Learning to build visual categories from Perception-Action associations

C. Joulain, P. Gaussier, A. Revel, B. Gas
ENSEA ETIS, 6 Av du Ponceau
95014 Cergy Pontoise Cedex, France
e-mail: gaussier or joulain@ensea.fr

Abstract

In this paper we describe how a mobile robot can autonomously learn and "recognize" simple objects present somewhere in an indoor visual scene. The experiment consists in transposing a classical conditioning experiment on a mobile robot. We propose the use of a selective attention mechanism to reduce the amount of computation involved by the complete image analysis. Objects are categorized according to their associated actions that are learned in accordance with a reward/punishment procedure. Our approach emphasizes the importance of a movement reflex mechanism based on the use of the same egocentric representation from the visual information to the motor output. Finally we highlight the impact of information coding in self organised topological maps on the robot performances.

1 Introduction

After decades of work in the field of autonomous robots, a lot of results have been obtained concerning indoor as well as outdoor robot navigation [3, 5]. However, everybody agrees on the lack of solution allowing robots to learn by themselves for real (on line learning).

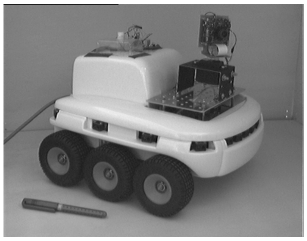

Figure 1: A prototype of the Koala robot

In our opinion, the main difficulty is linked to the "symbol grounding problem" [13]. We will show that an architecture for on line learning can be achieved by associating several features, namely: a self organized topological map, a neural architecture and a conditioning rule. We will illustrate this idea by an experiment which is the transposition of a classical instrumental conditioning (the robot must learn to react according to a reward/punishment procedure). More precisely, we present a right or left arrow in front of the robot camera and we want it to associate them with particular movements such as "turn 90 right" or "turn 90 left". Moreover, the robot must learn to go in the direction of a drawing representing a mushroom. The difficulty is that the relevant object is not alone in the scene. There are a lot of distractors, like doors, radiators, chairs that the robot must learn to discriminate from the objects to be learned. The robot must learn to build a "model" of the objects before using them. Moreover, even if we suppose the objects have been correctly learned, a global and invariant recognition method (like Fourier or Hough transforms) cannot be used directly because an "object" only represents a little part of the image (the result of a global analysis should be too noisy to be useful). For all those reasons, we have developed the PerAc (Perception - Action) architecture. In this architecture, we try to imitate the way information is computed in mammal brains. We separate the problem of recognizing "what" is present in the input flow and "where" are the relevant information [1, 6, 7]. In the first part of the paper, we point out the problems due to the unsupervised association learning process between an object and a movement. We describe how and which information must be extracted from a CCD camera. We show the interest of a movement reflex mechanism based on the use of the same egocentric representation from the visual information to the motor output. Next, we propose a neural architecture which isolates possible objects and learns to associate them with the correct movement. Finally, we show experimental results and discuss the interest of the generalization capabilities of a topological map model which allows on line categorization of visual shapes.

2 Object recognition mechanism

In order to reduce the image complexity and to work with more usable information, we extract edges to obtain a binary image. The boundaries are defined by the local maxima of a Nagao gradient operator. We assume those boundaries to be the frontiers of the objects (see figure 2) or at least, to contain enough information to distinguish two objects having different effects in the environment (ecological approach of vision [12]). In order to choose the areas that could be the location of possible objects, a mechanism of image exploration is used. This mechanism allows the robot to focus its "attention" on a particular area (a local view). For sake of simplicity, we use a corner detector providing focus points (like in figure 2). It is a simple difference of gaussian filters applied on the contour image (DOG filters see [7]).

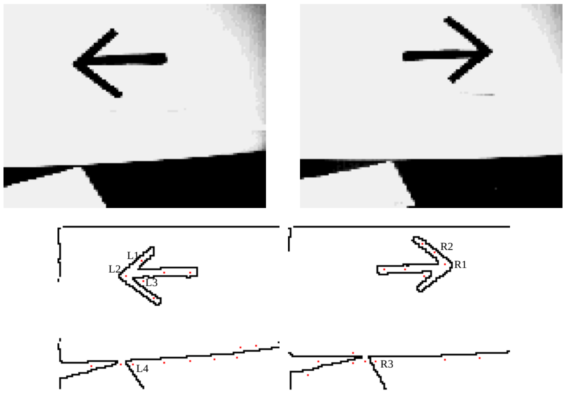

Figure 2: Grey level and contour image of a left and a right arrow. The labelled black points on the contour images represent explored focus points.

The resulting local view is then obviously independant of the object translations. Moreover, a log-polar $(log\rho, \theta)$ transformation which simulates the retinotopic projection of the retina on the primary cortical areas [6, 14] allows a weak scale and rotation invariance. This transformation gives more importance to the points close to the center of the focus of attention area. The resulting binary image is called the Visual Input (VI) of the Neural Network (N.N.).

The VI are stored on a Probabilistic Topological Map (PTM) [9]. It allows an immediate on-line learning and codes similar shapes on neighbor neurons (topological preservation). The neural map constitutes the Visual Output (VO) of the N.N. (see appendix A). A global term named vigilance controls its learning level and allows to select only the relevant local views. For instance, if a VI looks too much alike to a previously learned pattern (according to the vigilance parameter), then it is not stored. The vigilance parameter is controled by the reinforcement signal. If a punishment is emitted, then the vigilance must be increased to take into account the need of storing the "reason" why the punishment has appeared. Our vigilance parameter is equivalent to the one described in ART (Adaptive Resonance Theory [4]). For sake of simplicity, in the next paragraphs, the VO group will be used like a simple WTA (Winner Take All) or an ART model. Indeed, only a single neuron in VO will be active for each local view (VI).

3 Problem of action choice

In the next step, after the recognition of a local view, an action is performed. The mechanism is simple: several neurons are connected to VO, so that the activation of a local view involves an action. Each neuron in the Robot Motor Output group (RMO1) is devoted to a particular action: "turn left", "turn right" and "go ahead". As our robot only uses local views it has to focus on several points in order to explore the whole image. Moreover the exploration allows to propose several actions, but a robot can only perform one movement at a time. Then, only one action must be choosen. The addition of this sequential exploration is our major contribution to the PerAc architecture.

A first possibility to link a perception to a particular action, is to keep the action associated to the best recognized local view in the input image. Unfortunately, the "best" recognition in a visual scene can be something not relevant for the robot movements. For instance, on figure 2 the best recognized local view can be centered on the sheet of paper and not on the "right arrow". Indeed, during the recognition process, the robot focuses its attention on several feature points (the corners). It may recognize them or not, but at least, we want it to perform the movement associated to the recognition best linked to a movement. Thus, the choice of the winner must not be performed at the visual recognition group VO. Conversely, this choice must take into account the strength of the connections between the recognition and the action. The conditioning must be performed at the RMO1 group level. RMO1 represents the information of VO which is salient according to the motor aspect. In that purpose a Max operator is used to trigger a buffer group that stores the best proposed movement (see figure 4, the maximum is reset after each whole image exploration, i.e., after each robot movement). The N.N. mechanism involved in the image recognition is represented on figure 3. An example of movement decision is shown figure 4.

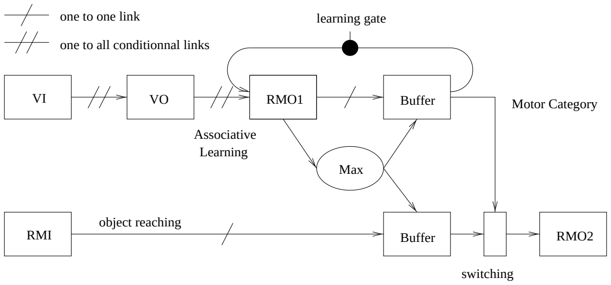

Figure 3: Simplified neural implementation. The robot first learns to build simple sensori-motor categories in RMO1 (Robot Movement Output 1) based on the recognition of visual shapes: VO (visual ouput). The reflex RMI indicates the movement to perform to reach the viewed object. Next, the robot uses RMO1 and the reflex to obtain RMO2

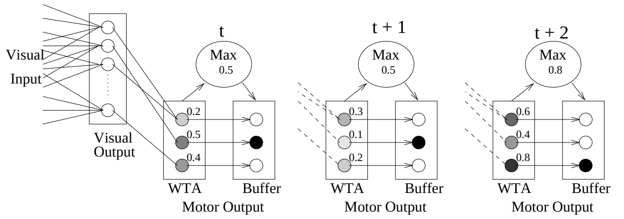

Figure 4: Mechanism of decision making during sequential exploration. The best associated movement is kept by the Max-Buffer operator

Thus, if the system has learned relevant sensori-motor associations between VO and RMO1, the recognition of the learned local view will imply the correct action. The presence of a distractor in the image will not involve a movement because the link between any action associated to a distractor will be weaker than those associated to relevant pictograms.

4 Associations Learning

The learning strategy assumes, at first, that all the explored local views in the perceived image can be relevant for the sensori-motor associations to be learned. The winner neuron in the motor group can simply learn to associate or dissociate all the local views from its action according to the sign of the reinforcement signal. A first problem is that neurons in the motor group must learn according to the effective robot action (i.e., the action proposed in the buffer group at the end of the exploration sequence). Otherwise, learning would have no sense since the reinforcement signal would not be associated to the finally involved neuron. Thus, at the end of the exploration sequence, when the action has been performed, the result of the buffer is forced on the neurons of the RMO1 WTA by opening the learning gate (see figure 3). Next, when the robot receives the reinforcement signal, the integrated input (the different visual recognitions -VO- created during the exploration of the focus points) and the effective ouput (the max of RMO1 which is in the buffer) are at its disposal.

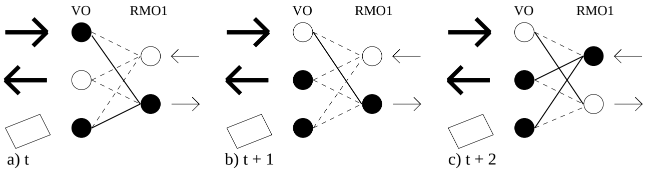

Figure 5: Example of sensori-motor links evolution during learning, a) t: correct association of the "right arrow", b) t+1: bad association for the "left arrow", c) t+2: the robot succeeds in performing the good association. The dotted lines represent the inhibitory links and the solid lines the activatory links.

4.1 Example of PerAc learning

On figure 5, we can see the evolution of learning for two visual scenes (see figure 2) where one distractor is present (a sheet of paper in this example). The problem is that the sheet of paper is not a relevant shape. The only effective views supposed to imply actions are the two arrows. The associative learning is performed between VO and RMO1. First, the "turn right" arrow and the "sheet of paper" are presented to the robot. By chance, the robot performs the correct action. It receives a positive reward so the "turn right" arrow and the distractor are both associated to the "turn right" action (see figure 5 a). Second, we present the "turn-left" arrow and the "sheet of paper", no association had been learned between an action and the "turn left" arrow. So, because of the distractor, the resulting action is "turn right". A punishment is emitted, the distractor is dissociated from the "turn right" action (see figure 5 b). Third, the "turn left" arrow and the distractor are presented once again. There is no link between the action and the distractor or the "turn left" arrow. So the chosen action only depends on the weak output neuron noise. In our case, by chance, the "turn left" action will be selected. So, a reward is given to the robot and both "turn left" arrow and the "sheet of paper" are associated to the "turn left" action (see figure 5 c). At the end of the exploratory sequence, we are sure that a single action is performed but we cannot be sure it has been performed due to our interpretation of the problem (the supposed correct reason !).

4.2 Proof of the algorithm

In order to understand how the system finds relevant views, two main cases must be studied. First, there is no intersection between the set of local views associated to the first movement and the set associated to the second one. Both sets will end by being associated to the correct movement. The learning problem is trivial but the robot will never be able to separate the distractors from the "important" objects because there is no way to reduce the links between the distractors and the correct action. Second, there is an intersection between both sets of local views. A problem will arise for the learning of the second set because the links between the shared distractors of both sets have been associated to the first action (fig 5 a and b). Learning will reduce the weights associating the distractors to the first action (fig 5 b). A second action may also be selected and it will be linked to the whole second set of patterns (fig. 5 c).

5 Control of precise movements

In the previous section, we have described how to select one movement from 3 possible movements. But, to control the robot precisely, we must be able to do the same thing with several tens of possible actions. A first solution could consist in learning which action must be selected if an object is present in a particular area of the image. For instance, if the mushroom is 47 degrees left from the camera direction (in left part of the image a figure 6), the robot should learn to turn 47 degrees left. This method could be simply coded in a look up table. However, if the robot proposes a particular movement and receives a negative reinforcement, it will have trouble to know if the punishment comes from a movement orientation error or a bad choice of the relevant object (or both). In our case, there are 32 possible movements and 32 locations in the image and at least 3 objects to be learned (in fact the system learns more than 50 shapes ! - see section 6), so the system would have to test at least 32 × 32 × 3 possibilities in order to learn which recognition at a particular position must be associated to a particular action ! The method is correct but obviously unserviceable.

In our PerAc architecture, so as to simplify the learning problem, all the movements are computed relatively to the object position. It allows an invariant learning of the movement direction according to the object position. This mechanism is implemented through the reflex pathway which provides a rough hardwired behaviour used as a bootstrap mechanism for the learning system. The reflex system uses the geometrical information given by the focalisation mechanism (Robot Movement Input (RMI) on figure 7 - see [11]) as a movement proposition for RMO1. To do so the position of the focus point is projected on the **x** axis (egocentric representation). This position is coded by a set of neurons that have the same topology as the image. The first neurons represent the lower **x** (the left part of the image) and the last neurons represent the higher **x** (the right part of the image). Figure 6 shows which neuron is activated in the RMI group according to the position of the focus point in the image. The reflex should allow the robot to go in the direction of the object of interest.

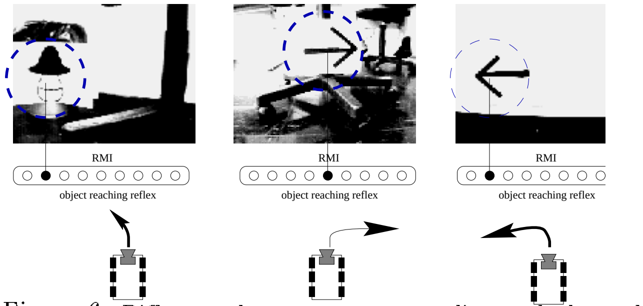

Figure 6: Different robot movements according to the learned sensory-motor associations and to the position of the object in the visual field. The circles represent the local area used by the VI group. The vertical lines indicate the projection of the focus points on RMI group to rotate the movement proposed by RMO1.

In fact, it would not be a good idea to use directly the RMI as a reflex input to the RMO1 group because the VO would be unable to control the RMO1 group easily. Indeed, RMO1 provides movement information corresponding only to "turn right or left" or "go ahead" while the RMI involves slight robot movements. They do not address the same kind of level. In fact they belong to two distinct levels: a high level corresponding to the recognition of a specific category and a low level corresponding to the object reaching problem. Moreover, if the robot is not in the same orientation than during learning, the recognition of a local snapshot allows the robot to perform a learned movement like "turn left" or "turn right" but the robot will be unable to learn a strategy like "follow this object". Indeed, the useful information is no more accessible in the VO group (the "what" and "where" information have already been separated by the focus of the attention). Thus, the RMI is used to rotate the result of the RMO1 group. The result is stored in the RMO2 group which corresponds to the real movement the robot must perform (figure 7). For instance, if an arrow involving a "turn left" action is presented at 10 degrees left of the image center, the robot turns 100 degrees left (90+10) and it moves forward a bit (like in figure 6).

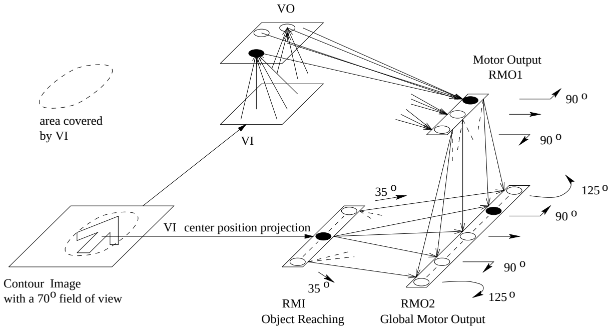

Figure 7: Details of the control of object reaching by a motor group that can choose between "turn left", "turn right" and "go straight ahead".

6 Experimental results

The architecture has been tested on our mobile robot. All the information come from a CCD camera (384 × 288). The digitalization resolution is 192 × 144 pixels gray scale. The camera visual field is about 70 degrees, so the first and the last neurons of the RMI group correspond to a 35 degrees angle respectively to the left and to the right. The extracted local view used as the Visual Input (VI) of N.N. is a 32×32 binary matrix. If an object in the image is subject to a 10 degrees rotation or if there is a scale variation of about 10 % the perceived image is exactly the same and the system cannot see any difference (exactly the same $(log \rho, \theta)$ image).

At the begining, the robot knows nothing, it has to learn, at the same time, the different shapes and their associations with a movement. In our experiment the different movements indicators are arrows and a mushroom, it could be anything else and we can have several different indicators for the same mouvement. The robot must turn according to the direction of the arrow and it must reach the mushroom. The indicators are drawing on a A4 sheet of paper ($21mm \times 29.7mm$) and the sheet is put in the robot visual field. The robot is brough back until it performs the good movement. It receives a punishment for a wrong movement and a reward for a good one. Each object needs about 2 or 3 presentations to be learned and then, the behavior seems to remain stable. The robot is able to recognize the object even if the distance is from two times shorter to 1.5 times longer. For instance, if "mushroom" is presented, the robot needs approximatly 3 trials to learn to go "straight ahead" in the direction of the mushroom. All those results come from real experiments on the robot.

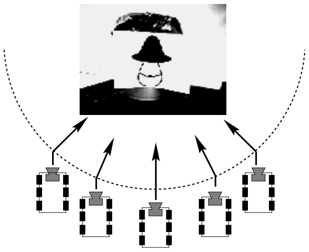

Figure 8: The simple "go ahead" order associated to the recognition of the mushroom allows to go always in the mushroom direction

The egocentric representation of the movements has also another interesting effect. As for the mushroom, the robot needs, in the worst case, 3 trials to associate a "turn right" arrow with the turn right movement. Then, whatever the position of the robot is (in the neighborhood of the arrow), the robot will tend to avoid the arrow area by turning right around it (see figure 9). Indeed, as soon as the robot is close enough to the learned position of the arrow, it will react by turning right but it will also add an angle more or less important according to the difficulty of avoiding it on the right. It is a very pleasant side effect of our algorithm.

7 Interest of the topological map

As noticed at the beginning of the paper the local views are learned on a topological map. The main interest of those maps is that two stimuli should produce close activity patterns in the map (figure 11) and close situations should be coded on nearby neurons. Unfortunatly, classical Kohonen maps need to separate learning and utilization phases and do not really allow on line and immediate learning. The Probabilistic topological Map (PTM) is an attempt to bring together features of a fast-learning algorithm and of a topology preserving map [10, 15, 9]. In order to obtain an immediate topology preserving map, we maintain a continuity on the map by allowing the immediate coding of the intersection of two learned patterns between their maximally responding cells. The winner neuron activity is diffused on the map to a maximum distance given by the diffusion radius value. Its action is meant to model the effect of lateral coupling of the winner with its neighbors: it makes them learn the input pattern, with an effect that decreases with distance. The neurons weights are adapted according to a probabilistic law that matches the diffusion function (the transformation of the PTM to a WTA was performed by reducing the diffusion radius of the PTM to a unitary distance). The generalization capabilities of PTM are very important. Typically, after the recognition process an image learned at a particular distance can be put two times nearer to the camera or 1.5 times farther from the camera without any change in the winner choice (it depends on the objects number and on the differences between objects). The PTM also allows a correct recognition even if the robot camera is +/- 50 degrees rotated from the orientation used during learning (see figure 8). The main interest of the topological map is to allow a good a priori generalization over the different learned shapes (two similar situations are coded on close neurons) and thus, to reduce the number of patterns that must be learned [15].

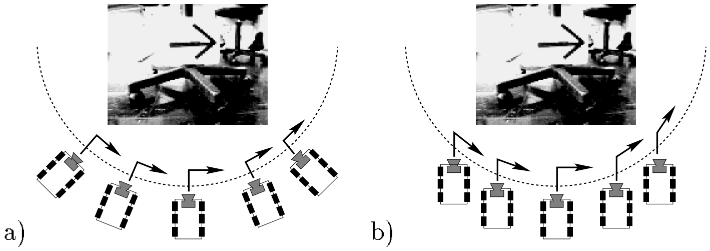

Figure 9: The simple "turn right" order associated to the recognition of the "turn right" arrow allows the robot to always avoid the "turn right" arrow by the left. The dotted circles represent the begining of the visual recognition of the arrow. a) The robot is always facing the arrow. b) The arrow appears at different positions in the CCD image.

The topology also brings a new element. When only one action is associated to different shapes, this action will always be activated (if there is no inhibition links on the map). As a matter of fact, a new shape is coded on a new neuron but due to the diffusion, the neurons in its neighborhood which have already learned something are also activated. Thus, the system cannot learn to associate the new shape with the action, because neurons in the neighborhood involve the action learned before. So, we also need to introduce inhibitory links between VO and RMO1 and thus each neuron in VO must have at the same time an activatory and inhibitory link. Those links can theoretically be strong at the same time but this situation cannot last because their probabilities evolution are exactly opposite. The Probabilistic Conditioning Rule (PCR) used to associate VO and RMO1 is described in appendix B. This association problem between a topological map and a WTA appears in the example figure 2. Only explored focus points are labeled. The Visual Input (VI) centered on points L1, L2, L3 are on the left arrows, they induce the "turn left" action. VI on R1, R2 have to imply "turn right" and VI on L4, R3 are not relevant (the labels will be used to design a visual pattern as well as the neuron coding the shape on VO). After several presentations when the robot performs the good action, the links between VO and RMO1 are as seen figure 10.

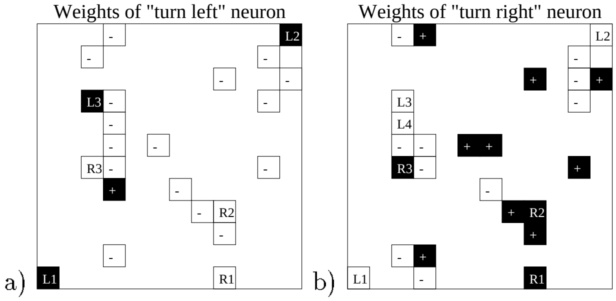

Figure 10: Representation of the motor links on the topological map. a) Links of the "turn left" neuron. b) Links of the "turn right" neuron. Black boxes correspond to activatory links and white boxes correspond to inhibitory links (no link elsewhere). Labels correspond to the position of neurons that recognized the local view shown figure 2

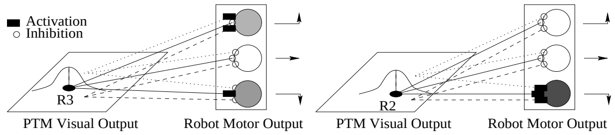

Figure 11: The topology diffusion, a movement can be involved by neurons in the neighbourhood of the winner.

In figure 2 b) the action "turn right" is involved by the VI on R2, as expected. To understand how it works we have to study the links (figure 10) and the topology (figure 11). When the robot learns to associate arrows, several VI are presented. A relevant VI for the "turn right" action (view centered on R2 or R1) will only be present in a "right arrow", so the links associated to R2 and to the neighbor shapes are the same. This can be seen on figure 10 a) and b). The labelled boxes represent the place where the shapes have been learned on the PTM. A white or black box indicates that the neuron at this position induces an inhibition or an activation to the associated motor neuron. A shape like the one represented by R2 is connected to the "turn right" action with an activatory link and to "turn left" action with an inhibitory link. But the most important is that the neurons in the neighborhood of the neuron coding R2 are connected to the "turn right" action with only positive links and to the "turn left" action with only negative links. So the activation of the "turn right" action will be higher than for the "turn left" action. Moreover, neurons in the neighborhood of a distractor like the sheet of paper (figure 2) coded on R3, are not solely linked to the "turn right"action with activatory links and also not solely linked to the "turn left" action with inhibitory links. Thus the contribution of those neurons to the activation of the "turn right" and "turn left" actions is negligible in comparison to the effect of the activity associated to a relevant shape like R2 (see figure 11).

Because each experiment needs about 10 min to be performed it is obviously impossible to test exhaustively the robustness of our stochastic PCR rule. So, from the real experiment, we have stored 6 images for the 3 differents cases to perform realistic simulations of the learning process. The indicator is on a dustbin in front of a radiator, so as the robot use edge detector for its perception there are several distractors in each image. Only in 2 of the 6 images the indicator is at the center of the field of view, for the rest, there are rotations and translations. For each image the robot explores 8 focus points so, there is a total of 144 (8 × 6 × 3) local views. In the simulation, there is no difference with the experimental process. The same image is presented until the movement is good. Images of the database are presented in a random order. When all the images have been shown a new random presentation is performed.

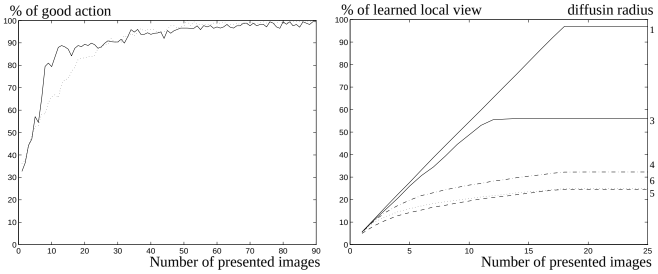

Figure 12: a) The rate of good action performed in function of the number of images presented (each image represents 8 local views). The dotted line corresponds to the case without topology (diffusion radius = 1), the full line is for a diffusion with radius = 3. b) Diffusion radius comparaison (the different curves). The pourcentage of views that are learned in function of time, the x axis is the number of presented images, performances are almost the same in time (not plotted).

In a first experiment, we only used the PTM as a WTA with 20 × 20 neurons (a vigilance of 0.7 and no diffusion : diffusion radius = 1) so, there is no benefit from the topology. For example, if we present 18 images (6 for each movement) with 8 focus points on each one, the network receives 144 local snapshots on the VI group. The WTA-PTM learns 140 differents views, the other ones where similar enough to an activated

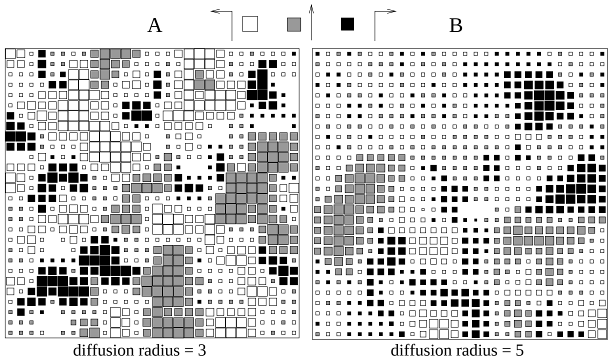

Figure 13: Each map represents the strength of a neuron to involve a special action, a) diffusion radius = 3, b) diffusion radius = 5.

one of the 140 learned neurons. Conversely, when the PTM is used with a diffusion radius of 5, the number of learned shapes is dramatically reduced (only 36 local views are learned - see figure 12) and the learning speed for the sensori motor association is higher [1] (see figure 12). Thus we can see that a priori generalization is efficient (the vigilance is still 0.7, see figure 13). To test the stability we have presented thousands of times the set of images. So we see that the learning is very stable, after 60 images presented the percentage of good actions is always greater than 95%. Even more, the percentage of good action is always increasing overall.

8 Conclusion

We have seen that an obstacle avoidance strategy "emerges" from the simple association of a single picture representing an obstacle and a particular movement. Moreover, we have shown that identification of the relevant information in an image is not really complex if the robot can use a mecanism to focus its attention on the different possible objects. This mechanism makes the robot "cognition" sequential and needs to be associated to an intermediate memory to come back to a parallel processing.

Another important conclusion is linked to the power of the topological representations and to the topology of the N.N. architecture. PTM introduces really interesting shortcuts in the learning process and allows to "take for free" information which are, most of the time, neglected in traditionnal IA systems.

Our different experiments with our Koala robot have also shown the interest of action to suppress the ambiguities or the errors. The present work is not a goal by itself. It is a first step toward the implementation of a real time navigation system that we have proposed in [11] and that we are currently testing. We hope it becomes now possible to really implement on autonomous robots, the concept of the ecological approach of vision proposed several decades ago by Gibson [12] to explain animals and humans visual capabilities.

A The PTM rule

The algorithm for the Probabilistic Toplogical Map is the following:

Present an input vector I to the map. Find the winner N^*, i.e. the map cell with highest activity. The similarity between I and the weight vector is first computed and the real neuron activity is processed trough an activation function f_k with variable selectivity $\tilde{D}_k(t)$. The activity in neuron N_k under presentation of binary input vector I is measured as follow:

$Act(N_k, I) = f_k[s_{input}(W_k, I)] + noise$, where if $\tilde{D}_k < 1$ then

$$s_{input} = \tfrac{1}{2} \cdot \left(\tfrac{1}{S} \sum_{\omega=1}^{P} V_k^\omega \cdot I_\omega + \frac{1}{P-S} \sum_{\omega=1}^{P} \overline{W}_k^\omega \cdot \overline{I}_\omega \right)$$

otherwise $$s_{input} = \frac{\sum_{\omega=1}^{P} V_k^\omega \cdot I_\omega + \sum_{\omega=1}^{P} \overline{W}_k^\omega \cdot \overline{I}_\omega}{\sum_{\omega=1}^{P} V_k^\omega + \sum_{\omega=1}^{P} \overline{W}_k^\omega}$$

where W_k is N_k's weight vector and $\overline{W}_k^\omega = 1 - W_k^\omega$, $\overline{I}_k^\omega = 1 - I_k^\omega$, $V_k^\omega = W_k^\omega + (\alpha - 1) \cdot \tilde{W}_k^\omega$,

$$f_k[x](t) = \tilde{D}_k(t) \cdot exp\left[-\tfrac{\nu+\epsilon}{2\sigma} \left(\tfrac{1-x}{1-\tilde{D}_k(t)+\epsilon} \right)^2 \right]$$

$\tilde{W}_k$ is the vector derived from W_k whose components $\tilde{W}_k^\omega$ are equal to 1 when W_k^ω has been reinforced more than once and 0 otherwise. P is the dimension of the input space and S is the presumed number of ones of the input vector (it is a constant value).

Diffuse the winner activity on the map according to a diffusion function.

$D_k(t) = D_k(N_k, N^*) = exp\left[-u \cdot d_{map}^2 (N^*, N_k) \right]$

If $D_k(t) > \tilde{D}_k(t)$ and $\tilde{D}_k(t) < \nu$ (the vigilance parameter) enable learning on the map: The N_k's selectivity parameter is modifed: $\tilde{D}_k(t+1) = D_k(N^*, N_k)$ If random $< D_k(t)$ then adapt N_k's weight W_k: $W_k^\omega(t+1) = I^\omega$. If $I^\omega = 1$ and $W_k^\omega(t) = 1$ then $\tilde{W}_k^\omega(t+1) = 1$ Else if the random draw fail $W_k^\omega(t+1) = W_k^\omega(t)$

In our experiment the different constants are: $\epsilon = 0.01$, $\sigma = 0.5$, $\nu = 0.7$, $P = 32 * 32 = 1024$, $S = 150$.

B The PCR Rule

In this section we only sum up the main caracteristics of our conditioning rule (see details in [8]). Its main interest is to allow one trial learning capabilities (all-or-none learning) and to provide pretty good results even in the case of a delayed reward. Our rule is a generalization of the Barto and Sutton [2] rule to the case of an environnement with objects not having the same appearance probability and with reinforcement signal which can be badly or not defined for a lot of situations. The PCR allows the robot to test hypotheses. Weights are not gradually changed. They stay at the same value during a time long enough to evaluate the interest of an hypothesis. The algorithm combines neural network

[1] examples of diffusion can be see on figure 11, i.e. a bubble constitued by a set of 9×9 neurons is activated with a maximum at its center.

associative capabilities and simulated annealing exploration methods.

Weights are only modificated by a probabilistic law when a reinforcement signal occurs. This law belongs to the same kind of rule we use for our probabilistic topological map. The PCR consists in using binary weights associated to a probability which measures the confidence on the weight value ($p \in [0,1]$). When the reinforcement signal varies, probabilities are updated and weights can be modified according to the confidence value. If there is no reinforcement variation, neither the probabilities nor the weights are modified, but information about the correlation between the input and the output of the weight go on being stored. To be able to modify confidence terms, a measure of input-output correlation must be stored. This term is updated at each time step. Three time integrated parameters are associated respectively to the input I_i, the output O_j and to the $input \cdot output$ product. Notations are: $\overline{I}$, $\overline{O}$ and $\overline{IO}$. We assume I_i belongs to $[0,1]$. The correlation is computed as $\mathcal{C} = \frac{\overline{\mathcal{IO}}}{\sqrt{\overline{\mathcal{I}} \cdot \overline{\mathcal{O}}}}$. $\mathcal{C}$ is an efficiency measure of correlation between input and output over a given span time. The algorithm is the following :

- $\overline{I_i}$,$\overline{O_j}$ and $\overline{IO_{ij}}$ are updated at each time step.
- If $\left|\frac{\partial P(t)}{\partial t}\right| > \xi$ and $\overline{I} \cdot \overline{O} \neq 0$ then the probabilities are updated as follows:
  $\Delta p_{ij}(t) = (\epsilon \cdot \overline{I_i} + \alpha \cdot \frac{\partial P}{\partial t}) \cdot \mathcal{C}_{ij} \cdot S\left(W_{ij}\right) - \lambda p_{ij}(t) \cdot \overline{I_i}$ and $p_{ij}(t+1) = p_{ij}(t) + \Delta p_{ij}(t)$.
- For the same conditions there is Random draw: if $Rnd > p_{ij}$ and $\overline{I} \cdot \overline{O} \neq 0$ then $W_{ij} = 1 - W_{ij}$ and $p_{ij} = 1 - p_{ij}$ and $\overline{IO}$, $\overline{I}$, $\overline{O}$ are reset to 0.

$P(t)$ is the global reinforcement signal. It represents a way to measure robot satisfaction over time. $S\left(x\right) = -1$ if $x = 0$ and $S\left(x\right) = 1$ if $x = 0$. α is the delayed conditioning learning rate. λ is the forgetting rate. ξ is a constant fixed by the experimenter. *Rnd* is a random value tossed in $[0,1]$.

In our case, we use PCR with analogic weights. The weight value is not the binary weight (W_{ij}), but $W_{ij} \cdot p_{ij}$. This product gives analogic weights corresponding to the confidence term. The ouput activity of a neuron is:
$O_j = max_{i=1}^{n}\left(W'_{ij} \cdot I_i\right) + noise$ with
$W'_{ij} = 1 + (2W_{ij} - 1) \cdot p_{ij}$

The noise can be as small as wanted. It is only used to allow a random choice when several neurons have the same output.

We also use the PCR for the inhibition links with little changes in the rules. First $W_{ij} \in \{0,1\}$ is changed to $W_{ij}^{-} \in \{-1,0\}$, $S(W_{ij}^{-})$ is changed to $S(-1) = -1$, $S(0) = 1$ and $W_{ij}^{-'} = 2W_{ij}^{-} + 1 - p_{ij}$. The rest of the adaptation rule is exactly the same. The ouput activity of a neuron become:
$O_j = max_{i=1}^{n}\left(W'_{ij} \cdot I_i\right) + min_{i=1}^{n}\left(W_{ij}^{-'} \cdot I_i\right) + noise$

The max is taken on the activatory links and the min on inhibitory links.

C Acknowledgements

We would like to thank I. Fijalkow and M. Quoy and S. Zrehen for their helpful comments. This work is supported by a GIS project (CNA43).

References

[1] I.A. Bachelder and A.M. Waxman. A view-based neurocomputational system for relational map-making and navigation in visual environments. *Robotics and Autonomous System*, 16(2-4), Dec 1995.

[2] A.G. Barto, R.S. Sutton, and C.J.C.H. Watkins. Learning and sequential decision making. Technical report, COINS-89-95, University of Massachusetts, Amherst, MA : Dept. of Computer and Information Science, Sep 1989.

[3] R.A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, R.A. 2(1):14–23, March 1986.

[4] G.A. Carpenter, S. Grossberg, and D.B. Rosen. Art2-a : an adaptive resonance algorithm for rapid category leaning and recognition. *Neural Networks*, 4:493–504, 1991.

[5] R. Chatila. Deliberation and reactivity in autonomous mobile robots. *Robotics and Autonomous System*, 16(2-4):197–211, Dec 1995.

[6] P. Gaussier. *Simulation d'un système visuel comprenant plusieurs aires corticales : Application à l'analyse de scènes.* PhD thesis, Université d'Orsay, 1992.

[7] P. Gaussier and J.P. Cocquerez. Simulation d'un système visuel comprenant plusieurs aires corticales : application à la reconnaissance de scènes complexes. *Traitement du Signal*, 8(6):441–466, 1992.

[8] P. Gaussier, A. Revel, C. Joulain, and S. Zrehen. Living in a partially structured environment: How to bypass the limitation of classical reinforcement techniques. *to appear in Robotics and Autonomous Systems*, 1997.

[9] P. Gaussier and S. Zrehen. The probabilistic topological map (ptm) : A self-organazing and fast learning neural map that preserves topology. *Accepted by Neural Networks.*

[10] P. Gaussier and S. Zrehen. A topological map for on-line learning : Emergence of obstacle avoidance. In *From Animals to Animats: SAB'94*, pages 282–290, Brighton, 1994. MIT Press.

[11] P. Gaussier and S. Zrehen. Perac: A neural architecture to control artificial animals. *Robotics and Autonomous System*, 16(2-4), December 1995.

[12] J.J. Gibson. *The ecological approach to visual perception.* Laurence Erlbaum Associates, Hillsdale, New Jersey / London, 1986.

[13] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[14] E.L. Schwartz. Spatial mapping in the primate sensory projection : analytic structure and relevance to perception. *Biological cybernetics*, 25:181–194, 1977.

[15] S. Zrehen and P. Gaussier. Why topological maps are useful for learning in an autonomous agent. In J.D. Nicoud and P. Gaussier, editors, *From perception to action conference*, Los Alamitos, CA, 1994. IEEE Press.