Active Perception and Map Learning for Robot Navigation

David Filliat* **Jean-Arcady Meyer**
AnimatLab-LIP6
8, rue du capitaine Scott
75015 PARIS
*david.filliat@lip6.fr

Abstract

This paper describes a simulated on-line mapping system for robot navigation. This system allows the autonomous creation of topological maps enhanced with metrical information provided by internal (odometry) and external (vision and sonars) sensors. Within such maps, the robot's position is represented and calculated probabilistically according to algorithms that are inspired by Hidden Markov Models. The visual system is very simple and does not allow reliable recognition of specific places but, used jointly with odometry, sonar recordings and an active perception system, it allows reliable localization even when the robot starts exploring its environment, and when it is passively translated from one place to another. Advantages and drawbacks of the current system are discussed, together with means to remediate the latter.

1. Introduction

In order to navigate within a large-scale environment, an autonomous robot needs to store spatial information gathered from exploration. Such information may derive from internal and external perceptions and may lead to two types of map - metrical and topological.

Metrical maps, such as occupancy grids (Moravec and Elfes, 1985), may be used for navigation in a very straightforward way, as they give the robot's position in an Euclidean space. In such a space, distances and directions are very easy to calculate and may be used directly to guide the robot toward a goal. Unfortunately, internal perception gained from an odometry device is subject to cumulative noise and cannot be used reliably over long periods of time. Another way of storing information about the environment is to use external perceptions to identify distinctive places. Such places may be stored in topological maps along with means to get from one place to another (Kuipers and Byun, 1991, Bachelder and Waxman, 1995). But this kind of map suffers from the *perceptual aliasing problem* (Whitehead and Ballard, 1991), i.e., from the fact that, in large environments, several places look the same from the robot's point of view.

The obvious solution for reliable navigation is to use both sources of information to create a map of the environment in which perceptually identical places are distinguished thanks to odometry (Thrun, 1999). From this point of view, biological systems may provide inspiration. For example, place cells found in rats respond mainly to perceptual cues, but it has been shown that they remain operational even in the dark, thus suggesting that their firing also depends upon odometry (Quirk et al., 1990).

Fusion of these two kinds of information may be realized in a very straightforward way in navigation systems based on Partially Observable Markov Decision Processes (POMDP) (Kaelbling et al., 1998). In these systems, movements and perceptions are used to update a probabilistic representation of position. However, a major drawback to this approach is that the map used to represent the environment has to be created prior to its utilization and cannot be extended on-line if the robot discovers a new part of its environment.

Figure 1: The Pioneer 2 mobile robot.

The navigation system described herein draws inspiration from both biology and POMDP models. It is based on a topological map in which additional data from odometry and vision are stored. The activity level of each node in the map represents the robot's probability of being in a given place. This activity is updated using techniques drawn from POMDP models. The map is created on-line, and its utilization is not separated from learning. Moreover, the system calls upon an active perception strategy which controls the robot camera in order to enhance map accuracy and position estimation.

This navigation system has been designed for a future implementation on a Pioneer 2 mobile robot manufactured by ActivMedia (Figure 1). This mobile robot is equipped with a 16 sonar belt, an orientable camera and a compass providing an absolute reference direction. However, this paper only reports simulation results obtained on map learning with the realistic simulator that is provided with the Pioneer 2 robot (ActivMedia, 1999).

2. Markov models for navigation

Navigation systems based on POMDP, like those developed by Simmons and Koenig (1995) or Thrun et al. (1998), rely on a discretization of space into a finite set of states S that cover all the environmental locations the robot can reach. The robot's position is represented by a probability distribution $P(s)$ over S. Actions and perceptions are also discretised and modeled probabilistically: a set of actions A and transition probabilities $p(s|s',a)$ is defined, where p is the probability of being in state s when the robot performs action a is state s'. Likewise, a set of perceptions O and probabilities $p(o|s)$ is defined, where p is the probability of acquiring perception o in state s.

$P(s)$ is updated in two ways whenever the robot acts or perceives. In the present context, acting means moving, and, for each move a, the updating equation used is:

$$p_{t+1}(s) = K \times \sum_{s' \in S} p(s|s',a) \times p_t(s') \qquad (1)$$

Likewise, for each perception o, the updating equation is:

$$p_{t+1}(s) = L \times p(o|s) \times p_t(s) \qquad (2)$$

where $p_t(s)$ is the robot's probability of being in state s at time t, and K and L are normalization factors ensuring that probabilities sum to 1 over S.

These POMDP systems, even if they rely on the assumption that the world is Markovian - which is obviously false - behave very well in the real world and are able to track the robot's position reliably. Learning of action and perception models (Simmons and Koenig, 1995) as well as off-line learning of POMDP structure (Thrun et al., 1998) is possible but, to our knowledge, current implementations do not allow the set of states of the POMDP to be modified on-line. Therefore, they are not able to deal with an environment that the robot should discover incrementally.

3. The current model

This work is inspired from Markovian models and makes it possible for a simulated robot to represent and estimate its position thanks to a fine-grained topological map augmented with metrical data, which is incrementally built during the exploration of the environment.

In each node in the map, perceptual data are recorded when the robot is located at the corresponding place in the environment. Likewise, links between nodes store the distance and absolute direction the robot has to move to get from one node to another, as measured by the robot's compass and odometry devices.

This kind of map combines advantages from both topological and metrical maps. From topological maps, it inherits their sparse representation of space and their ease of extension. Moreover, it can be used without knowing the robot's initial position, thus allowing the *lost robot problem* (Duckett and Nehmzow, 1997) to be tackled. Conversely, from metrical maps, it draws the easy use of odometry data and the ability to calculate new paths from one place to another through areas that remain unexplored (Trullier and Meyer, 1997, Trullier et al., 1997).

The activity of each node in this map assesses the robot's probability of being in the corresponding place in the environment. Accordingly, the map learning algorithm iterates the following steps each time the robot moves a fixed distance D :

- Calculate the activity of all the nodes in the map.
- Recognize the current node or create a new one.
- Update sonar and visual data stored in the recognized node using current data.
- Update connection parameters between previous and current nodes using odometry data.

D is set to approximately the robot's size in our experiments. The map structure and the successive steps of this algorithm are detailed in the next sections.

3.1 Map inputs

3.1.1 Sonar data

Data perceived through the 16 sonars are aggregated in 8 virtual sensors giving the distances of obstacles in 8 absolute directions. These data are calculated using compass information. The values of these 8 sensors are averaged and stored in the map node corresponding to the current place. In other words, the stored data that represent the environment around the place associated to a given node on the map are the means of all the values recorded during successive recognitions of this node.

The matching measure used to estimate the similarity of the robot's current location with any location stored in the map is simply :

$$Corr_s(S1, S2) = C - \sum_{k=1}^{8} abs(S1_k - S2_k)/8 \qquad (3)$$

where C is a constant which ensures that $Corr(S1, S2)$ remains positive, and $S1_k$ and $S2_k$ are the perceived and memorized obstacle distances in direction k. Thus, the matching measure is maximum if $S1 = S2$, and decreases when the differences between $S1$ and $S2$ increase.

3.1.2 Visual system

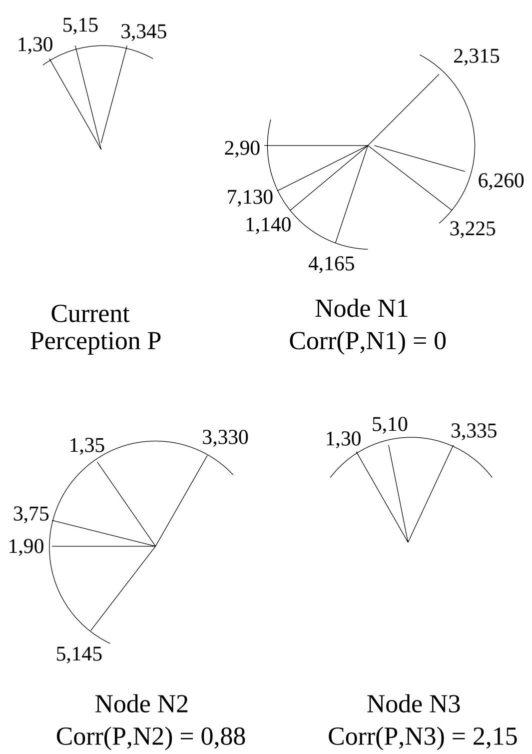

Figure 2: Examples of matching measures between currently perceived visual data and data stored in different nodes. Pairs of numbers indicate a landmark's category and heading. Arcs indicate, for each node, the directions for which visual data have already been recorded. In this example, the node that best matches the current perception is node $N3$.

The visual system that will ultimately be implemented on the real Pioneer 2 robot is inspired from that of Gaussier an Zrehen (1995). It will call upon specific landmarks - i.e. corners extracted from an image taken by the robot's camera - that will be classified into a predefined number of categories. According to these implementation perspectives, within each node in the current model, a list of landmarks - each characterized by a category number and an associated direction - is stored, along with how often each landmark has been detected.

When new landmarks are detected from the place associated with a given node, each newly perceived landmark is searched for in the list of memorized landmarks. If a landmark of the same category is found whose direction differs by less than 10 degrees from that of the new landmark, the latter is considered as being recognized. If not, the new landmark is added to the landmark list attached to the node.

Likewise, as a camera picture will cover only 40 degrees, the real robot will need to take several pictures to gain information about its whole environment. Because such movements will be time-consuming, instead of taking a full panoramic image for each node, the directions in which the camera has already been pointed will be recorded in each node. Such information will be used during subsequent recognitions of the place, in order gradually to obtain a full representation of the environment. An equivalent strategy has been implemented here, as an active perception procedure that orients the visual system in a specific direction:

- If the robot needs new data about its environment, the chosen direction is simply a direction for which there are no recorded data associated to the current node.
- If the robot needs to obtain a better localization estimate, the chosen direction is the direction for which the number of already recognized landmarks is maximum. This choice heads the visual system toward a direction where landmarks are numerous and easily recognizable.

The choice between these two alternatives is made by detecting the node with the highest activity in the map. If this activity is above a given threshold, the localization is considered successful, and new information is sought according to the first procedure. If this activity is below the threshold, the second procedure is used to improve the quality of localization.

The matching measure assessing the similarity of the perceived and stored visual data is:

$$Corr_l(L1, L2) = \sum_{l \in L1} G\Big(\min_{\substack{l' \in L2 \\ C(l') = C(l)}} abs(H(l) - H(l'))\Big) \qquad (4)$$

where $C(l)$ is the class of landmark l, $H(l)$ is the direction of landmark l and G a Gaussian of mean 0 and variance 10 given by :

$$G(x) = exp(-x^2/100) \quad (5)$$

The value of this function is 0 if there are no common landmarks in the perceived and stored data. It increases with the number and similarity of common landmarks (Figure 2).

3.2 Activity update

Activity is updated within the map using a two-step process, i.e., a first step taking into account the robot's move, and a second step taking into account the robot's perceptions.

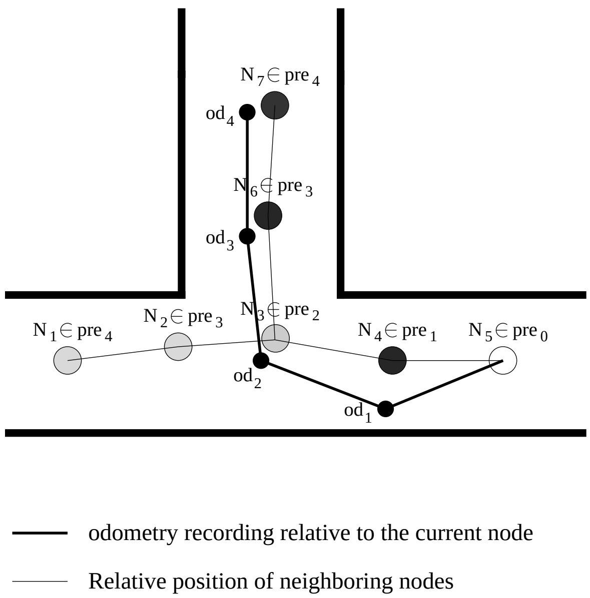

Figure 3: Propagation of activity in the map using odometry data. The gray level of each node belonging to pre_i shows the activity of the node i time steps earlier. Points od_i show the robot's position i time steps earlier relative to N_5. The update rule takes into account both the past activities of neighboring nodes and their positions relative to the current node (see text for details). In this example, the activity of the current node N_5 will be updated to that of node N_6, three time steps earlier, multiplied by a Gaussian transform of the distance $od_3 - rel_{N_6}$.

Recent information from odometry is used in a local map-matching procedure that takes into account a few past time-steps (Figure 3). For each node in the map, the relative position of neighboring nodes is calculated using odometry data stored in the connections. These relative positions are used to calculate the new activity according to equation:

$$act_t(n) = \max_{p\in[1..P]} \left(\max_{n'\in pre_p(n)} (act_{t-p}(n') \times D(rel_{n'}, od_p)) \right) \quad (6)$$

where od_p is the robot's relative position p time steps earlier as recorded by odometry, $rel_{n'}$ is the position of node n' relative to the current node, $pre_p(n)$ the set of map's nodes which can be reached from n using less than p connections, and $D(pos1, pos2)$, a Gaussian of mean 0 and variance K of the difference between the two relative positions $pos1$ and $pos2$:

$$D(pos1, pos2) = \exp(-\frac{(pos1 - pos2)^2}{K^2}) \quad (7)$$

K is set to half the distance D.

Thus, for each node N, this procedure seeks to determine from which more or less recently visited node M the robot is able to get closest to N, according to its recorded odometry. The new activity of node N is set to that of M, modulated by a function of the difference between the recorded odometry and the position of N relative to M as calculated using the map data. This function's value is 1 if the positions are identical and converges to 0 as their differences increase.

This update rule has the advantage of capitalizing upon past information to avoid the propagation of temporary wrong place identification. For example, if the robot reactively avoids a moving obstacle in a corridor, the current odometry recording will not correspond to the global direction of the corridor, whereas earlier recordings will still allow a good activity propagation along the corridor.

During the second step, sonar data are used to modulate the current node's activity according to the similarity of current perceptions with recorded perceptions:

$$act_t(n) = \frac{act_{t-1}(n) + Corr(S, S_n)/C_S)}{2} \quad (8)$$

where $Corr_s(S, S_n)$ is given by equation (3), C_S is the maximum of $Corr_s()$ over all map nodes, S is the current sonar data and S_n is the data memorized in node n.

At this time, the procedure described in the next section is used in order to recognize or create a new node. This node is used by the active perception procedure to choose a camera direction. Then, landmarks detected by the visual system modulate the node's activity using equation (8), where $Corr_s(S, S_n)$ is replaced by the function $Corr_l(L, L_n)$ given in equation (4).

3.3 Node recognition

If the highest activity in the map is above a given threshold, the most activated node is considered recognized as the current node, whereas if all activities are below this threshold, recent odometry is used to determine the current node. This is achieved by calculating the robot's position relative to the nodes previously recognized:

$$rel_0 = \sum_{p=1\rightarrow P} (od_p - rel_p)/P \quad (9)$$

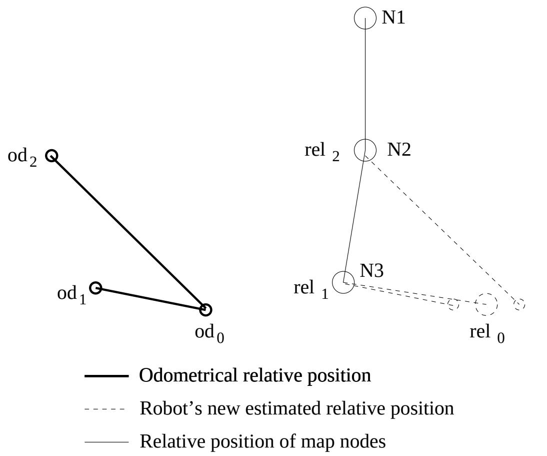

Figure 4: This figure shows the estimation of the new position of the robot relative to the last recognized node $N3$. The robot's current odometry recording is od_0. od_1 is the odometry recording one time step before, when the recognized node was $N3$. od_2 is the odometry recording two time steps before, when the recognized node was $N2$. The new position rel_0 is taken as the mean of the two relative positions that can be reached from $N3$ and $N2$ using odometry recordings od_1 and od_2.

where od_p is the position of the robot at time $t-p$ according to odometry and rel_p if the position of the node recognized at time $t-p$ relative to the node recognized at $t-1$ (Figure 4). In our experiments, P was set to 10.

This relative position is then used to verify whether a node already exists in the map at a distance smaller than D' where $D' < D$. If so, this node is considered recognized; if not, a new node is created.

After this first recognition step, the current node is used by the active perception procedure previously described to choose a new camera heading. An image is taken, and landmarks are extracted. Then a new update of each node's activity in the map is performed, using the perceived landmarks only. After this update, if the most activated node is different from the current one, and if its activity is above the threshold, this node is chosen as the new current node.

3.4 Map parameter update

Sonar and landmark data attached to the current node are updated according to current perceptions as described in section 3.1. The connection between the previously and the currently recognized nodes is then created, or updated if it already exists. The parameters are chosen so as to ensure map consistency (Figure 5). In this context, a map is considered to be consistent if, when two different paths link two nodes, the relative positions of these nodes, calculated by summing the connection data along these two paths, are identical.

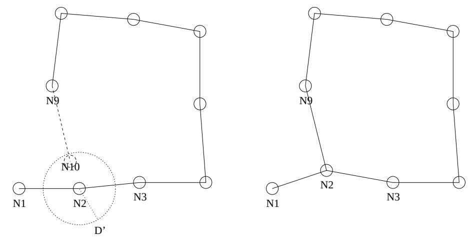

Figure 5: Procedure used to update connection parameters. In this example, the robot was previously in $N9$ and odometry suggests that $N10$ is the robot's new position. As the distance between $N10$ and $N2$ is smaller than D', $N10$ is not created, but $N2$ is recognized. Connections $(N1, N2)$ and $(N2, N3)$ are modified, and connection $(N9, N2)$ is created by a procedure which ensures map consistency (see text for details).

4. Experimental results

4.1 Experimental setup

The experiments reported in this section were conducted using the Pioneer 2 robot similator provided by Activmedia. This simulator calls upon realistic error estimates for sonar sensors and odometry. In particular, wheel encoder values are corrupted with a noise whose maximum is 1 percent of their value. Likewise, compass direction is corrupted with a random value whose maximum is 5 degrees. Lastly, sonar sensors are corrupted with a Gaussian noise and are sensitive to specular reflections when the angle between the sonar beam and the obstacle is too small.

The visual system has also been simulated using a program which provides the robot with the category and direction of the landmarks it perceives, according to its position and to the visual system's orientation in the simulated world. Landmarks are randomly positioned in the environment by the experimenter. In this simulator, the determination of the category and direction of landmarks is deterministic. However, as nodes in the map represent small areas in the environment by a single point, different pictures taken from the area surrounding this point are considered as being all taken from the point, thereby introducing noise in the perceived landmarks directions.

The robot is controlled by a low-level navigation module, which allows it to randomly explore the environment while reactively avoiding obstacles.

4.2 Map creation

Figure 6 shows three maps created in various environments. These maps exhibit some inconsistencies, such

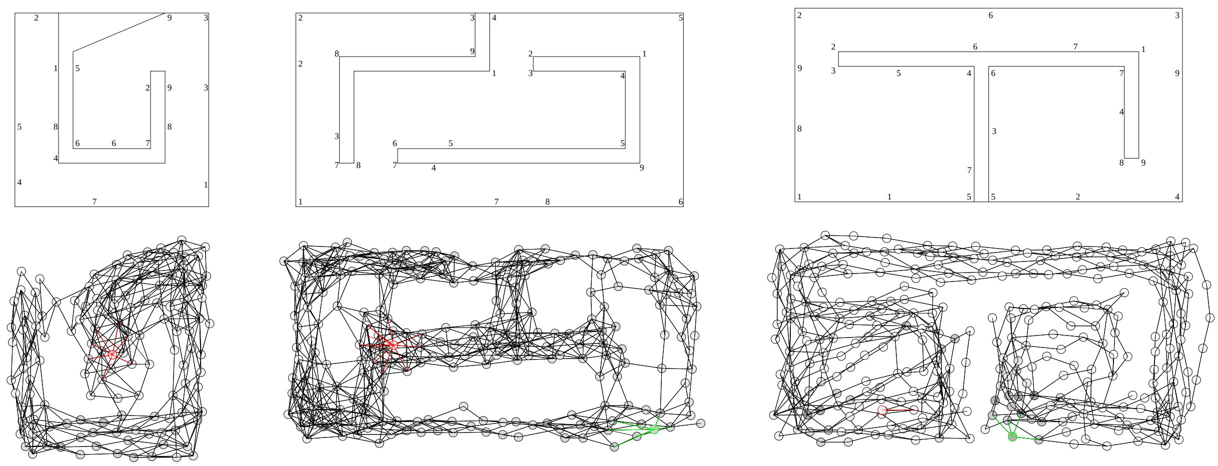

Figure 6: Examples of maps (bottom) created by the robot in various environments (top). Numbers indicate the category of the various landmarks that have been positioned in the environment by the experimenter.

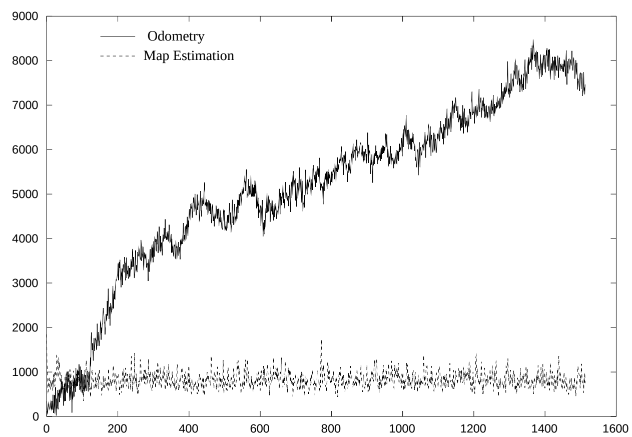

Figure 7: Evolution over time (abscissa) of the distance between the real location and the location estimated by odometry alone, on the one hand, and by the whole mapping system, on the other hand (ordinate). The environment and the map used are those of Figure 8. The environment size is 5000, the distance D' is 250, while the distance D and the robot's diameter are both 500 in arbitrary units. While odometry error keeps growing, the mapping system's error is restricted to a value slightly greater than the robot's size.

as inter-node links that cross walls. Most such inconsistencies result from erroneous node recognitions at the beginning of the map-building process, i.e., during a period when the representation of the visual environment associated with each node is not accurate enough to ensure good discrimination.

Note that once a map has been created with the right general structure, the perceptual similarity of different places is no longer a problem. As a matter of fact, in almost stabilized maps, individual perceptions are less important than in newly created ones because activity propagation using a robot's move is more efficient and makes the localization process more resistant to temporary lack of perceptual information. The relative number of inconsistent links, as compared to the total number of connections, is always observed to be low compared to the total number of connections. Consequently, the robot is able to localize itself accurately in the environment, even at the beginning of the exploration process (Figure 7), because activity propagation within the map is coherent with respect to the robot's actual moves (Figure 8).

4.3 Re-localization

Because this mapping system allows the robot to localize itself with no knowledge of its initial position, it remains efficient even when the robot is passively transported from one place to another in the environment.

Figure 9 indicates how the odometry and mapping errors evolve during two such experiments. Figure 10 shows the activity propagation within the map after such a transportation episode. It demonstrates that, after a few wrong place identifications, the robot succeeds in re-localizing itself correctly.

4.4 Map pruning

A drawback to the current mapping system is that it has a tendency to let the number of nodes slowly, but steadily, grow as the robot explores its environment, even should the exploration entail passing through places all known already. As such an increase in the number of nodes might prove a problem in a large and complex environment, dedicated procedures for detecting and pruning redundant nodes have been designed and implemented.

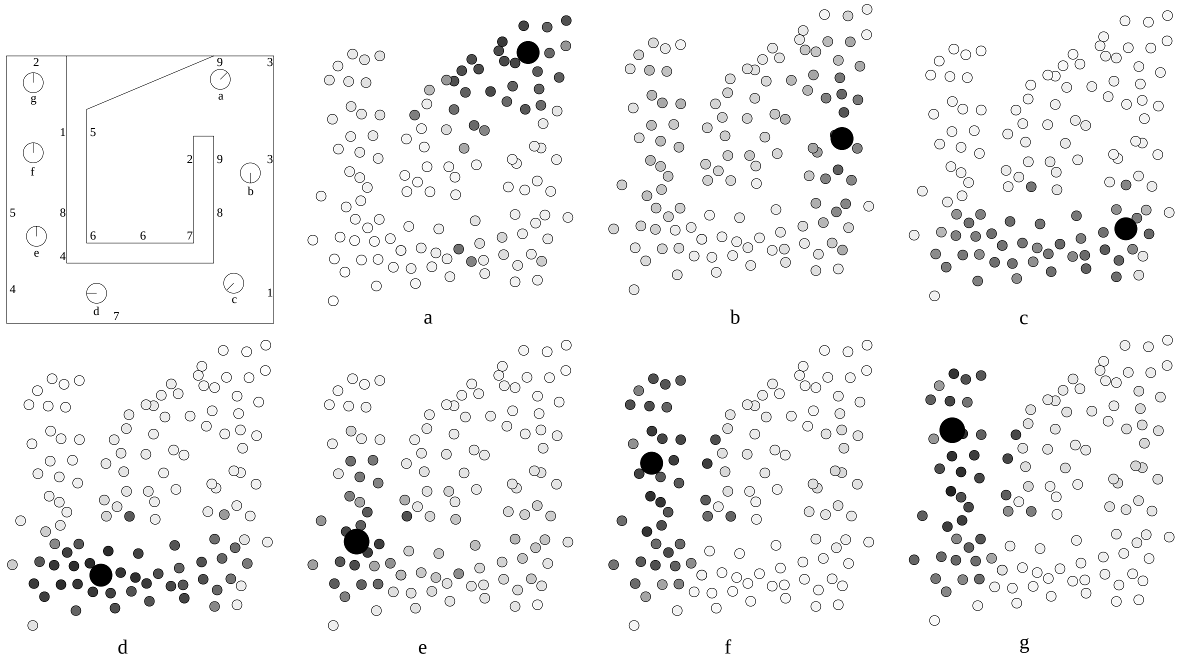

Figure 8: Activity updates within the map as the robot moves to successive places in its environment. Labels $a, b, ...g$ indicate both the actual position of the robot and the corresponding map activity. The grey level of each small node in a map indicates its activity, ranging from 0 for white nodes to 1 for black nodes. Larger black dots indicate the successfully recognized nodes.

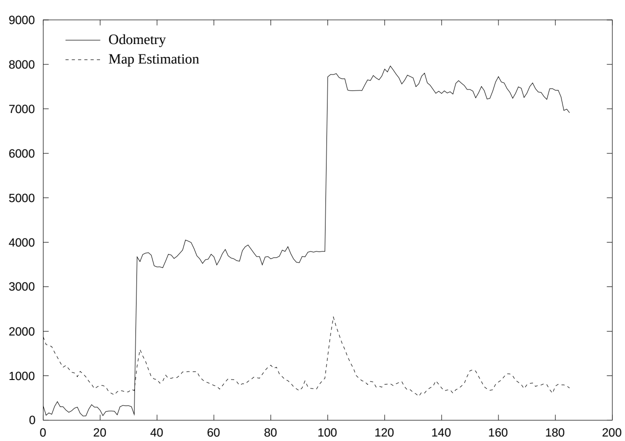

Figure 9: Evolution over time (abscissa) of the odometry and map estimation errors (ordinate) during two re-localizations. While the odometry error keeps growing, the mapping system's estimation error, after a temporary increase, restabilizes at its mean value.

For example, in the course of the experiment reported in Figure 11, each place in the environment is visited approximately 1500 times. The number of nodes created by the mapping system described so far attains 400 and results in the map shown on top of Figure 11. This map exhibits several erroneous nodes and connections (notably connections that cross walls) and, in particular, the whole left part of the map, which has been created at the beginning of the mapping process, is erroneous and useless. However, the second map on the figure has been created by the same mapping system, but with additional node-pruning procedures. This map contains only a few errors, and the number of its nodes quickly stabilizes around 170. One such procedure eliminates nodes near the current node if they have not been visited more recently than 300 time steps ago. Another procedure eliminates nodes than have not been visited since 1000 time steps. These implementation details have been determined empirically after examining the first map, and they surely depend on the environment size. However, they suggest that adaptive procedures can be set up that will cause the mapping system to stabilize automatically.

5. Discussion

The current simulated system exhibits some similarities with several other navigation systems, some of which having been implemented on real robots. For example, as already mentioned, it shares procedures with the work of Simmons and Koenig (1995) for position representation and estimation that are inspired by POMDP models. However, the present work has the advantage of allowing the corresponding map to be built incrementally during exploration. This has been made possible thanks to a procedure that adds new nodes to the map every time the representation of the environment needs to be refined.

Such a mapping procedure is highly reminiscent of

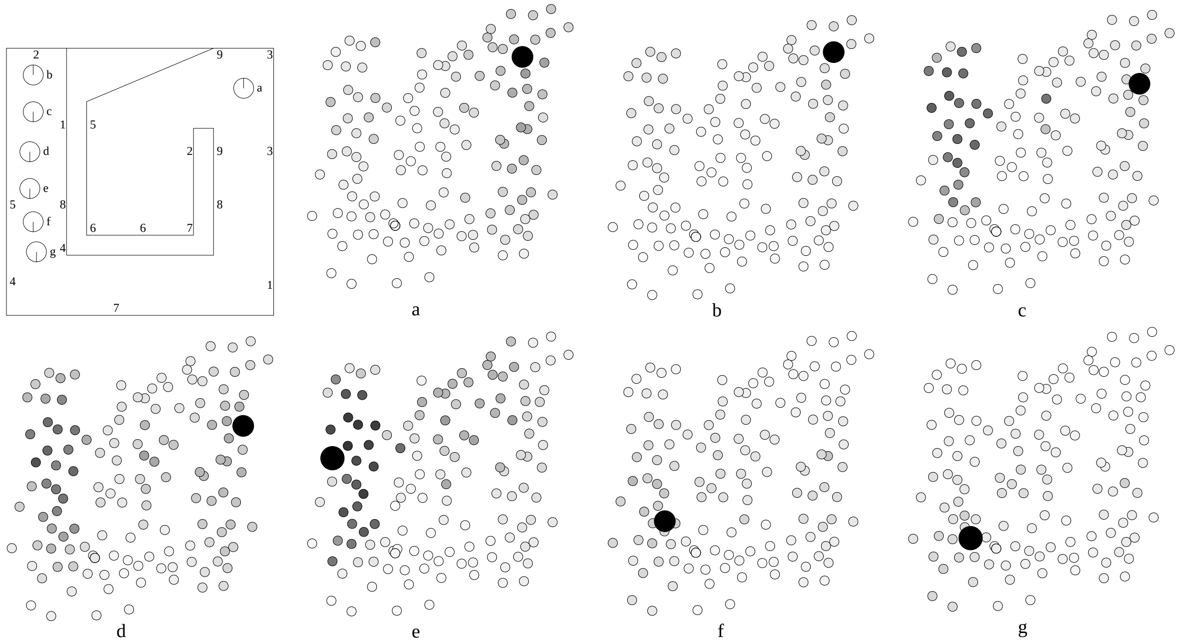

Figure 10: Activity updates within the map after a translocation process during which the robot has been moved passively from place a to place b in the environment. After a few wrongly recognized positions (from b to d), the robot correctly recognizes its actual position (e).

those used in biologically-inspired navigation systems [see Trullier and Meyer (1997) and Trullier et al. (1997) for reviews], with the fundamental difference that the activity of a given node in the present system codes for the robot's probability of being in the corresponding place in the environment. In other models, nodes work like so-called *place cells* in the hippocampus of rodents, i.e., they are deterministically active or not depending whether the robot is located in the corresponding place or not. The probabilistic approach that has been used here makes it possible to easily fuse vision with dead-reckoning - which is seldom the case in traditional biologically-inspired models - and this fusion very probably enhances the robustness of the current system with respect to sensory noise. Naturally, such a probabilistic approach raises questions about the traditional view regarding spatial representations in rodents. In particular, it suggests that some sort of population coding (Georgopoulos et al., 1986) might represent the animal's probability of being in a given place.

The current system is able to cope with the *chicken and egg* problem tackled by Yamauchi et al. (1999) and by Kurz (1995), which arises when a robot has to simultaneously build a map and estimate its position. In other words, the robot needs to know its position in order to build a map, whereas it needs a map to estimate its position. In these two research efforts, the problem is solved by continuously correcting the robot's position estimate using specific procedures. In our system, as in most systems relying on topological maps, the problem is avoided by taking into account recent odometry information and re-localizing the robot when there is enough evidence that the current estimation is wrong.

The current system also shares with those of Mataric (1991), Kuipers and Byun (1991), Donnart et Meyer (1996), the management of metric information together with the nodes and links of an inherently topological map. However, this system allows the robot to wander freely in an arbitrary environment, whereas Mataric's and Kuiper and Byun's systems are intended to work under the control of an underlying wall-following or corridor-following exploration procedure. Likewise, Donnart and Meyer's system cannot cope with an environment containing non-polygonal obstacles. Nevertheless, these and other systems (Nehmzow and Smithers, 1992) exhibit the interesting property of allowing the localization procedure to depend upon information acquired by moment-to-moment decisions taken by the underlying motor-control system. Such information could easily be added to the vision and dead-reckoning data already taken into account during activity updates in the map and would probably further enhance the robustness of the current system.

The procedures used here to evaluate the robot's position using odometry and to decide whether the robot is in a place already known or in one never seen before (see section 3.3) shows some similarities with those of Kurz (1995). Like the present system, his is able

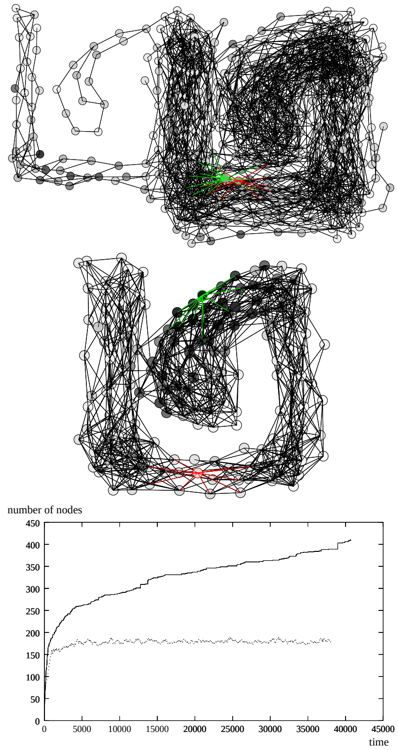

Figure 11: This figure reports a long-term experiment during which the robot explores the whole environment given in Figure 10 approximatively 1500 times. The map on the top exhibits several wrong nodes and links. The map below has been obtained with ad-hoc procedures that suppress useless nodes (see text for details) and is much more accurate. The chart at the bottom of the figure shows the evolution of the number of nodes in the map versus time for the mapping system alone (solid line) and for the mapping system with its pruning procedures (dashed line).

to create a topological map of large environments using low-accuracy sensors providing information that is merged with dead-reckoning data in order to disambiguate places. However, as the robot's position is represented by a single point in his mapping system, this system is not able to accumulate evidence over several time steps and, accordingly, is unable to recognize the robot's position without knowing its starting position.

Beyond the implementation differences that have just been discussed, the current system may also be compared to others with respect to its functionalities. It shares with Donnart and Meyer (1996) and with Yamauchi et al. (1999) the capacity of exploring and mapping an unknown environment while maintaining an accurate estimate of its position at all times. In the former work, the error between the real and estimated position of the robot is permanently maintained around a value roughly equal to the robot's size, as is the case in the results described above. Unfortunately, in the latter work, the size and evolution of this error cannot be deduced from available information. Likewise, the capacities of the current system to accurately localize itself extend to the case where the robot is passively translocated from one place to another in the environment. Similar capacities are evidenced in the work of Balakrishnan et al. (1999), but seem to require more motion steps to accurately position the robot once it has been moved to a new place. However, only systematic comparisons of the two systems could help in assessing their relative merits accurately.

While the actual system favorably compares to others, it should be emphasized that it nevertheless is subject to two limitations. First, it calls upon visual landmarks predefined by the experimenter, who provides the categories to which they belong, together with the directions in which they are seen. Although there is good reason to believe (Gaussier and Zrehen, 1995) that a robust procedure affording such functionalities can be implemented on a real robot, landmark detection and categorizing could call upon more autonomous procedures, like those of Tani (1998) or Gourichon (1999), which depend upon a methodology based on a so-called *mixture of experts*. Such procedures could afford an on-line and adaptive definition of landmarks.

A second drawback, mentioned in section 4.4, is that the mapping system has a tendency to let the number of nodes slowly, but steadily, grow. Ad-hoc procedures have already been tested which solve the problem in a given environment, but more adaptive criteria which would work in any environment have still to be designed and implemented.

6. Conclusions

The simulated mapping system described herein is able to create on-line a fine-grained topological map of an unknown environment. At each time-step, the system is able to track the robot's position reliably with an error that does not exceed roughly the robot's size, even when the odometry error is large. The system relies on an active perception mechanism that enables the robot to gradually refine its representation of the environment and to improve its position estimate.

Further improvements will aim at introducing mechanisms which will adaptively prune maps by removing useless nodes and connections, in order to prevent the map from continuing to grow once the environment is fully represented. Procedures for planning a trajectory from a given place to another will also be implemented in the near future.

Finally, this simulated Pioneer 2 system will be implemented on a robot, in order to check its ability to deal with real navigation problems.

References

ActivMedia (1999). *Pioneer 2 Mobile Robot. Saphira Manual.*

Bachelder, I. and Waxman, A. (1995). A view-based neurocomputational system for relational map-making and navigation in visual environments. *Robotics and Autonomous Systems*, 16:267–298.

Balakrishnan, K., Bousquet, O., and Honavar, V. (1999). Spatial learning and localization in rodents : A computation model of the hippocampus and its implications for mobile robots. *Adaptive Behavior*, 7(2):173–216.

Donnart, J. and Meyer, J. (1996). Spatial exploration, map learning, and self-positioning with monalysa. In Maes, P., Mataric, M., Meyer, J., Pollack, J., and Wilson, S., (Eds.), *From animals to animats 4. Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior.* The MIT Press.

Duckett, T. and Nehmzow, U. (1997). Experiments in evidence based localisation for a mobile robot. In Corne, D. and Shapiro, J., (Eds.), *Proceedings of the AISB 97 workshop on Spatial Reasoning in Animals and Robots.* Springer.

Gaussier, P. and Zrehen, S. (1995). Perac : A neural architecture to control artificial animals. *Robotics and Autonomous Systems*, 16((2-4)):291–320.

Georgopoulos, A., Schwartz, A., and Kettner, R. (1986). Neuronal population coding of movement direction. *Science*, (233):1416–1419.

Gourichon, S. (1999). Catégorisation distribuée et hiérarchisée de son environnement par un robot mobile. (in french). In Drogoul, A. and Meyer, J.-A., (Eds.), *Intelligence Artificielle Située.* Hermès.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101:99–134.

Kuipers, B. and Byun, Y. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8:47–63.

Kurz, A. (1995). Alef : An autonomous vehicle which learns basic skills and construct maps for navigation. *Robotics and Autonomous Systems*, 14:172–183.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angular sensors. In *Proceedings of the IEEE International Conference On Robotics and Automation.* IEEE Computer Society Press.

Nehmzow, U. and Smithers, T. (1992). Using motor actions for location recognition. In Varela, F. and Bourgine, P., (Eds.), *Toward a Practice of Autonomous Systems.* The MIT Press.

Quirk, G., Muller, R., and Kubie, J. (1990). The firing of hippocampal place cells in the dark depends on the rat's recent experience. *Journal of Neuroscience*, 10(6):2008–2017.

Simmons, R. and Koenig, S. (1995). Probabilistic navigation in partially observable environments. In Mellish, S., (Ed.), *Proccedings of IJCAI'95.* Morgan Kaufman Publishing.

Tani, J. and Nolfi, S. (1998). Learning to perceive the world as articulated: An approach for hierarchical learning in sensory-motor systems. In Pfeifer, R. and al., (Eds.), *From Animals to Animats 5. Proceedings of the Fifth Conferences on Simulation of Adaptive Behavior.* The MIT Press.

Thrun, S. (1999). Learning metric-topological maps maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71.

Thrun, S., Gutmann, S., Fox, D., Burgard, W., and Kuipers, B. (1998). Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proceedings of AAAI-98.* The MIT Press.

Trullier, O. and Meyer, J.-A. (1997). Biomimetic navigation models and strategies in animats. *AI Communications*, 10:79–92.

Trullier, O., Wiener, S., Berthoz, A., and Meyer, J.-A. (1997). Biologically-based artificial navigation systems: Review and prospects. *Progress in Neurobiology*, 51:483–544.

Whitehead, S. D. and Ballard, D. H. (1991). Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83.

Yamauchi, B., Schultz, A., and Adams, W. (1999). Integrating exploration and localization for mobile robots. *Adaptive Behavior*, 7(2):217–230.