

## PerAc: A neural architecture to control artificial animals

Philippe Gaussier <sup>a,\*</sup>, Stéphane Zrehen <sup>b</sup>

<sup>a</sup> *ENSEA ETIS, 6 av du Ponceau, 95014 Cergy Pontoise Cedex, France*

<sup>b</sup> *Laboratoire de Microinformatique, EPFL-DI, CH-1015 Lausanne, Switzerland*

---

### Abstract

In this paper, we propose a new neural architecture called PerAc which is a systematic way to decompose the control of an autonomous robot in perception and action flows. The PerAc architecture is used for the simulation of a vision system with a moving eye and then for landmark-based navigation on a mobile robot to learn without any a priori symbolic representation.

*Keywords:* Sensory-motor loop; Vision; Navigation; Neural building block

---

### 1. Introduction

Our main goal is to show how biological data from neurobiology and behavioral ethology can help to imagine simple neural models that explain complex behaviours commonly observed on animals in their environment. From the engineer's point of view, this approach leads to a lot of questions. If the problem is for instance to allow a robot to get from one point to another or more simply to return to an interesting position (our application) why not "just" trying to find a good algorithm that solves that task? After years of underestimation of the difficulties arising from such problems, classical Artificial Intelligence (AI) has evolved and now proposes technical solutions for those tasks [1,2] that are very different from the first motivation of imitating human intelligence. These recent successes seem to be linked to the forsaking of the top-down approach,

in favour of more bottom-up architectures. For instance, multi agent systems rely on pre-emption mechanisms that solve conflicts when different modules lead to contradictory conclusions ([1], subsumption [3,4]). What allows those systems to run correctly is not always the soundness of their theoretical background but the engineers' pragmatism who succeed step by step to solve the problems arising from the connection of low level processing (such as robot calibration, image segmentation, matching techniques, ...) with their high level counterpart (focus of attention, object recognition, path planning ...). The symbol grounding problem [5] can sometimes be solved for well defined applications but the price to pay is the use of a lot of black boxes that contain an ad hoc and hidden expertise of the engineers for each problem encountered. All the expertise developed in robotics has brought a lot of specialized algorithms but the difficulty to link low and high levels remains in most cases. The main reason is that the structures and their associated functions are not always separated where they

---

\* E-mail: gaussier@ensea.fr or zrehen@di.epfl.ch.

should. But this is not surprising since there is no general theory for deciding what is the correct division of the world — in the hypothesis that such a concept has any value — see Ref. [6]. In the present theoretical desert concerning that problem, one may further notice that several different entities can be built to match the same architecture (Fig. 1). Moreover, there is no essential restriction on the type of language used for programming them: it can be either a knowledge based system written in any computer language, or a neural language, such as the one we will present in this paper.

This difficulty to link functional entities justifies the will to return to atomic code elements that cannot be divided into more simple functions. A formal neuron (Fig. 2) is a good candidate if it processes only local information and does not use hidden variables. Its interest is that it can perform a lot of different operations necessary for “intelligent behaviour”: elementary logical operations, basic pattern recognition, decision taking, space transformation (rotations, projections) [7] (for the model of the cortical column as an atomic element, see [8]). One can group such elements (hereafter designated as neurons) into functional boxes even if the same neuron can belong to several functional boxes. The links between boxes are, thus, the set of links between all the neurons of the two boxes. They can be of any type: binary, positive, real-valued or discrete-val-

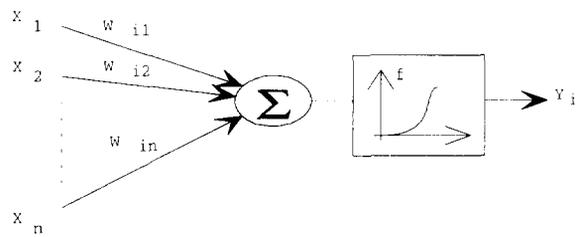


Fig. 2. A model of formal neuron.

ued. Thus, there is no more internal variables in a box made of neurons. Even if at the beginning the system is designed to take the information from the output neurons of a functional box and to use them as input for another functional box, neurons can learn links with hidden neurons belonging to another box. They will only take into account information according to their learning rules and so they will be able to modify the functionality of each box. If no input/output correlation appears between input and output neurons, then the link will remain weak. On the contrary, if a strong correlation appears, the computation is simplified by this kind of short cut in the normal “cognition” process. The possibility to have links between any neurons in the different functional boxes allows to redefine the system’s structure. Obviously, each possible link must be present from the beginning or must be created by a circuit between physically connected neurons.

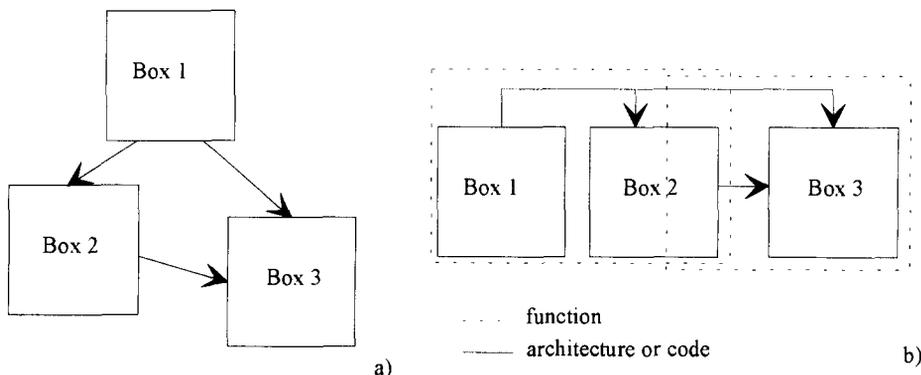


Fig. 1. The situation a) represents a top-down splitting of a problem in functional boxes. If during the functioning it appears that the box 3 needs internal information of box 2 there will be a problem. A neural implementation b) allows a splitting at the neuronal level and can solve the problem.

The lack of links between blocks will induce the same problems than people with brain lesions that cannot succeed in solving complete class of problems related to their disease (aphasy).

In our work, we put forward general principles for designing large but comprehensible N.N. that can play the role of a brain for a robot exhibiting complex behaviours and could be an interesting way to program new generations of parallel computers [9]. In particular, we will show that these biologically inspired systems are interesting from the computer science point of view. The exact same neural architecture will be used for complex visual scene recognition, and for landmark based navigation. We will also try to introduce the basis of a neural language and formalism. Such a framework is very important to point out the directions in which engineers must advance in order to build machines that overcome their current limitations. Most of current robots compute their actions from their perceived input by using models of their environment and are not able to imagine other models when they find themselves in unforeseen situations. They give pretty good results when their environment is adapted to their work but they are almost blind in a natural world. Too much data to analyze saturate the analysis capabilities of their “logical” brain. In the industrial domain too, each new model of arm manipulator needs to be modeled before it can be used to manipulate objects. Having them only learn their task like animals would certainly be a great achievement [10,11].

In this paper, we will present an autonomous mobile robot called Prometheus that can learn to return to an interesting place (its goal) in an unknown environment. The N.N. structure of Prometheus’ “brain” is based on the idea that Prometheus is intended to be in interaction with its environment, in accordance with the enactivist — or constructivist — paradigm [19–21]. Unlike an expert agent who knows how to reply to an arbitrary question about navigation problems or object manipulation, Prometheus is just an agent that learns to agree with its environment and its internal motivations. It has no global or complete representation of its world. It “keeps in memory” only what it has learned to act correctly in a

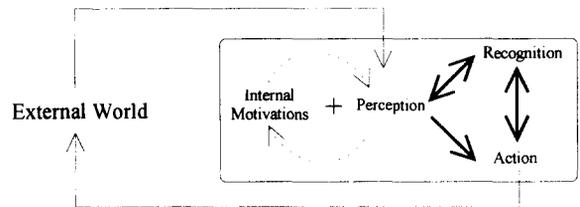


Fig. 3. The concept of PerAc architecture to control autonomous robots.

particular situation. Should the universe collapse, the robot’s memory would have no more meaning. For instance, a cognitive map involved in high level goal seeking or an object representation will be simulated in Prometheus with only few neurons in competition (representing a well chosen set of different places). This approach leads to a “from action to perception” scheme [22]. The concept of active perception will be applied at different levels in the paper to exploit the dynamics of the robot’s interactions with the environment (to simplify the robot’s task).

In the first part of the paper, we will present a neural architecture named PerAc (Perception-Action) (Fig. 3) which is inspired by studies by Albus [23], Brooks [3], Burnod [8], Carpenter and Grossberg [18], Edelman [24], and Hecht-Nielsen [25]. In a second part, we will show how the PerAc architecture can be used in the task of recognizing marks in a visual scene. Thus, we will describe the way the visual information can be used by another PerAc block to learn to return to a particular place. We will show that the problem of learning to recognize objects or scenes and to return to a previously discovered interesting location can be achieved with only two PerAc blocks push-pully connected. We will emphasize the importance of the choice of a coherent neural code that can be applied to code Prometheus’ eye saccades and the direction of Prometheus movements. Finally, we will conclude by showing how this architecture can be generalized to other tasks.

## 2. Basis of the PerAc model (Perception-Action)

Introspective reasoning gives rise to the intuition that the analysis of particular images is

easier to perform when we can build complex abstract data structure that represent the raw data. Indeed, if a system succeeds in extracting from an image “fundamental” information such as the presence of something with two legs, a body, two arms and a head, then its intelligent part will be able to deduce it is a man! This sort of representation is very interesting because it tries to be invariant for any kind of transformation of the original image [26]. Comparison between images becomes a graph matching problem but unfortunately, the problem of extracting information from raw data has been evaded. Animals have adopted quite different but nevertheless efficient solutions. They seem to use a “reasoning” procedure based more on image and memory. Indeed, a wide variety of insects but also mammals use directly snapshot information that they correlate with learned snapshot to take their decision. Moreover, the individual development of animals and humans seems to be based on relatively simple reflexes and conditioning mechanisms [27]. Previously conditioned or discovered behaviours become new reflexes on which new behaviours can settle. Thus, the neural codes associated to either low level (like obstacle avoidance) or more complex tasks (like navigation) must be compatible. The same goes for the association between different tasks of the same level. For instance, recognition of an object must be coded in such a way that it can help navigation.

### 2.1. Biological bases of the PerAc model

The ant is an interesting example of what a simple agent can succeed in doing in a social organization [28]. When studying collective behaviour, it is considered as a simple stochastic automaton. Nevertheless, the analysis of the behaviour of a single individual helps to understand some basic mechanisms that an animal must have in order to recognize an object or a place. For instance, the ant only uses direct visual information stored as a snapshot to retrieve a learned position. An experience illustrating that consists in placing a stick with two black rings around it just at the opening of the anthill [29] (Fig. 4). When the ants are accustomed to this object, if

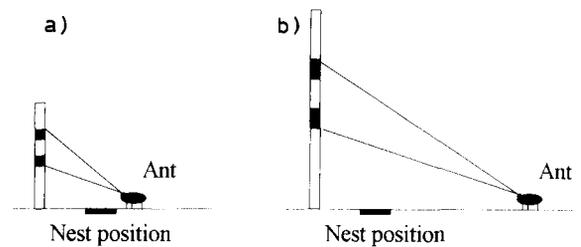


Fig. 4. (a) the ant learns the position of its anthill according to the visual aspect of a piece of wood. (b) the piece of wood is replaced by an object two times taller, the ant does not succeed in finding the nest, because it just tries to match the two perceived images [29].

the object is switched for an object twice as high, the ants will search the opening of the anthill just at the position that yields the same angular image. Thus, we can suppose that ants only try to find a perceived image that matches the stored snapshot. They just use a correlation method. A lot of works in psychology suggests that animals are able to use landmarks in their environment to locate themselves [30].

Snapshot recognition supposes the ability to locate correctly the matching mask on the current visual image. Obviously, Fourier transformation or other Gabor filters can provide position invariance for recognition but if several objects are present in the scene, overlapping problems appear in the parameter space and it becomes difficult to separate the signatures of the different objects. So high resolution recognition methods must also be used in the general case. The strategy apparently adopted by all the superior vertebrates consists in separating the recognition of an object (the WHAT problem) from finding its position (the WHERE problem). The temporal regions of the cerebral cortex are involved in the “what” pathway whereas the parietal regions try to find where the objects to analyze are [8,31,32]. The parietal system can then be regarded as an acting strategy to focus the attention of the system on a particular zone of the perceptive field. In conclusion, landmark based navigation or visual scene recognition are problems that can be divided in two subproblems: to recognize something in the perceptual flow and to learn to associate this recognition to a particular action

primarily proposed by a reflex acting flow. The concept of the Perception-Action system is now established. In the following section, we will see how to implement it simply as a neural network.

2.2. The PerAc (Perception-Action) block

The PerAc (Perception-Action) is a systematic neural structure that allows on-line learning of sensory-motor associations. It involves two data streams associated respectively to perception and action in each part of the robot controller (Fig. 5). From each perceived input, we suppose we can extract reflex information to control directly the robot action. There is also a mechanism for recognizing the sensory input patterns that can take control of the robot's actions and avoid the reflex pathway.

The neural boxes are competitive networks (Winner-Take-All or WTA). In such groups, only the neuron with maximal activation has a non-null activity after the competition is performed. They are used to code either the input vectors or the effector commands, as well as the "hidden" groups that can play the role of a memory. If  $V$  is an  $n$ -dimensional vector representing the activity of  $n$  neurons belonging to the same neural group

then we define the result of the competition between those  $n$  neurons as the vector  $V'$  of their activity after the competition process (to find the component of higher activity).

$$V' = \text{Max}^+ V \Leftrightarrow \forall i, i \in [0, n],$$

$$V'_i = \begin{cases} V_i & \text{if } V_i = \text{Max}_{j=1,n} V_j \text{ and } V_i > 0 \\ 0 & \text{otherwise} \end{cases}$$

The Reflex pathway: The action flow

Both the perceptual and the motor information are coded in egocentric coordinates. Each neuron in the motor groups corresponds to a particular movement orientation according to the current position of the considered system. In the same way, the visual input images are expressed in polar coordinates (as in the mammal visual system [33]). For instance, the ocular saccades of the robot's eye are represented as vectors associated to a grid of neurons that represents 32 orientations and 32 intensities of possible movements. The direction of the eye saccades is also expressed in the same coordinates. This simplifies the connectic problems of linking several neuron groups. Indeed, the retinal image directly pro-

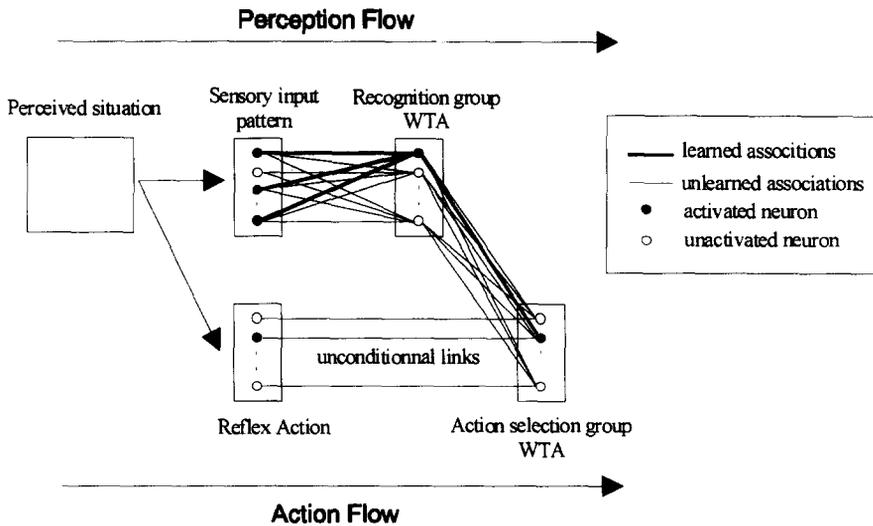


Fig. 5. Architecture of a PerAc Block. From the perceived situation a reflex action and a sensory input pattern are extracted. The action group learns to associate the recognized situation with the unconditional input. The system can then recognize a situation and react correctly even if the reflex mechanism is not activated.

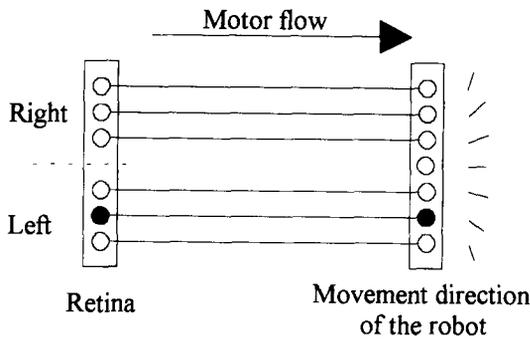


Fig. 6. representation of a reflex link in the motor flow of the PerAc architecture. When something is perceived on the retina the robot moves in that direction.

vides information for the activation of a particular saccadic eye movement in the retinal coordinates. This also makes possible the tracking of goals by the robot itself. Quantization precision is not really important because the use of probabilistic neurons that simulate neuron populations coding [34] allows to make movements with a precision that depends only on sampling time (see Appendix). Such a WTA group of neurons can be integrated simply in a reflex behaviour architecture such as those proposed by [3,11]. Simple reflexes can be easily constructed to control the ocular movement in the direction of “something” in the retinal image. In the same fashion, they can force the robot to move in the direction of that “thing” when it has recognized it as its goal (Fig. 6).

#### *The internal representation of perceptive information*

Because of the choice of the reflex structure, we cannot directly connect sensorial information to the action neurons. These latter neurons must compute a kind of logical “or” operation between their input: a movement must be performed if the reflex link is activated or if a pattern “A” or a pattern “B” ... is recognized. Unfortunately, the recognition of a pattern “A” for instance is a kind of “and” operation between the learned shape and the current visual input. Indeed, to recognize “A” the neuron must be sure that the first element of “A” is the same as the stored element and that the second element is also

correct and so on. It has been demonstrated by Minsky and Papert [12] that the same neuron cannot compute both “and” and “or” operations to perform any kind of logical equation. Therefore, we have introduced an unsupervised neural group to learn to recognize the perceptual situations. It is a self-organized and fast learning array of neurons that preserves locally the topology of its input. It is called the Probabilistic Topological Map (PTM) because the weights are adapted according to a probabilistic mechanism [13–15]. It allows having an a priori generalization for the new shapes coded on the map. If a new shape “A\*” similar to a previously learned shape “A” must be learned, it will be coded in the neighbourhood of the neuron coding “A”. Then the lateral diffusion of the activity of the neuron coding “A\*” will be sufficient to lead the motor action associated to “A” to be activated. For matters of simplicity and lack of space, we will not detail the PTM in the following (details about the interest of analogical and topological coding can be found in [14,16,17]). In Appendix A.2, the reader will find a simplified version of the PTM algorithm. It is a WTA model that uses a vigilance parameter to decide about learning a new shape (the algorithm can be replaced in the architecture by a classical ART-1 model [18]). The neurons compute the matching between their weight vector and the input data (point to point correlation). Then, a competitive mechanism lets find the winner neuron. The output vector of the neuron group  $Y$  is then defined by:

$$Y = \text{Max } f(W \cdot X)$$

$$\Leftrightarrow Y_i = \begin{cases} f(\vec{W}_j \cdot \vec{X}) & \text{if } f(\vec{W}_j \cdot \vec{X}) = \text{Max}_p f(\vec{W}_p \cdot \vec{X}) \\ & \text{and } f(\vec{W}_j \cdot \vec{X}) > 0 \\ 0 & \text{otherwise} \end{cases}$$

$X$  and  $W_j$  have the same size.

The vigilance parameter appears in the  $f$  function. If the vigilance is high each presented shape will tend to be considered as a new prototype of a new class. Conversely, if the vigilance is low, a neuron will learn a new shape only if it is significantly different from the previously learned shapes.

*The conditioning mechanism and the action group*

Learning associations between the recognition of a particular shape and the realization of a particular action is conditioned by a reinforcement signal that represents the internal motivations of the robot. A positive reinforcement is associated to the "pleasure" arising from solving a particular goal whereas a negative reinforcement will be emitted when the robot collides in a wall ("pain" signal) for instance. The pain signal provokes an increase of the random activity of the neurons, which allows the robot to quickly escape reflex solutions and to explore the whole action possibilities (for more efficient algorithms see [35]). In such a phase, the robot seems to be really stressed like a rat in a Skinner box when electric shocks are used to force it to discover and to learn a particular behaviour [36]. In the same way, pleasure increases the robot vigilance and allows it to learn what seems to have been the cause of the pleasure signal [13,37,38]. The reinforcement mechanisms also allows to modify the synaptic connections of the neurons in the action group. Their output is not the result of a weighted sum between input and weight vectors but a Max operator. Indeed, the weighted sum of several small input activities can produce a higher response than a strong well defined input activation and then involve an incorrect action. The output vector  $Y$  of the WTA action group is then defined as follow:

$$Y = \text{Max}([A] \cdot X)^+ - \text{Max}(-[A] \cdot X)^+ + I_0,$$

where  $[A]$  is the weight matrix of the Action group and  $I_0$  a constant vector that allows desinhibition.  $([A] \cdot X)^+$  represents the positive contributions while  $(-[A] \cdot X)^+$  represents the negative contributions. The matrix of the synaptic weights associated to the action group of neurons  $[A]$  takes into account the unconditional links [UL] related to the unconditional stimuli of the action group. The neurons in the action group are then able to learn conditional links [CL] according to the recognition result of the perception group.

$$[A] = [\text{UL}] + [\text{CL}] \text{ with } [\text{UL}] = \alpha \cdot Id$$

$$= \begin{pmatrix} \alpha & \dots & \dots & 0 \\ 0 & \alpha & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \alpha \end{pmatrix},$$

where  $\alpha$  is a constant small enough to ensure that recognition of a learned pattern will win over the reflex pathway. We use an Hebbian procedure to adapt the modifiable weights of the action group:

$$\Delta W_{ij} = \epsilon \cdot Y_i^{\text{win}} \cdot Y_j^{\text{win}} \left( 1 + \lambda \left[ \frac{d}{dt} \text{pleasure} \right]^+ \right)$$

$$\text{where } Y_i^{\text{win}} = \begin{cases} 1 & \text{if } Y_i = \text{Max}(Y_j) \\ 0 & \text{otherwise} \end{cases}$$

and  $\lambda \gg 1$  (the reinforcement term is much more efficient than the hebbian term).

$$\text{With } y = [x]^+ \Leftrightarrow y = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

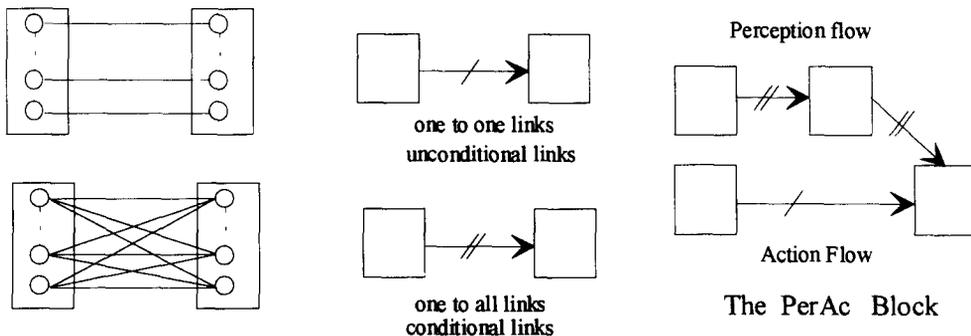


Fig. 7. two types of links to the formal design of our N.N. in Leto. In the second network, each neuron in the output group is linked to all the neurons in the input group. The sizes of the groups can be different.

Weights are reinforced when the reinforcement signal (pleasure) increases from time  $t - 1$  to time  $t$ . In a general way, the pleasure function is directly responsible for the emerging behaviour of the autonomous agent.

#### *Implementation of a PerAc block*

In Leto, our software used to design and create the N.N. that control Prometheus, a N.N. is represented by a set of boxes representing neural groups devoted to the same computation and using the same functioning rules. Each arrow represents a link between two groups of neurons. The arrows crossed with one short line represent one-to-one neuron links whereas the arrows crossed with two short lines represent one-to-all neurons links (Fig. 7). Commonly, the one-to-one links are reflex pathways and are considered as unmodifiable as in classical Pavlovian conditioning.

A token ring mechanism is used to update the activity of the neurons in each box. The activity of a box is computed only if all of its input groups have already been activated. To avoid the dead lock problems (in the case of recurrent or circular links) there is a special type of link which indicates that the presence of their input is not necessary to begin the process. In those cases, the input vector is considered as a null vector.

When new PerAc associations have been learned, they appear like new reflex which can support a new level of association (like a recursive mechanism). Learned links can, thus, be considered as meta-reflexes. Now, we will see how the PerAc block can be applied for object recognition and to control robot movements in a goal retrieval task.

### 3. Visual scene and landmark recognition

Prometheus' visual system computes the required information of the navigation system by emulating a moving eye. It performs learning and recognition of a local view associated to a landmark together with the angle between the landmarks. Its task is to learn several objects and to recognize them in a scene, where they can be scaled, rotated, deformed, occluded or noisy [39,40]. The first important feature of Prometheus that allows it to solve this problem is that it has a limited vision of the scene. It cannot see all the objects at once. It needs to move its eye from one object to the other. This limitation requires it to have a sequential functioning which simplifies learning and recognition.

Fig. 8. shows the general architecture of the vision system.

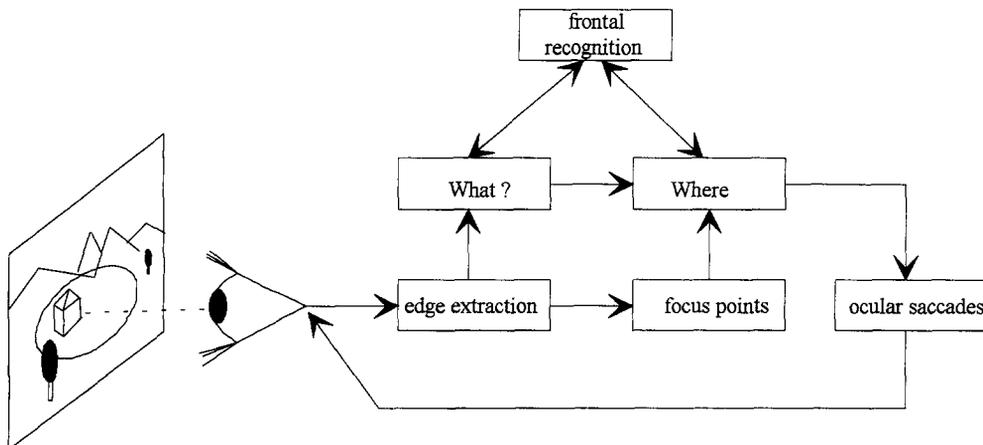


Fig. 8. General architecture of the vision system.

### 3.1. Biological model of vision system

Several cortical areas are involved in visual scene recognition [32,41]. The visual information preprocessed by the retina are used by the primary visual area to extract boundaries (V1), textures (V2), motion (MT) ... These primitives are integrated in more complex ones. They allow the preattentive control of the ocular saccades and of the focus of attention [42]. Next, the shapes are recognized by the temporal lobe whereas the parietal lobes control where to look.

Two connected levels of processing can be distinguished. The first one is involved in low level processing. It is massively parallel. It extracts the contours of the image [43] which are diffused to obtain local maxima that correspond to the characteristic points that attract the robot's attention [44,45]. The second one processes a state space transformation of the input picture, i.e., a log-polar transformation [33] which is tolerant to rotations and changes of scale but really dependent on shifts in position.

The local visual interpretation is performed by a mechanism that realizes mental rotations. Indeed, it has been shown that the visual recognition time depends on the angular variation between the learned object and the presented object [46]. Thus, we can imagine such a switching mechanism that would rotate objects to simplify their recognition and another one that would be

useful to build a scene representation that does not depend on the eye or head position.

On another level, the eye movements (ocular saccades) and the focus of attention are controlled by motor map [47]. Both visual and motor data are joined in the frontal areas where temporal integration is used to recognize sequences. They define a *non-symbolic* mental representation of the studied object.

The recognition of an object can be performed very quickly and does not need any ocular saccade but the recognition of a complex scene (and its recognition by a human subject) is more precise when the presentation time increases [48]. Indeed, humans seems explore the same parts of a visual scene when they explore it for the first time than later when they look again at the same scene (scan-path learning [49]).

### 3.2. A PerAc network for the visual system

#### *Primitives and architecture of the visual system*

Prometheus' visual system tries to emulate the behaviour of the biological models depicted in the previous section. From the image of a CCD camera ( $256 \times 256$  pixels), the contours are extracted and a simple filter allows to find a proposition on where the robot should focus its attention. We have chosen to use angles between edges as focus points. They are extracted by a

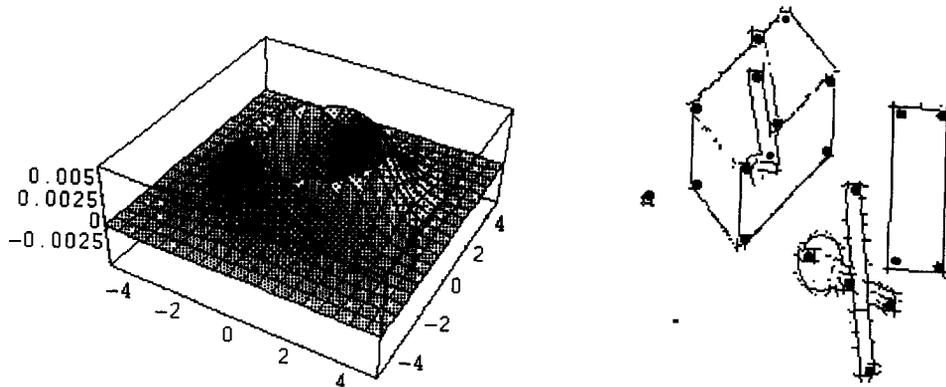


Fig. 9. (a) The filter used to find corners: it is the difference of gaussian mask: a OFF-Center cell. (b) an example of feature point extraction on contour image. Big black dots represent features points.

kind of OFF-Center cell (Fig. 9) that provides a maximum response when there is a sharp corner in the neighbourhood. A competition mechanism identical to that involved to extract edges is used to find feature points at a particular resolution.

We prefer the robot's eye to focus its attention on a corner rather than on the gravity center of objects, because with the second solution, if the object is occluded, the position variation of the object's center of gravity is huge and makes recognition impossible. When the robot glances at an object's corner, there is not recognition of the limited viewing zone. It only risks to lose a few focus points from all the focus points used to recognize the object. The sequential object exploration is then a good method to provide redundancy and movement information to help recognition. To sum up, a perceptive data stream identifies the contour image around the focus point, and a motor one guides ocular saccades. Both interact with each other. The scheduling memory of the local recognitions and actions can explain attentional processes that lead us to first explore one possibility before "thinking" about the next one.

As far as the final task is to allow the robot to navigate, we can suppose there is no rotation problem if the robot camera is always horizontal<sup>1</sup> and so simplify the explanation about the visual system (information about the mental rotations can be found in [39,40]).

Due to these considerations, Prometheus' vision system does not need any complex hierarchical structure to recognize objects. Moreover, the object concept in Prometheus is not linked to the need to analyze a closed region in the image. An object can be composed of several isolated parts. So a scene with all or a part of its most relevant objects can be considered as a single object. Its recognition will depend on the robot's capability to recall the scan path used during learning to go

<sup>1</sup>The attitude of the camera could be controlled by a gyroscopic mechanism like the biological vestibular system and by a mechanism of mental rotation like those performed to rotate the landmarks to retrieve the learned angles.

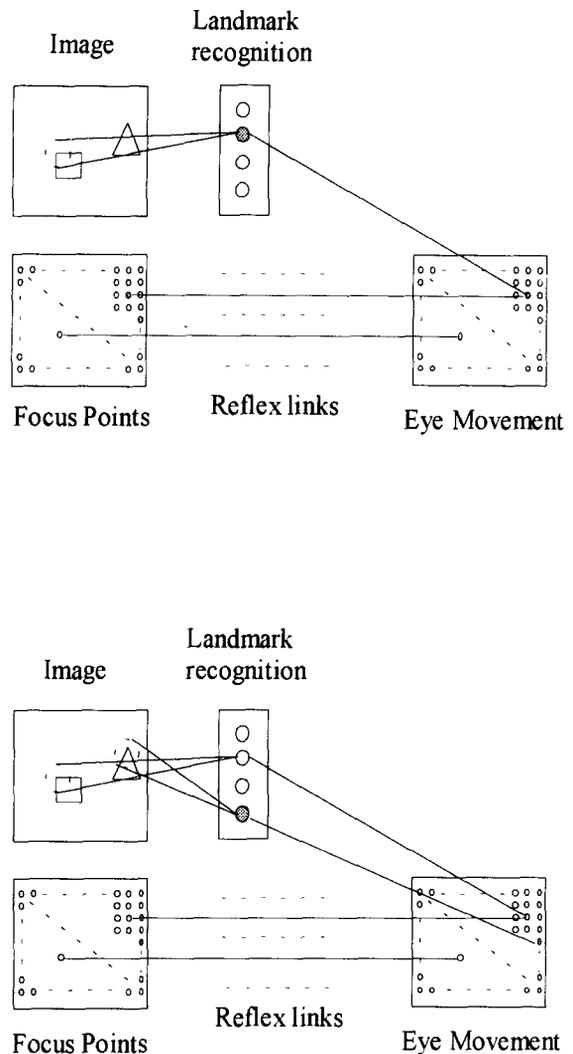


Fig. 10. Functioning of the PerAc block used for vision. Four neural groups are involved. Prometheus focuses its eye on one of the square's vertices. The ocular saccade it will perform is due to the combined activation of one neuron in the local recognition group and a neuron in the proposed eye movement group. The performed saccade thus corresponds to the one learned when exploring this square's vertex for the first time.

from one focus point belonging to one piece of object to the next.

#### *Learning and recognition of a visual scene*

During training, the robot extracts the characteristic points in the scene and it performs an

invariant transformation from each of these points. During interpretation, the robot focuses its eye on a characteristic point (a corner), it performs an invariant transformation (i.e., a polar logarithmic transformation) and then a mental rotation to match the present target with the learned representation. To complete its interpretation or to remove any ambiguity, the robot focuses on the other characteristic points used during learning according to learned saccadic movements (Fig. 10). At last, a mechanism of time integration is introduced to simulate a short term memory. Thanks to it, Prometheus will be able to interpret a particular area according to the previous interpretation.

When a characteristic point has been chosen, an inhibition mechanism prevents the robot from choosing it all the time. However, a problem remains. The points to inhibit are in  $\{\log(\rho), \theta\}$  space and when the robot changes its focus points, it loses the origin of the transformation. There is a new mapping in the state space. If a simple feedback is performed, it is not the neuron corresponding to the previous mapping which will be inhibited. Consequently, we assume that the brain has a local mapping of the picture expressed in coordinates invariant with respect to the eye movements. This space must be like an *internal universe* and we also need an inverse polar transformation. For details, see [40]. The complete architecture of the vision part is shown on Fig. 11.

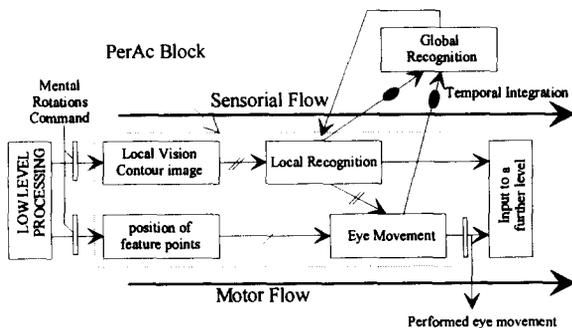


Fig. 11. PerAc architecture for visual scene interpretation. Each block is a group of neurons. The global recognition can be suppressed in the navigation task if all the landmarks are different from each others.

The eye movement group is a WTA (Winner Takes All), with input in the perceptive and motor flow: the position-of-feature-points group proposes a movement, and the local recognition group is associated to a given movement. There is a global recognition group, which learns with the help of a teacher, and which works according to a counterpropagation algorithm [25]. However, it does not belong to the studied unit block and it is not necessary to solve the complete robot task.

### 3.3. Experimental results

In the following example, the robot has learned 3 objects (Fig. 12a): a key, a cube, and a cigarette. The objects have been presented just in front of a gray level CCD camera ( $256 \times 256$  pixels). The edges are extracted by a N.N. inspired from [43]. A simple Nagao contour extractor can also be used [50]. The resolution of the local views after the log-polar transformation is  $32 \times 32$  pixels. The learning time only allows the robot to learn 4 local views of each object. So, the N.N. stores  $4 \times 32 \times 32 = 4096$  bits of information for each object (the compression rate is 128). Later, a scene with several of these objects is presented to the robot. The edges and the focus points are also extracted. The robot recognizes the learned objects well even if they are rotated, occluded or seen from a little different angle (Fig. 12b). When it finds a learned local view, it focuses in the direction of the supposed position of the following learned view to verify if its first interpretation was correct.

Other experiments on our mobile robot indicate that the polar transformation and the pattern matching mechanism allow to recognize a planar object over a distance that varies of  $\pm 1/3$  and also bear that the CCD camera should be horizontally oriented at  $\pm 50^\circ$  from the learned position (the robot faces the object) [51] (for similar measures see [52]).

### 3.4. Discussion about the visual system

Obviously, if we want to use information about object integrity it may be difficult, and the system

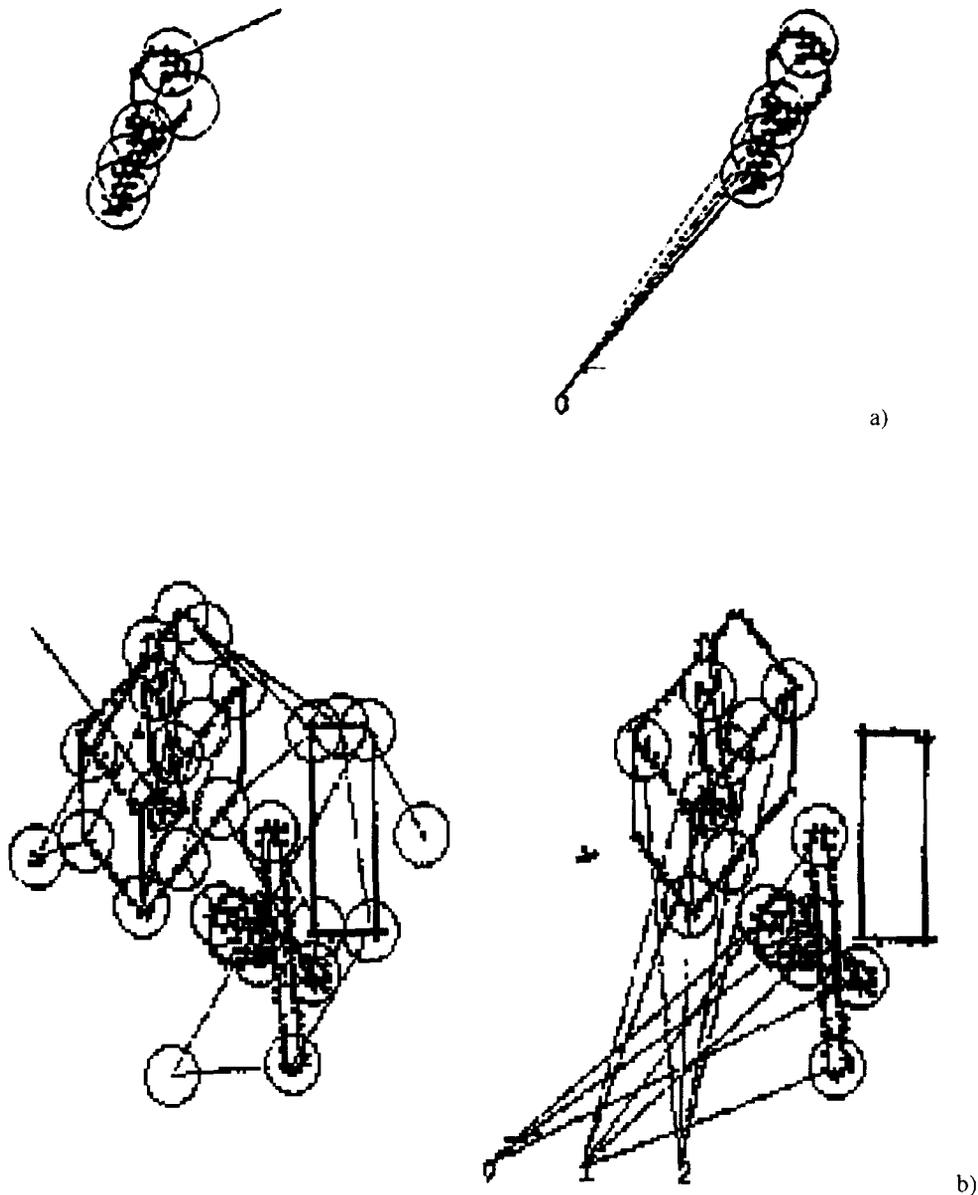


Fig. 12. (a) Learning of a key labeled as object 0. (b) Observation of a complex scene: scan path of the eye (ocular saccades) and interpretation of each zone pointed out.

would certainly need additional groups. But if we simply consider that each scene is an object to recognize, all the information needed is already available. Nevertheless, some precautions must be taken. For instance, horizontal and vertical movements do not have the same meaning for the navigation system. But the information about the

apparent size of the object and about its angular position could be combined. All those things are not yet implemented but are currently being investigated.

The size of the local view (snapshot view) can be adapted to the complexity of the problem. If the landmarks are all the same then the informa-

tion is perhaps the “drawing” realized by a subset of landmarks. So in that case the visual field must be larger to take into account the low resolution information of the image. The temporal information can also be used to avoid ambiguities. It is represented by the Global Recognition group of Fig. 11. Its feedback loop to the Local Recognition group then allows to distinguish the vision of the same “snapshot” by their order (temporal aspect).

Today, the main problem is the difficulty to analyze the complexity of the learning and recognition task. If only few objects have to be recognized and are “sufficiently” different, the competition mechanism will allow a very good generalization of size and orientation variations. At the opposite, if a lot of objects that look like to each others must be recognized, the generalisation will be less important and a preprocessing that allows to take away the shape from each others should be added.

#### 4. Target retrieval using landmarks

Most of the present navigation systems use odometric information to know where they are on a cartesian map. A lot of path finding algorithms have been developed based on the classical A\* algorithm or potential field techniques [54] (Refs. [17,53] for a neural approach of the potential fields). Unfortunately, odometry is not precise in a long run and it must be recalibrated by other sources of information such as particular visual patterns called landmarks [1]. These robots much

more work like surveyors. In other approaches, based on proximity sensors (ultrasounds ...), the different places are difficult or impossible to identify and the robot must take into account its movements sequence to decide what its current position is. These algorithms almost all separate the learning phase of the different places from the learning of the links between them: can I go from “A” to “B” ...? They somehow succeed in building a cognitive map of the environment [55]. An interesting subsumption implementation that does not need to produce a cartesian map can be found in [56,57]. In our view, their main problem is linked to the learning criteria. How can the robot decide when to learn a new place? If the robot forgets to learn a place, it will be unable to reach particular places (cut in the graph of its cognitive map). Conversely, if it learns too many places it will have memory problems and will be unable to realize that two nodes in its map are in fact associated to the same physical place. In the case of an environment of heterogeneous complexity, a self-adaptation of the learning criteria seems to be difficult to implement without knowledge stemming from a reinforcement signal (goal reached, “important” places, ...). Moreover, these algorithms do not answer the question about how to reach a place when an infinity of pathways can be taken as in an open area.

##### 4.1. Biological models of navigation system

First of all, we can ask the question of whether animals use a “map” (Cartesian map) of their environment with information about the position

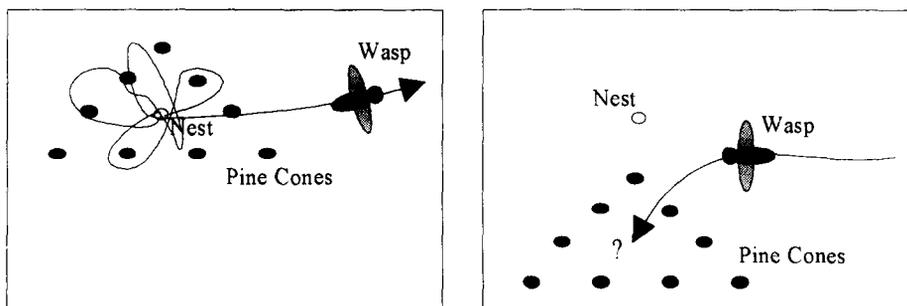


Fig. 13. When a wasp goes away from its nest. (a) It begins by circling around the nest position. (b) If the position of the pine cones around the nest are translated then the wasp will try to find the nest where it should be relative to the landmarks [58].

of each landmark. Tinbergen had realized an experiment that brings light to that question [58]. He was interested in the manner a wasp succeeds in retrieving its nest which can be difficult to see. Due to the wind, proprioceptive mechanisms cannot alone explain the nest localization. For that reason, he put around a wasp's nest several pine cones in a triangular fashion (Fig. 13). He noticed that the insect circles around its nest for several seconds, before departing for journeys as long as an hour. Before the wasp returned, he moved the triangle to another location. Then, when the wasp returned in the neighbourhood of its nest, it went in the direction of the pine cones and tried to find the nest at the same position where it had been between the pine cones. If the pine cones are far enough from the nest then the wasp will never succeed to retrieve its nest because it is not easily visible. It, thus, appears that the wasp does not need a map of its environment but, like the ant, it "just" tries to retrieve landmarks at the same position they were learned (see Section 2.1).

Different models have been proposed to explain all those complex behaviours from only direct perceived image treatments. They show that animals do not need complex internal representation of the world (generally associated to a map). For instance, Cartwright and Collett [59] have proposed a model of bee navigation. In their model, the bee proposes a movement direction that lowers the discrepancy between the perceived image and a snapshot taken at the target position. The main drawback is that all the landmarks must be the same, and circularly symmetrical, such as the cylinders they use. Their model cannot be generalized to more complex landmarks.

Obviously mammals can use more sophisticated methods for navigation. But it would be unreasonable not to use the same principle, if it is compatible with the biological data about the mammals and if it can also explain their more complex behaviors. For instance, Morris [60] proposed an experiment in which a rat is trained to swim in a tank toward an invisible platform. Fixed marks on the walls of the tank are visible from any point in the tank, and they constitute the only information available to the rat for its localiza-

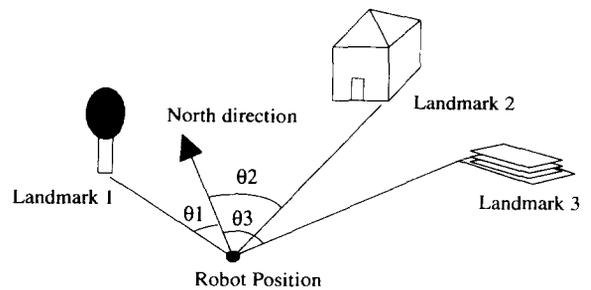


Fig. 14. Example of a landmark configuration that the robot can use in a localization task.

tion. Other experiments by O'Keefe [61] show that a rat can find a goal in a X maze by using familiar objects as a lamp or a window as landmarks (Fig. 14).

These experiments have also shown that the brain's hippocampus plays an important role in this work of target retrieval in mammals [62]. They have found that particular cells in the hippocampus respond maximally when the rat is at a particular position and that their activity decreases as the rat is displaced. It also seems that this response does not depend of the rat's orientation in its environment. This means that the rat must be able to rotate all its visual information in order to present it all the time in the same orientation. This switching mechanism can be explained by the presence of head-direction cells, whose response depends on an absolute direction of the rat throughout the environment. Nevertheless, the real role of the hippocampus in place recognition is still not clear. The only indisputable thing is that the hippocampus merges or correlates information coming from different cortical areas in the brain. Thus, it provides a multimodal representation that can link the recognition of visual landmarks with the movement to go from one landmark to the other. Other proprioceptive information also allows hippocampal cells to react when the animal is in the dark [63]. In conclusion, the navigation of animals can be explained without the need of a cartesian map of the environment. Their internal "map" can be very sparse and bear no reference to the topology of the external universe.

#### 4.2. A PerAc network for the navigation system

Obviously, more and more systems take into account these biological considerations and navigate directly from 2D perceived images to reduce their algorithmic complexity and to increase their robustness (qualitative navigation [64], visual homing [65]). The PerAc network for navigation somehow looks like the system proposed by Bachelder and Waxman [66,67] but it also allows the robot to decide which movement to do to reach a particular place. The main difference is that we do not want to learn each position in the environment. Our robot must only discover an interesting place and learn by itself how to return there from any other point. It can also generalize efficiently to other places in its environment. We will show that if a compass is available our algorithm can allow a robot to navigate correctly even if it is situated far away from the learned position (with the limitation that the landmarks must stay visually recognizable). If no absolute direction is available other simulations will show that a landmark can be used as referential but then the generalization capability to long distances is reduced. In that case, the algorithm can correctly model navigation in a closed room or an outdoor navigation limited in a closed area limited to the envelope of the landmarks. Moreover, we will also show that the place recognition is not more

complex when the landmarks have different aspects according to the robot's point of view.

#### A hippocampus-like system that correlates incoming information

In Prometheus, the position of the robot in its environment is coded as a snapshot image of the landmarks containing their bearings. We suppose that these angles can be known as the result of either ocular saccades or head movements. In the previous section, we have proposed a mechanism for providing this type of information. Both types of data can be joined to provide information about "what" are the landmarks and "where" they are. Simple product or logical AND neurons can be used to merge these different types of information in a map of neurons that reacts only if a particular landmark is recognized at a particular place (Fig. 15).

This model seems to be biologically plausible and to agree architectures and the navigation models based on the hippocampus [68-70]. In our model, however, a place is not coded in hippocampal neurons but in a cortical area. The hippocampus is only used as a relay to allow the information to be associated. A short term memory represented by recurrent positive feedback links (or by the intrinsic synaptic properties) is used to obtain a spatial image of the position of the different landmarks in the observed environ-

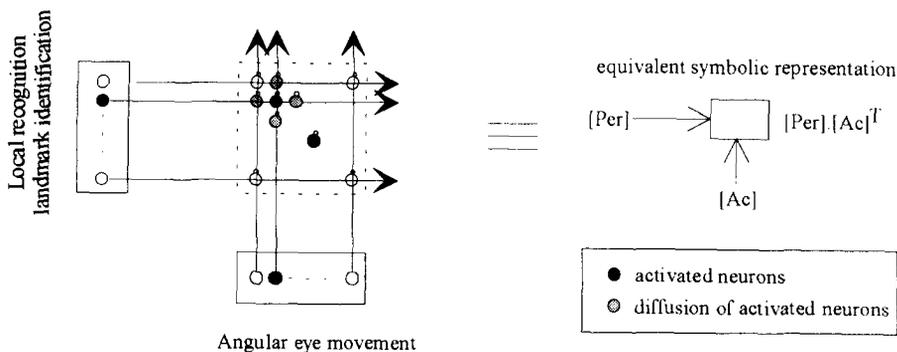


Fig. 15. Recombination of visual and motor flow as an input to the place cells.

ment from the sequence of input activation. This sort of merging of two vectors [Per] and [Ac] (representing the recognition of the perceived image and the action flow) can be formally written as a [Hip] matrix:

$$[\text{Hip}](t+1) = f([\text{Hip}](t)) + [\text{Per}] \cdot [\text{Ac}]^T,$$

where  $f$  is a function that allows a short term memorization of the previous vectors.  $f$  can be a scalar function. Then, we have:

$$[\text{Hip}](t+1) = \alpha \cdot [\text{Hip}](t) + [\text{Per}] \cdot [\text{Ac}]^T$$

with  $0 < \alpha \leq 1$ .

If  $\alpha = 1$  then [Hip] is the sum through time of all the What/Where associations. So a clear mechanism must be introduced to reset [Hip] when the robot changes its position and must compute again its location. This matrix [Hip] defines an array of neurons representing the robot's position in which a line represents the "identity" of the landmark, and a column its bearing in head-centered coordinates. The bearing is discretized, in order to have a binary vector to learn. The activation of place cells could be computed as the inverse of a Hamming distance between their weight vector and the hippocampus activity:

$$\text{match}(W, \text{Hip}) = 1 - \sum_{i=1}^m \sum_{j=1}^n \frac{|W_{ij} - \text{Hip}_{ij}|}{n \cdot m},$$

but then, all topology information *about* the landmarks scene would be lost. Indeed, let us suppose that the input vector of Fig. 16a. has been learned by a place cell. If the same vector shifted by a given amount is presented (Fig. 16b

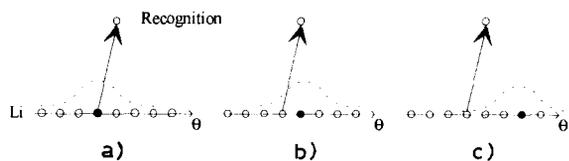


Fig. 16. Displacement of the neural activity according to angular position of the considered landmark (Li) in the visual field. The response of the recognition neuron between case a) and case b) should be higher than between a) and c). This is obtained by using a diffusion of the active neuron (black circle).

or 16c) the activity of the place cell does not depend on the shift value. This is a pity because two close sets of landmark bearings suggest close positions.

If the active neurons in the input pattern are also diffused on their neighbours (in the horizontal direction), it is possible to overcome that problem, as shown on Fig. 16. Indeed, the diffusion induced activity on the neurons learned for pattern a is higher for pattern b than pattern c ( $\text{match}(a,b) > \text{match}(a,c)$ ). Obviously, we must suppose that the visual system can differentiate landmarks. We cannot afford having the same landmark found twice in the same panoramic view. Otherwise the system would not succeed in knowing which angle is associated to which landmark. So, in the case all the landmarks are the same kind of cylinder, we suppose the visual system will use information about the neighbourhood or will choose a particular landmark to index all the others by reference to it. This implies to learn a sequence and not just to recognize a snapshot. Fortunately, this is exactly what the visual system of Prometheus does [40] (see the previous section). Thus, the experiments with identical landmarks are not relevant for our navigation system because the problem must be solved by the vision part of the system.

To sum up, we do not use the hippocampus has a structure that identifies the place but as a correlator between informations coming from motor and sensory areas. Why these correlations are not directly performed by the neurons in the cortical areas seems to be due to practical problems [71]. As a matter of fact, the important number of neurons in the different cortical areas does not allow systematic interconnections between them (each neuron has approximately 10 000 synapses, if we consider the human brain with 10 000 000 000 neurons, the information can have to pass, in the absolute, throw 2 or 3 neurons to join any pair of neurons). Therefore, they cannot detect easily the correlation of their activity with the activity of other neurons located in a very distant area. As the hippocampus receives projections from all cortical areas, it could be the structure that decides if a situation is enough different from others to be learned. A more long

term learning could then explain the learning of the temporal sequence of landmarks recognition in frontal cortical areas (linked to the time to make the cortico-cortical connections). That will then explain why subjects with hippocampal lesions can continue to live normally but have difficulty to learn new information [60].

*Learning how to return to a particular place*

At the beginning of the exploration phase, we suppose Prometheus moves randomly, looking for something interesting. When it finds "food", it first eats a piece of it and then moves around in order to find various positions in the food proximity. At these places, it will learn both the landmarks configuration and the direction that leads to the food. Later, when the robot wants to find "food", it considers the information of the place cells and moves in the direction associated to the most activated place cell (competitive mechanism) to reach the food. Thus, at each time, the distance to the target is reduced (Fig. 17) and it returns inevitably to the learned position of the food. The interest of such a mechanism is that we only need to learn a few number of places in the immediate neighbourhood of the goal and the robot generalizes to all the area in which the landmarks are visible (see Fig. 24).

The learning phase is the most complex because it is an unsupervised and an on-line process. When Prometheus eats "food", it triggers a

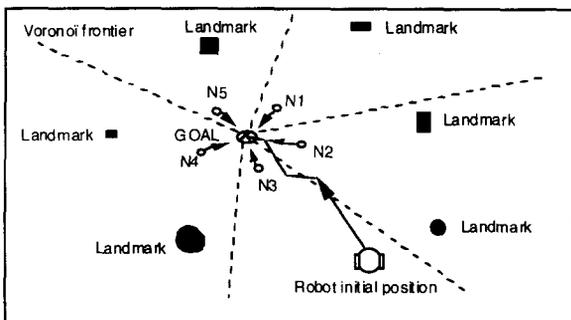
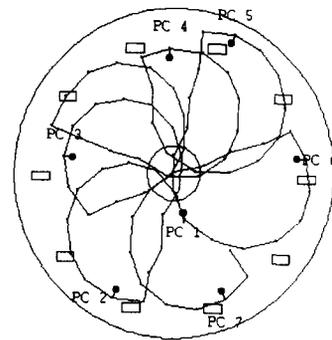
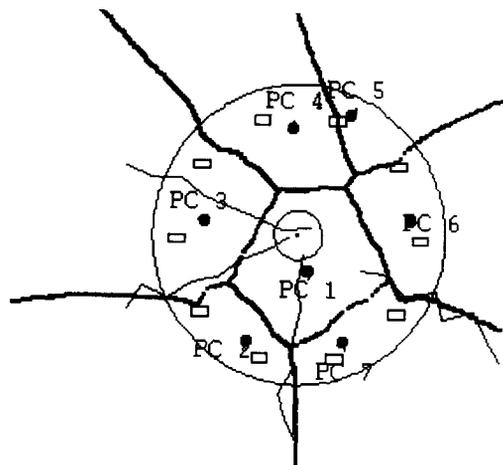


Fig. 17. Local exploration around the target represented by the intersection of the dot lines. The agent records at certain points (represented by small circles, N1 to N5) their relative position to the landmarks (represented by squares) and the direction to the target.



a)



b)

Fig. 18. (a) Local exploration around the target represented by the large black circle. The robot records at certain points (represented by small circles) their relative position to the landmarks (represented by squares) and the direction to the target. The numbers correspond to the place-field number in its neuron group. (b) Different trajectories. The Place-cells (PC) are indexed by their order during exploration. The Voronoi tessellation is represented by the thick lines, the landmarks by the rectangles and the target by the inner circle. The large circle represents the limit beyond which the target is not perceived. Thin lines represent trajectories from various starting-points.

reflex which allows to somehow circle around the food at a certain distance, in order to visit evenly placed locations around it. At each of these well chosen locations, a place cell learns the relative position of the robot according to the landmarks,

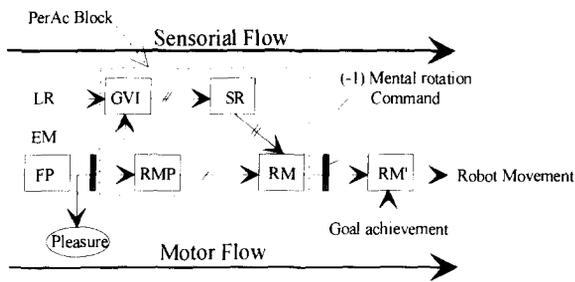


Fig. 19. The navigation neural network. SR is the Scene Recognition group. Its input is the Global Visual Input group which corresponds to the Landmarks Recognition associated to the Eye Movement. The Robot Movement group (RM) is a WTA. When the food is visible (Food Proposal group), the chosen direction in RM corresponds to the food position, because of high-valued one-to-one links between RMP and RM groups. The RM' group is also a WTA and it corresponds to the Robot Movement in the environment. When goal achievement is activated, it activates through a high intensity reflex a particular neuron in RM', causing the robot to turn in a given direction, thus giving rise to ellipsoid trajectories. The black rectangles represent a shifting mechanism used either to provide an invariant representation of the input, or to transform invariant representations into extracorporeal ones.

and the direction heading towards the target (Fig. 18). We shall now detail the neural network used for landmark-based navigation.

As usual when using the PerAc block, four neuron groups are involved in the navigation task (Fig. 19). Inside the PerAc block, the neuronal groups used for the input and the actions must correspond to invariant representations with respect to the robot's orientation (Fig. 20). The switching mechanism that provides that invariance must be used at the input of the block, while the inverse transformation is applied at its output. Thus, movements to go from the location learned by a place cell to the food are learned independently of the robot's orientation (inside the PerAc block) but the movement actually performed (outside the block) takes the orientation into account.

When a movement direction is selected, the robot makes one step of a given length in that direction. The input to this network are the north direction, and the food and landmarks positions in the robot's visual space. We assume that a compass is available. It could be replaced by a vestibular system or a gyroscopic mechanism that would produce low precision information about the body orientation. A local landmark could also be used but it reduces the generalization capabilities of the robot to very distant situations (see

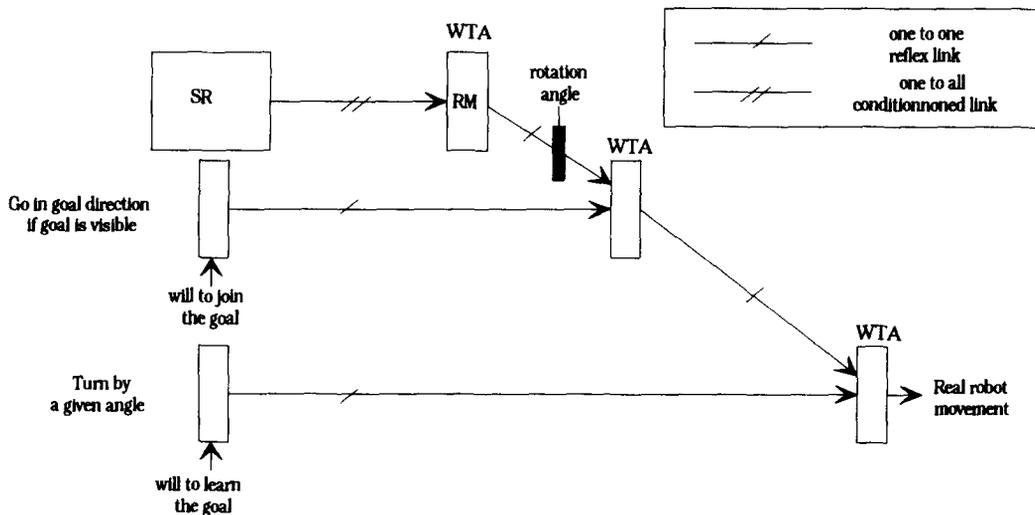


Fig. 20. The different reflex information added to the learned robot movement RM. The inverse rotation is performed to take into account the rotation of the input data of the scene recognition SR toward the absolute or relative landmark used for the angle measure. The Real Robot Movement is obtained through the competition of the learned information and the reflex of going in the direction of the goal or turning by a given angle. Both pieces of information can be directly linked to the same box.

experimental results Fig. 24). Just as for humans and most mammals, we assume that the immediate visual angle is limited. Therefore, food is perceived only when it is located in a given orientation ahead of the robot. The same goes for the landmarks, but we assume that when a position must be recorded, Prometheus rotates in order to see in all directions. This supposes that when exploring a scene, it can make ocular saccades and move its head as well, thus, spanning the whole surrounding space.

The functioning of the N.N is easier to understand when starting from the end, that is the one-dimensional neural map corresponding to the movements. We used two different maps, because the "exploration" reflex must activate a "turn left by a certain angle" from the current angular position of Prometheus. This reflex, thus, activates the group coding movements located outside the PerAc block (Fig. 21).

When food is in sight (food recognized), a neuron corresponding to its angular position relative to the robot's facing position is activated in the Food Position Map (we suppose that the robot has previously learned to what the food looks like). The shifting mechanism activates a neuron in the Robot Movement Proposal (RMP) by adding an angle corresponding to the angle between the robot and the north. If there is pleasure at that moment, a place cell learns the invariant landmarks position, and the association with the robot movement in RM due to the reflex link from RMP. The inverse shifting mechanism is applied to the output of that group, by subtracting the same angle. This activates the neuron in the effective RM' map which corresponds to the actual movement to be performed by Prometheus.

The achievement of the robot's goal (to eat food) triggers a movement reflex that remains active for a certain amount of time (Fig. 20). The provoked trajectories after reaching food, thus, take an ellipsoidal shape, which ends after a while. As soon as food is in sight (given a limited visual angle) the position of the landmarks is recorded. This supposes that when pleasure is active, the robot moves its "head" in order to see landmarks in all possible directions.

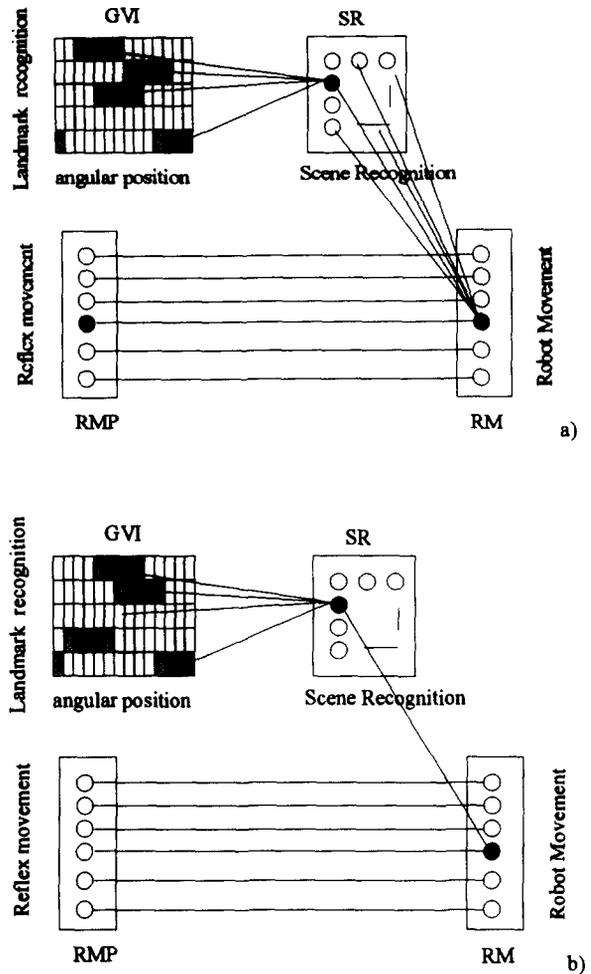


Fig. 21. Example of place situation learned by the N.N. (a) GVI represents a configuration of landmarks seen from particular angles. One neuron in SR learns that situation (black circle). The visual system devoted to the recognition of the target provides the direction of the goal in the group RMP. The unconditional link between RMP and RM allows one neuron to win in RM.

To sum up the robot behaviour, we must not forget the timing of the learning process (Fig. 22). A learning cycle begins the first time the robot discovers or sees its goal, i.e., the "food". At that time, we suppose that to see the goal causes pleasure and therefore an increase of a vigilance parameter that controls learning. It is a global parameter of the Neural Network simulator. It modifies the functioning of each neuron in the network (see Appendix 7.2.3). Much like in ART

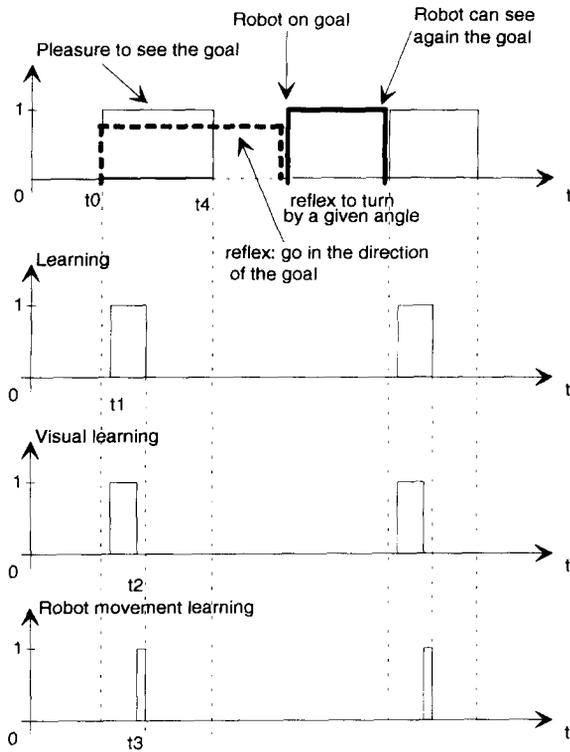


Fig. 22. Scheduling of the learning and reflex signals.

networks [18], the vigilance level controls if a new pattern must be learned or not. If it is high, the network will tend to learn all presented patterns. Thus, when pleasure is present, the vigilance causes the SR group to learn the presented landmark panorama. If vigilance is low, a lot of patterns are not learned, because their matching with already learned patterns is too high. When the vigilance is low the neurons can generalize and produce an activity that tends to be monotonically dependent of the matching. So the competition mechanism can produce a well adapted answer.

Thus, when the robot sees the goal at time  $t_0$ , it activates learning at time  $t_1$  that allows to extract and learn information about the landmarks and their bearings. This process ends at time  $t_2$  and then the information is available for the navigation part of the robot's "brain". At time  $t_3$ , the GVI can be learned by the SR group and be associated to the activation of the RM group due to the reflex link from RMP. At the

end, the robot performs the movement in the direction of the goal. During the time between  $t_3$  and  $t_4$  the robot moves according to the reflex movement in the direction of the visible goal. There is no learning during that time because the learning rate and the vigilance have returned to low values. Both parameters can be computed as the positive derivative of the pleasure signal. When the pleasure signal appears they are high but return to zero if the pleasure remains at a constant value.

When this learning phase is over, it becomes possible to launch the robot from a place where it is not supposed to see the food, and it appears from the simulation results (Fig. 18 and part 4.3) that the robot always takes the right direction, whatever its starting point. The distance from the place cells recorded positions from which the robot can be launched grows with the angular resolution and with the width of the diffusion applied to the input.

### 4.3. Simulation results

We have simulated the navigation network on several test situations. These experiments are divided in two groups. The first series concerns examples in which the landmarks have the same interpretation from any point of view but are considered as different from each other. The second series use complex landmarks that do not have the same aspect according to the robot point of view.

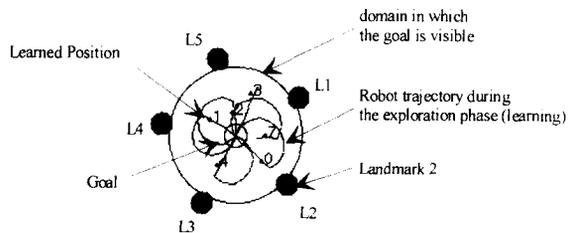
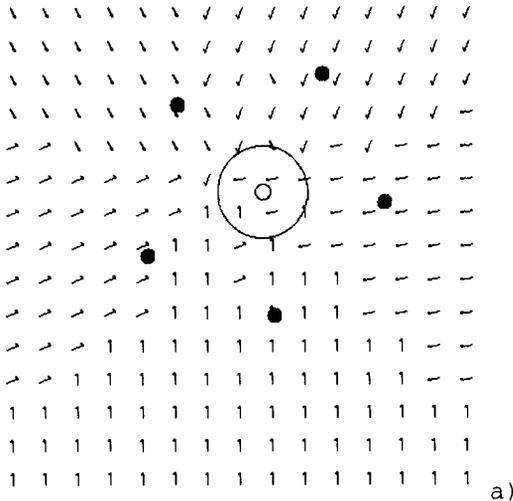


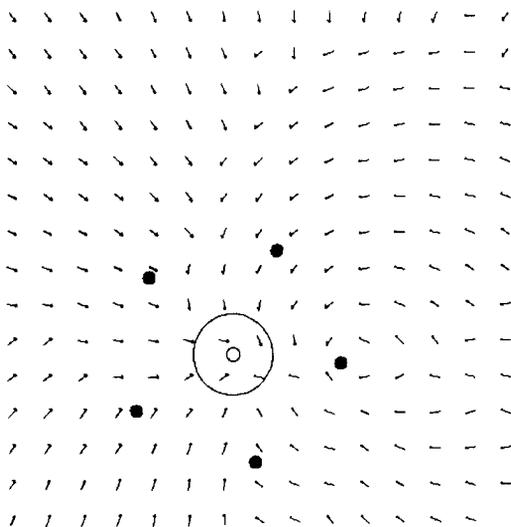
Fig. 23. Robot trajectory to learn to return to the goal represented by the little circle (in white). The largest circle represent the maximum distance from which the goal is visible. The full disks represent the landmarks. All the landmarks are different from each other even if they are all represented on the screen by the same symbol.

*Case of cylindrical landmarks*

Fig. 23 represents the exploration phase of the robot to learn how to return to the goal. In all the experiences, the goal is represented by an empty circle in a larger circle representing the area in which the goal is visible. When the robot is out-



a)



b)

Fig. 24. Vector field representing the robot movement direction: (a) when using an absolute direction to measure the angles between the landmarks, (b) when using a landmark as origin of the angle measures. As reference direction (null angle), we take the direction of the Easter landmark.

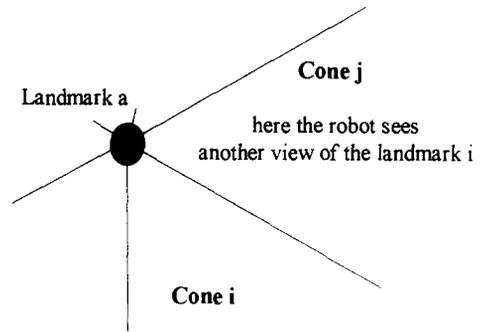


Fig. 25. When a landmark is seen from the area or cone *i*, it is recognized as a different image than from the cone *j*.

side this largest circle it cannot see the goal and therefore cannot use its visual reflex to move towards it. It can only use the recognition of the place to decide which movement to perform. Fig. 23 shows the 8 places that the robot has learned during the exploration phase. In the Scene Recognition group 8 neurons (or place cells) have learned the positions.

Fig. 24a and 24b represents the movement the robot will propose from all the possible positions in the environment in the case the landmark configuration has been dilated. Fig. 24a represent the case when the robot uses a compass (an absolute direction) to commute the measured bearings. The frontier between the different domains associated to the different proposed movement are lines. On the other hand, when the robot use the East landmark as a reference to compute the angles then the proposed movement seem to turn around the goal (Fig. 24b). The Voronoï frontiers are more complex but the robot nevertheless navigates correctly. So the model allows the robot to navigate correctly with landmark dilatation and absolute or local point of reference to measure the angles.

*Generalization to complex landmarks*

When the landmarks are not cylindrical or do not have the same visual aspect from any point of view, the previous results can be generalized if we consider that each landmark view is considered as an independent landmark (Fig. 25). This idea boils down to consider more "snapshot landmarks" as input for the navigation N.N. [52].

In fact, that simplifies the computation of the robot location: neurons that code situations which differ in the presence of one or several “snapshot landmarks” will have more distinct activities than in the case studied before (for a detailed analysis of the geometrical properties of such concepts see [72]). The competition mechanism can then

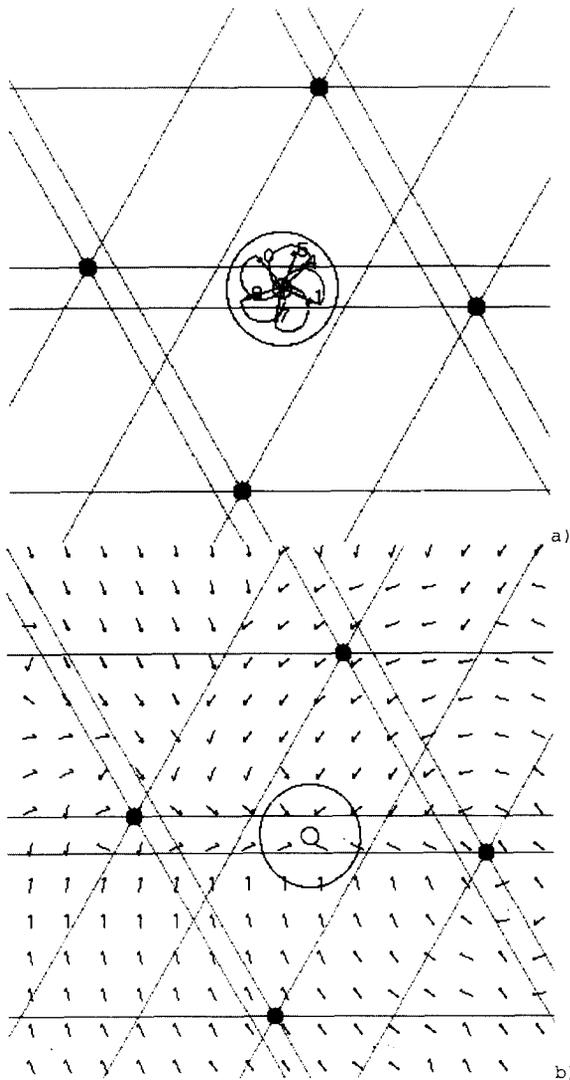


Fig. 26. (a) Learning of a place in the case of landmark which are associated to 6 different visual aspects. The robot is supposed to be able to recognize one visual aspect of the object only if the object is seen from in the cone made by two consecutive lines. (b) vector field when using the east landmark as origin.

be less precise. Fig. 26 represents the case in which the landmarks have different visual aspects when they are seen from different points of view (the more common situation in natural environment). The lines that start from the landmarks represent the limits of the domains associated to a particular view associated to a landmark ( Figs. 25 and 26). To simplify, we suppose that the 360 degrees around the object are divided into 6 angular sectors. That means that one landmark is in fact represented by 6 visual landmarks which are at the same location but only one of them can be recognized at each position. Fig. 26 shows that the robot can reach the goal even if it loses a lot of information about the learned landmarks. For instance in the upper left part of the domain on Fig. 26, the robot can only recognize the learned view of the landmark 1 and 2. Whatever it succeeds in going in the good direction because the associated neuron in the Scene Recognition group (place cells group) has won over the other neurons. In that respect, the image of the different sectors shows that the recognition of the right place is easier when landmarks cannot be visible. In most of the distant positions from the goal, the robot can only recognize 2 landmarks, and only few neurons have learned usable views. Thus, the eligible neurons are only good ones.

#### 4.4. Discussion about the navigation system

Our model verifies some biological findings such as the capability to join the goal when the landmarks are dilated or when they are rotated [59,68]. It supports the lack of landmarks or a misinterpretation of few of the landmarks. There is no need for a particular number of landmarks (more than two). To learn or to recognize the place, the precision will only grow with the number of landmarks. The main interest of the use of the PerAc block in the navigation system appears to be that it realizes an approximation of a potential field function. With only 8 neurons, the N.N. allows the robot to deduce the correct direction of movement to reach the goal. So there is no need to code each position of the environment (in a real Cartesian map) and their associated movement as done in most the navigation models.

More realistic trajectories can be obtained if the movement is performed according to a probabilistic vote rather than a determinist WTA mechanism.

Moreover, the same N.N. can be used to allow the avoidance of particular zones or to introduce other goals. For instance, if the robot finds another interesting area, it can learn it on unused neurons of the Scene Recognition group. Then, it may associate these last neurons to movements to perform in order to reach the new goal. The goal might change when pain is encountered, for instance when the robot collides in an obstacle (for more information about the goal level in the neural network [15,38,40]).

Obviously, the 3D vision aspect is very much simplified because we only use information about the relative position of the landmarks. We also suppose the landmark recognition is not a problem (the system can withstand miss-interpretation of a few landmarks). Here, we have simulated the possibility that a landmark can be associated to different snapshots as would happen with 3D objects but we have supposed the apparent object size (in the robot brain) does not vary too much when the distance to the robot changes [68]. Our current experiments on a real robot and results of other groups [67] indicate that the robustness of the vision system can be sufficient for simple shapes (log-polar transformation, see part 3). In any case, the recognition problem of really complex objects in the frame of autonomous robots

using CCD camera is still opened. The invariant recognition of objects that need high resolution analysis should request a complete understanding of all the visual brain areas (primitive extraction, fill in mechanisms, focus of the attention, depth, movement ...) and will surely be more associative than the simple pattern matching we perform [26].

### 5. Conclusion

Our aim for designing the PerAc block and the neural architecture that goes with it is to find the simplest possible solution for a large set of problems related to animal and human intelligence. We stressed particularly the simplicity principle because we think the simpler an architecture is, the less we have to put our own intelligence in it. Only proscriptive constraints implemented in reinforcement signals (pain, pleasure) are used to control the system [21]. Therefore, we can hope to design really general architectures to understand cognitive processes. We also keep in mind that insects and mammal give precious information about the neural code they use (even if it is not the same), and we hope our model can provide interesting information for biological research.

Throughout this paper, we have insisted on the importance of active perception. We have shown that using action simplifies the interpretation of

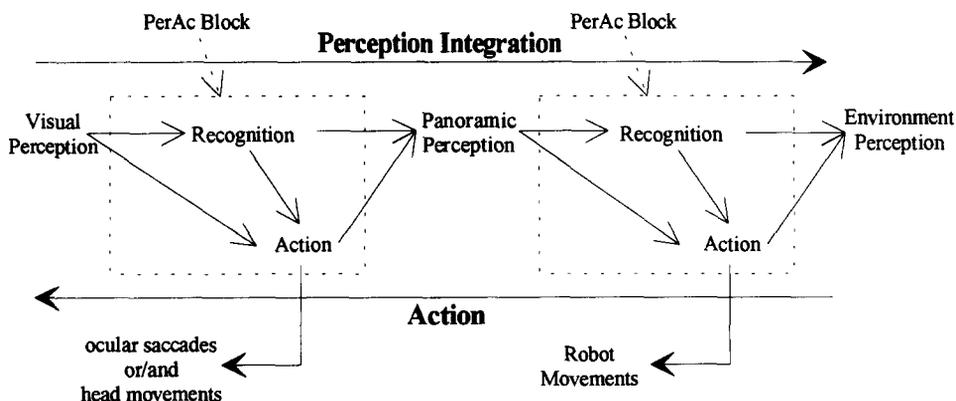


Fig. 27. The global neural network for visual scene recognition and navigation. It is made of two PerAc blocks, the first one for vision and the second for navigation.

perception: each action is a choice and conditions entirely the future of the robot. The greatest advantage of this type of approach is that it makes cognition sequential, thereby avoiding the possible large duplications and relaxation mechanisms needed by massively parallel systems such as the connexionist systems proposed by Feldman [73] or by the PDP group [7].

Prometheus proves that a complete autonomous navigation system has no need for an explicit symbolic representation: high level capacities use all the emergent phenomena due to the lower levels. The robot learns to categorize its external world according to what is relevant to it and not to us. What it stores only depends on its action capabilities and on its perception of the world complexity. Prometheus' "brain" architecture is summarized on Fig. 27. It appears to be a sort of basic building block and a systematic tool to combine motor and perceptive information. In addition, the PerAc architecture takes into account the dynamical aspect of the robot's behaviour and solves some robot control problems in which the "autonomy" is needed. Indeed, the PerAc architecture relies on the postulate that the recognition of any cue can be simplified if the system can act on it. That justifies the splitting of any perceived cue into two parts: a motor part and a cognitive part. The first is the result of a hardwired conventional processing, and is implemented as a reflex link. The cognitive part proposes to learn to recognize problems or simply important situations in order to allow a quicker adaptation of the system's response.

Furthermore, our model proposes an alternative to the classical scheme of hierarchical classification, because we propose to integrate not only static recognition information but also motor information provided by the input cue or/and the

local recognition. For instance, in the recognition problems or in the classification of high dimensional data, a commonly accepted method to avoid losing topology information consists in classifying local features before taking the results as inputs to higher levels. That constitutes a bottom up architecture with a pyramidal shape: the higher the level is, the less there are nodes to code the more abstracted information [74,75] (Fig. 28). From this point of view, the PerAc concept allows to greatly reduce the number of steps between the interface with the real world to "sufficiently" abstract levels. For instance, in our vision system, there are only three levels. The first level corresponds to low level processing, the second one is associated to the recognition of local features (local snapshots) and the third one deals with the global object recognition (by recognition of the temporal sequence of local recognitions).

To solve the same kind of task Fukushima [75] needs a number of layers that will directly depend on the invariance expected in its image analysis. In PerAc, the reduction of the task complexity is due to the *a priori* knowledge we introduce about the nature of the input image and about the relevance of the focus point. But that *a priori* information has nothing to do with the information required by methods of recognition by modelization. Here the information can be explained by two things. The first is the ontogenesis of the system, during which some value system could be evolved [76] (a review of artificial evolving can be found in [77]). The second is that we suppose that inputs have their own topology and that simple competitive/cooperative mechanisms can always be used to locate important features in any perceived cue. Moreover, our model agrees the motor theory of speech recognition which postulates we recognize speech signals

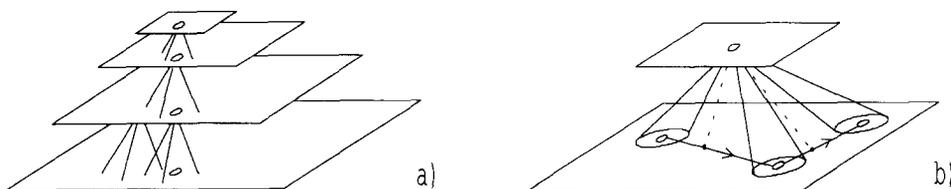


Fig. 28. (a) classical pyramidal structure for hierarchical classification. (b) PerAc structure (less levels).

by trying to imitate the heard sound [78]. The information used for the recognition would be the sequence of articulations to imitate the sound. It would be obviously more variable than the original sound and must take into account the mechanical limitations of our phonatory system as well as of our knowledge about the possible succession of actions order that produces intelligible words and sentences.

Obviously, the PerAc or any other one architecture is nothing without a good model for all the neural groups involved. Consequently, we estimate important to simultaneously improve the design of interesting neural group and the architecture they fit in. For instance, topology must be preserved at each level of the system (see [14,15]) in order to simplify task planning ([16,53]). Besides, the architecture described in this paper only allow the robot to react immediately without taking into account motivational information. The robot can only do tasks which can be reduced to finding sensorimotor regularities. In other papers [15,38,40], we describe how motivational nodes can influence the shape recognition. This will for instance let the robot to prefer doing particular things such as homing or feeding. For generalization to even more abstract tasks, see [79]. In addition, associative searches and goal propagation found at the cortical level [8,26,76] in the brain should be added to really allow the architecture to be very general. This process would lead to the definition of an explicit parallel language to “program” animal robots with adaptation and autonomy capabilities.

From another point of view, all those examples lead naturally to the question of the definition of emergence, central to the constructivist paradigm [19] in which it represents an alternative to the classical cognitivist paradigm [6]. At first sight it corresponds to the application of the holistic principle — the whole possesses feature that cannot be found in any of its subcomponents —, but there is no clear undiscussed definition of emergence. Nevertheless, we have given two examples of phenomena which exhibit some kind of emergence, as the features of those systems cannot be explained by any of their components.

In the vision application, the system in which

emergence appears is the vision system plus the image itself. The approach is based on Gestalt Theory, according to which the image as a whole contains more information than the parts of the image. This includes all the possible ambiguities and optical illusions which are not present in subcomponents of the image. Moreover, the optical illusions are due to the vision system performing its operations on the image. This is a good example of a structural coupling between the system and its “environment” cherished by Maturana and Varela [19]. All these examples, which by no means pretend to propose a clear definition of emergence, at least show that it is necessary to set up global solutions to cognitive problems. One cannot be content with studying only a function to be approximated, or the behaviour of a single processing element, since it cannot be known a priori which role or how important this element should be in the whole system. Moreover, by studying only subcomponents, one loses the opportunity to use the important dynamical properties of the system [14].

Future work will be concerned by finding ways to extend that kind of networks to more complex tasks, always relying on the constructed level to obtain the next. A particular attention will be paid to introduce goal generation and resolution [8] and to improve their cheap limbic systems which is the major element to control the robot motivations and to allow leaning and adaptation capabilities. Next, an obstacle avoidance mechanism must be added [13] and conflicts between the navigation level and the reflex level of obstacle avoidance should be solved. At last, the introduction of imitation [80] and communication capabilities (in the same bottom up approach as the rest of our development) could allow the robot to learn the basis of a language and then to access to symbolic objects or representations which would be a way to drastically enhance the robot’s cognitive capabilities. We know the wasp nervous system contains about 100 000 neurons and Prometheus’ N.N. involves about 60 000 neurons and about 1000 times more connexions (part of them have been rewritten in C programs to reduce the computation time). Henceforth, it appears that the design of robots with the intelli-

gence of an insect or a rat may not be completely out of reach. Obviously, this would require the proper use of the advantages provided by the algorithms discovered in computer sciences or in classical AI to short cut some of the simulation problems of N.N.

## Acknowledgements

This research is supported by the Swiss National Funds PNR 23 program.

We would like to thank Arnaud Revel for his help, and J.P. Banquet and Y. Burnod from the neuroscience institute at Jussieu (Paris VI) for their helpful comments on the hippocampus modelization.

## Appendix A

### A.1. A diffusion mechanism to find the focus points

The process which extracts the focus points from the contour image [C] is equivalent to the convolution of the contour image with a local filter. Its shape is a difference of two gaussian functions [DOG] and allows to detect corner of a special resolution according to the gaussian parameters. The diffusion equation is:

$$D_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}}.$$

Thus, the mask associated to the feature points extraction is:  $DOG(x, y) = D_{\sigma_1}(x, y) - D_{\sigma_2}(x, y)$  where  $\sigma_1$  and  $\sigma_2$  are two positive constants.

The convolution between the contour image and the DOG filter is:  $[F] = [C] * [DOG]$

$$F(x, y) = \sum_{j=1}^{j_{\max}} \sum_{i=1}^{i_{\max}} DOG(i, j) \cdot \left( x + \frac{i_{\max}}{2} - i, y + \frac{j_{\max}}{2} - j \right),$$

where  $i_{\max}$  and  $j_{\max}$  define the size of the DOG mask.

A competition between neurons on [F] map belonging to a  $(i_{\max} j_{\max})$  neighbourhood allows to find local maxima. Their intensity grows with the corner's sharpness and well-definedness at the filter's resolution.

### A.2. Formal description of the perac blocks

#### Basic operators: rotation, projection and selection

In both vision and navigation N.N., we use a mechanism to simulate a space transformation of the manipulated vectors: we either select one part of a huge vector (information flow) or process a rotation. Those mechanisms can be implemented in a neural fashion by the use of "switching" neurons such as sigma Pi units [7], or by any appropriate learning or connecting mechanism, or by an action like rotating the head or focusing the attention on a detail of a panoramic image. Mathematically speaking, these operations can all be reduced to a matrix product between the input vector and a projection matrix. In the case of the selection of a part of the input vector according to a translation parameter  $t$ , we have:

$$Y = [\text{Proj}] \cdot X$$

$$\text{with } [\text{Proj}] \text{ defined by } \begin{cases} \text{Proj}_{i+t, i} = 1 \\ \text{Proj}_{i, j \neq i-t} = 0 \end{cases}$$

$\dim(Y)$  can be different from  $\dim(X)$  in the case we want to extract a sub-picture from  $X$ .

In the PerAc architecture the  $t$  parameter comes from the action flow. It is associated to the selection of a part of the analyzed image.

#### A probabilistic Winner Take All

The quantization precision is not really important because the use of probabilistic neurons allows to make movements with a precision that only depends on sampling time. For example, if the robot can only move forward or 90° left or right, it can move in the direction of 25° if the left-neuron is activated randomly twice more than the straight-ahead-neuron. The precision of such a probabilistic control can be very efficient and seems to explain human and animal manipulation precision [34] (Fig. 29).

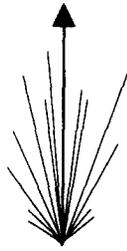


Fig. 29. Example of vectors movements that can be proposed by the WTA. The vertical line represent the average direction of the set of vector. Its precision is higher than the precision of the vector resolution [34].

It is an example in which the time integration can compensate a low instantaneous resolution. If  $S_i$  represents the analog activity of the neuron before the competition then the neuron  $i$  is activated if  $p_i(t) = 1$ . That mechanism can be written as follows:

$$p_i(t) = \begin{cases} 1 & \text{if } \text{rnd} < \left( \frac{S_i(t)}{S_{\max}(t)} \right)^k \\ 0 & \text{otherwise} \end{cases}$$

where  $S_{\max}(t) = \text{Max}\{S_i, i = 1, \dots, n\}$  and  $k$  is a constant. If  $k$  is high (typically around 100) then  $p_i$  tends to give the same result as a deterministic WTA. If  $S_i$  is the maximum then  $p_i = 1$ , otherwise  $p_i = 0$ . The trajectory of the movement from a point  $M$  at time  $t_0$  to a point  $N$  at time  $t_1$  is:

$$MN \vec{N} = \sum_{t=t_0}^{t_1} \sum_{i=1}^n p_i(t) \cdot V_i$$

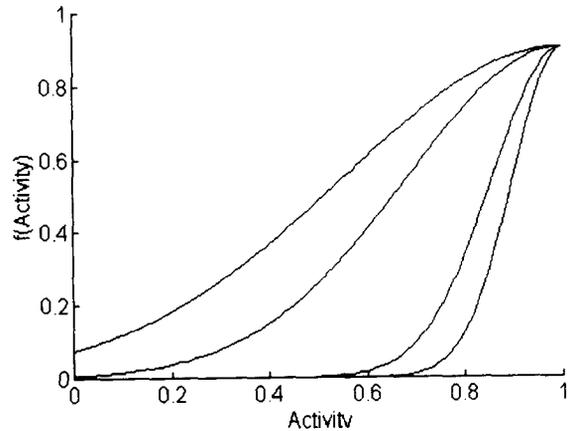


Fig. 30. Activation function for different vigilance values (1, 0.5, 0.1, 0.05) in the case of a neuron that has previously learned a shape ( $\bar{D}_k = 0.9$ ).

*Neural pattern matching*

The matching can either be computed as a scalar product or a distance measure:

$$\text{Act} = \frac{W \cdot X}{\sum W} \text{ or } \text{Act} = 1 - \frac{|W - X|}{N}$$

where  $N$  is the number of input.

The normalization of the activity can be more complex in the case of the scalar product because of the possibility of having inhibitory weights. The activation function  $f$  of a neuron  $k$  is:

$$f_k[\text{Act}](t) = \bar{D}_k(t) \cdot e^{-\frac{\text{vigilance}}{2\sigma} \left( \frac{1 - \text{Act}}{1 - \bar{D}_k(t)} \right)^2}$$

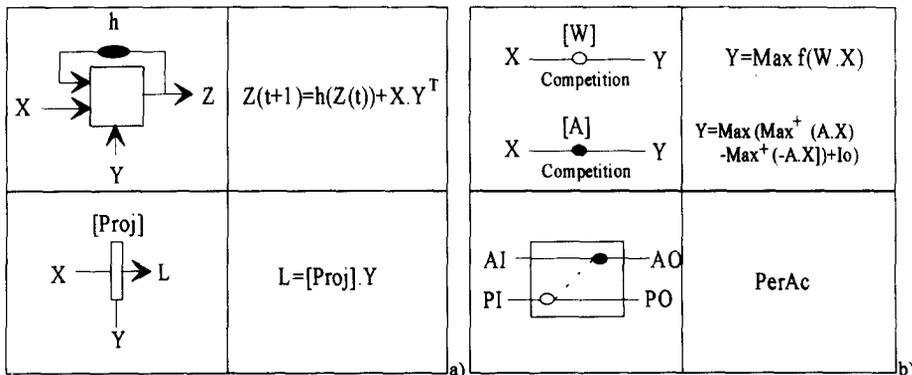


Fig. 31. (a) different operators used to construct the PerAc architecture. (b) Summary of the PerAc block. AI and AO are, respectively, the action input and output whereas PI and PO are the perception input and output.

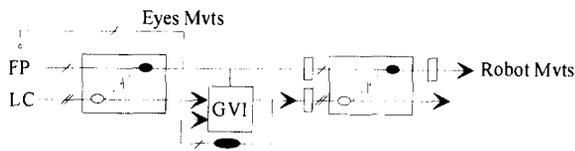


Fig. 32. symbolic representation of the N.N. used to control Prometheus in the case all the snapshots associated to each landmarks are different from each others.

where  $\bar{D}k(t)$  depends of the neuron history. If the neuron has learned a shape  $\bar{D}k(t)$  is high and the neuron is highly selective. At the beginning, we take  $\bar{D}k(t) = 0.5$  and after learning its value become  $\bar{D}k(t) = 0.9$  for instance.

The mean square value of the  $f$  function is  $\sigma$  when vigilance = 1 and tends to infinity when vigilance tends to 0. When the vigilance is low the curve tends to look like a line because all the values are on the higher part of the curve. Nevertheless, the competition between the neurons allows to find the maximum value of activation but such a high precision process does not seems us to be biologically possible. It should be simpler to consider that the nature of the non linearity depends on the vigilance value (Fig. 30).

### A.3. Synthesis

Fig. 31 sums up the different elements used in and around the PerAc boxes.

Fig. 32 represents the vision and the navigation networks used to simulate the robot's "brain". It is clear that both networks are the same. We only need to add preprocessing mechanisms.

## References

- [1] R. Chatila, Control architecture for autonomous mobile robots, *Proc. of Perception to Action*, Lausanne, P. Gaussier and J.D Nicoud, eds. (IEEE Computer Society Press, 1994) 254–265.
- [2] J.C. Latombe, *Robot Motion Planning* (Kluwer, Dordrecht, 1991).
- [3] R.A. Brooks, A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation* 2 (1986) 14–23.
- [4] J.S. Albus and W.G. Rippey, RCS: A reference model architecture for intelligent control, *Proc. of Perception to Action*, Lausanne, P. Gaussier and J.D Nicoud, eds. (IEEE Computer Society Press, 1994) 218–229.
- [5] S. Harnad, The symbol grounding problem, *Physica D-42* (1990) 335–346.
- [6] G. Lakoff, *Women, Fire and Dangerous Things: What Categories Reveal about the Mind* (The University of Chicago Press, Chicago, 1987).
- [7] D.E. Rumelhart et al., *Parallel Distributed Processing* (MIT-Press, Cambridge, CA, 1986).
- [8] Y. Burnod, An adaptive neural network: The cerebral cortex, *Masson*, Paris (1989).
- [9] H. Kitano, Challenges of massive parallelism, *IJCAI93* (1993) 813–834.
- [10] D. McFarland, Animal robotics: from self-sufficiency to autonomy, *Proc. of Perception to Action*, Lausanne, by P. Gaussier and J.D. Nicoud, eds. (IEEE Computer Society Press, 1994) 47–54.
- [11] T.L. Anderson and M. Donath, Animal behaviour as a paradigm for developing robot autonomy, *Robotics and Autonomous Systems* 6 (1990) 145–168.
- [12] M. Minsky and S. Papert, *Perceptrons* (MIT Press, Cambridge, CA, 1969).
- [13] P. Gaussier and S. Zrehen, A topological neural map for on-line learning: emergence of obstacle avoidance in a mobile robot, *SAB 94*, Brighton, 1994.
- [14] S. Zrehen and P. Gaussier, Why topological maps are useful for the design of autonomous agents, *Proc. of Perception and Action*, Lausanne, P. Gaussier and J.D Nicoud, eds. (IEEE Computer Society Press, 1994) 230–241.
- [15] S. Zrehen, Elements of brain design for autonomous agents, *PhD dissertation*, EPF-Lausanne (1995).
- [16] L. Steels, Exploiting analogical representations, *Robotics and Autonomous Systems* 6 (1990) 71–88.
- [17] G. Schöner, M. Dose and C. Engels, Dynamics of behavior: theory and applications for autonomous robot architectures, *Robotics and Autonomous Systems* (1995) this issue.
- [18] G. Carpenter and S. Grossberg, A massively parallel architecture for a self-organizing neural pattern recognition machine, *CGVIP* 37 (1987) 54–115.
- [19] Maturana and F. Varela, *The Tree of Knowledge*, Shambhala, ed. (1987).
- [20] J. Stewart, Life = Cognition: The epistemological and ontological significance of artificial life, *Proc. SAB'91 Paris*, P. Bourguine and F. Varela, eds. (MIT Press, Cambridge, MA, 1991) 475–483.
- [21] J. Stewart, The implications for understanding high-level cognition of a grounding in elementary adaptive systems, *Robotics and Autonomous Systems* (1995) this issue.
- [22] R. Pfeifer and C. Scheier, From perception to action: the right direction?, *Proc. of Perception and Action*, Lausanne, P. Gaussier and J.D Nicoud, eds. (IEEE Computer Society Press, 1994) 1–11.

- [23] J.S. Albus, Outline for a theory of intelligence, *IEEE Trans. on Syst. Man and Cybern.* 21(3) (1991) 473–509.
- [24] G. Edelman, *Neural Darwinism: The Theory of Neuronal Group Selection* (Basic Books, New-York, 1987).
- [25] R. Hecht-Nielsen, Counterpropagation networks, *Applied Optics* 26(23) (1987) 4979–4984.
- [26] D.H. Ballard, Cortical connections and parallel processing: Structure and Functions, *Behavioral and Brain Sciences* 9 (1986) 67–120.
- [27] J. Piaget, *La naissance de l'intelligence chez l'enfant* (Delachaux et Niestle Editions, Geneve, 1936).
- [28] D.L. Deneubourg et al., The dynamics of collective sorting: robot-like ants and ant-like robots, *From Animals to Animats, Proc. of Simulation of Adaptive Behavior* (MIT Press, Cambridge, CA, 1991).
- [29] R. Whener and F. Råber, Visual spacial memory in desert ants, cataglyphis bicolor (Hymenoptera: Formicidae), *Experientia* 35 (1979) 1569–1571.
- [30] J.L. Gould and W.F. Towne, Honey bee learning, *Advances in Insect Physiology* 20 (1979) 55–86.
- [31] C.D. Gilbert, Microcircuitry of the visual cortex, *Annual Review of Neuroscience* 6 (1983) 217–247.
- [32] D.C. Van Essen and J.H.R. Maunsell, Hierarchical organization and functional streams in the visual cortex, *TINS* (1983) 370–375.
- [33] L. Schwartz, Computational anatomy and functional architecture of striate cortex: a spacial mapping approach to perceptual coding, *Vision Research* 20 (1980) 645–669.
- [34] A. Georgopoulos, Neural interpretation of movement: role of motor cortex in reaching, *FASEB Journal* 13 (1989) 2846–2857.
- [35] P. Gaussier and A. Revel, Performing and testing hypotheses: A neural brain model to control animats, Internal report ETIS, 1995.
- [36] A. Lieberman, *Learning: Behavior and Cognition*, Brooks and Cole, eds. (Publishing Compagny, 1994).
- [37] P. Gaussier and S. Zrehen, *A Constructivist Approach for Autonomous Agents*, N. Thalmann, ed. (Wiley, New York, 1994).
- [38] P. Gaussier and S. Zrehen, Navigating with an animal brain: a neural network for landmark identification and navigation, *Proc. of the Intelligent Vehicles'94 Symposium*, Paris (1994).
- [39] P. Gaussier and J.-P. Cocquerez, Neural networks for complex scene recognition: simulation of a visual system with several cortical areas, *Proc. of IJCNN*, Baltimore, Vol. 3 (1991) 233–259.
- [40] P. Gaussier, Simulation d'un système visuel comprenant plusieurs aires corticales, *Doctoral Thesis*, Paris XI, Orsay (1992).
- [41] M.W. Oram and D.I. Perret, Modeling visual recognition from neurobiological constraints, *Neural Networks* 7(6/7) (1994) 945–972.
- [42] A. Treisman, Features and objects: the fourteenth Bartlett memorial lecture, *The Quarterly Journal of Experimental Psychology* 40A(2) (1988) 201–237.
- [43] S. Grossberg and E. Mingolla, Neural dynamics of surface perception: boundary webs, illuminants, and shape-from-shading, *CVGIP* 37 (1987) 116–165.
- [44] M. Seibert and A.M. Waxman, Spreading activation layers, visual saccades, and invariant representations for neural pattern recognition systems, *Neural Networks* 2 (1989) 9–27.
- [45] R.J. Watt and M.J. Morgan, The recognition and representation of edge blur: evidence for spacial primitives in human vision, *Vision Research* 23(12) (1983) 1465–1477.
- [46] M.J. Farrah and K.M. Hammond, Mental rotation and orientation-invariant object recognition: Dissociable processes, *Cognition* 29 (1988) 29–46.
- [47] S. Grossberg, F. Guentlu, D. Bullock and D. Greve, Neural representations for sensory-motor control, II: Learning a head-centered visuomotor representation of 3D target position, *Neural Network* 6 (1993) 43–67.
- [48] R.J. Baron, Visual memories and mental images, *International Journal of Man-Machine Studies* 23 (1985) 275–311.
- [49] D. Norton and L. Stark, Eye movements and visual perception, *Scientific American* 224(6) (1971) 34–43.
- [50] D. Demigny, J. Devars, L. Kessal and J.F. Quesne, Implantations temps réel du filtre de lissage d'image de Nagao, *Revue de Traitement du Signal* 10(4) (1993) 319–329.
- [51] C. Joulain, Apprentissage d'une tâche de ramassage d'objet par un robot autonome, *Mémoire de DEA TIS* (1995).
- [52] A. Baloch and A. Waxman, Visual learning, adaptive expectations, and behavioral conditioning of the mobile robot MAVIN, *Neural Networks* 4 (1991) 271–302.
- [53] G. Schöner and M. Dose, A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion, *Robotics and Autonomous Systems* 10 (1992) 253–267.
- [54] R.C. Arkin, Motor schema-based mobile robot navigation, *International Journal of Robotics Research* (1987) 92–112.
- [55] E.C. Tolman, Cognitive maps in rats and men, *Psychological Review* 55(4) (1948) 189–208.
- [56] M.J. Mataric, Learning a distributed map representation based on navigation behaviors, *Proc. USA-Japan Symposium on Automation*, Kyoto, Japan (1990) 499–506.
- [57] M.J. Mataric, Navigating with a rat brain: A neurobiologically-inspired model for robot spacial representation, *Proc. Ist SAB Conference* (1990) 169–175.
- [58] N. Tinbergen, *The Study of Instinct* (Clarendon Press Publishers, Oxford, 1969).
- [59] B.A. Cartwright and T.S. Collett, Landmark learning in bees, *Journal Comp. Physiology* 151 (1983) 521–543.
- [60] R.G.M. Morris et al., Place navigation impaired in rats with hippocampal lesions, *Nature* 297 (1982).
- [61] J. O'Keefe, Computations the hippocampus might perform, *Neural Connections, Mental Computation*, Nadel et al., eds. (MIT Press, Cambridge, CA, 1989) 225–284.
- [62] J. O'Keefe and L. Nadel, *Hippocampus as a Cognitive Map* (Clarendon Press Publishers, Oxford, 1978).
- [63] G.J. Quirk, R.U. Muller and J.L. Kubie, The firing of

- hippocampal place cells in the dark depends on the rat's recent experience, *Journal of Neuroscience* 10(6) (1990) 2008–2017.
- [64] T. Levitt, D. Lawton, D. Chelberg, K. Koitzsch and J.W. Dye, Qualitative navigation 2, *Proc. DARPA Image Understanding Workshop*, Los Altos, CA (1988) 319–326.
- [65] R.C. Nelson, Visual homing using an associative memory, *Biol. Cybern.* 65 (1991) 281–291.
- [66] I.A. Bachelder and A.M. Waxman, Mobile robot visual mapping and localization: A view-based neurocomputational architecture that emulates hippocampal place learning, *Neural Networks* 7(6/7) (1994) 1083–1099.
- [67] I.A. Bachelder and A.M. Waxman, A view-based neurocomputational system for relational map-making and navigation in visual environments, *Robotics and Autonomous Systems* (1995) this issue.
- [68] D. Zipser, A computational model of hippocampal place fields, *Behavioral Neuroscience* 99(5) (1985) 1006–1018.
- [69] B.L. McNauton, Neural mechanisms for spacial computation and information storage, *Neural Connections, Mental Computation*, Nadel et al., eds. (MIT Press, Cambridge, CA, 1989) 285–350.
- [70] N. Burgess, M. Recce and J. O'Keefe, A model of hippocampal function, *Neural Network* 7(6/7) (1994) 1060–1081.
- [71] M. Tsukada, A proposed model of Hippocampal-cortical memory system and temporal pattern sensitivity of LTP in hippocampal neurons, *Concepts in Neuroscience* 3(2) (1992) 213–224.
- [72] M.A. Penna and J.W. Wu, Models for map building and navigation, *IEEE Trans. on Systems, Man, and Cybern.* 23(5) (1993).
- [73] J.A. Feldman, Connectionist models and parallelism in high level vision, *CVGIP* 31 (1985) 178–200.
- [74] R. Linsker, From basic network principles to neural architectures: emergence of spatial-opponent cells, *Proc. Natl. Ac. Sci.* 83 (1986) 7508–7512.
- [75] K. Fukushima and S. Miyake, Neocognitron: A new algorithm for pattern recognition tolerant on faults and shifts in position, *Pattern Recognition* 15(6) (1982) 455–465.
- [76] G. Edelman, Bright air, brilliant fire: on the matter of mind, *Basic Books* (1992).
- [77] J. Kodjabachian and J.A. Meyer, Evolution and development of control architectures in animats, *Robotics and Autonomous Systems* (1995) this issue.
- [78] S. Maeda, Improved articulatory model, *Journal of the Acoustical Society of America* 81(S1) (1988).
- [79] J.G. Taylor, Goals, drives and consciousness, *Neural Networks* 7(6/7) (1994) 1181–1190.
- [80] K. Dautehahn, Getting to know each other: Artificial social intelligence for autonomous robots, *Robotics and Autonomous Systems* (1995) this issue.



**Philippe Gaussier** is born in 1967 in Marseille, France. He received the M.S. degree in electronic from Aix-Marseille university and a Ph.D. degree in computer science from the University of Paris XI (Orsay) in 1989 and 1992, respectively. From 1992 to 1994, he conducted research in Neural Network applications and in control of autonomous mobile robots at the Swiss Federal Institute of Technology. He was the Program chairman of the Perception Action (PerAc'94) conference organized in September 94 in Lausanne. He is now an assistant Professor at the ENSEA (national engineering school in electronics and its applications) in France. He has worked on various problems related to industrial applications of N.N., visual recognition, robot navigation, neural modeling ... Currently his research interests include modelization of visual and navigation systems and their implementation on real robots.



**Stephane Zrehen** was born in 1967 in Paris, France. He received his Physics Engineer degree from EPFL in 1990. For two years, he worked in consulting for industrial companies in Switzerland. Since 1992, he has been working as a research and teaching assistant at the Microcomputing lab at EPFL. He was in charge of the Postgraduate Course on Biological and Artificial Neural Nets. He received his PhD from EPFL in June 1995. His research interests are to keep on developing animal-like intelligent control systems for mobile robots.