Second Order Hidden Markov Models for Place Recognition: New Results

Olivier AYCARD, Jean-François MARI & François CHARPILLET
LORIA, B.P. 239
54506 Vandœuvre CEDEX, FRANCE
Phone: +33 3.83.59.20.84 Fax: +33 3.83.41.30.79
E-mail: {aycard, jfmari, charp}@loria.fr
WWW : http://www.loria.fr/{˜aycard, ˜jfmari, ˜charp}

Keywords: Hidden Markov Models, Robotics, Sensor Interpretation.

Abstract

Second order hidden Markov models have been used for a long time in pattern recognition, especially in speech recognition. Their main advantages over other methods (neural networks ...) are their capabilities to model noisy temporal signals of variable length. In a previous work, we proposed a new method based on second order hidden Markov models to learn and recognize places in an indoor environment by a mobile robot, and showed that this approach is well suited for learning and recognizing places. In this paper, we propose major modifications to increase the global rate of places recognition. Results of experiments on a real robot with distinctive places are given.

1 Introduction

The automatic recognition of places is an important issue that determines the capability of a mobile robot to localize itself in its environment. Place recognition is useful in a variety of tasks such as the automatic construction of topological maps [4]. Place learning and place recognition have been addressed by different approaches such as analytical methods or pattern classification methods. In the first approach, the problem is studied as a reasoning process. For instance, [9] uses ultrasonic sensors to build evidence grids [3] associated with places and defines an algorithm to match two places. On the contrary, in the second approach, stochastic [1] or neuronal [7] models of places are built from training sensor data. The recognition process is envisioned as the association of the signal acquired from sensors with a model of the place to identify.

The purpose of this paper is to present a pattern classification approach of the recognition process in which places are modelized using second order hidden Markov models (HMM2). This is a robust approach for handling the large variability of complex temporal signals such as the one perceived by a mobile robot. One of the main advantage of this approach is the capability of automatically building stochastic models from temporal signals even if they are noisy. HMM2s have been shown to be efficient models for capturing temporal variations in speech [5] and in many cases they surpass first order hidden Markov models when trajectory in the state space has to be accounted for.

In robotics, [8] used stochastic models based on Partially Observable Markov Decision Process (POMDP) for developping navigation methods. Their objective is to find, given an observation, the best action to reach a predefined goal, whereas our goal is to find, given an observation, the place recognized. Moreover, they do not use Markov models for sensor interpretation, and include actions of the robot in their model. On the contrary, in our work, we use Markov models for sensor interpretation, but planification and execution of actions are performed by other techniques [6].

In a previous work [1], HMM2s have been shown to be efficient for the place recognition problem. Even if a good recognition rate has been obtained (75%), some limits of our approach have to be overcame. We propose in this paper a substantial improvement of our previous approach : the new recognition rate reaches 92%.

In this paper, we briefly describe our mobile robot in section 2. In section 3, we summarize the HMM2s and the algorithms for training and recognition of places. Section 4 describes the methodology used in [1], presents the results obtained and discuss the limits of these results. In section 5,

[0] Thanks to Richard Washington (richw@ptolemy.arc.nasa.gov) for useful comments and corrections on earlier drafts of this paper

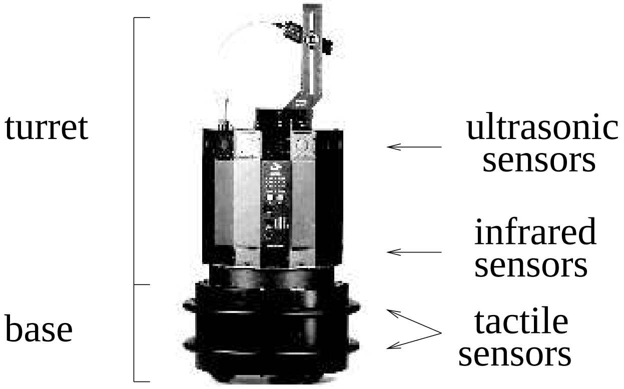

Figure 1: Our mobile robot

we present majors modifications we performed to increase the global rate of recognition. We give some conclusions and perspectives in section 6.

2 Description of our robot

Our robot is a Nomad200 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.

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 The Second Order Hidden Markov Models

In a HMM2, the underlying state sequence is a second-order Markov chain. Therefore, the probability of a transition between two states at time t depends on the states in which the process was at time $t-1$ and $t-2$.

A second order hidden Markov model λ is specified by:

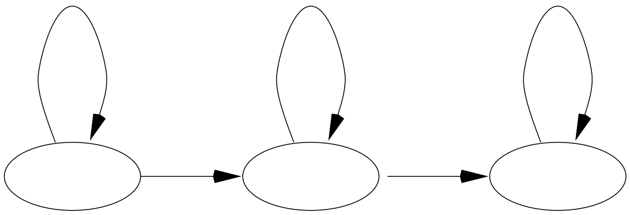

Figure 2: Topology of states used for each model of place

- a set of states called S;
- a 3 dimensional matrix a_{ijk} over S x S x S

$$a_{ijk} = Prob(q_t = s_k / q_{t-1} = s_j , q_{t-2} = s_i)$$

$$= Prob(q_t = s_k / q_{t-1} = s_j , q_{t-2} = s_i , q_{t-3} = ...)$$

$$with \sum_{k=1}^{N} a_{ijk} = 1 \quad for\ 1 \leq i \leq N \ , \ 1 \leq j \leq N$$

  where N is the number of states in the model and q_t is the actual state at time t ;
- each state s_i is associated with a mixture of Gaussian distributions.

In this formalism, each place to be recognized is modeled by an HMM2 whose topology is depicted in figure 2.

In this experiment, we have to face several major issues: designing efficient algorithms for training and recognition purposes; collecting a corpus of observations during several runs and labelling this corpus by finding temporal borders of each item that the robot has observed during its run.

3.1 The recognition phase

The recognition is carried out by the Viterbi algorithm. This algorithm is a dynamic programming search that determines the most likely state sequence (ie, the most likely place sequence) given a sequence of observations. The most likely state sequence is obtained by keeping track of back pointers for each computation of which previous transition leads to the maximal partial path probability. By tracing back from the final state, we get the most likely state sequence (ie, the most likely place sequence).

The robot's environment is described by means of a grammar that enables some sequence of models and restrict other ones. According to this grammar, a bigger HMM2, constituted by a sequence of HMM2 of places, is built on which the Viterbi algorithm is used. Then, the best sequence of states determines the ordered list of places that the robot saw during its run. It must be noted that the list of models is known only when the run is completed.

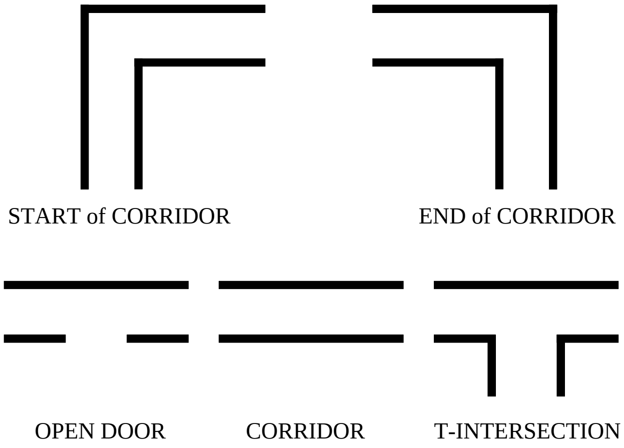

Figure 3: The 5 places to learn

3.2 The learning phase

The learning of the models is performed with the Baum-Welch algorithm using the maximum likelihood estimation criteria that determines the best model's parameters according to the corpus of items. Intuitively, this algorithm counts the number of occurrences of each transition between the states in the training corpus. Each count is weighted by the probability of the alignment (state, observation).

It must be noted that this criteria does not try to separate models like a neural network does, but only tries to increase the probability that a model generates its corpus independently of what the other models can do.

More details on these two algorithms and their extensions for second order hidden Markov models can be found in [1].

4 Application to mobile robotics

In [1], we chose to model five distinctive places (figure 3) that are representative of our office environment: a corridor, a T-intersection, a start of corridor when the robot reaches the start of a corridor, an end of corridor when the robot reaches the end of a corridor and an open door. This set of items is a complete description of what the mobile robot can see during its run. All other unforeseen objects, like people wandering along in a corridor or static obstacles, are treated as noise.

4.1 The corpus collecting and labeling

We built a corpus to train a model for each place. For this, our mobile robot makes 50 passes (back and forth) in a long corridor (approximatively 30 meters). This corridor contains two curves (one at the start of the corridor and one at the end), a T-intersection and some open doors (at least four, and not always the same). The robot ran with a reactive navigation algorithm [2] which enables the robot to cross a corridor staying as far as possible in the middle of the corridor in a direction parallel to the two walls constituting the corridor while avoiding unknown obstacles. While running, on each pass all the robot's ultrasonic sensor data are stored in a file. The 100 files corresponding to the 50 passes constitute the learning corpus. The acquisitions are done in real conditions with people wandering in the lab, doors completely or partially opened and static obstacles like shelves.

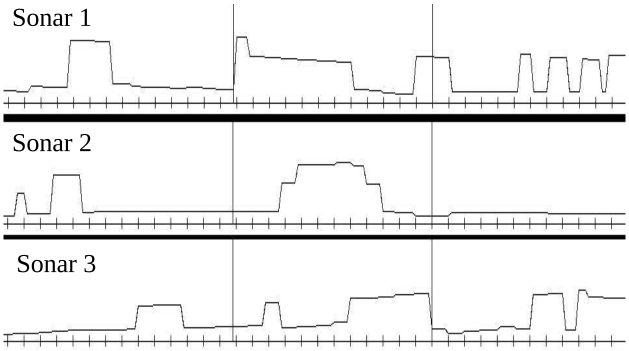

Figure 4: The segmentation corresponding to a T-intersection

A pass in the corridor contains not only one place but all the places seen while running in the corridor. To learn a particular place, we need to segment passes in distinctive places. Moreover, we need to select the pertinent sensors' measures to observe a place. This task is more complex because the sensors' measures are noisy and when there is a place on the right side of the robot, there is another place on the left side of the robot. For these reasons, we choose to segment passes to use for each side, the sensor perpendicular to each wall of the corridor and its two neighbor sensors. These three sensors normally give valid measures. So for each pass, we have two segmentations: one for the right side, and one for the left side.

As all places except the corridor cause a noticeable variation on these three sensors over time, we define the beginning of a place when the first sensor's measure suddenly increases, and the end of a place when the last sensor's measure suddenly decreases. Figure 4 shows an example of the segmentation on the right side with these three sensors of a part of an acquisition corresponding to a T-intersection. The first line segment is the beginning of the T-intersection (sudden increase on the first sensor), and the second line segment is the end of the T-intersection (sudden decrease on the third sensor). The left part of the first line and the right part of the second line are a corridor place. Figure 5 shows the position of the robot at the beginning and at the end of the T-intersection and the measures of the

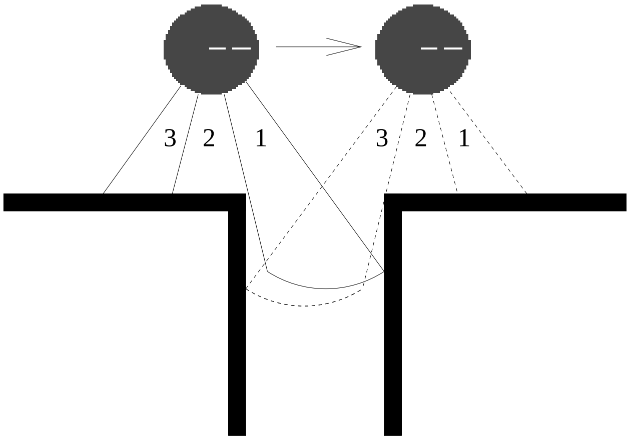

Figure 5: The three sonars used for the segmentation of a T-intersection

three sensors used at these two positions for the segmentation.

4.2 The models' training

As said earlier, we chose three coefficients corresponding to the three sensors' measures. Because the segmentation was made using the variations on these three sensors' measures, we use the first derivative of the three sensors' measures as input of the model. The topology used to train each model is shown in figure 2. Intuitively, we can think that the first state will contain the strong increase of the signals corresponding to the beginning of the place, the second state will contain the stationary part of the signals (where the derivative is nearly equal to zero) and the third state will contain the end of the place where the signal decreases strongly.

The training has been performed twice for each pass, once using the first derivative of the three right sensors to learn places on the right side and once using the first derivative of the three left sensors to learn places on the left side.

Two different kinds of training are performed. The first training uses segmented data and each model is trained independently on these data. The second training uses the former models and estimates them on unsegmented data like in the recognition phase. It means that we build a bigger model (constitued of a sequence of the models of places) according to the sequence of observed places and train the resulting model with the unsegmented data.

4.3 The recognition phase

The goal of the recognition process is to spot the recognized places in the corridor. A model for the corridor is used because the Viterbi algorithm has to map each frame to a model during the recognition. The corridor model connects 2 items like a silence between 2 words in speech recognition.

During this experiment, the robot uses its own reactive algorithm [2] to navigate in the corridor and must decide which places have been encountered during the run. We took 10 new passes (back and forth) and used the five models trained to perform the recognition.

The recognition has been performed twice for each pass, once using the first derivative of the three right sensors to make a recognition on the right side and once the first derivative of the three left sensors to make a recognition on the left side.

A place is recognized if it has been detected by the corresponding model. But different types of errors can occur:

Insertions: the robot has seen 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 another.

4.4 First results and discussion

The results are presented in the confusion matrix 1.

	start corridor	end corridor	T-inter.	open door	Ins.
start corridor	15	0	0	0	8
end corridor	0	19	0	0	17
T-inter.	1	1	16	38	67
open door	0	0	2	77	0
deletions	0	0	0	1	0
Total	16	20	18	116	
% reco.	93	95	89	63	

Table 1: Previous results

An element c_{ij} at row i and column j is the number of times the model j has been recognized when the right answer was the place i.

Start and end of corridor place are well recognized. Theses places have a very particular pattern[1], and are difficult to confuse with another.

[1] We define the pattern of a place as the pattern of the graphic representation of observations of the place over time as in figure 4.

T-intersections are globally well recognized, though sometimes confused with open doors.

Open doors are often confused (1 time of 3) with T-intersections for several reasons:

- The pattern of the signal when the robot passes in front of an open door is very similar to the pattern of the signal when the robot passes in front of a T-intersection. Moreover, since HMM2s do not have a great power of discrimination, they cannot easily make the distinction between two models.

- We have only one T-intersection, but several open doors each different from the others. In addition, we worked in a real environment, where the doors are completly or partially opened. The HMM of the open door is thus a model which contains variable signals, which increase the confusion with T-intersection.

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 the three sensors which are interpreted as places.

Places are globally well recognized (over 85% of recognition, for each place, except open doors), and 75% of global recognition. The major problem is the insertions (54%) of places. Let us note omissions occur with a very low probability (less than 1%).

5 Improvement of the recognition rate

To improve the rate of recognition, we have to resolve two major problems:

- Reduce the rate of confusion between open-doors and T-intersections. For this, we have to find a way to improve the discrimination between these two HMM2s. These two types of places have similar pattern but different width. An open door has a width of 90 centimeters, and a T-intersection has a width of 120 centimeters. So, the most important criteria which allows an open door to be distinguished from a T-intersection is the width, but at the moment it is not really effective. Currently, observations are done each time the robot has moved. As the robot has a variable speed depending on its local environment, the distance travelled between two observations is not constant. So we have to find a way to take into account the width of each place during learning and recognition. Our idea is to transform observations, to simulate that the robot always moves the same distance between two observations. This will help HMM2s to discriminate between T-intersections and open doors using a width criteria.

- Reduce the rate of insertions. At the moment, the learning and recognition of places are performed on each side independently using only three sensors on each side (ie, six sensors in all). This technique has two drawbacks:

  - We lose information about the environment using only 6 sensors (2 sets of 3 sensors). Our mobile robot has 16 sensors. Informations provided by front and rear sensors are not currently taken into account. We have insertions of start or end of corridor in the middle of the corridor, which makes no sense. Front sensors could avoid these insertions.
  - The learning and recognition are performed on each side independently. This does not permit learning and recognition of places in a global way. A more global learning and recognition could be useful for several reasons. For example, the influence of one side's recognition on the other could be taken in account, to eliminate some obvious insertions.

  Our idea is to take into account all the 16 sensors' observations.

The two next subsections present the two major modifications we used to improve the rate of recognition. In the third subsection, we discuss the new results.

5.1 Transformation of observations

To be able to easily distinguish the open-door place and T-intersection place by their width, it's necessary that between two acquisitions the distance travelled by the robot will always be the same. On our mobile robot, it is impossible to program automatic acquisitions as a function of the distance travelled, but it is possible to acquire observations in the same way we did in the first experiment (section 4.1) and modify them before learning phase and recognition phase to simulate that between two acquisitions a constant distance has been travelled. So, we analyzed the distance travelled between two acquisitions supposing that the error of the odometric position estimate was insignificant between two observations. We notice that:

- In a cluttered area, the robot approximatively moves 1 centimeter between two acquisitions.

- In a free area, the robot approximately moves 5 centimeters between two acquisitions.

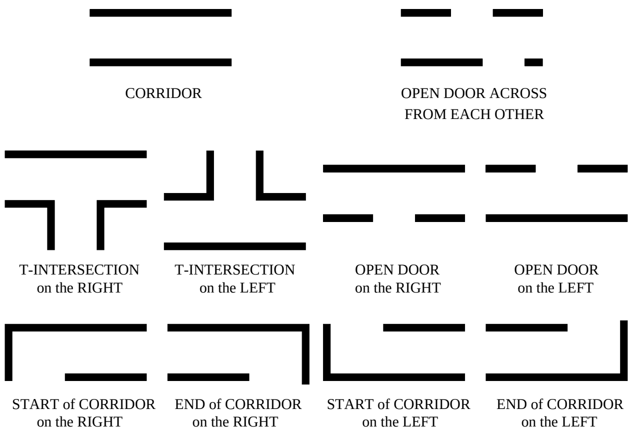
CORRIDOR
OPEN DOOR ACROSS FROM EACH OTHER
T-INTERSECTION on the RIGHT
T-INTERSECTION on the LEFT
OPEN DOOR on the RIGHT
OPEN DOOR on the LEFT
START of CORRIDOR on the RIGHT
END of CORRIDOR on the RIGHT
START of CORRIDOR on the LEFT
END of CORRIDOR on the LEFT

Figure 6: The 10 global places

If we choose to simulate (interpolating from real data) that acquisitions are done at least 5 centimeters apart, we are sure to always have at least one real acquisition of the robot within the interval, and thus have different measures for each simulated acquisition. We decided to simulate that the robot will make an acquisition each 7.5 centimeters. We voluntary chose a bigger distance than 5 centimeters to be sure that the robot will always make at least a real acquisition between two simulated acquisitions. We chose a very simple algorithm to transform acquisitions. The main idea to simulate an acquisition for each sensor, is to make an average for each sensor every 7.5 centimeters travelled based on all real acquisitions done in between. We simulate too that the robot moves exactly 7.5 centimeters between two acquisitions. For this, we compute the distance that the robot travelled since its last simulated acquisition, and we compute the position where the robot was supposed to be after a travell of exactly 7.5 centimeters.

5.2 Using 16 sensors for learning and recognition phase

As said previously, we modified our approach to use all the 16 ultrasonic sensors instead of 3 to build an HMM for each place. Using 16 sensors, we take into account the pattern of each place in a more general way.

But, previously defined places (figure 3) are useless. Now, as we consider the environment in a global way, it makes no sense to consider 5 places that can be seen on each side. We have to define more global places taking into account what can be seen on the right side and on the left side.

To build the new global places, we combine the 5 previous places (figure 3) observable on the right side with the 5 places observable on the left side. As we do not consider the combination of open doors with T-intersections or with start or end of corridor as they nearly never appear in our environnement, we define 10 new places (figure 6):

- A corridor, where there is at the same location a corridor seen by the 3 left sensors and a corridor seen by the 3 right sensors.
- An T-intersection on the right (resp. left), where there is at the same location a corridor seen by the 3 left (resp. right) sensors and an intersection seen by the 3 right (resp. left) sensors.
- An open door on the right (resp. left), where there is at the same location a corridor seen by the 3 left (resp. right) sensors and an open door seen by the 3 right (resp. left) sensors.
- A start of corridor on the right (resp. left), where there is at the same location a corridor seen by the 3 left (resp. right) sensors and a start of corridor seen by the 3 right (resp. left) sensors.
- An end of corridor on the right (resp. left), where there is at the same location a corridor seen by the 3 left (resp. right) sensors and an end of corridor seen by the 3 right (resp. left) sensors.
- Two open doors across from each other, where there is at the same location an open door seen by the 3 left sensors and an open door seen by the 3 right sensors.

5.3 New results and discussion

As we consider the environment in global way, we do not have two (one on each side) segmentations, two training runs and two recognitions for each pass. We will do only one segmentation using the definitions of the 10 new places. Training and recognition are done once with the first derivative of the 16 sensors' measure as inputs of each HMM2.

We label the same previous learning corpus (section 4.1) using the rules defined in section 5.2. We perform the learning phase of the 10 new places using the new segmentation. The recognition phase is performed using the same previous recognition corpus (section 4.3). The goal of the recognition process is to spot the 9 places (end of corridor on the left, end of corridor on the right, intersection on the left, intersection on the right, open door on the left, open door on the right, open doors across from each other, start of corridor on the left and start of corridor on the right) in the corridor. Results are given in the confusion matrix 2.

We notice that the rate of confusion between open doors (on the left or on the right) and T-intersections (on the left or on the right) decreased. Due to the transformation of observations, the number of observations for an open door

	right	right	right	right	left	left	left	left	door	Ins.
	start	end	inter.	door	start	end	inter.	door	door	
right start	7	0	0	0	0	0	0	0	0	0
right end	0	7	0	0	0	1	0	1	0	0
right inter.	0	0	7	0	0	0	0	0	0	0
right door	0	0	0	42	0	0	0	1	1	25
left start	0	0	0	0	8	0	0	0	0	0
left end	0	0	0	0	0	6	0	0	0	0
left inter.	0	0	0	0	0	0	8	0	0	0
left door	1	0	0	4	0	0	1	43	1	34
door door	0	0	1	0	0	0	0	1	2	1
oublis	0	1	0	0	0	0	0	0	0	0
Total	8	8	8	46	8	7	9	46	4	
% reco.	88	88	88	91	100	86	89	93	50	

Table 2: New results

(resp. T-intersection) is nearly always the same. As the number of observations for open doors is different of the number of observations for intersections, the discrimination between these two types of places is easier.

We had previously a significant number of insertions of T-intersections. We have actually no more insertion of T-intersection, but a higher number of insertions of open doors. In fact, a T-intersection has a higher width than an open door, and so insertions of small width corresponding to a reflection on the sensors can only rarely be recognized as T-intersections, but rather as open doors. On the other hand, we had 67 insertions of T-intersections without transformation of the observations, against currently 59 insertions of open door. Now, reflections of small duration are assimilated as noise and can not be recognized as open door or intersections, so the global rate of insertions has decreased.

The use of the 16 sensors during the learning phase eliminated insertions of start of corridor (resp. end of corridor) when the robot is not at one end of the corridor. Observations of sensors situated on the front of the robot are very different when the robot is in the middle of the corridor, or at the end of the corridor. So, the models of start of corridor (resp. end of corridor) could be recognized only when observations of front and rear sensors correspond to the start of a corridor (resp. the end of a corridor), which will rarely occur when the robot is in the middle of the corridor. So, it is nearly impossible to have insertions of start of corridor (resp. end of corridor) in the middle of a corridor.

The rate of recognition of two open doors across from each other is mediocre (50%) for several reasons :

- We did not make the distinction between when the first open door is on the left and the second on the right or the contrary.
- We did not take into account that the two open doors can overlap themselves during a few observations or during many observations.
- As said previously, we worked in a real environment with several open doors each different from the others. Moreover, some doors are completely opened and others partially opened. The model has to take in account this diversity.

The 3 previous reasons make that the HMM2 of this place is a model which contains variable signals, as opposed to one trained to recognize a very well specified signal.

With our two modifications, the global rate of recognition is increased from 75% to 92%. Insertions of places decreases from 54% to 42%. Omissions stay very low probability (less than 1.5%).

6 Conclusion and perspectives

In this paper, we have presented a new method to learn and recognize places in an indoor environment with second order hidden Markov models. One of the main interests of this work is the specification of an automatic learning algorithm of the environment that allows the recognition of typical places. The transformation of observations to simulate constant frequency of observation improve the global rate of recognition by decreasing the confusion between open doors and T-intersections. The use of 16 sensors instead of

2 sets of 3 allows the environment to be learned and recognized in a more general way, which decreases the rate of insertions by eliminating insertions of start or end of corridor (left or right) in the middle of a corridor. This method gives good results, and has a good robustness to noise. The results can be improved by adding more models to decrease the intra-class variability (especially for open doors across from each other) and to take into account contextual information. This method has two drawbacks. Like in [4], a place can only be recognized when it has been completely seen. So, the robot has to go back to turn at a T-intersection, for example. Moreover, it must be noted that the list of places is known only when the run is completed.

References

[1] O. Aycard, F. Charpillet, D. Fohr, and J.F Mari. Place learning and recognition using hidden markov models. In *proceedings of IEEE International Conference on Intelligent Robots and Systems*, 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 International Symposium on Computational Intelligence in Robotics and Automation*, 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] D Kortenkamp and T Weymouth. Topological mapping for mobile robots using a combination of sonar and vision sensing. In *Proceedings of AAAI*, 1994.

[5] 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.

[6] P. Morignot, O. Aycard, and F. Charpillet. A pair of heterogeneous agents in a unique vehicle for object motion. In *proceedings of IEEE International Conference on Tools on Artificial Intelligence*, pages 508–513, 1997.

[7] C Rompais, J.F Barret and G Pradet. Neural classifier for environnement recognition in mobile robotics. In *proceedings of the IEEE/ICANN*, pages 463–468, 1995.

[8] R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *proceedings of IJCAI*, 1995.

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