Navigation and Planning in an Unknown Environment Using Vision and a Cognitive Map

Nicolas Cuperlier, Mathias Quoy, and Philippe Gaussier

ETIS-UMR 8051
Universit de Cergy-Pontoise - ENSEA
6, Avenue du Ponceau
95014 Cergy-Pontoise France `cuperlier@ensea.fr`

Summary. We present a framework for Simultaneous Localization and Map building of an unknown environment based on vision and dead-reckoning systems. An omnidirectional camera gives a panoramic image from which no a priori defined landmarks are extracted. The set of landmarks and their azimuth relative to the north given by a compass defines a particular location without any need of an external environment map. Transitions between two locations are explicitly coded. They are simultaneously used in two layers of our architecture. First to construct, during exploration (latent learning), a graph (our cognitive map) of the environment where the links are reinforced when the path is used. And second, to be associated, on an another layer, with the integrated movement used for going from one place to the other. During the planning phase, the activity of transition coding for the required goal in the cognitive map spreads along the arcs of this graph giving transitions (nodes) an higher value to the ones closer from this goal. We will show that, when planning to reach a goal in this environment is needed, the interactions of these two levels can lead to the selection of multiple transitions corresponding to the most activated ones according to the current place. Those proposed transitions are finally exploited by a dynamical system (neural field) merging these informations. Stable solution of this system gives a unique movement vector to apply. Experimental results underline the interest of such a soft competition of transition information over a strict one to get a more accurate generalization on the movement selection.

1 Introduction

Path planning requires from the agent or the robot to select the appropriate action to perform. This task might be complex when several actions are possible, and so different approaches have been proposed to choose what to do next. Experiments carried out on rats have led to the definition of cognitive maps used for path planning [21]. Most of cognitive maps models are based on graphs showing how to go from one place to an other [2, 3]. They mainly differ in the way they use the map in order to find the shortest path, in the way

H.I. Christensen (Ed.): European Robotics Symposium 2006, STAR 22, pp. 129–142, 2006.

they react to dynamical environment changes, and in the way they achieve contradictory goal satisfactions. One can refer to [9, 12] for a comparative review of localisation and mapping models. Many methods rely on the combination of different algorithms that have to be triggered appropriately (and concurrently) when necessary. For instance, localisation may involve different sensors (laser, ultra-sound, visual feature recognition ...) that have to be chosen appropriately. Some works use ruled-based algorithms, classical functional approach, that can exhibit the desired behaviors, we will not discuss them in this paper, but one can refer to [8].
Instead, other works try to look at what the nature does by taking inspiration from neurobiology to design control architectures. There are at least two reasons for this:

- first, getting robust, adaptive, opportunistic and ready-made solutions for control architecture.
- second, if robotic results can be compared to experimental results involving several parts of the brain, which are generally difficult to study due to its complexity, it can help neurobiologists to understand how a neurobiological model behaves.

Hence we propose here a unified neuronal framework based on an hippocampal and and prefrontal model where vision, place recognition and dead-reckoning are fully integrated (see Fig. 4 for an overview of the architecture). This model relies on a topological map: the environment is coded via a set of distinctive nodes and by the way a robot can go from one node to another. In our work, those nodes are not directly places of the environment but rather the transition between two of them. No cartesian metric informations and no occupancy grid are used to construct the map. Localisation is achieved using a biomimetic model designed to emulate the neural activity of particular neurons found in the rat hippocampus named place cells. Those cells learn direction and identity (recognition) of punctual landmarks leading to a place definition. A key point for the understanding of this model is the distinction between transition coding for the succession of two place cells at the recognition (visual) level and motor transition encoding, on a motor level, the integrated movement performed to go from one place to the other. Whereas these two kinds of transition are strongly dependent and linked in a unique way, they do not have the same modality: one is related to vision coding and the other is related to motor coding. Keeping this basic distinction in mind, we can list the assets of this model:

- autonomous landmarks extraction based on characteristic points (section 2)
- autonomous place building via place-cells-like neurons: there are no a priori predefined squares, or world model (section 3)
- autonomous creation of transitions at recognition level (recognition transitions). A neuron codes the succession of two recognized place cells without any combinatorial explosion (section 4).

- autonomous cognitive map building based on those recognition transitions between places, giving topological informations like adjacency of two transitions linked (i.e: if transitions AB and BC are linked: adjacency of the destination place cell AB with the place cell of origine of transition cell BC). But, this map can also give a kind of metric via the value of the arc of this graph. (section 5).
- autonomous association of recognition transitions with integrated movement giving motor transitions which can be used for planning (section 6).
- autonomous planning using both the cognitive map (graph of recognition transitions) and the corresponding motor transitions (section 7)
- stable movement given by the stable fixed point solution of a dynamical system (section 8)

Drawbacks will be left for conclusion.

2 Autonomous Landmark Extraction Based on Characteristic Points

Images are taken by a panoramic camera at a low resolution. This allows to handle lighter images that may be processed in a real time. In order to eliminate problems induced by luminance variability, we only use the gradient image as input of the system (a 1500×240 pixels image extracted from the 640×480 pixels panoramic image which is originally circular). Two processes then occur in parallel:

- First, curvature points are extracted from this gradient image by Difference Of Gaussian (D.O.G) filtering. Those curvature points are robust focal points due to the low resolution. Each focal point is the center of a 32x32 pixels small image giving the local visual area extracted around it (cf. circles on Fig. 1) . This image is binarized through a log-polar transform [19, 11] and next it is learned on a neuron coding for this landmark. A soft competition between landmark neurons, allowing several interpretations of a given local snapshot, is then computed
- Second, each landmark is linked with its angular position relative to the north given by a compass [20, 14].

In a panoramic image, 32 (landmark, azimuth) pairs are extracted (see Fig. 1). Thus, this visual system provides both a *what* and a *where* informations: the recognition of a 32×32 pixels snapshot in log-polar coordinates, and the azimuth of the focal point. *What* and *where* informations are then merged in a product space that memorizes the incoming inputs during a given time. The number of landmarks needed is a balance between the robustness of the algorithm and the speed of the process. If all landmarks were fully recognized, only three of them will be needed. But as some of them may not be recognized in case of changing conditions (luminance; occlusion), taking a greater number

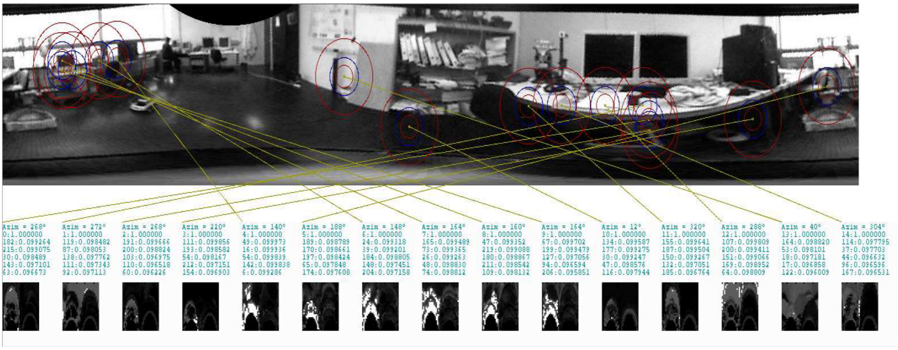


Fig. 1. Image taken from a panoramic camera. Below are 15 examples of 32 × 32 log-polar transforms taken as landmarks and their corresponding position in the image.

is enough to guarantee the robustness. Moreover the log-polar transform gives some rotation and depth robustness.

3 Autonomous Place Building

Each set of (landmark, azimuth) pairs, merged in the product space, is learned and thus characterizes one location. The neuron coding for this location is called a "place cell" (PC) as the one found in the rat's hippocampus [14]. Place cell's activity is the result of a matching function computing the distance between the learned set and the current set (the distance is computed only on the recruited neurons). Thus, the activity of the k^{th} PC can be expressed as follows:

$$P_k = \frac{1}{l_k}\left(\sum_{i=1}^{N_L} \omega_{ik}.f_s(L_i).g_d(\theta_{ik}^L - \theta_i)\right) \tag{1}$$

with $l_k = \sum_{i=1}^{N_L} \omega_{ik}$ the number of landmarks used for the k^{th} PC, where $\omega_{ik} = \{0, 1\}$ expresses the fact that landmark i has been used to encode PC k, with N_L the number of learned landmarks, L_i the activity of the landmark i, $f_s(x)$ the activation function of the neurons in the landmark recognition group, θ_{ik}^L the learned azimuth of the i^{th} landmark for the k^{th} PC, θ_i the azimuth of the current local view interpreted as the landmark i. d is the angular diffusion parameter which defines the shape of the function $g_d(x)$. The purpose of $f_s(x)$ and $g_d(x)$ is to adapt respectively the dynamics of *what* and *where* groups of neurons. They are defined as follow :

$$g_d(x) = \left[1 - \frac{|x|}{d.\pi}\right]^+$$
$$f_s(x) = \frac{1}{1-s}\left[x - s\right]^+$$

where $[x]^+ = x \ if \ x > 0$, $and\ 0\ otherwise.$

The s parameter rescales the activity of the landmark neuron over s between 0 and 1. The d parameter modulates the weight of the angular displacement.

More information on the neural model for place cell coding may be found in [10, 7].

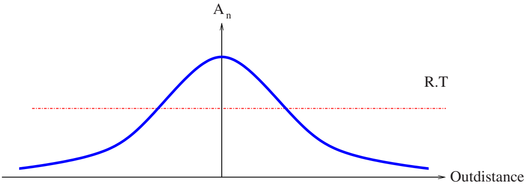


Fig. 2. Idealised shape of a place cell (A_n) response, according to the distance to the exact location where it has been learned. This cell has its maximal response in this place.

Recruitment of a new place cell for encoding a new location is performed autonomously, without any external signal. If the activities of all previously learned place cells are below a *given* recognition threshold (R.T), then a new neuron is recruited for coding this recently discovered location (see Fig. 2). Hence, the density of location learned depends on the level of this threshold. But it depends also on the position of the robot in the environment. Namely, more locations are learned near walls or doors because a fast change in the angular position of near landmarks, or in the (dis)appearance of landmarks may occur. In other locations, small changes produce a small variation in the place cell activity (see Fig. 2) . If at a given place, several cells responds with an activity greater than the R.T, a competition takes place so that the most activated cell wins and codes this location. These cells are created during the exploration of the unknown environment. We use random exploration but naturally other kind of algorithms can be used. At the end of this task, the environment is fully covered by place cells, so that in any part of it a place cell responds specifically for it (see Fig. 3.).

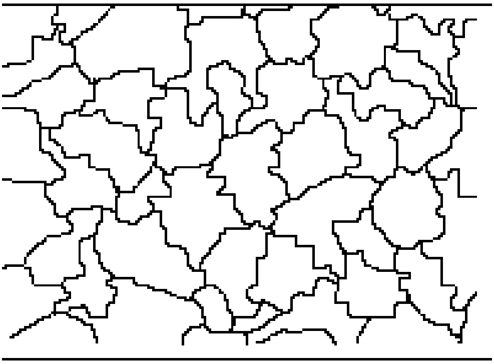


Fig. 3. A simulated environment fully explored. Each region represents the response domain for which a particular place cell win the competition for recognition. After a full exploration the entire environment is covered by the place cell population.

4 Autonomous Building of Transitions at Recognition Level

Two successively reached places are coded by a *transition cell* (see Fig. 4). Hence two successively reached places (A, then B) are coded by a $t$ransition cell (AB). A relevant question is about the growth of the number of these cells. Before to show the experimental results, we already can make two important remarques about this growth.

- First, this growth is intimately linked with the number of place cells. This last one mainly depends on two parameters:
  - the value of R.T: the higher R.T is the bigger is the number of cell created.
  - the complexity of the environment: the number and the location of its landmarks and the number of obstacles found inside.
- Second, we do not create all possible transitions but only physically feasible transitions between “adjacent” place cells. And since the number of a place cell neighbours is necessary limited (see Fig 3), the number of transition created is also limited.

Hence, we have studied the ratio between created transition cells over created place cells for three environments of increasing complexity according to their obstacle configuration (a single, a two and a four room environment). The tests have been performed in simulation using a virtual robot (or animat). For each simulation, 10 animats have explored their environment for 50000 cycles. This number has been chosen high enough to be sure that the animat has learned a complete cognitive map. The results shown here are the average on these 10 animat results. The ratio remains stable around the mean value 5.45 for all environments (see table 1.).

Env / RT	0.97
nbp	133.8(2.85)
nbt	735.8(19.80)
ratio	5.49(0.06)
nbp	606.2(6.89)
nbt	3389.2(56.38)
ratio	5.59(0.08)
nbp	643.7(9,88)
nbt	3281,2(48,80)
ratio	5.09(0,04)

Table 1. Ratio of the number of place cells (nbp) created over the number of transitions created (nbt) according to the number of room in the environment: with one room (top line), with two rooms (middle line) and four rooms (bottom line). Standard deviation is given into brackets. This ratio remains stable. There are five times more transition cells than place cells.

Indeed, only a few transitions can be created from a given place cell, since a transition is a link between "adjacent" place cells and since the number of a place cell neighbours is necessary limited. So there is no combinatorial explosion on the number of created transitions and we do not need a "full matrix" to create the transitions: a matrix of only M*N is enough. With M the number of place cells, and N the maximal number of possible neighbours.

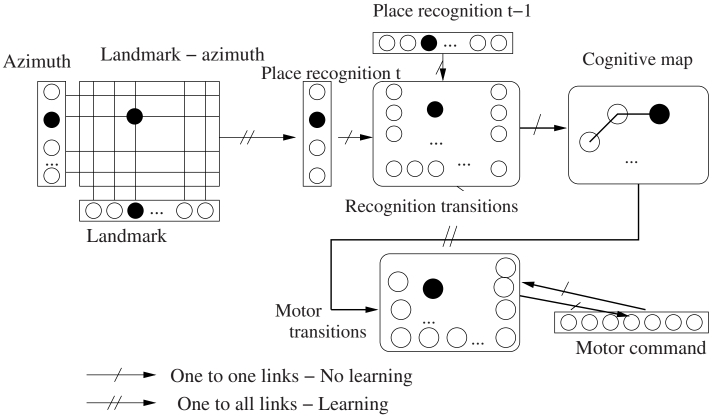


Fig. 4. Sketch of the model. From left to the right: merging landmarks and their azimuth in a product space, then learning of the corresponding set on a place cell. Two successive place cells (the one at time t and the previous one at $t-1$) define a transition cell. They are used to build up the cognitive map and are also linked with the corresponding motor transition that integrated the movement performed.

5 Autonomous Cognitive Map Building

Since our robotic model is inspired from the animat approach [13] we use three contradictory animal like motivations (eating, drinking, and resting). Each one is associated with a satisfaction level that decreases over time and increases when the robot is on the proper source. When a level of satisfaction falls bellow a given threshold, the corresponding motivation is triggered so that the robot has to reach a place allowing to satisfy this need. Hence this place becomes the goal to reach. More sources can be added and one can increase the number of sources associated with a given motivation. Each time a new transition is created, a new node is recruited in the cognitive map. This node is then linked with the previous transition used. When a transition leads to a place cell where a source can be found, a link between the corresponding motivation and the most active node on the map is created and set to one (latent learning), otherwise this link is set to zero. After some time, exploring the environment leads to the creation of the cognitive map (see Fig. 5). This map may be seen as a graph where each node is a transition and the arcs the fact that the path between these two transitions was used. We can give a fixed value (lower than one) to each link at the creation time. This value is increased if the link is used, and decreased if it is not. After some time passed in the environment, some links are reinforced. These links correspond to paths that are often used. In particular, this is the case when some particular locations have to be reached more often than others (see section 7) [15].

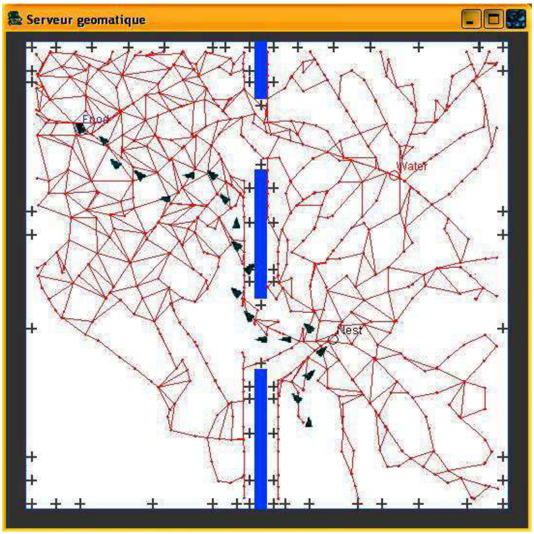


Fig. 5. Cognitive map (in red) build by exploration of the environment. The triangles give the successive robot positions starting from the right to the goal (on the left). Landmarks are represented by blue crosses. The blue rectangle are obstacles.

6 Autonomous Motor Transitions Creation

Each of these cells is linked with the direction used to go from the starting location to the ending location. For instance, going from place A to place B creates a recognition transition cell AB and the corresponding node on the map. In the same time another transition cell is created on the motor level. This motor transition associates the direction (relative to the north) for going from A to B with the node AB on the map. This direction integrates all direction changes performed between A and the creation of B using robot wheel encoders to compute elementary displacement vectors. Direction changes are induced by a new movement vector generated by the exploration mechanism (random exploration) or due to the obstacle avoidance mechanism. Hence the integrated vector (I.V) takes care of all these movement changes. Each time this transition is performed, the I.V. is averaged with the corresponding learned direction. The I.V is then reset when entering a different place cell. The norm of this vector is also computed in the same way.

7 Autonomous Planning Using the Cognitive Map and Motor Transitions

When a goal has to be reached, the transitions leading to it are activated via the links learned during exploration between those transitions and the corresponding motivation. This activation is then diffused on the cognitive map graph, each node taking the maximal incoming value which is the product between the weight on the link (lower than one) and the activity of the node sending the link. After stabilization, this diffusion process gives the shortest path between all nodes and the goal nodes. This is a neural version of the Bellman-Ford algorithm [5, 17] (see Fig.6).

When the robot is in a particular location A, all possible transitions beginning with A are possible. The top-down effect of the cognitive map is to bias the possible transitions such that the ones chosen by the cognitive map have a higher value. This value reflects a topological distance measure: the number of intermediate node to get in touch with to actually reach the goal. This small bias is enough to select/filter the appropriate transitions via a competition mechanism (however see section 8). This mechanism realizes a soft competition: several motor transitions are proposed at this level. They allow to compute a more accurate direction than a strict competition since transitions starting only from places really close to the current one and topologically close to the goal (on the graph) are selected. Final selection of the motor action results from the merging of these global decisions with local constraints such as obstacle avoidance, robot inertia...

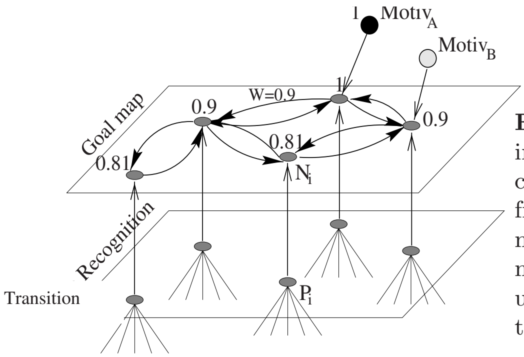


Fig. 6. Diffusion of the activity in the graph corresponding to the cognitive map. Diffusion is starting from the goal. Each node keeps the maximal activity coming from its neighbours. The activity is the product between the value of the link and the activity of the sending node.

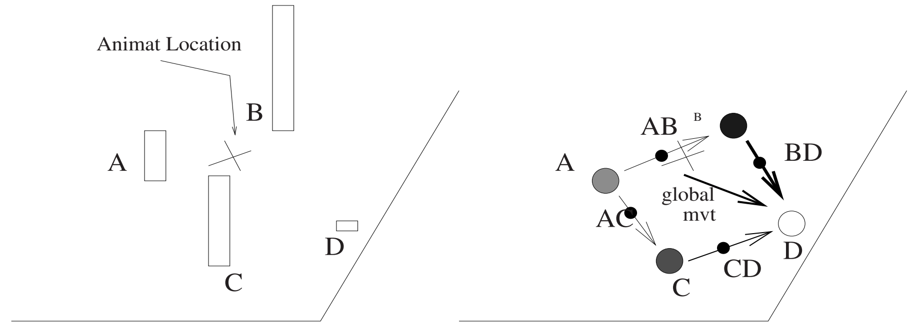
8 Movement and neural field dynamics

As seen in section 7, after planning, different movements are proposed. a simple competition mechanism selects the neuron with the higher value. But how may these informations be used ? We have first tested strict competition between them, but why do not also use other transitions that contains interesting informations about the agent location context ? So we now have several transitions to be taken into account for the movement. The solution used for having a stable continuous direction to follow is to define a dynamical system where the stable fixed point solution is the direction to follow. This is achieved using a neural field [1, 18, 16].

$$\tau.\frac{df(x,t)}{dt} = -f(x,t) + I(x,t) + h + \int_{z \in V_x} w(z).f(x-z,t)dz \tag{2}$$

Where $f(x,t)$ is the activity of neuron x, at time t. $I(x,t)$ is the input to the system. h is a negative constant. τ is the relaxation rate of the system. w is the interaction kernel in the neural field activation. A difference of Gaussian (DOG) models these lateral interactions that can be excitatory or inhibitory. V_x is the lateral interaction interval that defines the neighbourhood. Without inputs the constant h ensures the stability of the neural field homogeneous pattern since $f(x,t) = h$. In the following, the x dimension is an angle (direction to follow according to the north).

This equation allows the computation of attractors corresponding to fixed points of the dynamics and to local maxima of the neural field activity. A stable direction to follow is reached when the system is on any of the attractors. The angle of a candidate transition is used as input. The intensity of this input depends on the corresponding goal transition activity, but also on its origin place cell recognition activity (see Fig 7). If only one transition is proposed, there will be only one input with an angle $x_{targ} = x^*$ and it erects only one



level of place recognition in the neighborhood

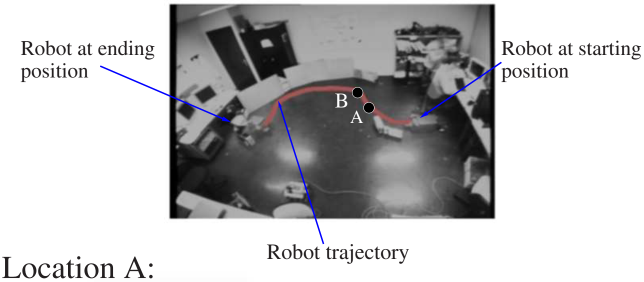
level of place recognition in the neighborhood
+ transitions cells and actions

Fig. 7. The merging mechanism allows to get a better direction (global movement) than the use of the single information obtained from current transition (BD). It takes into account the previous movement performed and the transitions predicted from close enough place cell (C).

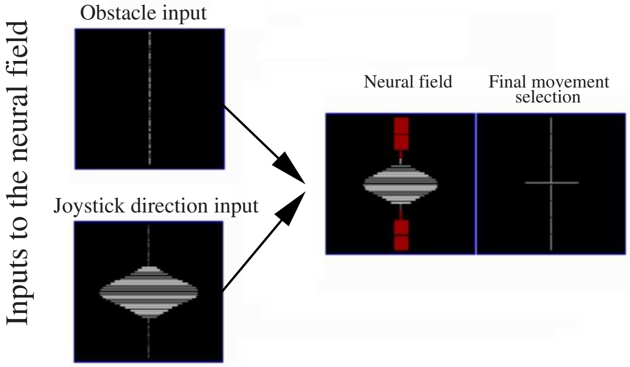
attractor $x^* = x_{targ}$ on the neural field. If x_c is the current orientation of the robot, its rotation speed will be proportional to $w = \dot{x} = \frac{df(x,t)}{dt}|_{x_c}$.
Merging of several transition informations depends on the distance between them. Indeed if the inputs are spatially close, the dynamics give rise to a single attractor corresponding to the average of them. Otherwise, if we progressively amplify the distance between inputs, a bifurcation point appears for a critical distance, and the previous attractor becomes a repellor and two new attractors emerge.
Obstacles are detected by 12 infra-red sensors. A reflex behavior is triggered by a Braitenberg-like architecture [6]. When an obstacle is detected on a given direction, the reflex system will generate a *negative* input in this orientation. Hence the bubble of the neural field activity will move. Consequently the computed direction will reflect this change and allow to avoid this obstacle (see Fig 8). Oscillations between two possible directions are avoided by the hysteresis property of this input competition/cooperation mechanism.
It is possible to adjust this distance to a correct value by calibrating the two elements responsible for this effect: spatial filtering is obtained by the convolution of the Dirac like signal coming from motor transition information with a Gaussian and taking it as the input to the system. This combined with the lateral interactions allows the fusion of distinct input as a same attractor. The larger the curve, the larger the merging will be.

9 Conclusion

Our model currently running on robots (Koala robots and Labo3 robots) has interesting properties in terms of autonomous behavior. However, this autonomy has some drawbacks:



Location A:



Location B :

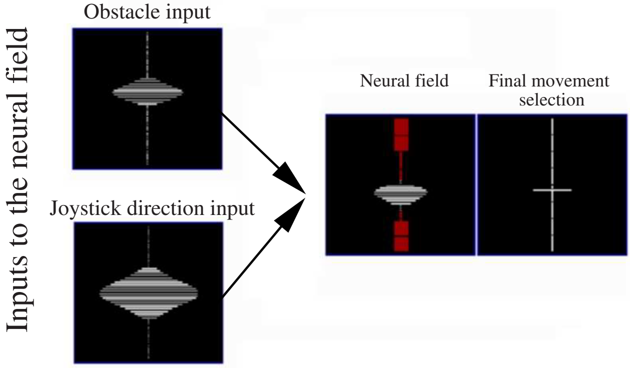


Fig. 8. Top: Trajectory of a Labo3 robot in an open environment with obstacles. The movement direction is given by a joystick input. We focus on the final action selection on two points in this environment: A and B. During all the trajectory, the movement ordered by the joystick is go straight. Turn movements in the robot trajectory are only due to obstacle avoidance. Middle (A): Neural field activity without any obstacle. The direction taken corresponds to the joystick input. Bottom (B): Neural field activity with an obstacle. The obstacle shifts the neural field maximal activity leading to a turning move.

- we are not able to build a cartesian map of the environment because all location learned are robot centered. However, the places in the cognitive map and the directions used give a skeleton of the environment.
- we have no information about the *exact* size of the rooms or corridors. Again, the cognitive map only gives a sketch of the environment.
- some parameters have to be set: the recognition threshold (section 3) and the diffusion size of the interaction kernel of the neural field (section 8). The first one determines the density of build places. The higher the threshold, the more places are created. The second parameter has to be tuned for each robot depending on its size and on the position of the infra-red sensors for obstacle avoidance. For instance, a too high diffusion value prevents the robot from going through the doors.

The transition used in this model may also be the elementary block of a sequence learning process. Thus, we are able to propose a unified vision of the spatial (navigation) and temporal (memory) functions of the hippocampus [4].

Acknowledgments

This work is supported by two french ACI programs. The first one on the modeling of the interactions between hippocampus, prefontal cortex and basal ganglia in collaboration with B. Poucet (CRNC, Marseille) JP. Banquet (INSERM U483) and R. Chatila (LAAS, Toulouse). The second one (neurosciences integratives et computationnelles) on the dynamics of biologically plausible neural networks in collaboration with M. Samuelides (SupAero, Toulouse), G. Beslon (INSA, Lyon), and E. Dauce (Perception et mouvement, Marseille).

References

1. S. Amari. Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87, 1977.
2. M. Arbib and I. Lieblich. Motivational learning of spatial behavior. In J. Metzler, editor, *Systems Neuroscience*, pages 221–239. Academic Press, 1977.
3. I. Bachelder and A. Waxman. Mobile robot visual mapping and localization: A view-based neurocomputational architecture that emulates hippocampal place learning. *Neural Networks*, 7(6/7):1083–1099, 1994.
4. J. Banquet, P. Gaussier, M. Quoy, A. Revel, and Y. Burnod. A hierarchy of associations in hippocampo-cortical systems: cognitive maps and navigation strategies. *Neural Computation*, 17(6), 2005.
5. R. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
6. V. Braitenberg. *Vehicles : Experiments in Synthetic Psychology.* Bradford Books, Cambridge, 1984.
7. J. B. C. Giovannangeli, P. Gaussier. Robot as a tool to study the robustness of visual place cells. In *I3M'2005: International Conference on Conceptual Modeling and Simulation*, Marseille, 2005.

8. J. Donnart and J. Meyer. Learning reactive and planning rules in a motivationnally autonomous animat. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(3):381–395, 1996.
9. D. Filliat and J.-A. Meyer. Map-based navigation in mobile robots - I. a review of localisation strategies. *Journal of Cognitive Systems Research*, 4(4):243–282, 2003.
10. P. Gaussier, A. Revel, J. Banquet, and V. Babeau. From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological Cybernetics*, 86:15–28, 2002.
11. C. Joulain, P. Gaussier, and A. Revel. Learning to build categories from perception-action associations. In *International Conference on Intelligent Robots and Systems - IROS'97*, pages 857–864, Grenoble, France, September 1997. IEEE/RSJ.
12. J.-A. Meyer and D. Filliat. Map-based navigation in mobile robots - II. a review of map-learning and path-planing strategies. *Journal of Cognitive Systems Research*, 4(4):283–317, 2003.
13. J.-A. Meyer and S. Wilson. From animals to animats. In M. Press, editor, *First International Conference on Simulation of Adaptive Behavior*. Bardford Books, 1991.
14. J. O'Keefe and N. Nadel. *The hippocampus as a cognitive map*. Clarendon Press, Oxford, 1978.
15. M. Quoy, P. Gaussier, S. Leprêtre, A. Revel, C. Joulain, and J. Banquet. *Lecture Notes in Artificial Intelligence Series, 1812*, chapter A planning map for mobile robots: speed control and paths finding in a changing environment, pages 103–119. Springer, ISBN 3-540-41162-3, 2000.
16. M. Quoy, S. Moga, and P. Gaussier. Dynamical neural networks for top-down robot control. *IEEE transactions on Man, Systems and Cybernetics, Part A*, 33(4):523–532, 2003.
17. A. Revel, P. Gaussier, S. Leprêtre, and J. Banquet. Planification versus sensory-motor conditioning: what are the issues ? In *From Animals to Animats : Simulation of Adaptive Behavior SAB'98*, pages 129–138, 1998.
18. G. Schöner, M. Dose, and C. Engels. Dynamics of behavior: theory and applications for autonomous robot architectures. *Robotics and Autonomous System*, 16(2-4):213–245, December 1995.
19. L. Schwartz. Computational anatomy and functional architecture of striate cortex: a spatial mapping approach to perceptual coding. *Vision Res.*, 20:645–669, 1980.
20. N. Tinbergen. *The study of instinct*. Oxford University Press, London, 1951.
21. E. Tolman. Cognitive maps in rats and men. *The Psychological Review*, 55(4), 1948.