the ART network is a form of novelty detection depending on a vigilence threshold.

Neural models of memory can also detect prediction error by learning sequences of Place Cells that provides a simple means of representing pathways through the environment [36].

More recently, a model of artificial curiosity proposed by Kaplan & Oudeyer [37] allows an agent to focus on novel stimuli to improve its learning in challenging situations, avoiding well known and totally unknown ones.

In the following we present a navigation strategy based on our previous model of visual Place Cells and its limitation for navigating complex environments.

II. FROM TWO SENSORIMOTOR NAVIGATION STRATEGIES

A. A model of Place Cells to perform sensorimotor navigation

In previous works, we developed a biologically plausible model of the hippocampus in order to obtain visual Place Cells (VPCs) [1] that allowed controlling mobile robots for visual navigation tasks [2], [3]. The embedded pan-tilt camera allows the capture of 15 images over a 360 degrees panorama. Gradient images convolved with a difference-of-gaussian filter allows to highlight a set of salient points in the scene. A log polar transform of local views of $16 \times 16$ pixels centered on each focus point is computed in order to improve pattern recognition against small rotations and scale variations.

Local views correspond to the "what" information coded in the postrhinal cortex or in other areas of the ventral visual pathway of the rat temporal cortex [5]. The azimuth of these local views (the "where" information) is provided by the parietal cortex through the parahippocampal region. The merging of “what” and “where” information may be performed in the superficial layer of the entorhinal cortex or in the postrhinal cortex [7], [6]. A neural network learns to categorize constellations of landmarks and their azimuths in the scene (5 landmarks per images) (see Fig.1) to produce place cells. Activities of the different place cells depend on the recognition level of corresponding constellations. Robustness comes from the large number of local views extracted (75 per panorama) and the use of a competition between place cells (see [4] for more details).

A neural network learns to associate a particular PC with an action (a direction to follow in our case). This sensory-motor architecture (Per-Ac [8]) allows the system to learn robust behaviors.

The activity of the cells, even in outdoor conditions, shows a peak for the learned locations (see Fig. 2) and generalizes quite correctly over large distance (2 to 3 m inside and 20 to 30 m outside). These broad activities are perhaps similar to the broad place cells formed in the Entorhinal Cortex.

Even if our architecture has been successfully tested in small sized environments (typically one room), our visual-only based mechanism shows limitations when trying to scale to larger and more complex ones (multi-room, outdoors). First the large number of Place Cells needed to cover this kind of environment introduces a computational problem that highly decreases the robustness of the localization. Moreover, we encounter some situations in outside environment where the large number of trees all around the system does not leave enough available landmarks to recognize (the entire panorama is full of green leaves that only represent noise for the system) and the only way to overcome such problem is to follow the road below. We propose to overcome these
issues by adding to our current architecture a biologically plausible road following strategy. Such strategy allows the robot to follow roads rather than learning Place Cells, in situations where it is neither necessary, nor efficient to do so. Providing two different strategies to the robot is not sufficient by itself to navigate autonomously. The system also needs a metalearning mechanism that evaluate both strategies to be able to select the right one in a given situation.

B. A simple algorithm to perform road following behavior

In this section we present a fast and robust biologically plausible road following strategy. Nowadays, several methods trying to solve this problem consist in extracting road boundaries. To detect boundaries, several methods have been proposed such as Laser-based methods [9], Radar-based [10], stereovision-based [11], color cue based [12], methods using Hough transform [13], steerable filters [14] and spline model [15]. All these methods are not suitable when the road does not exhibit significant markings or boundaries. Kong & al. [16] propose an other algorithm which allows to find vanishing points and their confidence level using road segmentation techniques. But such method requires Gabor filters to extract textural features and road boundaries.

Here, we propose a simpler but efficient method that could be biologically plausible. Our algorithm consists in finding, among N (fixed number) potentially vanishing points, the one that is the best candidate. As an example, if five potentially vanishing points are proposed, there will be one in the center of the image, one on the right (resp. left) and extreme right (resp. left). Then, the robot will orient itself towards the best potential vanishing point. To find the best potential vanishing point, our method consists in determining four preferred orientations for the gradient of each pixel in the image.

The algorithm is the following one :

1. acquisition of a low resolution image (160 x 120 pixels) by a wide field camera (90 degrees field of view).
2. edge extraction and gradient transformation.
3. selection of the potential vanishing points: division of the image in N bins over the horizontal axis. Each point is placed in the center of a bin. A fixed parameter defined the skyline elevation while there is no automatic skyline detection at the moment (in following experiments, this value is set at 20% from the top of the image height).
4. Four preferred directions are choosen for each vanishing point (see Fig. 3)

- Each vanishing point \( V_n \) is then coded by a neuron \( ACT(n) \) which activity is done by the following equations :

\[
ACT(n) = \text{dir}(n, 1) \times \text{dir}(n, 2) + \text{dir}(n, 3) \times \text{dir}(n, 4)
\]

(1)

\[
\text{dir}(n, i) = \sum_{i=1}^{N} f(\cos(\theta_{vp}(i, n, l))) \times (1 - \left| \cos(\theta_g(n, l)) \right|)
\]

(2)

with :

1. \( f(x) = \begin{cases} 
x & \text{for } 0.9 < x < 1 \\
0 & \text{for } x < 0.9 \\
1 & \text{for } x > 1
\end{cases} \). The value of 0.9 have been choosen to allow an error of 25° on each preferred direction.

2. \( \theta_{vp}(i, n, l) \) corresponds to angle between a preferred direction and the vector from pixel to potential vanishing point.
3) \( \theta_g(n, l) \) corresponds to angle between the vector from pixel to vanishing point and the vector corresponding to gradient orientation

4) \( \text{dir}(n,1) \) is the preferred directions top-right of neuron \( n \), \( \text{dir}(n,2) \) top-left, \( \text{dir}(n,3) \) bottom-left and \( \text{dir}(n,4) \) bottom-right.

- Then, a simple winner takes all competition selects the best candidate between the \( N \) neurons.

The motor control of our model is directly inspired by control theories of Braitenberg vehicles [19]. This control is quite simple : when a vanishing point is detected on the right (resp. left), the robot will turn right (resp. left). Convergent behavior emerges from sensorimotor interactions between the system and its environment, without any need for an internal representation of the environment, or inference. Consequently, angular precision is less important than sample rate in such control.

We tested this algorithm on real images of road (See Fig.4) in several situations.

A - Centered vanishing point
B - Centered vanishing point
C - Vanishing point on the left
D - Vanishing point on the right

Fig. 4. Results obtained with our vanishing points detection algorithm on real images. For each cases: Up – Real image of a road. Down – Activity levels of 41 neurons, each one firing for a particular vanishing point on the image. A – In a simple case (road with boundaries), the vanishing point is well detected in the center of the image and generalized quite correctly to neighborhood B – Without boundaries, the vanishing point remain salient since there is a significant gradient between road and grass. C– For a twisting road, our algorithm detect two potential vanishing points, a centered one and the other on the left side. Nevertheless, the more active is the one on the left. D – Example of a vanishing point detected on the right side.

Thanks to this method, our system is able to follow any types of vanishing points such as roads, corridors, paths or railways. Furthermore, this algorithm has a satisfying framerate of 20 images per seconds (for 41 vanishing point neurons) and this framerate could increase if we consider less neurons. Such high framerate is obtained because only the higher gradients are considered in our algorithm. Therefore, the intensity of the gradient have been normalized by using the cosinus of the angle. So, in Fig.4 (case B) the gradient of road edges is not really high even though the vanishing point is detected.

A drawback of our method is the adjustment of the skyline position. Moreover, in some environments, the vanishing lines over the skyline could be an information (like in a corridor or in forest) although high relieves or clouds that can disturb localization of the vanishing point.

III. TOWARDS A FIRST METHOD FOR SELF-MONITORING

Here we present a generic model for self-assessment based on novelty detection techniques. Our model consists in two steps. First, to learn sensorimotor contingencies induced by the navigation strategy involved in a normal situation (training set), second to be able to detect extraneous sensorimotor patterns in novel situations.

Our model is as follow (see Fig.5):

Let's denote \( y \), a vector of \( n \) neurons \( y_i \) relative to agent’s own sensations. It can be both place cells or vanishing point cells in our case. \( y \) can be viewed as a set of random variable \( y_i \).

\( x \) is a vector of neurons \( x_i \) relative to agent’s proprioception, where the winning neuron code for the current orientation.

\( P \) is the tensorial product between \( x \) and \( y \) with recurrent connections of weight \( \alpha \). \( P \) codes a short term memory of the agent’s perception, where \( P_i, j \) denote the particular tuple of both \( x_i \) and \( y_j \) neurons.

Here we defined Perception as the integral of all Sensations and Actions [31]:

\[
\text{Per}(t) = \int_{-\infty}^{t} \text{Sen}(t) \cdot \text{Ac}(t)
\]

(3)

Where \( \text{Sen} \) denotes a vector of sensations (the input vector \( y \) in our case), and \( \text{Ac} \) denotes a vector of actions (the proprioceptive vector \( x \) coding for the current orientation). In our model, the matrix \( P \) estimates the robot’s perception by integrating sensations and actions in a finite shifting temporal window defined by the recurrent weight \( \alpha \):

\[
P(t + 1) = \alpha . P(t) + (1 - \alpha) . \text{Sen}(t) . \text{Ac}(t)
\]

(4)

\[
P(t) = \sum_{i=0}^{t} \alpha^{t-i-1} . (1 - \alpha) . \text{Sen}(t) . \text{Ac}(t)
\]

(5)

Basically, it means that recent inputs have a higher weight in our process than older ones. This type of filter has been tested by Richelieu and Manzanera [17] in a motion detection context. The parameter \( \alpha \) is used in order to attach more importance to the near past than to the far past.
Then, a vector $\hat{y}$, same size as $y$, estimates the mean $E[y]$ of the current sensation $y$ from the perception matrix $P$ by a least mean square algorithm (LMS) [33]. We make the assumption that $y$ follows a Gaussian distribution required by least-squares. An absolute difference between $y$ and $\hat{y}$ represents the instant error vector $e$. In the same manner, a vector $\hat{e}$, estimates the first moment about the mean, $\mu_2 = E[e]$, of the current error $e = y - \hat{y}$ from the perception matrix $P$ by a LMS algorithm. The second order error is defined as $e_2 = e - \mu_2$. The second moment about the mean is defined as $\mu_3 = E[(e - \mu_2)^2]$

The third order error is defined as $e_3 = e_2 - \mu_3$. Novelty $N$ is defined as the global fourth moment about the mean. $N$ is a single neuron that integrates all $e_3$ neurons activities: $N = |\sum_{i=1}^{n} e_3|$. $N$ is summarized by:

$$N = E[|\sum_{i=1}^{n} (|y_i - E[y_i]| - E[|y_i - E[y_i]|]) - E[|y_i - E[y_i]| - E[|y_i - E[y_i]|]|]$$

(6)

$N$ represent the prediction error of the network, that will be used to detect unexpected events. The different moments about the mean $\mu_2$, $\mu_3$ and $\mu_4$ represent respectively the pseudo-variance, the pseudo-skewness and the pseudo-kurtosis (while their measure follows the L1-norm rather than the L2-norm).

Our architecture is thus able to learn an internal model of the dynamical interactions the system has with the external world.

IV. EXPERIMENTS AND RESULTS.

We have tested our model of self-evaluation in several situations for both strategies. Following experiments run in 2 different 3-dimensional simulated environments. We used a simulated robotic platform that is 40*40cm wide and equipped with 2 wheels, proximity sensors for obstacle avoidance and a pan-tilt camera used to extract points of interest in the visual panorama.

The place/action strategy has been tested in a simulated room (Setup 1: see Fig.6) of 15*15m with a uniform floor and salient landmarks on walls. The robot is trained by an human teacher (supervised learning) to perform a round path by learning Place Cell/Action associations. As a stereotypical human/dog training interaction, the teacher uses a leash to pull the robot in the desired direction. Thus, the robot is detecting prediction error by comparing human order to its own will. This novelty detection neuromodulates the vigilance of the system so that it decides to recruit a new place cell and learns the association to its current orientation. Following such interactions, the robot is able to learn the path the human is teaching. A prescriptive learning (correcting the system rather than showing it the path) is necessary to get a stable and robust attraction field [34]. No more than 8 place/action associations are sufficient to perform a robust round trip in our experiment. The evaluation mechanism learns the sensorimotor contingencies while the robot performs its round trip in a normal situation (similar to the training set). In this setup, the vector of sensation $y$ is defined by the vector of 8 Place Cells learned by the system. We set the recurrent weight $\alpha$=0.95 empirically, based on the frequency of changes in sensations. The sensorimotor loop of that strategy is quite slow since states only change when the robot navigates from one place to an other (it mainly depends on the distance between 2 places and the robot’s linear velocity). Indeed, an $\alpha$ near to 1 results in a long temporal window (old states are more important than recent ones). 3 laps were necessary for the evaluation mechanism to completely predict its sensation from all sensorimotor situations perceived during the trip. Indeed, learning is completed only when the novelty level reaches a minimum (typically below 0.4) and stagnates in all places.

The road following strategy has been tested in a simulated outside environment (Setup 2: see Fig.6) of 40*40m with white roads over grass on the floor and trees texture on walls. The system is then able to correctly follow roads when one is in its field of view. On the other side, navigating on grass results in random movements since there is no stable and well-defined vanishing point to follow. The evaluation mechanism learns the sensorimotor contingencies while the robot performs road following in a normal situation (training set). In this setup, the vector of sensation $y$ is defined by 13 vanishing point neurons processed by the system. We set the recurrent weight $\alpha$=0.7 empirically, based on the velocity of the sensorimotor loop. Indeed, the sensorimotor loop of that strategy is significantly faster than for the place/action strategy while vanishing point states change at a speed that directly depends on the robot’s angular velocity. 5 minutes of navigation were necessary for the system to completely predict its sensation from sensorimotor situations perceived while following a road. Learning is completed when the novelty level reaches a minimum of 0.4 and stagnates.

A. Results relying on Place/Action experiments

After the system has completely learned the desired trajectory (Setup 1) and sensorimotor predictions relative to this trajectory, we tested it in several situations to show the
The novelty level shows a sudden offset when the system is avoiding the obstacle, since it predicts well its sensation while navigating on both roads, normal and abnormal sensorimotricity quite correctly since the evaluation mechanism is able to differentiate between the self-evaluation mechanism on a 1 hour navigation experiment. This time, the system evolves in the simulated outside environment previously described (Setup 2) and tries to follow the road (Setup 2). The purpose of this experiment is to test the evaluation mechanism in almost all different situations. Results show that the evaluation mechanism is able to differentiate between normal and abnormal sensorimotricity quite correctly since it predicts well its sensation while navigating on both roads, and detects novelty anywhere else (on grass).

V. DISCUSSION

In this incremental design approach, one objective of our robotic experiments is to show the limitations of models. Hence, since our previous works on modeling place cells
allows our system to perform simple sensorimotor behaviors in small size environments, we highlight the need to take into account solutions to overcome some issues we met while trying to navigate more complex ones. First, we extended our architecture by adding a robust and biologically plausible road following strategy. Such strategy allows the robot to naturally converge into visible roads, following it on a robust manner. It provide a second strategy to navigate in situations where learning Place Cells is neither necessary, nor efficient. Then, as using two different strategies is not sufficient by itself to navigate autonomously, we showed that the system also needs a mechanism that evaluate both strategies in order to regulate these strategies. We proposed a model for self-assessment based on statistical outlier detections in a dynamic point of view. We defined perception as an internal model of the sensorimotor interactions the system has with the external world. Such model of perception provide a generic grounding to perform predictions on agent’s sensation. Novelty is then defined as the prediction error at an n’th level, depending on the desired accuracy. Results show that our model presents good generalization capabilities since the same architecture can work for at least two different sensorimotor strategies.

Current work focuses on using this self-assessment mechanism as an active judge to monitor (select or inhibit) different strategies in particular contexts. The mechanism should therefore select useful strategies, while discarding others, based on its own evaluation of a given behavior in a specific context. An emotional controller (involving basic emotions like frustration and satisfaction), will be necessary to take decisions. Moreover, the robot may stop and call for help if no relevant strategies are found (if switching strategy does not increase any progress at all). We will evaluate the model performance on long range outside experiments (navigating several kilometers) on a real outdoor robot.
ACKNOWLEDGMENT

We would like to thank AUTO-EVAL, the NEUROBOT French ANR project. This project was also supported by the Centre National de la Recherche Scientifique (CNRS) and the Delegation Generale pour l’Armement (DGA).

REFERENCES


