Learning to automatically detect features for mobile robots using second-order Hidden Markov Models

Olivier Aycard
GRAVIR - IMAG - Universite Joseph Fourier
38 000 Grenoble, France
Olivier.Aycard@imag.fr

Jean-François Mari
LORIA - Universite Nancy2
54 506 Vandœuvre Cedex, France
jfmari@loria.fr

Richard Washington*
NASA Ames Research Center
Moffett Field, CA 94035, USA
richw@email.arc.nasa.gov

keywords: sensor data interpretation, Hidden Markov Models, mobile robots

Abstract

In this paper, we propose a robust method based on Hidden Markov Models to interpret temporal sequences of sensor data from mobile robots to automatically detect features.

Hidden Markov Models have been used for a long time in pattern recognition, especially in speech recognition. Their main advantages over other methods (such as neural networks) are their ability to model noisy temporal signals of variable length.

We show in this paper that this approach is well suited for interpretation of temporal sequences of mobile-robot sensor data. We present two distinct experiments and results: the first one in an indoor environment where a mobile robot learns to detect features like open doors or T-intersections, the second one in an outdoor environment where a different mobile robot has to identify situations like climbing a hill or crossing a rock.

1 Introduction

A mobile robot operating in a dynamic environment is provided with sensors (infrared sensors, ultrasonic sensors, tactile sensors, cameras...) in order to perceive its environment. Unfortunately, the numeric, noisy data furnished by these sensors are not directly useful; they must first be interpreted to provide accurate and usable information about the environment. This interpretation plays a crucial role, since it makes it possible for the robot to detect pertinent features in its environment and to use them for various tasks.

For instance, for a mobile robot, the automatic recognition of features is an important issue for three main reasons: 1) it determines the capability of a mobile robot to locate itself in its environment [Borenstein *et al.*, 1996]. 2) This is also the first step in the construction of cognitive maps [Kuipers, 2000]. 3) Features can be used by a mobile robot as subgoals for a navigation plan [Lazanas and Latombe, 1995].

In semi-autonomous or remote, teleoperated robotics, automatic detection of features is a necessary ability. In the case of limited and delayed communication, such as for planetary rovers, human interaction is restricted, so features detection can only be practically performed through on-board interpretation of the sensor information. For all these reasons, features detection has received considerable attention over the past few years. This problem can be classified with the following criteria:

Natural/artificial The first criterion is the nature of the feature. The features can be artificial, that is, added to the existing environment. [Becker *et al.*, 1995] define a set of artificial features[2] located on the ceiling and use a camera to detect them. Other techniques use natural features, that is, features already existing in the environment. For instance, [Kortenkamp *et al.*, 1992] use ultrasonic sensors to detect natural features like open doors and T-intersections.

Using artificial features makes the process of detection and distinction of features easier, because the features are designed to be simple to detect. But this approach can be time-consuming, because the features have to be designed and to be positioned in the environment. Moreover, using artificial features is impossible in unknown or remote environments.

Analytical/statistical methods Feature detection has been addressed by different approaches such as analytical methods or pattern classification methods. In the analytical approach, the problem is studied as a reasoning process. A knowledge based system uses rules to build a representation of features. For instance, [Kortenkamp *et al.*, 1992] use rules about the variation of the sonar sensors to learn different types of features and adds visual information to distinguish two features of the same type. In contrast, a statistical pattern-classification system attempts to describe the observations coming from the sensors as a random process. The recognition process consists of the association of the signal acquired from sensors with a model of the feature to identify. For instance, [Yamauchi, 1995] uses ultrasonic sensors to build evidence grids [Elfes, 1989]. An evidence grid is a grid corresponding to a discretization of the local environment of the mobile robot. In this grid, Yamauchi's method updates the probability of occupancy of each grid tile with

*NASA contractor with RIACS.

[2]The features are patterns composed of 3x3 squares, and each square is colored in black or white.

several sensor data. To perform the detection, he defines an algorithm to match two evidence grids.

These two approaches are complementary. In the analytical approach, we aim to understand the sensor data and build a representation of these data. But as the sensor data may be noisy, so their interpretation may not be straightforward; moreover, overly simple descriptions of the sensor data (e.g., "current rising, steady, then falling") may not directly correspond to the actual data.

In the second approach, we build models that represent the statistical properties of the data. This approach naturally takes into account the noisy data, but it is generally difficult to understand the correspondence between detected features and the sensor data.

A solution that combines the two approachs could build models corresponding to human's understanding of the sensor data, and adjust the model parameters according to the statistical properties of the data.

Automatic/manual feature definition The set of features to detect could be given manually or discovered automatically [Thrun, 1998]. In the manual approach, the set is defined by humans using the perception they have of the environment. Since high level robotic system are generally based loosely on human perception, the integration of feature detection in such a system is easier than for automatically-discovered features. Moreover, in teleoperated robotics, where humans interact with the robot, the features must correspond to the high level perception of the operator to be useful. These are the main reasons the set is almost always defined by humans. However, properly defining the features so that they can be recognized robustly by a robot remains a difficult problem; this paper proposes a method for this problem. In contrast, when features are discovered automatically, humans must find the correspondence between features perceived by the robot and features they perceive. The difficulty now rests on the shoulders of the humans.

Temporally extended/instantaneous features Some features can only be identified by considering a temporal sequence of sensor information, not simply a snapshot, especially with telemetric sensors. Consider for example the detection of a feature in [Kortenkamp *et al.*, 1992] or the construction of an evidence grid in [Yamauchi, 1995]: these two operations use a temporal sequence of sensor information. In general, instantaneous (i.e., based over a simple snapshot) detection is less robust than temporal detection.

This paper describes an approach that combines an analytical approach for the high-level topology of the environment with a statistical approach to feature detection. The approach is designed to detect temporally extended features that have been manually defined. The feature detection uses Hidden Markov Models (HMMs). HMMs are a particular type of probabilistic automata. The topology of these automata corresponds to a human's understanding of sequences of sensor data characterizing a particular feature in the robot's environment. We use HMMs for pattern recognition. From a set of training data produced by its sensors and collected at a feature that it has to identify — a door, a rock, ... — the robot adjusts the parameters of the corresponding model to take into account the statistical properties of the sequences of sensor data. At recognition time, the robot chooses the model whose probability given the sensor data — the *a posteriori* probability — is maximized. We combine analytical methods to define the topology of the automata with statistical pattern-classification methods to adjust the parameters of the model.

The HMM approach is a flexible method for handling the large variability of complex temporal signals; for example, it is a standard method for speech recognition [Rabiner, 1989]. In contrast to dynamic time warping, where heuristic training methods for estimating templates are used, stochastic modeling allows probabilistic and automatic training for estimating models. The particular approach we use is the second-order HMM (HMM2), which have been used in speech recognition [Mari *et al.*, 1997], often out-performing first-order HMMs.

This paper is organized as follow. We first define the HMM2 and describe the algorithms used for training and recognition. Section 3 is the description of our method for feature detection combining HMM2s with a grammar-based analytical method describing the environment. In section 4, we present an experiment of our method to detect natural features like open doors or T-intersections in an indoor structured environment for an autonomous mobile robot. A second experiment on a semi-autonomous mobile robot in an outdoor environment is described in section 5. Then we report related work in section 6. We give some conclusions and perspectives in section 7.

2 Second-order Hidden Markov Model

In this section, we briefly present the HMM2 and the algorithms used for learning and recognition. A very complete tutorial on HMMs and their applications to speech can be found in [Rabiner, 1989].

2.1 Definition

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 HMM2 λ is specified by:

- a set of states, S;
- a 3 dimensional matrix A over $S \times S \times S$

$$\begin{aligned} a_{ijk} &= Prob(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i) \qquad (1) \\ &= Prob(q_t = s_k / q_{t-1} = s_j, q_{t-2} = s_i, q_{t-3} = ...) \end{aligned}$$

  with $\sum_{k=1}^{N} a_{ijk} = 1$ for all $1 \leq i, j \leq N$, where N is the number of states in the model and q_t is the actual state at time t;
- a mixture of Gaussians associated with each $s_i \in S$:

$$b_i(O_t) = \sum_{m=1}^{M} c_{im} \mathcal{N}(O_t; \mu_{im}, \Sigma_{im}), \qquad (2)$$

where $\sum_{m=1}^{M} c_{im} = 1$ and O_t is the input vector (i.e., the observation[1]) at time t.

The probability of the state sequence $Q = q_1, q_2, ..., q_T$ is defined as

$$Prob(Q) = \pi_{q_1} a_{q_1 q_2} \prod_{t=3}^{T} a_{q_{t-2} q_{t-1} q_t} \tag{3}$$

where π_{q_1} is the probability of state s_{q_1} at time $t = 1$ and $a_{q_1 q_2}$ is the probability of the transition $s_{q_1} \rightarrow s_{q_2}$ at time $t = 2$.

2.2 The Viterbi Algorithm

Recognition of a given sequence of observations is performed by the Viterbi algorithm [Forney, 1973], which determines the most likely state sequence given an observation sequence. The Viterbi algorithm uses dynamic programming to compute the best partial state sequence to time t for all states. The most likely state sequence $q_1, ..., q_T$ is obtained by keeping track of back pointers for each computation of which previous transition leads to the maximal partial path probability.

2.3 The Baum-Welch Algorithm

Model learning is performed using the maximum likelihood estimation criterion that determines the best model parameters according to the corpus of items. It must be noted that this criterion does not try to separate models like a neural network, but only tries to increase the probability that a model generates its corpus independently of what the other models can do. 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).

3 Application to mobile robotics

The method presented in this paper performs feature detection by combining HMM2s with a grammar-based description of the environment. To apply second order Hidden Markov Models to automatically detect features, we must accomplish a number of steps. In this section we review these steps and our approach for treating the issues arising in each of them. In the following sections we expand further on the specifics for each experiment.

The steps necessary to apply HMM2s to detect features are the following:

1. Defining the number of distinct features to identify and their characterization.

   As Hidden Markov Models have the ability to model signals whose properties change with time, we choose a set of sensors (as the observations) that have noticeable variations when the mobile robot is observing a particular feature. The features are chosen for the fact that they are repeatable and human-observable (for the purposes of labeling and validation). So, we define coarse rules to identify each feature, based on the variation of the sensors constituting the observation to identify each feature. These rules are for human use, for segmentation and labeling of the data stream of the training corpus. The set of chosen features is a complete description of what the mobile robot can see during its run. All other unforeseen features are treated as noise.

[1] An observation is defined as the measure of one or several sensors at a given time.

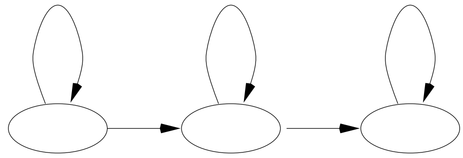

Figure 1: Topology of states used for each model of feature

2. Finding the most appropriate model to represent a specific feature.

   Designing the right model in pattern recognition is known as the model selection problem and is still an open area of research. Based on our experience in speech recognition, we used the well known left-right model (figure 1), which efficiently performs temporal segmentation of the data. Recognition begins in the leftmost state, and each time an event characterizing the feature is recognized it advances to the next state to the right. When the rightmost state has been reached, the recognition of the feature is complete.

   The number of states is generally chosen as a monotone function of the length of the pattern to be identified according to the state duration probabilities.

   The state duration in a HMM2 is governed by two parameters: the probability of entering a state only once, and the probability of visiting a state at least twice, with the latter modeled as a geometric decay. This distribution fits a probability density of durations [Crystal and House, 1988] better than the classical exponential distribution of an HMM1. This property is of great interest in speech recognition when a HMM2 models a phoneme in which a state captures only 1 or 2 frames.

   This choice gives generally high rate of recognition. Sometimes, adding or suppressing one or two states has been experimentally observed to increase the rate of recognition. The number of states is generally chosen to be the same for all the models.

3. Collecting and labeling a corpus of sequence of observations during several runs to perform learning.

   The corpus is used to adjust the parameters of the model to take into account the statistical properties of the sequences of sensor data. Typically, the corpus consists of a set of sequences of features collected during several runs of the mobile robot. So, these runs should be as representative as possible of the set of situations in which features could be detected. The construction of the corpus is time-consuming, but is crucial to effective learning.

   A model is trained with sequences of sensor data corresponding to the particular feature it represents. Since a run is composed of a sequence of features (and not only one feature), we need to segment and label each run. To perform this operation, we use the previously defined coarse rules to identify each feature and extract the relevant sequences of data. Finally, we group the segments of the runs corresponding to the same feature to form a corpus to train the model of that feature.

4. Defining a way to be able to detect all the features seen during a run of the robot.

   For this, the robot's environment is described by means of a grammar that restricts the set of possible sequences of models. Using this grammar, all the HMM2s are merged in a bigger HMM on which the Viterbi algorithm is used. This grammar is a regular expression describing the legal sequences of HMM2s; it is used to know the possible ways of merging the HMM2s and their likelihood. More formally, this grammar represents all possible Markov chains corresponding to the hidden part of the merged models. In these chains, nodes correspond to HMM2s associated with a particular feature. Edges between two HMM2s correspond to a merge between the last state of one HMM2 and the first state of the other HMM2. The probability associated with each edge represents the likehood of the merge.

   Then, the most likely sequence of states, as determined by the Viterbi algorithm, determines the ordered list of features that the robot saw during its run. It must be noted that the list of models is known only when the run is completed. We make the hypothesis that two or more of the features cannot overlap. The use of a grammar has another important advantage. It allows the elimination of some sequences that will never happen in the environment. From a computational point of view, the grammar will avoid some useless calculations.

   The grammar can be given apriori or learned. To learn the grammar, we use the former models and estimate them on unsegmented data like in the recognition phase. Specifically, we merge all the models seen by the robot during a complete run into a larger model corresponding to the sequence of observed items and train the resulting model with the unsegmented data.

5. Evaluating the rate of recognition.

   For this, we define a test corpus composed of several runs. For each run, a human compares the sequence of features composing the run, using knowledge of the environment, with what has been detected by the Viterbi algorithm. A feature is recognized if it is detected by the corresponding model close to its real geometric position. A few types of errors can occur:

   Insertion: the robot has seen a non-existing feature (false positive). This corresponds to an over-segmentation in the recognition process. Insertions are currently considered when the width of the inserted feature is more than 80 centimeters;

   Deletion: the robot has missed the feature (false negative);

   Substitution: the robot has confused the feature with another.

   In the experiments that we have run, the results are summarized first as confusion matrices, where an element c_{ij} is the number of times the model j has been recognized when the right answer was feature i, and second with the global rate of recognition, insertion, substitution and deletion.

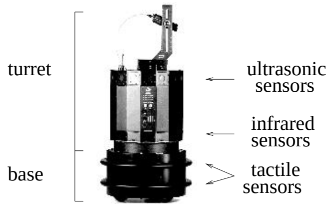

Figure 2: The Nomad200 robot

Figure 3: The Marsokhod rover

In the two following sections, we present two experiments where we used second-order Hidden Markov Models to detect features using sequence of mobile-robot sensor data. In each section, after a brief description of the problem and the mobile robot used, we explain the specific solution to each of the issues introduced in this section.

4 First experiment: Learning and recognition of features in an indoor structured environment

In this first experiment, we used second order Hidden Markov Models to learn and to recognize indoor features such as T-intersections and open doors given sequences of data from ultrasonic sensors of an autonomous mobile robot. These features are generally called *places*. A complete description of this experiment can be found in [Aycard *et al.*, 1997a; 1998b].

4.1 The Nomad200 mobile robot

In this experiment, we used a Nomad200 (figure 2). It is composed of a base and a turret. The base consists of 3 wheels and tactile sensors. The turret is an uniform 16-sided polygon. On each side, there is an infrared and an ultrasonic sensor. The turret can rotate independently of the base. The Nomad200 senses its environment using 16 ultrasonic sensors. The angle between two ultrasonic sensors is 22.5 degrees, and each ultrasonic sensor has a beam width of approximately 23.6 degrees. By examining all 16 sensors, we can obtain a 360 degree panoramic view fairly rapidly. 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.

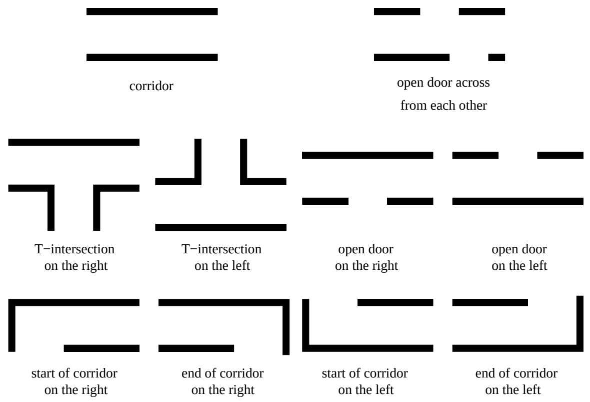

Figure 4: The 10 places to recognize

4.2 Specifics of HMM2 application to indoor place identification

Here we discuss the specific issues arising from applying HMM2s to the problem of indoor place identification, along with our solutions to those issues. The numbering corresponds to the numbering of the steps in section 3.

The set of places

Currently, we model ten distinctive places that are representative of an office environment: a corridor, a T-intersection on the right (resp. left) of the corridor, an open door on the right (resp. left) of the corridor, a "starting" corner on the right (resp. left) when the robot moves away from the corner, an "ending" corner on the right (resp. left) side of the corridor when the robot arrives at this corner, two open doors across from each other (figure 4). 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, are treated as noise.

To characterize each feature, we define coarse rules observing variations on sensors on one side of the robot. These rules will be used for segmentation and labeling the corpus for training and evaluation.

The model to represent each place

In the formalism described in section 2, each place to be recognized is modeled by an HMM2 whose topology is depicted in figure 1.

Corpus collecting and labeling

We built a corpus to train a model for each of the 10 places. For this, our mobile robot made 50 passes (back and forth) in a very long corridor (approximately 30 meters). This corridor contains two corners (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 simple navigation algorithm [Aycard *et al.*, 1997b] to stay in the middle of the corridor in a direction parallel to the two walls constituting the corridor. While running, the robot stored all of its ultrasonic sensor measures. The acquisitions were done in real conditions with people wandering in the lab, doors completely or partially opened and static obstacles like shelves.

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 must manually segment and label passes in distinctive places. The goal of the segmentation and the labeling is to identify the sequence of places the robot saw during a given pass. To perform this task, we use the rules defined to characterize a place. We segment and label each run.

Finally, we group the segments from each pass corresponding to the same place.

The recognition phase

The goal of the recognition process is to identify the 9 places in the corridor. We use a tenth model for the corridor because the Viterbi algorithm needs to map each frame to a model during recognition. The corridor model connects 2 items much like a silence between 2 words in speech recognition. During this experiment, the robot uses its own reactive algorithm to navigate in the corridor and must decide which places have been encountered during the run. We took 40 acquisitions and used the ten models trained to perform the recognition. The recognition is independently processed on each side.

4.3 Results and discussion

Results are given in table 1 and 2.

We notice that the rate of recognition are very high, and the rate of confusion are very low. This is due to the fact that each place has a very particular pattern, and so it is very difficult to confuse it with an other. In fact, HMM2 used hidden characteristics (i.e, characteristics not explicitly given during the segmentation and the labelization of places) to perform discrimination between places. In particular, a place is characterized by variations on sensors on one side of the robot, but too with variations on sensors located on the rear or the front of the robot. Observations of sensors situated on the front of the robot are very different when the robot is in the middle of the corridor than 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 the start of a corridor (resp. end of corridor) in the middle of the corridor.

HMM2 have been able to learn this type of hidden characteristics and to use them to perform discrimination during recognition.

But, we see that T-intersection and open doors have very similar characteristics using sensor information, and there is nearly no confusion between these two places. An other characteristic has been learned by the HMM2 to perform the discrimination between these two places. The width of open doors is different from the width of intersections, the discrimination between these two types of places is improved because of the duration modeling capabilities of the HMM2, as presented above and as shown by [Mari *et al.*, 1997].

The rate of recognition of two open doors across from each other is mediocre (50%). There exists a great variety of doors that can overlap and we only define one model that represents all these situations. So this model is a very general model

	right start	right end	right inter.	right door	left start	left end	left inter.	left door	door door	Ins.
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
deletions	0	1	0	0	0	0	0	0	0	0
Total	8	8	8	46	8	7	9	46	4	60
% reco.	88	88	88	91	100	86	89	93	50	

Table 1: Confusion matrix of places

	number	%
Seen	144	100
Recognized	130	90
Substituted	11	9
Deleted	2	1
Inserted	60	42

Table 2: Global rate of recognition

of two doors across from each other. Defining more specific models of this place would lead to increase the associate rate of recognition.

The major problem is the high rate of insertion. 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, and in the middle of the corridor. These conditions cause reflections on some sensors which are interpreted as places. A level incorporating knowledge about the environment should fix this problem.

Finally, the global rate of recognition is 92%. Insertions of places are 42%. Deletions are at a very low probability level (less than 1.5%).

5 Second experiment: Situation identification for planetary rovers: Learning and Recognition

In a second experiment, we want to detect particular features (which we call *situations*) when an outdoor teleoperated robot is exploring an unknown environment. A complete description of this experiment can be found in [Aycard and Washington, 2000].

This experiment has main differences with the previous one:

1. the robot is an outdoor robot;
2. the sensors used as the observation are of a different type than in the indoor experiment;

5.1 Marsokhod rover

The rover used in this experiment is a Marsokhod rover (see figure 3), a medium-sized planetary rover originally developed for the Russian Mars exploration program; in the NASA Marsokhod, the instruments and electronics have been changed from the original. The rover has six wheels, independently driven,[2] with three chassis segments that articulate independently.

The Marsokhod is controlled either through sequences or direct tele-operation. In either case the rover is sent discrete commands that describe motion in terms of translation and rotation rate and total time/distance. The Marsokhod is instrumented with sensors that measure body, arm, and pan/tilt geometry, wheel odometry and currents, and battery currents. The sensors that are used in this paper are roll (angle from vertical in direction perpendicular to travel), pitch (angle from vertical in direction of travel), and motor currents in each of the 6 wheels.

The experiments in this paper were performed in an outdoor “sandbox,” which is a gravel and sand area about 20m x 20m, with assorted rocks and some topography. This space is used to perform small-scale tests in a reasonable approximation of a planetary (Martian) environment. We distinguish between the small (less than approx. 15cm high) and large rocks (greater than approx. 15cm high). We also distinguish between the one large hill (approx. 1m high) and the three small hills (0.3-0.5m high).

5.2 Specifics of HMM2 application to outdoor situation identification

Here we discuss the specific issues arising from applying HMM2s to the problem of outdoor situation identification, along with our solutions to those issues [Aycard and Washington, 2000]. The numbering corresponds to the numbering of the steps in section 3.

The set of situations

Currently, we model six distinct situations that are representative of a typical outdoor exploration environment: when the robot is climbing a small rock on its left (resp. right) side, a big rock on its left side,[3] a small (resp. big) hill, and a default situation of level ground.

This set of items is considered to be a complete description of what the mobile robot can see during its runs. All other

[2] For the experiments, the right rear wheel had a broken gear, so it rolled passively.

[3] The situation of a big rock on the right side was not considered because of the non-functional right-side wheel.

	BL	SL	SR	BH	SH	Ins
BL	19	3	1	-	-	9
SL	3	25	-	-	-	12
SR	1	2	31	-	1	26
BH	1	-	-	20	2	15
SH	-	-	-	1	23	28
Del	1	1	-	-	-	-
Total	25	31	32	21	26	90
Reco	76%	81%	97%	95%	88%	

Table 3: Confusion matrix of situations

unforeseen situations, like flat rocks or holes, are treated as noise.

From the sensors described in section 5.1, we chose eight sensors: roll, pitch, and the six wheel currents, to define coarse rules to identify each situation (used by humans for segmentation and labeling the corpus for training and evaluation).

The model to represent each situation

In the formalism described in section 2, each situation to be recognized is modeled by a HMM2. This topology is well suited for the type of recognition we want to perform. In this experiment, each model has five states to model the successive events characterizing a particular situation. This choice has been experimentally shown to give the best rate of recognition.

Corpus collecting and labeling

We built six corpora to train a model for each situation. For this, our mobile robot made approximately fifty runs in the sandbox. For each run, the robot received one discrete translation command ranging from three meters to twenty meters. Rotation motions are not part of the corpus. Each run contains different situations, but each run is unique (i.e., the area traversed and the sequence of situations during the run is different each time). A run contains not only one situation but all the situations seen while running. For each run, we noted the situations seen during the run, for later segmentation and labeling purposes.

The recognition phase

The goal of recognition is to identify the five situations (small rock on the left or right; big rock on the left; small or big hill) while the robot moves in the sandbox. The default situation model connects two items much like silence between two words in speech recognition.

During the recognition phase, the robot was operated as for corpus collecting. We took approximately 40 acquisitions and used the six trained models to perform the recognition.

Results and discussion

In the confusion matrix(table 3), the acronyms used are: BL = big rock on the left, SL = small rock on the left, SR = small rock on the right, BH = big hill, and SH = small hill.

As each situation can be easily distinguished from the others, the global rate of recognition is excellent (87%) (see tables 3, 4). Small (resp. big) rocks on the left are sometimes confused with big (resp. small) rocks on the left; the signal provided by the sensors does not contain the information necessary to discriminate these two models. In fact, the variations on the sensors are nearly the same. The only criterion which distinguishes these two models is the amplitude of the variation on the three left wheels, and visibly it is not sufficient. The small rocks on the right are perfectly recognized. This situation has a very distinctive pattern, and only with difficulty can it be confused with another. The fact that we could not learn and recognize a situation where the robot is crossing a big rock on its right avoids any confusion.

	#	%
Seen	135	100
Recognized	118	87
Substituted	15	11
Omitted	2	2
Inserted	90	67

Table 4: Global rate of recognition

The major problem is the high rate of insertion. This rate is due to the noise of the sensors being recognized as a situation. This is especially the case for situations characterized only by small variations on a part (or all) of the set of sensors, in particular the crossing of a small hill.

6 Related work

Markov models have been widely used in mobile robotics. [Thrun, 2001] reviews techniques based on Markov models for three main problems in mobile robotics: localization, map building and control. In these techniques, a Markov model represents the environment, and a specific algorithm is used to solve the problem. Our approach is different in a number of ways. We address a different problem: the interpretation of temporal sequences of mobile-robot sensor data to automatically detect features. Moreover, we use very little *a priori* knowledge: in particular, the topology of the model reflecting the human's understanding about sequences of sensor data characterizing a particular feature. All the other parameters of the model are estimated by learning. On the contrary, the techniques presented in [Thrun, 2001] need some *preliminary knowledge*: a map of the environment, a sensor model and an actuator model. Usually, there is no learning component in these techniques.

The most well-known work including a learning component is by [Koenig and Simmons, 1996]. They start with an *a priori* topological map that is translated into a Markov model before any navigation takes place. An extension of the Baum-Welch algorithm reestimates the Markov model representing the environment, the sensor and actuator models. There are a number of differences with this work:

- They use a Markov model to model the environment, whereas we use a Markov model to model the sequence of events composing a particular feature;
- They need some *a priori* knowledge: a topological map of the environment, and sensor and actuator models;

- They make hypotheses on the value of some parameters to reduce the number of parameters to estimate; we do not make any such hypothesis;
- The observations they use are discrete, symbolic and unidimensional. There are obtained by an abstraction (based on some hypothesis) of the raw data of several sensors. Discrete symbolic and unidimensional observations are the result of our method. They are obtained by interpretation of a sequence of raw data from several sensor without any prior hypothesis.

Our work can be seen as a preliminary step for all of the work presented in [Thrun, 2001]. We have previously built a sensor model based on the recognition rates reported in this article; the model allowed robust localization in dynamic environments [Aycard *et al.*, 1998a].

7 Conclusion and future directions

In this paper, we have presented a new method to learn to automatically detect features for mobile robots using second-order Hidden Markov Models. This method gives very good results, and has a good robustness to noise, verifying that HMM2s are well suited for this task. We showed that the process of recognition is robust to dynamic environment. Features are detected even if they are quite different from learned features: for instance, an open door is recognized even if it is completely or partially opened. Moreover, features are detected even if they are seen from a different point of view. For instance, in contrast to Kortenkamp et al [Kortenkamp *et al.*, 1992], features are detected even if the robot is not at a given distance from a wall and doesn't move in a direction perfectly parallel to the two walls constituting the corridor. Finally, our approach has been successfully tested in an outdoor environment.

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. Another criterion that could improve the results is to choose a different number of states for each feature.

Moreover, the method takes advantage of analytical methods and pattern classification methods. First, we analyze the sensor data and define a model to represent the patterns in the data. Secondly, the learning algorithm automatically adjusts the parameters of the model using a learning corpus. Moreover, the learning algorithm was able to extract more complex characteristics of a feature than simple variations of sensor data between two consecutive moments. For instance:

- The length of a sequence[4] of observations was taken into account in the first experiment to detect the difference between a T-intersection and an open door;
- In the first experiment, the gradual decrease (resp. increase) of the value of sensors located in front (resp. in the rear) of the robot during time has been used to characterize a start (resp. an end) of corridor;
- The algorithm can find correlation between data from sensors of different types to characterize a feature. For example, the correlation of the roll, pitch and wheel current sensors is used to characterize a situation in the second experiment.

However, our method has two drawbacks:

- As in Kortenkamp et al [Kortenkamp *et al.*, 1992], a feature can only be recognized when it has been completely visited. For example, the robot would have to go back to turn at a T-intersection after it had recognized it.
- Moreover, using the current technique, the list of places is known only when the run has been completed. To detect features online during navigation, we can use a variant of the Viterbi algorithm called Viterbi-block [Kriouile *et al.*, 1990]. This algorithm is based on a local optimum comparison of the different probabilities computed by the Viterbi algorithm during time-warping of a shift-window of fixed length in the signal and the different HMMs. This algorithm can detect features a few meters after they have been seen. We have used this algorithm to perform localization in dynamic environment [Aycard *et al.*, 1998a].

[4]the number of observations composing the sequence

References

[Aycard and Washington, 2000] O. Aycard and R. Washington. State identification for planetary rovers: Learning and recognition. In *Proceedings of the IEEE/ICRA*, 2000.

[Aycard *et al.*, 1997a] 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.

[Aycard *et al.*, 1997b] 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.

[Aycard *et al.*, 1998a] O. Aycard, P. Laroche, and F. Charpillet. Mobile robot localization in dynamic environment using place recognition. In *Proceedings of IEEE/ICRA*, pages 3135–3140, 1998.

[Aycard *et al.*, 1998b] O. Aycard, J.-F. Mari, and F. Charpillet. Second order hidden markov models for places recognition: New results. In *proceedings of IEEE/ICTAI*, 1998.

[Becker *et al.*, 1995] C Becker, J Salas, K Tokusei, and J-C Latombe. Reliable navigation using landmarks. In *proceedings of IEEE/ICRA*, 1995.

[Borenstein *et al.*, 1996] J. Borenstein, B. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.

[Crystal and House, 1988] T.H. Crystal and A.S. House. Segmental Durations in Connected Speech Signals: Current Results. *Journal of Acoustical Society of America*, 83(4):1553 – 1573, April 1988.

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

[Forney, 1973] G.D. Forney. The Viterbi Algorithm. *IEEE Transactions*, 61:268–278, March 1973.

[Koenig and Simmons, 1996] S. Koenig and R.G. Simmons. Unsupervised learning of probabilistic models for robot navigation. In *proceedings of IEEE/ICRA*, pages 2301–2308, 1996.

[Kortenkamp *et al.*, 1992] D Kortenkamp, L Douglas Baker, and T Weymouth. Using gateways to build a route map. In *proceedings of IEEE/IROS*, 1992.

[Kriouile *et al.*, 1990] A. Kriouile, J.-F. Mari, and J.-P. Haton. Some Improvements in Speech Recognition Algorithms based on HMM. In *Proceedings of IEEE/ICASSP*, pages 545 – 548, Albuquerque, April 1990.

[Kuipers, 2000] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 2000.

[Lazanas and Latombe, 1995] A Lazanas and J-C Latombe. Landmark-based robot navigation. *Algorithmica*, 1995.

[Mari *et al.*, 1997] 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.

[Rabiner, 1989] 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.

[Thrun, 1998] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 1998.

[Thrun, 2001] S. Thrun. Probabilistic algorithms in robotics. *AI Magazine*, 2001.

[Yamauchi, 1995] B Yamauchi. *Exploration and spatial learning in dynamic environnements*. PhD thesis, Case Western Reserve University, 1995.