Global localization and topological map learning for robot navigation

David Filliat, Jean-Arcady Meyer

Global localization and topological map-learning for robot navigation

David Filliat[*]
[*]DGA/Centre Technique d'Arcueil
16 bis, av. Prieur de la Côte d'Or
94114 Arcueil Cedex - France
david.filliat@etca.fr

Jean-Arcady Meyer[**]
[**]AnimatLab-LIP6
8, rue du capitaine Scott
75015 Paris - France
jean-arcady.meyer@lip6.fr

Abstract

This paper describes a navigation system implemented on a real mobile robot. Using simple sonar and visual sensors, it makes possible the autonomous construction of a dense topological map representing the environment. At any time during the mapping process, this system is able to globally localize the robot, i.e. to estimate the robot's position even if the robot is passively moved from one place to another within the mapped area. This is achieved using algorithms inspired by Hidden Markov Models adapted to the on-line building of the map. Advantages and drawbacks of the system are discussed, along with its potential implications for the understanding of biological navigation systems.

1 Introduction

The word navigation refers to all the strategies that may be used by a robot to purposely move in its environment. Such strategies range from simple visible goal heading behavior to complex map-based navigation that allows the planification of movements to arbitrary distant goals (Trullier et al., 1997). Using the latter strategies basically raises three sub-problems : *map-learning*, which concerns the construction of a map representing the environment, *localization*, which concerns the estimation of the robot's position inside this map and *planification*, which concerns the design of a plan to reach a given goal.

Every navigation strategy may call upon two sources of information. The first is the idiothetic source that provides information about the robot's movements using internal sensors such as accelerometers. This information can be directly expressed in a metrical space. The second one is the allothetic source that provides information about the robot's position inside its environment using external sensors such as sonar sensors or a camera. The characteristics of these two sources are complementary : while idiothetic information suffers from cumulative errors that make it unreliable for long-term position estimation, allothetic information suffers from the *perceptual aliasing* problem that prevents the robot from distinguishing between two places. Therefore, the efficiency of a navigation system usually relies on its capacity to efficiently combine these two types of information.

It is important to note that allothetic information can be used in two different ways. The first makes use of a metrical model of the sensors, which permits the allothetic data to be expressed in the metrical space of idiothetic information. This is, for example, the case for sonar data used to estimate the position of obstacles in a metrical map of the environment (Moravec and Elfes, 1985). The second way avoids any use of metrical models of the sensors and directly resorts to allothetic information to compare and recognize different positions. This is, for example, the case when the colors of the environment are used to recognize a position in a topological map (Ulrich and Nourbakhsh, 2000). This paper will limit itself to methods that use allothetic sensors without any associated metrical model. Indeed, this choice makes a much more general use of allothetic data possible, as it does not require sensors measuring metrical properties of the environment. This way, information like a color, an odor or a temperature can be used to map the environment. Moreover, such simple use of allothetic data seems more representative of the way an animal like a rat builds an internal model of its environment.

Without metrical models of the sensors, however, a navigation system will have to cope with some limitations. Most of these limitations stem from the fact that it is impossible to infer what should be perceived at a distant position without actually going there. For example, it is easy, with a metrical sensor model, to infer that a wall perceived two meters away will be perceived as being one meter away if the robot moves one meter in the direction of this wall. On the contrary, such an inference is impossible without using a metrical sensor model. Consequently, a map-learning system will only provide information about positions that have already been visited at least once. As will be shown in the remainder of this paper, this limitation must be dealt with by the map-learning and localization procedures.

The main issue with map-based strategies lies in the necessity of simultaneously tackling localization and map-learning problems. The difficulty arises from the *chicken and egg* status of these problems (Yamauchi et al., 1999). In other words, a map is necessary to estimate the position, while knowing the position is necessary to update the map. It is true that the localization problem when a map is given a priori has been given efficient solutions (Thrun et al., 1999). Notably, some models are able to tackle the *lost robot problem*, i.e. the estimation of the robot's position without any initial cues about its position. Unfortunately, the corresponding models that are able to *globally localize* a robot are difficult to extend to on-line map-learning.

In the context of animat research, strong emphasis is placed on autonomy. A map-based navigation system should therefore make it possible to accurately localize an animat in any, eventually initially unknown, environment without human intervention. These requirements are met by global localization models that build environmental maps on-line. The model described in this paper affords solutions to such requirements. Moreover, for the reasons stated above, this model does not make use of any metrical sensor model. It draws inspiration from the literature on bio-mimetic navigation systems, on the one hand, and from purely robotic navigation systems, on the other hand. Several improvements to the simulation model presented by Filliat and Meyer (2000) will be described here, together with new results that were obtained with a real robot implementation.

2 Global localization and map-learning

Localization models described in the literature basically pertain to three categories called respectively *direct-position inference*, *single-hypothesis tracking* and *multiple-hypothesis tracking* (Filliat and Meyer, 2002).

2.1 Direct-position inference

These models (e.g., Franz et al., 1998, Gaussier et al., 2000) call upon environments and sensory capacities that are not subject to perceptual aliasing. Allothetic information is supposed to directly provide an unambiguous estimate of the position, without the need to use any idiothetic information. These models therefore heavily rely on perceptual systems that are able to discriminate between a great number of positions. However, such an hypothesis about the absence of perceptual aliasing within a whole environment is hard to assume *a priori* in any initially unknown environment.

2.2 Single-hypothesis tracking

These models (e.g., Smith et al., 1988, Dedeoglu et al., 1999) take the perceptual aliasing issue into account and solve it by using idiothetic information to disambiguate positions. This information is used to estimate the current position relative to the previous one, and this estimate is used to limit the search space of the position that corresponds to current allothetic data. Assuming that the restrained search area no longer exhibits perceptual aliasing, the corresponding position is unique. This mechanism allows a single position hypothesis to be tracked, as the alternative positions that would correspond to the same allothetic data are simply discarded.

This method is *local* in the sense that the current position is searched for only in the vicinity of the previous position estimate and not over the whole map. As a consequence, an initial position estimate has to be provided to the system either by a separate direct position inference mechanism or by an operator. This requirement limits the robot's autonomy and moreover precludes future correct position estimation if the current estimate should accidentally prove false.

2.3 Multiple-hypothesis tracking

A solution to avoid the dependence on an initial position estimate in perceptually aliased environments is to track multiple hypotheses of the robot's position. According to this scheme, instead of discarding the positions corresponding to current allothetic data that do not match the previous position estimate, these positions are memorized as alternative hypotheses of the robot's position. All these hypotheses are subsequently tracked in parallel and their relative credibilities are monitored. At every moment, the most credible hypothesis is considered as the robot's current position.

This approach allows a *global localization* that is not tied to an initial position estimate. Moreover, the set of concurrent hypotheses may be empty and may be initialized with all the positions that correspond to the first allothetic information gathered in the environment. Therefore, this approach solves the *lost robot problem*, and it affords a high degree of autonomy to the localization process.

The corresponding implementation may call upon the explicit process of monitoring several possible positions in parallel (Piasecki, 1995), or it may call upon Partially Observable Markov Decision Processes (Simmons and Koenig, 1995, Fox et al., 1998). These latter solutions may be viewed as implicit multiple-hypothesis tracking, where each possible position in the map is considered as a position hypothesis. This solution already yielded highly successful robots operating in challenging environments (Thrun et al., 1999).

2.4 Map-learning

From a recent review of map-learning strategies in robots (Meyer and Filliat, 2002), it appears that combining map-learning with direct position inference is relatively straightforward as it simply entails adding to the map allothetic situations that have never been seen before.

A lot of models also combine map-learning with single-position tracking methods (Arleo and Gerstner, 2000, Dedeoglu et al., 1999) because this approach still works when the robot gets outside the area already mapped. Indeed, in such case, it is straightforward to insert a new position in the map, because it is defined relatively to a previously known position.

On the contrary, combining map-learning with multiple-hypothesis tracking algorithms is more difficult. The reason is that these algorithms rely heavily on the completeness of the map to estimate the relative credibilities of the different position hypotheses. This estimation entails comparing what the robot currently perceives with what it should perceive in each of the possible positions monitored. Therefore, when the map is incomplete - which is the case during map-learning - this estimation is difficult, as the robot may be either inside or outside the currently mapped area. If it is inside, the global localization procedure can estimate the robot's position; if it is not, this procedure cannot be used.

Various attempts have been made to overcome this difficulty while nevertheless combining global localization with map-learning. A first method is to use off-line mapping algorithms that build a map corresponding, with the highest possible probability, to a set of data gathered by the robot (Shatkay and Kaelbling, 1997). However, this method does not meet our requirement of autonomy because localization and map-learning are to be separated.

A second method that works on-line is to use powerful distance sensors, along with associated metrical models, in order to prevent the robot from traveling outside the mapped area (Thrun et al., 2000). Indeed, as argued in the introduction, metrical sensor models make it possible to build a map that extends beyond the current robot's position. Accordingly, frequently estimating the robot's position guarantees that it always remains within the mapped area.

A third method will be used here, which combines global localization and map-learning without resorting to any metrical sensor model. This method entails frequently checking whether the robot is in the mapped area or not. If such is the case, a global localization algorithm can be used directly. If not, a single hypothesis tracking method based on the previous positions is used temporarily, until the robot re-enters the mapped area. To decide between these two alternatives, Filliat and Meyer (2000) proposed to simply use the credibility of the most credible among the concurrent position hypotheses. Should this credibility fall below a given threshold, the robot would be considered to be outside the mapped area. However, additional experiments with such a procedure showed it to be brittle, because the corresponding threshold needed to be changed according to the particular environment mapped. Moreover, large uncertainties in the robot's position, which lead to low credibilities of the concurrent hypotheses, always led to believe that the robot was outside the mapped area, thus rendering the mapping process quite unstable.

This paper describes an updated model where the decision between the two cases calls upon an heuristic based on the variation of the sum of credibilities of the various hypotheses. This heuristic, that will be described later on, efficiently detects when the robot exits the mapped area, thus affording the model a substantial gain in robustness, notably because the corresponding parameters become independent of the environment.

3 The-model

This section outlines a simplified version of the model that assumes that panoramic sensors are used. Experimental results presented further were obtained with directional sensors and active perception strategies described in Filliat and Meyer (2000) and Filliat (2001).

3.1 Structure

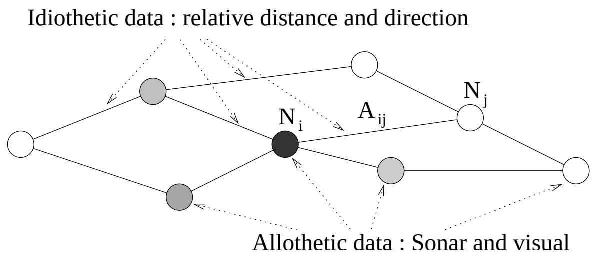

Figure 1: The topological map used in the model.

The map built by the system is a dense topological map, the nodes of which represent close positions in the environment (with a mean spacing of 25 cm). Each node stores the allothetic data that the robot can perceive at the corresponding place in the environment. A link between two nodes memorizes at which distance and in which direction the corresponding places are positioned relatively to each other, as measured by the robot's idiothetic sensors (Figure 1). All the directions used in the model are absolute directions, assuming a fixed reference direction given by a magnetic compass. The robot's position is represented by an activity distribution over the nodes : activity A_i of node i represents the probability that the robot is at the corresponding position. These probabilities are estimated using allothetic and idiothetic data gathered by the robot, as will be described in section 3.3.

The model iterates the following steps that are explained in the paragraphs below :

- Update the activity of each node in the map;
- Recognize a node as corresponding to the robot's current position or create a new one;
- Update visual and sonar data stored in the recognized node using the current allothetic data;
- Update the idiothetic data stored in the links;
- Choose the direction of the next move in order to explore the environment or to reach a goal.

3.2 Model inputs

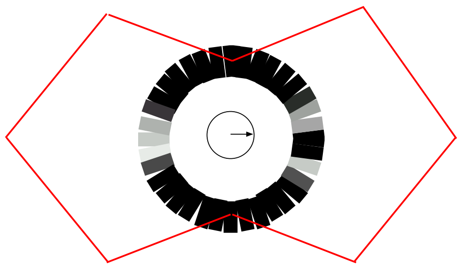

Figure 2: Schematics of allothetic data used in the model. The broken line joins the points detected by sonar sensors in eight absolute directions. The rectangles arranged on a circle indicate the mean grey-level perceived in the corresponding direction by the camera.

Two series of allothetic data are used in the model : sonar data and visual data (Figure 2). Sonar data are gathered through a 16-sonar belt and aggregated into eight virtual sensors that provide distances to obstacles in eight absolute directions. Visual data are gathered by an omnidirectional camera and down-sampled to the values of 36 virtual sensors that measure the mean grey-level of the environment in 36 absolute directions.

Both sonar and visual allothetic data are associated with a procedure P_O that compares two perceptions O_M and O_P. This procedure, which returns 1 if the two perceptions are identical, and decreases to 0 more quickly the more the perceptions are different, is used to estimate the probability that the robot is at a position characterized by data O_M, given the currently perceived allothetic data O_P. In the experiments described below, we used the following function[1] :

$$P_O(O_M/O_P) = \sqrt[l]{\prod_{k=1}^{l} F(O_M^k - O_P^k)}$$

[1]This procedure is adapted to the case of partial data when a directional camera is used. See Filliat (2001) for details.

where O_M^k and O_P^k are the values of allothetic data in the absolute direction k, l is the total number of directions for the considered sensor - i.e., eight for sonar data and 36 for visual data - and F is a Gaussian function given by $F(x) = e^{-x^2/K^2}$. The parameter K is chosen empirically for each sensor so as to give $P_O = 10^{-6}$ for maximally different sensor values. The model seems robust with respect to this parameter, since the same value was efficiently used for all simulated and real experiments.

Idiothetic data are used to estimate the probability that the robot has moved from one node in the map to another. Given a displacement of direction θ_{od} and length r_{od} measured by the robot's odometry, the probability of having moved from node A to node B is :

$$P_D(AB/od) = E_1 \times E_2$$

with :

$$E_1 = exp\left(\frac{-(\theta_{od} - \theta_{AB})^2}{L^2}\right)$$

$$E_2 = exp\left(\frac{-(r_{od} - r_{AB})^2}{M^2}\right)$$

where θ_{AB} and r_{AB} are the direction and length of the link between nodes A and B, L and M are empirically set to $L = 30$ *degrees* and $M = 20$ *cm* through statistics gathered on the moves interspersed with activity updates. Here also, the same values have been used in all simulated and real experiments.

3.3 Activity updates

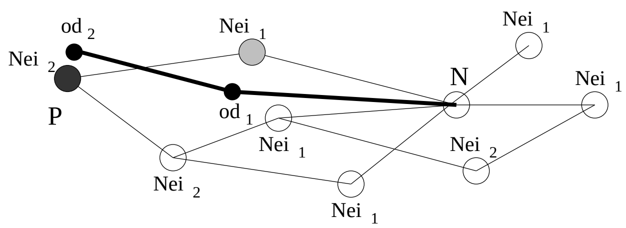

Figure 3: Illustration of the use of idiothetic data for activity updates. Nei_k is the set of all the nodes linked to node N by k connections, and od_k is the position of the robot at time $t-k$ as measured by the odometry relatively to node N. In this example, the activity of node N will be a function of the activity of node P, at time $t-2$ (see text for details).

The activity of each node is updated each time the robot has moved by a given distance (50 cm in the experiments). Such updates are directly inspired by the equation used in POMDP-based navigation models (Simmons and Koenig, 1995) and are adapted to the irregular structure of our model. Idiothetic data are first integrated using the equation :

$$A_i(t) = \max_{k\in[1..K]} \left(\max_{j\in Nei_k(i)} (A_j(t-k) \times P_D(ij/od_k)) \right)$$

where $Nei_k(i)$ is the set of all the nodes linked to node i by k connections, $A_j(t-k)$ is the activity of node j at time $t-k$, and od_k is the position of the robot at time $t-k$ as measured by the odometry relatively to node i.

The effect of this equation is to estimate the probability of the robot's being at node i, taking into account the node j that best fits the robot's path over K past time-steps (see Figure 3). The sum S_a of the activities of all the nodes is then calculated. It will be used to decide whether the robot is in the mapped area or not (see next section).

Then, allothetic data O_P are integrated using :

$$A_i(t+1) = A_i(t) \times P_O(O_i, O_P)$$

The effect of this equation is to increase the activities of nodes characterized by allothetic data that match the current perceptions, and to decrease the activities of the other nodes. Activities are then normalized such that their sum equals 1.

3.4 Position estimation

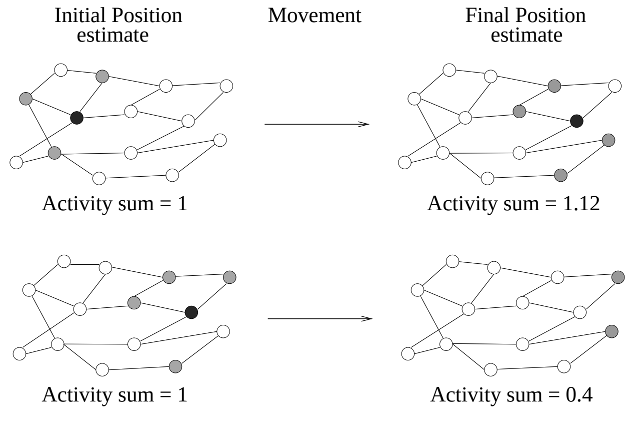

Figure 4: Illustration of the heuristic used to detect whether the current position is in the mapped area or not. When the robot is in the mapped area, the sum of the activities remains approximately constant (top half of the figure) while, if the robot exits the mapped area, the sum decreases (bottom half of the figure).

The model presented so far estimates the robot's most probable position, assuming that this position is part of the map. However, during map-learning, the robot can get out of the mapped area. To decide if the robot has exited the mapped area, an heuristic based on the variation of the sum of the activities before and after the integration of idiothetic cues is used. The idea underlying this heuristic is that, when the robot exits the mapped area, the sum S_a of the activities should suddenly decrease (see Figure 4). If the robot remains in the mapped area, on the contrary, this sum should either increase or remain stable.

Taking into account that the sum of activity is 1 before idiothetic cue integration, the algorithm used to estimate the position is then :

- If $S_a \geq 1$, the node with the highest activity is recognized as the current position.
- If $S_a < 1$, the robot's position is estimated using odometry information gathered since the last recognized node. If this position falls close to an existing node, this node is recognized; otherwise, a new node is added to the map[2]. Such a procedure amounts to temporarily using a single-position tracking method.

3.5 Map updates

Once the node corresponding to the current position has been determined, the allothetic data that characterize it are updated using the newly perceived data.

The direction and distance that correspond to the link between the previously recognized node and the current one are also updated using the newly measured displacement. To achieve map consistency, the values of all the links in the map are then updated using the relaxation algorithm of Duckett et al. (2000). 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. Basically, the relaxation algorithm "shakes" the relative positions of all the nodes in the map so as to make these relative positions as close as possible to their measured values, thereby resulting in a globally coherent map.

3.6 Exploration strategy

Once the map has been updated, the exploration of the environment resumes. The exploration strategy used in the model aims at limiting localization errors and at ensuring exhaustive exploration. As global localization is efficient only when the robot is in the mapped area, the exploration strategy limits the distance that the robot may travel in an unmapped area. This is implemented thanks to a mechanism that retraces the recent route backwards if the model consecutively creates five nodes, i.e. if the heuristic mentioned above detects that the robot is outside the mapped area during five consecutive

[2]It should be noted that the heuristic thus used has a tendency to over-estimate the novelty of a position, which results in having any unmapped position always being correctly recognized as new. However, it also often causes a position previously mapped to be classified as new. This over-estimation is compensated for by verifying the existence of a node close to the position estimated before creating a new one.

time-steps. When this mechanism is not active, on the contrary, the direction of movement is chosen towards the less explored area, i.e. the direction free of obstacles where there are fewer nodes in the map, so as to ensure exhaustive exploration.

3.7 Path planning

If a goal is assigned to the robot, a movement is planned towards this goal. To achieve this, a *policy*, determining in which direction D_i to move from each node i of the map to reach the goal, is calculated using a simple spreading-activation algorithm starting from the goal. The direction of the next move is then chosen according to a voting method (Cassandra et al., 1996). A score is accordingly calculated for 36 sectors of 10 degrees surrounding the robot. This score is the sum of the activities of the nodes whose associated direction falls in this sector :

$$V(d) = \sum_{d-5<D_i<d+5} A_i$$

where $V(d)$ is the score of the sector of direction d, D_i is the direction of the goal associated with node i, and A_i is the activity of node i. The direction to be taken by the robot corresponds to the sector that achieves the highest score.

A detour mechanism may also be triggered when the planned trajectory to the goal turns out to be blocked by an unforeseen obstacle (Tolman, 1948). In such a case, the contradiction between planned movements that would lead the robot to cross the obstacle and the local obstacle-avoidance procedures that repel the robot from this obstacle generates an oscillatory behavior in front of the obstacle. These oscillations are detected by a continuous check of the robot's progression and a threshold is used to detect when too low a progression indicates it is probably impossible to reach the goal. The nodes that are close to the robot's position are then excluded from the planning process, which is entirely repeated. This results in a new policy that avoids the blocked position and leads the robot to the goal by a different route whenever possible (Filliat, 2001).

4 Experimental results

The model has been implemented on a Pioneer 2 mobile robot (see Figure 5). This robot is equipped with 16 sonar sensors and a directional camera. Although a magnetic compass could be used to estimate the absolute direction, this sensor turned out to be inefficient in our environment because of numerous magnetic disturbances. In the current system, the direction is therefore estimated using the robot's odometry, and its error is periodically compensated for by manually aligning the robot with a reference direction. This correction is made every 50 time-steps, i.e. approximately every 10 minutes. A set of low-level procedures allows local obstacle-avoidance during navigation.

Figure 5 shows a map obtained by the system in the corridors of our laboratory, this map being superimposed on an architectural sketch of the environment. It was created in 2000 time-steps in approximately six hours of operation, most of this time being consumed in stopping and starting the robot and in orienting the camera at each time-step. This time could be significantly reduced by the use of an omnidirectional camera that would allow the system to operate without stopping the robot at each time-step. Be that as it may, the map thus obtained correctly reproduces the structure of the laboratory and permits the robot's position to be estimated precisely. Figure 6 shows part of the robot's trajectory, as estimated either by the whole localization system or by a sub-part of this system that called upon the robot's odometry only. The trajectory estimated by the whole system is closer to the real trajectory, because it remains in the open area and does not cross any wall, thus demonstrating that the localization system is efficient.

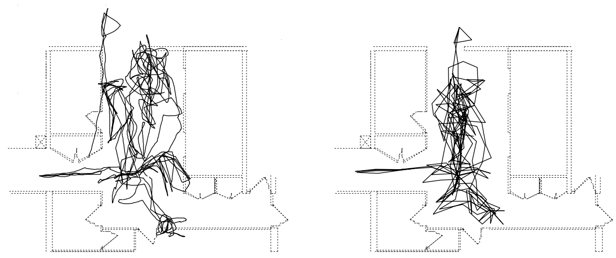

Figure 6: Comparison of two procedures to estimate the robot's trajectory. Left: results obtained with odometry alone. Right: results obtained with the full navigation system.

Moreover, the localization algorithm effectively achieves global localization most of the time. Indeed, it frequently computes the robot's position using node activities instead of using the position-tracking method that is temporarily triggered when the navigation system detects that the robot is outside the mapped area (Figure 7).

We carried out specific experiments to demonstrate this global localization capacity. In particular, we stopped the localization system when the robot was correctly localized at position A and subsequently manually moved it to position B in the environment of Figure 5. The standard localization and exploration process were then resumed without providing the system any cue about this displacement. Figure 8 shows the error in the estimation of the position during the subsequent lo-

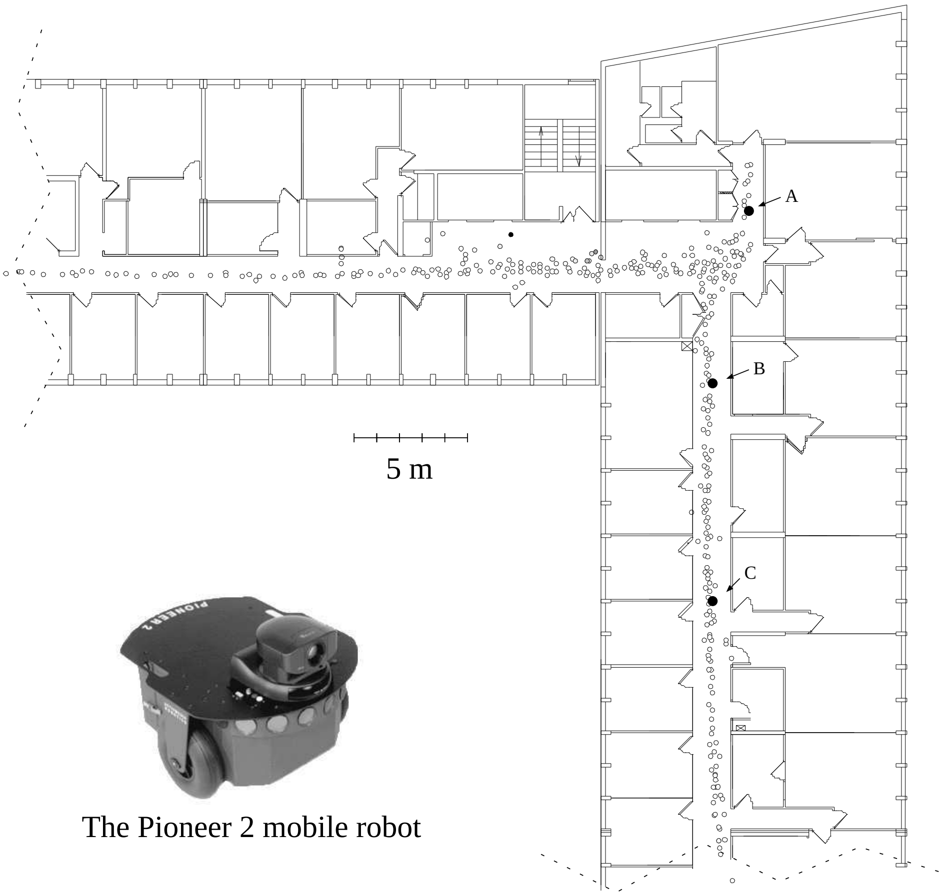

Figure 5: An example of a map created in the corridors of our laboratory. The map is superimposed on an architectural sketch of the environment.

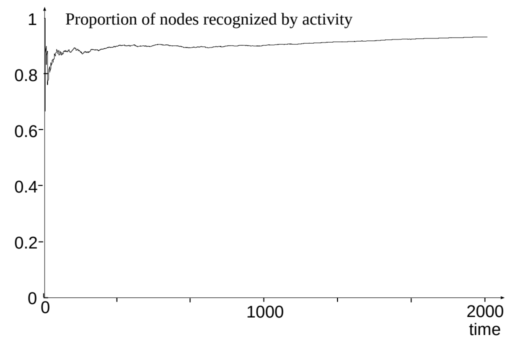

Figure 7: Proportion of the number of nodes that are recognized by the global localization system using node activities relatively to the total number of recognized nodes.

calizations. It thus turns out that the robot succeeds in getting correctly re-localized after 10 time-steps, when the localization error returns to its initial value, typically equivalent to the robot's diameter (50 cm). The large augmentation of the error between the third and seventh time-steps is caused by perceptual aliasing that causes the environment near position B to look very similar to the environment near position C. Consequently, while the robot is effectively positioned near position B, the system wrongly estimates that there is a high probability of its being near position C. Such an incorrect inference gets corrected after 10 time-steps when the robot is far enough from position B for the environment to be sufficiently different from what it looks like near position C.

It is important to note, however, that, contrary to what was demonstrated in simulation in a previous paper (Filliat and Meyer, 2000), such a re-localization capacity may temporarily prove to be inefficient. The main reason is that the real vision system is much noisier than the simulated one, which enhances perceptual aliasing difficulties. As a consequence, information provided by sonar sensors and by idiothetic cues about the structure of the environment is assigned much greater importance in actual case than in simulation. This causes the re-localization procedure to become inefficient on the real robot when, for instance, a wrongly estimated position belongs to the same corridor as the real one. In this case, re-localization is not effective until the robot has entered an open area or a different corridor. Unfortu-

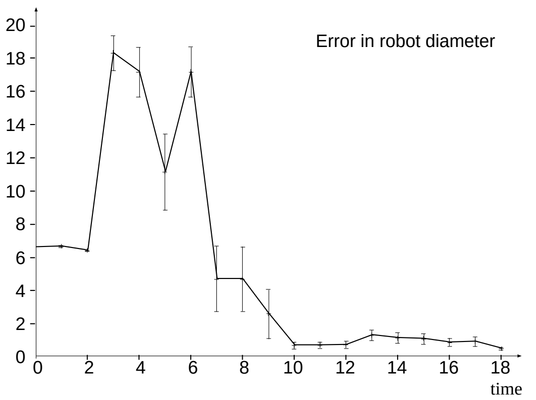

Figure 8: Evolution of the error in the estimation of the position after a passive displacement of the robot from point A to point B (Figure 5).

nately, the exploration strategies employed in the model emphasize strong local exploration in order to avoid localization errors. When such re-localization issues are encountered, local exploration prevents movements that would rapidly lead the robot out of a corridor and that would make prompt re-localization possible.

A solution to this problem would be to implement an active navigation strategy that would guide the robot toward areas where re-localization would be efficient. This suggestion is supported by the fact that, in the current system, manually assigning a goal to the robot when it is temporarily lost entails getting out of the corridor in question and permits a rapid re-localization.

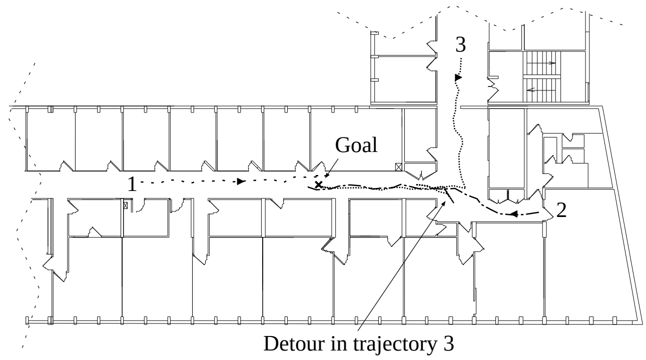

Figure 9: Three examples of goal-directed trajectories starting from three different positions. Trajectories 1 and 2 are direct, while trajectory 3 entails a re-planification leading to a slight detour.

Finally, the model makes it possible to efficiently reach any goal position in the environment. To demonstrate this, we performed ten trials to reach a fixed goal, starting from different positions. Among these trials, one failed due to the robot's getting trapped into a narrow dead-end. The nine other trials were successful, because the robot either directly reached the goal in five trials, or after the use of the above-mentioned re-planning procedure in the last four ones (Figure 9). The mean precision of the final positions in these nine trials was 50 cm, all trials ending less than 80 cm from the goal. These data are representative of the performance obtained with any other goal in the environment.

5 Discussion

The navigation system presented here therefore affords important autonomy capacities to mobile robots by combining global localization with map-learning. Its performances are achieved using relatively simple sensors and without resorting to metrical models for these sensors. The localization precision thus obtained (50 cm) is sufficient for most navigation tasks in common office environments[3]. In cases where it wasn't, the existing procedures could be supplied with additional short-range visual guidance algorithms, as demonstrated by current research efforts (Gourichon and Meyer, 2002).

The absence of any metrical model for the sensors is compensated in our model by the need for an exhaustive exploration of the environment. Indeed, the navigation system strongly relies on careful exploration to avoid localization instabilities during the map-learning process.

The capacities of the system have been demonstrated on a real robot in an environment mostly made up of hallways. Experiments in simulation indicate that navigating in open environments will be possible without any loss of precision using an omnidirectional camera. However, when a directional camera is used, as is the case in this paper, the system could present instabilities in the mapping process due to the higher rate of localization failures caused by the incompleteness of available data. In this case, the structure of the environment provided by the corridor is important, as shown by the mentioned limitations to the re-localization capacity. Further experiments in wider environments and using an omnidirectional camera will be conducted in the context of a new application within the AnimatLab, the Psikharpax project.

Complete autonomy of the system would be achieved if the robot were able to monitor its direction, along with its position. Indeed, the current method - which entails estimating the direction through odometry and periodically correcting the resulting error through an external reference - could be automated if the robot were able to learn how to associate the relative positions of some landmarks with its current orientation. Encouraging results have already been obtained with a preliminary implementation of such a capacity. This implementation entails first detecting colored landmarks from the initial position (using the method described in Gourichon and Meyer, 2002) and memorizing the directions

[3]For example, it allows a door to be reached correctly.

of these landmarks in the first map-node. The robot is then periodically guided by our navigation system toward this initial position where its direction estimate is reset using the perceived direction of these landmarks. Improvements to this scheme should entail memorizing such landmarks in several nodes of the map, so as to be able to reset the direction estimate in several positions and to avoid recurrent visits to the start node.

As mentioned in the previous section, the system could also be improved by the implementation of active navigation strategies to enhance the re-localization capacity. Such strategies could, for example, guide the robot, according to current position hypotheses, toward areas where the positions corresponding to the various hypotheses would be easy to differentiate.

With respect to other navigation models, this one shares several features with the ELAN model presented by Yamauchi and Beer (1996). However, the authors report that the latter model, which was functional in simulation, failed in real robot experiments. We believe that three main differences with respect to ELAN allow our model to work on a real robot and that they are therefore important for robustness :

- the regular correction of the direction by an external procedure that avoids large direction estimation errors and allows meaningful activity estimation,
- the use of vision instead of range sensors to reduce perceptual aliasing,
- the use of a dedicated heuristic to decide when to add a new node to the map.

This third point is particularly interesting, as the use of the heuristic mentioned by Yamauchi and Beer, i.e., a threshold on the most activated node, leads in our model to a severe loss of robustness. Comparisons of our model with other approaches can be found in Filliat and Meyer (2000).

Finally, this model is highly reminiscent of several biologically inspired navigation models described in the literature (Trullier et al., 1997). Indeed, nodes that have been used herein may be viewed as counterparts of *place-cells* found in the hippocampus of the rats. Our approach, however, relies on global localization, while most existing biologically-inspired models (e.g., Balakrishnan et al., 1999; Arleo and Gerstner, 2000) simply call upon single-hypothesis tracking and upon special procedures for the initial estimation of the position. Nevertheless, there are some indications that rats might in fact resort to global localization procedures also. For example, Zemel et al. (1997) describe a method to encode arbitrary probability distributions in the activities of a population of neurons. This technique potentially allows multiple-position hypotheses to be encoded in place-cell activities in a way very similar to what is done in our model. Another paper (Zhang et al., 1998), also demonstrates that deducing the position of a rat in a maze from place-cell recordings is much more precise when a probabilistic framework similar to that underlying this model is used, instead of resorting to a standard method like population vector coding.

In other words, such cues suggest that it might be useful to interpret the functioning of the hippocampus of rats during navigation within a probabilistic framework similar to the one used in this article.

6 Conclusion

The navigation system presented herein allows a high degree of autonomy by integrating global localization and map-learning processes with minimal human intervention. Moreover, this integration has been achieved using simple sensors, without resorting to any metrical sensor model, through the implementation of dedicated heuristics. Its capacities have been demonstrated on a real mobile robot operating in an unmodified office environment. Current research efforts to further enhance the autonomy of the system already provided encouraging results.

There is also good reason to think that the inner workings of the model could bear some resemblance to their biological counterparts found in the rat.

7 Acknowledgements

This work was supported by Robea, an interdisciplinary program of the French Centre National de la Recherche Scientifique.

References

Arleo, A. and Gerstner, W. (2000). Spatial cognition and neuro-mimetic navigation : A model of hippocampal place cell activity. *Biological Cybernetics, Special Issue on Navigation in Biological and Artificial Systems*, 83:287–299.

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.

Cassandra, A. R., Kaelbling, L. P., and Kurien, J. A. (1996). Acting under uncertainty : Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Dedeoglu, G., Mataric, M., and Sukhatme, G. S. (1999). Incremental, online topological map building with a mobile robot. In *Proceedings of Mobile Robots XIV - SPIE.*

Duckett, T., Marsland, S., and Shapiro, J. (2000). Learning globally consistent maps by relaxation. In *Proceedings of the International Conference on Robotics and Automation (ICRA'2000)*.

Filliat, D. (2001). *Cartographie et estimation globale de la position pour un robot mobile autonome (in french)*. PhD thesis, Université Pierre et Marie Curie.

Filliat, D. and Meyer, J. A. (2000). Active perception and map-learning for robot navigation. In *From Animals to Animats 6. Proceedings of the Sixth Conferences on Simulation of Adaptive Behavior*. The MIT Press.

Filliat, D. and Meyer, J.-A. (2002). Map-based navigation in mobile robots - I. A review of localization strategies. *Submitted for publication*.

Fox, D., Burgard, W., Thrun, S., and Cremers, A. B. (1998). Position estimation for mobile robots in dynamic environments. In *Proceedings of the Fifteenth National Conference on Articial Intelligence (AAAI-98)*.

Franz, M., Scholkopf, B., Georg, P., Mallot, H., and Bulthoff, H. (1998). Learning view graphs for robot navigation. *Autonomous Robots*, 5:111–125.

Gaussier, P., Joulain, C., Banquet, J., Lepretre, S., and Revel, A. (2000). The visual homing problem : An example of robotics/biology cross-fertilisation. *Robotics and autonomous systems*, 30(1-2):155–180.

Gourichon, S. and Meyer, J.-A. (2002). Using coloured snapshots for short-range guidance in mobile robots. *Special Issue on Biologically Inspired Robots - International Journal of Robotics and Automation*, Submitted for publication.

Meyer, J.-A. and Filliat, D. (2002). Map-based navigation in mobile robots - II. A review of map-learning and path-planing strategies. *Submitted for publication*.

Moravec, H. and Elfes, A. (1985). High resolution maps from wide angular sensors. In *Proceedings of the IEEE International Conference On Robotics and Automation (ICRA-85)*. IEEE Computer Society Press.

Piasecki, M. (1995). Global localization for mobile robots by multiple hypothesis tracking. *Robotics and Autonomous Systems*, 16:93–104.

Shatkay, H. and Kaelbling, L. P. (1997). Learning topological maps with weak local odometric information. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*.

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

Smith, R., Self, M., and Cheeseman, P. (1988). Estimating uncertain spatial relationships in robotics. In Lemmer, J. F. and Kanal, L. N., (Eds.), *Uncertainty in Artificial Intelligence*, pages 435–461. Elsevier.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Haehnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). MINERVA : A second generation mobile tour-guide robot. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-1999)*.

Thrun, S., Burgard, W., and Fox, D. (2000). A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55:189–208.

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.

Ulrich, I. and Nourbakhsh, I. (2000). Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-2000)*.

Yamauchi, B. and Beer, R. (1996). Spatial learning for navigation in dynamic environments. *IEEE Transactions on Systems, Man, and Cybernetics-Part B,Special Issue on Learning Autonomous Robots*, 26(3):496–505.

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

Zemel, R. S., Dayan, P., and Pouget, A. (1997). Probabilistic interpretation of population codes. In Mozer, M. C., Jordan, M. I., and Petsche, T., (Eds.), *Advances in Neural Information Processing Systems*, volume 9, page 676. The MIT Press.

Zhang, K., Ginzburg, I., McNaughton, B., and Sejnowski, T. (1998). Interpreting neuronal population activity by reconstruction : A unified framework with application to hippocampal place cells. *Journal of Neurophysiology*, 79:1017–1044.