Solving the Global Localization Problem for Indoor Mobile Robots

Leonardo Romero[1], Eduardo Morales[2], and Enrique Sucar[2]

[1] UMSNH, Morelia, Mich., 52000, Mexico, lromero@zeus.umich.mx
[2] ITESM Campus Cuernavaca, Mor., 62589, Mexico
{emorales,esucar}@itesm.mx

Abstract. Global localization is the problem of determining the position of a robot under global uncertainty. This problem can be divided in two phases: 1) from the sensor data (or *sensor view*), determine clusters of hypotheses where the robot can be; and 2) devise a strategy by which the robot can correctly eliminate all but the right location. In the second phase, previous approaches consider an ideal robot, a robot with a perfect odometer, to predict robot movements. This paper introduces a non deterministic prediction approach based on a Markov localization that include an uncertainty model for the movements of the robot. The non deterministic model can help to solve situations where a deterministic or ideal model fails. Hypotheses are clustered and a greedy search algorithm determines the robot movements to reduce the number of clusters of hypotheses. This approach is tested using a simulated mobile robot with promising results.

1 Introduction

Global localization is the problem of determining the location of the robot under global uncertainty. This problem arises, for example, when a robot uses a map that has been generated in a previous run, and it is not informed about its initial location within the map.

The global localization problem can be seen as consisting of two phases: hypothesis generation and hypothesis elimination [4]. The first phase is to determine the set of *hypothetical locations* H that are consistent with the sensing data obtained by the robot at its initial location. The second phase is to determine, in the case that H contains two or more hypotheses, which one is the true location of the robot, eliminating the incorrect hypotheses. Ideally, the robot should travel the minimum distance necessary to determine its exact location.

This paper presents an approach to solve the global localization problem in a known indoor environment modeled by an occupancy grid map, a two dimensional map where the environment is divided in square regions or cells of the same size. This approach is an improved version of the global localization approach given in [10]. We use a Markov localization (see [7,10]) in both phases, to represent and update the set H of hypotheses, and predict movements of the robot in order to eliminate hypotheses. The main contribution of this paper is to predict

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 416–423, 2003.

movements using a *Markov localization that includes an uncertainty model for the movements of the robot*. Previous approaches [8,5,10] only consider a determinist or ideal model for the robot (a robot with a perfect odometer) during the prediction process.

The rest of this paper is organized as follows. Section 2 describes relevant issues of our approach to generate hypotheses [10]. Section 3 presents the framework of Markov localization. Section 4 explains our approach to eliminate hypotheses. Experimental results using a mobile robot simulator are shown in Section 5. We choose a simulator because it is easy to create complex environments, with many similar places, to test the robustness of our approach. Finally, some conclusions are given in Section 6.

2 Hypotheses Generation

In this section a simple occupancy grid map is used as an example to show the ideas behind the proposed approach. Figure 1 (a) shows this simple map built using a mobile robot simulator. Figure 1 (b) shows a local map view, extracted from the map for the position of the robot shown in (a), considering that the robot direction is aligned with a global fixed direction (pointing downwards in this case). Figure 1 (c) shows the sensor view (generated by the simulator) considering that the robot is aligned with the global direction. Both views have an angular resolution of 0.5 degrees, a common value found in laser range sensors, and include the robot position for reference as a black cell. Perceptual limitations are taken into account setting a maximum range of 3 meters. The problem consist in estimating the set H of possible locations that have local map views consistent with the sensor view.

2.1 Polar Correlation

To have a simple model of robot motion and hence a small state space in the Markov localization, we assume that the robot should be in one of 8 possible directions ($\theta_i = 45 * i$ degrees, $i = 0, ..7$), with respect to the global fixed direction, one for each adjacent cell. A polar correlation, using a sum of absolute

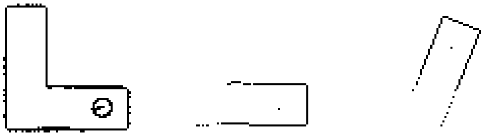

Fig. 1. A simple environment. From left to right: (a) Occupancy grid map. (b) Local map view computed from the map and the robot location showed in (a). (c) Actual sensor view

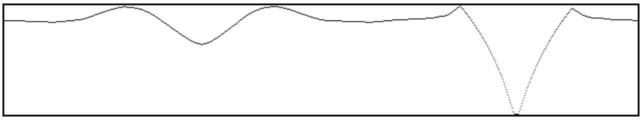

Fig. 2. Correlation results. Angular displacements are from 0 (left) to 359.5 degrees (right)

differences, can be used to find the match between a local map view and the sensor view. Figure 2 shows the correlation results for all the possible angular displacements of the sensor view against the local map view shown in Figure 1. From the minimum difference an angular displacement can be computed to align the robot with one of the directions θ_i. Obviously, the right angular displacement should be indicated considering the most probable position of the robot. In the case of Figure 2 the angular displacement corresponds to -21 degres, and the best estimated direction is 270 degrees. As the Markov localization needs a probabilistic value $p(s|l)$ of perceiving a sensor view s given that the robot is at location $l = (< x, y >, \theta_i)$, a difference value $d(s, v(l))$ can be computed from the correlation results between the sensor view s and the local map view at the cell $< x, y >$, denoted by $v(l)$, and then a probabilistic value can be obtained from $d(s, v(l))$. We compute $d(s, v(l))$ as the minimum difference (in the correlation results) for an angular interval with center at θ_i. The desired probability is computed by $p(s|l) = e^{-\alpha d(s,v(l))}$ where α is a positive real number.

Given that this procedure is expensive, the next section shows a fast way to find a small set of candidate cells to apply this procedure, instead of all the free cells in the map.

2.2 Roadmap

Following the ideas described in [9], the set of possible cells where the robot is allowed to move, tries to keep a fixed distance k to obstacles. Figure 3 (a) shows the full set of free cells where the robot can be as white pixels, while Figure 3 (b) shows the cells that form the roadmap. There is a significant reduction in the number of cells. To get a robust procedure, our approach considers a *thick roadmap* (see Fig. 3 (c)) which include the cells in the neighborhood of the *thin roadmap* (see Fig. 3 (b)). The idea is to use the thin roadmap to predict movements of the robot (in the case of more than one group of hypotheses) and to use the thick roadmap to restrict the possible locations where the robot can be. The following section describes the process to update the probability of hypotheses after the robot senses or moves.

3 Markov Localization

Following [5], the key idea of Markov localization is to compute a probability distribution over all possible locations in the environment. $p(L_t = l)$ denotes the

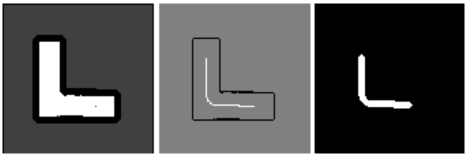

Fig. 3. A roadmap. From left to right: (a) Full set of free cells. (b) Thin roadmap for $k = 1$ m. (c) Thick roadmap

probability of finding the robot at location l at time t. Here, l is a location in $x - y - \theta_i$ space where x and y are Cartesian coordinates of cells and θ_i is a valid orientation. $p(L_0)$ reflects the initial state of knowledge and it is uniformly distributed to reflect the global uncertainty. $p(L_t)$ is updated whenever:

1. *The robot moves.* Robot motion is modeled by a conditional probability, denoted by $p_a(l|l')$. $p_a(l|l')$ denotes the probability that motion action a, when executed at l', carries the robot to l. $p_a(l|l')$ is used to update the belief upon robot motion:

$$p(L_{t+1} = l) \leftarrow \frac{\sum_{l'} p_a(l|l')p(L_t = l)}{p(s)} \tag{1}$$

   Here $p(s)$ is a normalizer that ensures that $p(L_{t+1})$ sums up to 1 over all l. An example of $p_a(l|l')$ is shown in Figure 4, considering that the robot moves forward in the thick roadmap, and the orientation of the robot is aligned to one of the possible 8 directions θ_i. Circles denote grid cells and the most probable transition in Figure 4 is labeled with 4/10.
2. The robot senses. When sensing s,

$$p(L_{t+1} = l) \leftarrow \frac{p(s|l)p(L_t = l)}{p(s)} \tag{2}$$

   Here $p(s|l)$ is the probability of perceiving s at location l. In order to get an efficient procedure to update the probability distribution, cells with probability below some threshold u are set to zero.

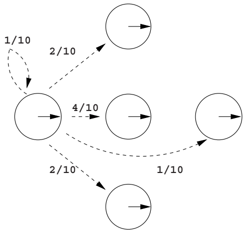

Fig. 4. Robot motion for one direction

4 Hypothesis Elimination

Locations, l, with $p(L_t = l) \geq u$ are clustered into n clusters or groups according to their location l using AutoClass [3], a Bayesian clustering technique.

The idea to eliminate hypotheses is to move the robot through the thin roadmap trying to reduce the number of clusters.

To get an efficient procedure, our approach considers that the mobile robot is at the most probable location and then considers all the cells of the thin roadmap as valid movements for the robot. If we assign to locations of the thin roadmap the number of possible groups of hypotheses, a good movement to eliminate hypotheses is to direct the robot toward the nearest cell with less than n groups. Let this cell be called the *goal* cell.

To compute the similarity between local map view associated to two cells of the thin roadmap we introduce a *similarity matrix*. Let c_i, $(i = 1, .., m)$ denote the m cells of the thin roadmap, and $sim(c_i, c_j)$ be the similarity between the local map views associated to cells c_i and c_j. A similarity measure sim can be computed using the correlation technique previously presented,

$$sim(c_i, c_j) = max_{i=0,..,7}\{P(s = v(c_i)|l = (c_j, \theta_i))\} \tag{3}$$

$sim(c_i, c_j)$ for all $i, j = 1, ..., m$ form a *similarity matrix* $\mathbf{S}$ that can be computed from the map and the roadmap, before the localization process starts.

If there are more than one group of hypotheses, we can predict a robot movement in two different ways: 1) using an ideal model for the robot movements and 2) using a model that include uncertainty. Let these types of prediction be called deterministic and non deterministic prediction respectively.

In both cases we use a Markov localization representation $p'(L)$, to track groups of hypotheses under the possible set of *virtual* movements of the robot; and $p(L)$ to represent the set of hypotheses of the location of the robot, given the set of *real* movements. When a prediction process starts, $p'(L)$ and $p(L)$ are the same. Once a goal cell in $p'(L)$ is computed, assuming that the robot location is given by the most probable hypothesis, the robot can move towards the goal cell, updating $p(L)$. The prediction process is repeated if there are more than one group of hypotheses, until there is only one group.

4.1 Deterministic Prediction

If the movements of the robot are considered deterministic or ideal, they can be represented like a rotation followed by a translation [5]. Let c_b be the most probable location where the robot can be. After a given *virtual movement* v from c_b to a cell c_i of the thin roadmap, the transformation given by v can be applied to all locations of the probability distribution. Let c_j be the new position for one hypothesis, after transformation v. After this virtual movement, a *virtual sensing* is applied. Here we use the similarity matrix to estimate $p(s|l)$, assuming that the robot is at location $l =< c_j, \theta_j >$, and that s corresponds to the local map view from cell c_i (the most probable):

$$P(s|l)) = \begin{cases} sim(c_i, c_j^{'})\ if\ c_j \in\ thick\ roadmap \\ 0 \qquad\qquad\quad otherwise \end{cases} \tag{4}$$

where $c_j^{'}$ is the cell of the thin roadmap closest to c_j.

4.2 Non Deterministic Prediction

An improvement over the deterministic prediction is to include an uncertainty model for the virtual movements of the robot (e.g. the model illustrated in Figure 4). A virtual movement from a cell c_b to c_i (as in the deterministic prediction) is split in a sequence of $< (m_1, s_1), (m_2, s_2)... >$, where m_i indicates a motion step to an adjacent cell, and s_i is a sensing step. In other words, instead of considering only a target cell where the robot can be, there will be a set of cells (limited by the trimming process after the sensing step). The sensing step is the same as in the deterministic prediction.

Considering that the thin roadmap is usually of one or two cells wide, results from an adjacent cell can be used to compute further results, giving a fast algorithm. In the implementation we use a breadth first search over the thin roadmap.

5 Experimental Results

This section presents preliminary results obtained using a mobile robot simulator. The robot simulates sonars and a low cost laser range sensor, implemented with a laser line generator and a camera. The laser sensor gives good measurements within a range of 3 m. The simulated robot has an uniform random error on displacements of ±10% and ±5% on rotations. We present two experiments to test the deterministic and non deterministic prediction for two complex environments.

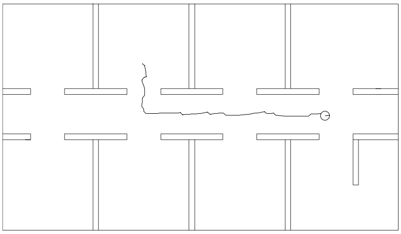

Fig. 5. A complex environment

Figure 5 shows a complex simulated environment of 17.5×10 m. At the beginning there were 6 groups of hypotheses, one per room except in the two rightmost rooms. After moving through 5 cells (of 10 × 10 *cm*) towards the predicted goal cell, a sensing step is inserted and the prediction process is repeated until there is only a group of hypotheses. At the end of the path followed by the robot, there is only a group of hypotheses, the right one. In this case, both types of predictions, deterministic and non deterministic, solved the localization problem and lead to similar paths.

Figure 6 shows another simulated environment of 12 × 15.5 *m* where the corridor on the left is slightly longer than corridor on the right. At the beginning there were 2 groups of hypotheses, C_0 and C_1. The deterministic prediction computes a goal cell P_d near the intersection of corridors, while the non deterministic prediction indicates a cell P_{nd} in the bottom part of the roadmap. In this case, the non deterministic prediction solves the localization problem and the deterministic prediction fails. The deterministic prediction is faster (2 seconds versus 6 seconds on a PC Pentium III 733Mhz) but it fails to solve the global localization problem.

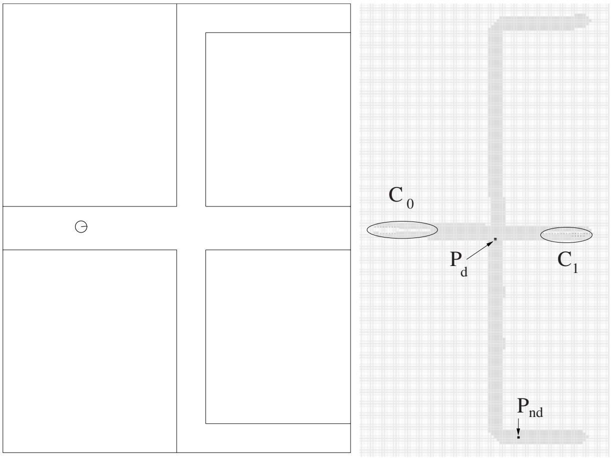

Fig. 6. (left) The simulator. (right) Roadmap with clusters C_0 and C_1 and results from the deterministic (P_d) and non deterministic prediction (P_{nd})

6 Conclusions

A robust approach to solve the global localization problem in indoor environments has been presented. It can be seen as the application of two Markov localization representations: one to track probable locations of the robot; an another to predict movements of the robot when there is more than one group of hypotheses. The second Markov representation can use a deterministic or a non determinist model for the movements of the robot. As the experiments confirm, a non deterministic prediction is more robust than a deterministic one, specially when the odometer of the robot or its sensors are not very accurate, or there are long corridors in the environment. In these cases the non deterministic prediction (using a model for the uncertainty of the robot), succeeds while a determinist prediction (modeled by a single rotation followed by a translation) can fail.

In the future, we plan to test this approach using real robots and environments with long corridors and similar places.

References

1. J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A.K. Peter, Ltd., Wellesley, MA, 1996.
2. J.A. Castellanos and J. D. Tardos. *Mobile robot localization and map building: a multisensor fusion approach*. Kluwer Academic Publishers, 1999.
3. P. Cheeseman and J Stutz. Bayesian classification (autoclass): Theory and results. In U.-M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, 1996.
4. G. Dudek, K. Romanik, and S. Whitesides. Localizing a robot with minimum travel. *SIAM J. Comput*, 27(2):583–604, 1998.
5. D. Fox and W. Burgard. Active markov localization for mobile robots. *Robotics and Autonomous Systems*, 1998.
6. L. Guibas, R. Motwani, and P. Raghavan. The robot localization problem. In K. Goldberg, D. Halperin, and J. C. Latombe, editors, *Algorithmic Foundations of Robotics*. A. K. Peters, 1995.
7. J.-S. Gutmann, W. Burgard, D. Fox, and K. Konolige. An experimental comparison of localization methods. In *Proc. International Conference on Intelligent Robots and Systems (IROS'98)*, 1998.
8. L. P. Kaelbling, A. R. Cassandra, and J. A. Kurien. Acting under uncertainty: Discrete Bayesian models for mobile-robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1996.
9. L. Romero, E. Morales, and E. Sucar. A robust exploration and navigation approach for indoor mobile robots merging local and global strategies. In M. C. Monard and J.S. Sichman, editors, *Advances in Artificial Intelligence, LNAI 1952*. Springer, 2000.
10. L. Romero, E. Morales, and E. Sucar. A hybrid approach to solve the global localization problem for indoor mobile robots considering sensor's perceptual limitations. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, Seatle, WA, 2001.
11. R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proc. International Joint Conference on Artificial Intelligence*, 1995.
12. S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, 1998.
13. S. Thrun, A. Bucken, W. Burgar, et al. Map learning and high-speed navigation in rhino. In D. Kortenkamp, R. P. Bonasso, and R Murphy, editors, *Artificial Intelligence and Mobile Robots*. AAAI Press/The MIT Press, 1998.
14. G. Weib, C. Wetzler, and E. Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Intelligent Robots and Systems*, 1994.