Mobile Robot Localization in Dynamic Environment using Places Recognition

Olivier AYCARD, Pierre LAROCHE & François CHARPILLET
LORIA, B.P. 239
54506 Vandœuvre-lès-Nancy CEDEX
FRANCE
Phone: +33 3.83.59.20.84 Fax: +33 3.83.41.30.79
E-mail: {aycard, laroche, charp}@loria.fr

Abstract

In this paper, we present a new method to localize a mobile robot in dynamic environments. This method is based on places recognition, and a match between places recognized and the sequence of places that the mobile robot is able to see during a run from an initial place to an ending place. Our method gives a coarse idea of the robot's position and orientation. Moreover, we can determine the actual state of places (i.e open doors, closed doors).

1 Introduction

When a mobile robot navigates, the knowledge of its position and orientation relatively to its environment is usefull. These informations are crucial to know if a specified goal has been reached, or to know the position of the mobile robot on a predefined path.

Dead reckoning is a simple method for mobile robot that integrates wheel translation and rotation to determine the robot's Cartesian location. However, due to slippage between the robot's wheel and the ground, the translations and rotations measured by the wheel encoders may not reflect the robot's actual motions.

Some studies have been done to avoid this drift. This can be done in two different ways :

- In the first technique called "Map-based positioning", the robot uses its sensors to create a map of its local environment. This local map is compared to a global map previously stored in the memory. If a match is found, then the robot can compute its actual position and orientation in the environment. [4] uses ultrasonic sensors to build a geometric map of the environment and an Extended Kalman Filter to compute the drift in position and orientation.
  The great advantage of this method is that at any time the mobile robot have a precise orientation and position. But this method is only usefull when the environment contains enough stationnary easily distinguishable features that can be used for matching. Moreover, we should note that currently most work in map-based positioning is limited to laboratory settings and to relatively simple and nearly static environments.
- The second method is called "Place-based positioning". Places like T-intersections or open doors have a fixed and known position, relative to which a robot can localize itself. The robot uses its sensors to recognize places in its local environment. So, the main task in localization is then to recognize the places reliably and to calculate the robot's position. [8][5] define the environment as a graph, where each node corresponds to a place, and arc are paths to go from one place to an other. [8] represents each place by an evidence grid[3]. Using dead reckoning and a grid-matching algorithm, the mobile robot computes the drift in position and orientation when its mobile robot passes in a previously learned place. [5] defines a number of distinctive places, and defines rules to recognize two distinctive places by ultrasonic sensors, and attaches a visual signature to differenciate two places of the same class. The recognition of places is performed using sensors and vision. As its mobile robot navigates with a simple wall-following in an environment constitued of corridors, the goal of localization is to compute the robot position in the crossed corridor.
  This technique only performs a periodical localization, but as it is based on places recognition,

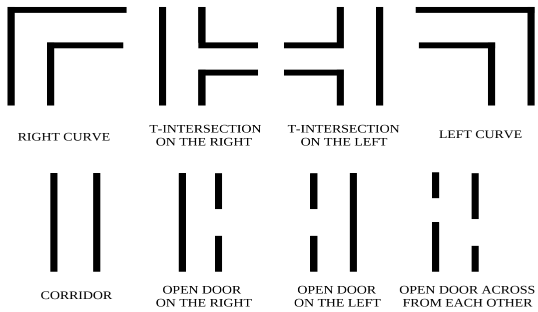

Figure 1: The 8 global places

it is more robust to changes in environment.

In this paper, we present a new method to localize a mobile robot using "Place-based positioning". We too represent the environment as a graph, where each node corresponds to a place, and arcs are paths to go from one place to an other. As [5], we define a certain number of distinctive places. We define 8 classes of places (figure 1). We only use ultrasonic sensors to recognize places. Our goal is to have a coarse idea of the robot position during a run from an initial place to an ending place. So we extract from the graph the sequence of places that the robot will see and compare them with the sequence of places recognized during the run. This comparison is based on the rate of confidence we have in places recognition. It gives the robot position even if some places are not visible, for instance an open door which has been closed. Moreover, the comparison is usefull to determine the actual state of the environment (i.e the opened doors, the closed doors and the free T-intersections), and to update the graph representing the environment.

This paper is organized as follow. In section 2, we give a short presentation of our mobile robot. In section 3, we summarize previous work [2] on place learning and recognition using second-order Hidden Markov Models. Section 4 is the description of our methodology. We discuss results in section 5 and give some conclusions and perspectives in section 6.

2 Description of our robot

Our robot is a Nomad200 (figure 2) manufactured by Nomadic Technologies. It is composed of a base and a turret. The turret can rotate independently of the base.

The base is formed by 3 wheels and a ring of 20 tactile sensors. They detect contact with objects. They are only used for the emergency cases. They are associated with low-level reflexes such as emergency stop and backward movement.

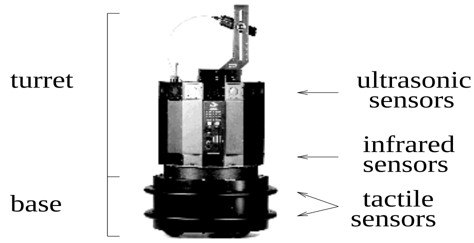

Figure 2: Our mobile robot

The turret is an uniform 16-sided polygon. On each side, there is an infrared and an ultrasonic sensor. The ultrasonic sensors give range information from 17 to 255 inches. But the quality of the range information greatly depends on the surface of reflection, and the angle of incidence between the ultrasonic sensor and the object. The infrared sensors measure the difference between emitted light and reflected light. They are very sensitive to the ambient light, the object color, and the object orientation. Since we assume that for short distances, the range information is acceptable, we just use infrared sensors for the areas shorter than 17 inches, where the ultrasonic sensors are not usable.

3 Previous work

In previous work [2], we presented a new method to learn and recognize 8 distinctive type of places (figure 1) based on second-order Hidden Markov Models (HMM2s). HMM2s have been shown to be efficient models for capturing temporal variations in speech [6] and in many cases they surpass first order Hidden Markov Models (HMM1) when the trajectory in the state space has to be accounted for. We used them to learn and recognize places by a mobile robot running in an indoor environment. For this, we built a model for each place, and trained it using a learning corpus.

To evaluate the rate of recognition, we compare what the robot had to see, and what it had really recognized. Our mobile robot makes 20 passes (back and forth) in a very long corridor (approximatively 30 meters) composed of a combination of the 8 classes of places. During this experiment, the robot uses its own reactive algorithm [1] to navigate in the corridor.

A place is recognized if it has been detected by the corresponding model and it has been found close to its real geometric position. But different types of errors occured:

	right curve	right inter	right door	left curve	left inter	left door	door door	Ins.
right curve	17	0	0	0	0	1	0	0
right inter.	0	9	0	0	0	0	0	1
right door	0	0	42	0	0	1	1	24
left curve	0	0	0	14	0	0	0	1
left inter.	0	0	0	0	8	0	0	0
left door	0	0	2	1	1	43	1	32
door door	0	0	0	0	0	1	2	4
ommissions	1	0	1	0	0	0	0	0
Total	18	9	45	15	9	46	4	62
% reco.	94	100	93	93	89	93	50	

Table 1: Confusion matrix

Insertions: the robot has recognized a non existing place. This corresponds to an over segmentation in the recognition process. Insertions are actually considered when the width of the place is more than 80 centimeters;

Deletions: the robot has missed the place;

Substitutions: the robot has confused the place with an other.

The results are presented in a confusion matrix (table 1). An element c_{ij} at row i and column j is the number of time the model j has been recognized when the right answer was the place i. For instance over 133 corridors seen, 126 have been recognized as corridors, but 3 have been recognized as right doors, 3 have been recognized as left doors and has not been recognized.

Most of the insertions are due to the inaccuracy of the navigation algorithm and to the unexpected obstacles. Sometimes the mobile robot has to avoid people or obstacles, and in these cases it does not always run parallel to the two walls, in the middle of the corridor. These conditions cause reflections on sensors which are interpreted as places.

4 Our approach

If we want to use our previous results to localize our mobile robot during a run, we have to face to several problems. On one hand, the rate of recognition is not perfect. If we recognize a place, we will not know if it is a right recognition, a wrong recognition or an insertion. So we can not use it without comparing the places recognized with the places seen, to detect insertions and bad recognitions. On the other hand, it is impossible, due to the dynamic of the environment, to know **a priori** what sequences of places the robot is able to see during a run from an initial place to an ending place. For instance, when extracting the informations from the graph, we determine that during a run, the mobile robot has to recognize a sequence of places in which there are 3 open doors on the left, and if at least one of these doors has been closed, the sequence of places extracted from the graph will not correspond to the actual state of the environment, and the comparison will have no sense.

On the other hand, given the confusion matrix (table 1) we can know which confidence we give to a place that has been recognized. Moreover, given an initial place and an ending place, we can easily extract the sequence of places the robot will see. If we can take into account that some places are not visible (i.e a closed door), it is possible to determine the actual state of the environment and the position of the robot. Describing the sequence of places visible using an automaton (figure 6), each time a place is recognized we will be able to determine, function of the confidence according to this recognition, the state of the automaton where the mobile robot has the highest probability to be.

Our problem of localization can be cast in a classic HMM problem : given a sequence of recognized places o, find the optimal state sequence. This problem is usually solved using the forward-backward algorithm. In the next subsections, we present HMMs, and how we applied them to build an HMM representing the sequence of places that the robot will see during its movement. In the last subsection, we present the modifications we performed to adapt the forward-backward algorithm to solve our problem of localization.

4.1 Definition of Hidden Markov Model

A very complete tutorial on Hidden Markov Models and their application can be found in [7].

An Hidden Markov Model is defined by :

- A set of states including an initial state and a final state defining the topology of the Model.
- A matrix A of probabilities of transitions from one state to another, where a_{ij} is the probability of going from state i to state j.
- A matrix B of probabilities of observations associated with each state, where $b_i(j)$ is the probability of observing symbol j in the state i.

4.2 Use of confusion matrix

To build the matrix of confidence according to diverse recognized places (table 2), we only take the transposate of the confusion matrix (table 1) and normalyze it. In fact, we want here to determine what

	corridor	right curve	right inter	right door	left curve	left inter	left door	door door	Omm.
corridor	84%	0%	0%	4%	0%	0%	4%	0%	1%
right curve	0%	94%	0%	0%	0%	0%	0%	0%	6%
right inter.	0%	0%	90%	0%	0%	0%	0%	0%	0%
right door	1%	0%	0%	58%	0%	0%	2%	0%	2%
left curve	1%	0%	0%	0%	93%	0%	1%	0%	0%
left inter.	0%	0%	0%	0%	0%	100%	4%	0%	0%
left door	0%	6%	0%	1%	0%	0%	51%	17%	0%
door door	0%	0%	0%	3%	0%	0%	1%	17%	0%
Ins	14%	0%	10%	33%	7%	0%	37%	66%	

Table 2: Confidence matrix

has been really seen as a function of what has been recognized.

4.3 Construction of the topology of the model

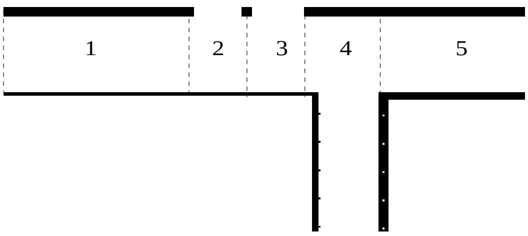

Figure 3: Description of a part of a corridor

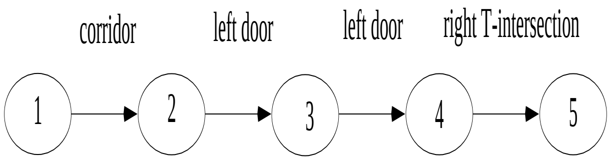

Figure 4: Part of the graph corresponding to the description

Given the description of a part of a corridor, where the mobile robot has to go from the location 1 to the location 5, the topology of our model (figure 4) is a part of the graph representing the environment. At this moment, our topology is only a direct path going from the initial place to the last place.

To take into account insertions, we add a transition which allows to stay in the same state of the automaton (figure 5).

All the places are not obligatory visible : for instance, a door can be closed or a T-intersection blocked. In this case, when the mobile robot will pass in front of these places it will see a corridor. So, when the mobile robot has recognized a corridor, we must

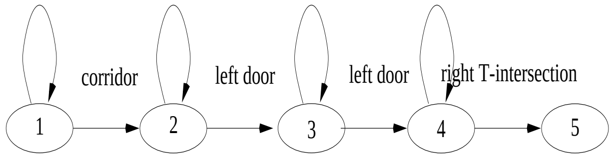

Figure 5: Taking into account insertions

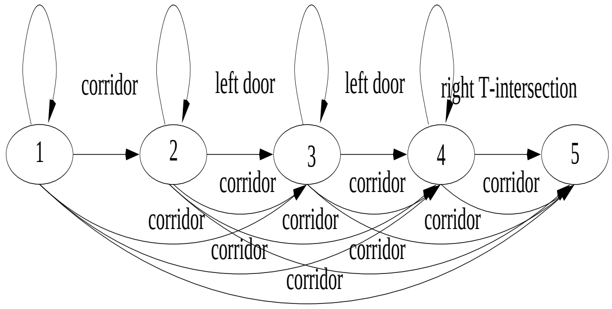

Figure 6: Taking into account that some places are not visible

take into account the fact that one or some places are not visible. For this, we add some transitions which permits to the mobile robot when it recognizes a corridor to go from the current state to all other next states (figure 6).

4.4 Localization

Using the forward-backward algorithm, the probability of the robot being in state i at time t can be expressed as :

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

As we want to know the most likely state after each recognized place, we only need to know the forward variable. As a consequence, our problem has to be rewritten :

$$\gamma_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)}$$

This forward variable has to be computed inductively, as follows :

1. Initialization

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), \begin{array}{l} 1 \leq j \leq N \\ 1 \leq t \leq T-1 \end{array}$$

Here π_i is the probability of being in state i, a_{ij} is the probability to reach the state j starting from state i and $b_i(o)$ expresses the probability of having recognized the place o in state i. So our problem of localization will be solved easily if we know the transition and the observation probabilities.

4.4.1 The transition probabilities

We have three classes of transitions : self-transitions, one-step transitions and several-steps ones.

- self-transition : the robot is in state i, it recognizes a place o. To stay in the same state, o has to be an insertion. As each place has a different probability to be inserted, the self-transition probability will be dependent of the place recognized : we write $a_{ii(o)} = Pr(Inserted, o)$.
- one-step transition : this is the "normal" transition. The robot is in state i, it recognizes a place o and goes to state j. In fact, this is a little more complicated, because closed doors and blocked intersections are recognized like a corridor. The probability to go from one state to the next one is also dependent of the place recognized :

$$a_{i(i+1)(o)} = \begin{cases} \text{if } l_i \neq Corridor \\ Pr(l_i, o) + Pr(Corridor, o) \\ \\ Pr(l_i, o) \text{ otherwise} \end{cases}$$

  where l_i is the place that has to be recognized to leave the state i.
- multiple steps transitions : this occurs when one or several places have been omitted or one or several places are blocked (closed doors or blocked T-intersections are recognized as corridors). To take into account the probability of a place to be blocked, we divide the probability to be in a corridor by the number of blocked places. In this case, we have

$$a_{ij}(o) = \left[\prod_{k=i\to j-2} Pr(l_k, Ommission)\right] * Pr(l_{j-1}, o) + \frac{Pr(Corridor,o)}{j-i-1}$$

4.4.2 The observation probabilities

As we have seen in the previous section, the transition probabilities are dependent of the places recognized and the observations corresponding to the states. So we don't need anymore the b function used in the forward algorithm, which is rewritten as follows :

1. Initialization

$$\alpha_1(i) = \pi_i, 1 \leq i \leq N$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{i<=j} \alpha_t(i) a_{ij(o_{t+1})}\right], \quad \begin{array}{l} 1 \leq j \leq N \\ 1 \leq t \leq T-1 \end{array}$$

4.4.3 A brief example

To explain how our system works, we will develop it on a simple example. The robot crosses the corridor (figure 3), beginning in state 1 (with probability 1), and recognizes the following places : right door, corridor, left door, right T-intersection, left door, corridor. Figure 7 shows the most likely state after each observation. The first right door has to be an insertion, then the first left door is closed (cause we only recognize a corridor).

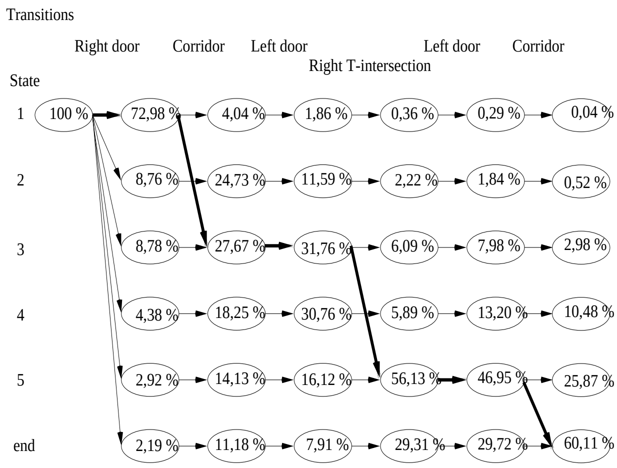

Figure 7: Recognition of the state of the environment

5 Experiments and results

In this section, we give the detail of a mission where the robot ran from the begin of the corridor to the T-intersection (figure 8). The recognition is performed online. The table 3 gives the recognized places, a deduction of the actual state of the environment and approximation of the position of the robot.

As long as the places are recognized, the comparison with the description of the environment permits to know the position of the mobile robot and the actual state of the places of the environment. All the open doors, the T-intersection and the curve at the beginning of the corridor have been recognized. On the

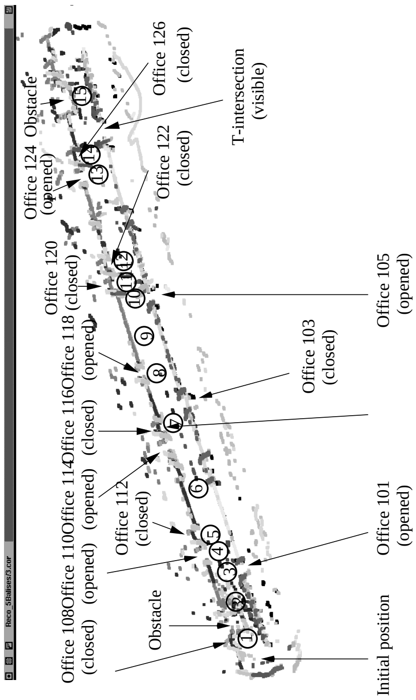

Figure 8: State of the environment

New balises recognized	position of the robot	state of the environment
Corridor Right curve	1	Insertion Initial curve visible
Left door	2	Office 108 opened
Corridor	3	
Left door Right door	4	Insertion Office 101 opened
Left door	5	Office 110 opened
Corridor	6	Office 112 closed
Left door	7	Office 114 opened
Corridor	8	Office 116 closed
Left door	9	Office 118 opened
Corridor	10	
Right door	11	Office 105 opened
Left door	12	Office 120 opened
Corridor	13	Office 122 closed
Left door	14	Office 124 opened
Corridor right inter.	15	Office 126 closed right inter. visible

Table 3: Use of places recognized to deduce the actual state of the environment, and the approximative position of the robot

other hand, the robot has recognized two open door (Office 108 and 120) instead they were closed. Each of these two doors are near an other door, and in this configuration, an insertion is difficult to detect.

Each time, a new place has been recognized, and the corresponding place in the environment has been found, we can be sure the robot is recognizing the next place of the sequence representing the corridor. So, the mobile robot is in front of the place, it is recognizing.

6 Conclusion

In this paper, we presented a new method to localize a mobile robot using "Place-based positioning". Our method gives a coarse idea of the robot position during a run from an initial place to an ending place. Moreover, it determines to the actual state of the environment (i.e the opened doors, the closed doors and the free T-intersections), and to update the graph representing the environment.

On the other hand, if some places are not visible and at the same time, the mobile robot inserts some places, we will find a wrong state of the environment and a wrong localization. This problem can be resolved using distance informations, and a Partially Observed Markov Decision Process, which will take into account the length of the displacement of the robot, and moreover, will choose the best motion to increase the knowledge about the environment.

References

[1] O. Aycard, F. Charpillet, D. Fohr, and J.F Mari. Place learning and recognition using hidden markov models. In *proceedings of IEEE/IROS*, pages 1741–1746, 1997.

[2] O. Aycard, F. Charpillet, and J.P Haton. A new approach to design fuzzy controllers for mobile robots navigation. In *proceedings of IEEE/CIRA*, pages 68–73, 1997.

[3] A Elfes. Using occupancy grids for mobile robot perception and navigation. *IEEE Computer*, 22(6):46–57, June 1989.

[4] J.J.Leonard and H.F.Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publishers, 1992.

[5] D Kortenkamp and T Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *proceedings of the 1994 AAAI Conference*, 1994.

[6] J.-F. Mari, J.-P. Haton, and A. Kriouile. Automatic Word Recognition Based on Second-Order Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, 5, January 1997.

[7] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech recognition. *IEEE Trans. on ASSP*, 77(2):257 – 286, February 1989.

[8] B Yamauchi. Mobile robot localization in dynamic environments using dead reckoning and evidence grids. In *proceedings of the IEEE/ICRA*, pages 674–680, 1996.