Environment Mapping as a Topological Representation

Alberto Vale, M. Isabel Ribeiro

Instituto Superior Técnico / Institute for Systems and Robotics
Av. Rovisco Pais 1, 1049 - 001 Lisboa, Portugal
{vale,mir}@isr.ist.utl.pt

Abstract

Environment representation and mapping play a key role in the navigation of mobile robots, in particular in outdoors scenarios that are becoming increasingly important fostered by a large number of challenging applications. The main goal of this paper is to use sensor fusion and feature extraction to provide the minimal and sufficient topological representation of an environment to support the navigation of a mobile robot in unstructured environments. Experimental results illustrate the performance of the representation procedure.

1 Introduction

The choice of the environment representation is essential in the design of any navigation algorithm for mobile robots, mainly in hostile environments. A topological map is a representation of an environment with no metric information. Topological maps provide useful abstractions of an environment, showing natural or artificial features that characterize particular locations or places.

Given the complexity of unstructured environments and its inherent uncertainty, the probabilistic approach to robotics scales better to real-world applications than the deterministic algorithms. The robots interact with the world through the actuators and get feedback from the sensors. To perform a mission, the robots need an abstraction of the world to localize themselves and to reach a given target according to the allocated mission. As referred in [4] the probabilistic paradigm pays tribute to the inherent uncertainty in robot perception and control relying on the selected representation of uncertainty when determining what to do.

The main goal of mapping is to provide a suffice accuracy map for autonomous navigation which can be achieved by a probabilistic approach on the world representation.

In [5], the authors describe an high accuracy outdoor navigation system based on standard dead reckoning sensors, laser range and bearing information. The main tool is the Extended Kalman Filters (EKF). The application provides the pose (position and orientation) of the vehicle, and the position of the trunks (natural landmarks). Due to the landmark selection, the algorithm is extremely dependent on the matching performance. Moreover, the algorithm is computational expensive being quadratic in the number of landmarks.

In [2] this computational burden is overcome by the implementation of a Fast SLAM algorithm. While Kalman filter-based algorithms are quadratic in the number of landmarks to incorporate each sensor observation, the FastSLAM algorithm estimates recursively the full posterior distribution over robot pose and landmarks, this estimation being logarithmically scaled with the number of landmarks. The algorithm is based on the Rao-Blackwellized representation of the integrating particle filter and on Kalman filter representations. The algorithm shows limitations, in particular the number of landmarks increases when compared to the EKF. These limitations are also drawbacks for outdoors applications embodying natural features.

In the challenging applications of mobile robots in large outdoors scenarios, where complexity and uncertainty are present, it is necessary to overcome the drawbacks imposed by the large number of landmarks and/or features available. All the available information has to be condensed and recorded into a map able to provide an abstraction of the environment for autonomous navigation.

The novelty of this paper is a procedure to map outdoors environments in a topological representation, using a probabilistic approach. The topological map collects environment features and properties provided by the sensor fusion procedure. The algorithm is intended to adapt to the available sensors, this meaning that adding or removing different types of sensors enlarges or reduces the number of properties available to the algorithm. Based on the amount of available information, the algorithm builds the environment representation for mobile robot navigation as a set of states, where each state is represented by a set of Gaussians. These pdfs translate numerically the features that represent the environment by their means and covariance matrices. The main tool used to accomplish the mapping procedure is a modified version of the Expectation-Maximization algorithm, where the standard version is referred in [6] and [8].

The paper is organized as follows. Section 2 presents an overview of topological world representations and introduces the notation of the map's parameters that will be used to support map building. With the adopted world representation, the map building is discussed in Section 3. Experimental results obtained in a real indoors environment are presented in Section 4. Section 5 concludes the paper and presents directions for further developments, in particular the extension of the experiments for outdoors environments.

2 Topological Maps

A topological map is a representation of an environment with no metric information available, showing physical (natural or artificial) features that characterize particular locations or places. The map expresses a functional relationship among relevant features with a resolution that is proportional to the complexity of the environment's representation. The structure of a topological map relies on a set of nodes that, in this work, represent places in an outdoor environment. Each node is defined as a state of the map and is characterized by a set of relevant features to support the state identification and to avoid mismatching.

The notation used to define a topological map is the following:

- s_i is the state i of the map,
- $S = \{s_1, \ldots, s_N\}$ is a set of N states of the map,
- v_j is the j^{th} feature or attribute, $j = 1, \ldots, M$, that may classify any state s_i,
- $v_j \in V_j$, i.e., the feature j takes values in the set V_j,
- $s_i(v_j)$ is the value of the attribute v_j at state s_i; $s_i(v_j) = \emptyset$ means that the attribute v_j is unknown at state s_i.

The connection between states is described in [3].

3 Map Building

In the present work, a topological map is built to support the navigation of a mobile robot. To perform a symbolic representation of the environment the robot perceives it with its on-board sensors and the acquired data is processed aiming at extracting the most relevant features of the environment. The built map provides the essential information for the navigation process. The robot perception is condensed in observations, o_t, that represent the information obtained from the processing of the raw data acquired at each time instant t. For a time interval, T, the result is a sequence of observations, O_T. An observation is a vector where each component is related with a different feature, v_j. For instance, a feature defined as "Colour" might have the values "red", "green" or "blue". This characterization is translated in numerical values, as the colours can be written in RGB format.

The notation used to characterize the observations is the following:

- $o_t = [o_t(v_1)o_t(v_2)\ldots o_t(v_M)]$ is a M-dimensional observation vector referred to time instant t,
- $o_t(v_j)$ is the value of the attribute v_j extracted from the observation o_t,
- $o_t(v_j) = \emptyset$ states that the observation of the attribute v_j, at time instant t, is not achieved,
- $O_t = \{o_{t_0}, o_{t_1}, \ldots, o_t\}$ is an observation sequence from t_0 to t.

The different components of the observations reflect that the robot is able to perceive a diversity of attributes of the environment. These different levels of perception have to be recorded in each state of the map. According to the uncertainty of the measurements and sensors, each state s_i is represented by a Gaussian pdf, characterized by a mean vector μ_i and a covariance matrix, R_i,

$$s_i \sim \mathcal{N}(\mu_i, R_i).$$

As previously referred, a map is composed by a set of states s_i and, consequently, is represented by a set of Gaussian pdfs, each one represented by its mean and covariance matrix, as shown in (1):

$$S = \{s_1, \ldots, s_N\} \sim \{\mathcal{N}(\mu_1, R_1), \ldots, \mathcal{N}(\mu_N, R_N)\} \quad (1)$$

The Figure 1 shows an example of a map, where each state s_i is symbolically represented by the plot of the values of μ_i. The representation in Figure 1 does not provide any information on the state spatial distribution. The bindings represented by the grey arrowed lines express the probability transitions between states. In the proposed frame-work, these transitions result from the Hidden Markov Model approach described in [3].

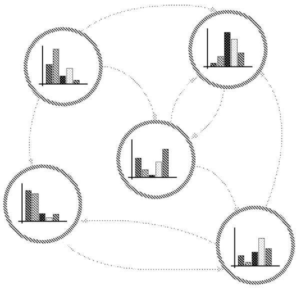

Figure 1: An example of a topological representation, where each state is characterized by 5 features.

With this map characterization, the mapping procedure estimates the mean vectors and the covariance matrices in (1) that maximize the probability of all observations given the environment model, i.e., that maximize the likelihood function (2),

$$\begin{aligned} p(O \mid S) &= p(o_{t_0}, o_{t_0+1}, \ldots o_t \mid S) \\ &= p(o_t \mid S) \cdot p(o_{t_0}, \ldots, o_{t-1} \mid S) \\ &= p(o_t \mid S) \cdot p(o_{t-1} \mid S) \cdot \ldots \cdot p(o_{t_0} \mid S) \\ &= \prod_{i=1}^{t} p(o_i \mid S) \qquad (2) \end{aligned}$$

or, equivalently, its logarithmic representation,

$$L(S) = \log(p(O \mid S)) = \sum_{i=1}^{t} \log(p(o_i \mid S)). \qquad (3)$$

Given that S is a set of states, any observation is a measurement of the state s_k with a probability c_k, for $k = 1, \ldots, N$. Therefore, the probability $p(o_i \mid S)$ in (3) can be written as a combination of $p(o_i \mid s_k)$, $k = 1, \ldots, N$ yielding

$$L(S) = \sum_{i=1}^{t} \log\left(\sum_{k=1}^{N} c_k \cdot p(o_i \mid s_k)\right). \qquad (4)$$

3.1 Expectation-Maximization (EM) Algorithm

The maximization of the likelihood function in (4) is a hard problem to solve. A way to overcome the computational burden associated to it is by changing the function $L(S)$ by the expectation of the likelihood given a previous estimation of the model, S^{old}, i.e.,

$$F(s) = E\left\{\log(p(O \mid S)) \mid S^{old}\right\} \qquad (5)$$

this corresponding to the use of the Estimation and Maximization algorithm, as referred in [4], [6] and [8].

Given a previous estimate of the model, it is assumed that it is possible to evaluate the probability w_{ij} that the observation o_i belongs to the state s_j. Accordingly, $\log(p(o_i \mid S))$ in (3) can be written as $\log(c_j \cdot p(o_i \mid s_j))$ with uncertainty w_{ij}. Therefore, the likelihood function (5) becomes

$$\begin{aligned} F(S) &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \log(c_j \cdot p(o_i \mid s_j)) \\ &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \log(c_j \cdot \mathcal{N}(o_i, \mu_j, R_j)) \\ &= \sum_{j=1}^{N}\sum_{i=1}^{t} w_{ij} \left[\log(c_j) - \log\left((2\pi)^{\frac{M}{2}}\sqrt{|R_j|}\right)\right. \\ &\qquad \left. -\frac{1}{2}(o_i - \mu_j)^T R_j^{-1}(o_i - \mu_j)\right] \qquad (6) \end{aligned}$$

where $\mathcal{N}(o_i, \mu_j, R_j)$ is a Gaussian pdf.

Expectation Step

The parameter w_{ij} in (6) is evaluated using the values of the previous estimation as shown in (7), where η is a pdf normalization factor,

$$w_{ij} = \eta \cdot c_j^{old} \cdot \mathcal{N}(o_i, \mu_j^{old}, R_j^{old}). \qquad (7)$$

This corresponds to the Expectation Step of EM algorithm, as described in [9].

Maximization Step

The next step of the EM algorithm is the maximization of (6). According to the available tools for maximization, the Lagrangean of the likelihood function is defined as,

$$Q(S) = F(S) + \lambda(\sum_{j=1}^{N} c_j - 1).$$

The equations:

$$\frac{\partial}{\partial \mu_j} Q = 0, \frac{\partial}{\partial R_j} Q = 0, \frac{\partial}{\partial c_j} Q = 0$$

provide the values for μ_j, R_j and c_j parameters yielding

$$\mu_j = \frac{1}{\sum_{i=1}^{t} w_{ij}} \sum_{i=1}^{t} w_{ij} o_i, \qquad (8)$$

$$R_j = \frac{1}{\sum_{i=1}^{t} w_{ij}} \sum_{i=1}^{t} w_{ij}(o_i - \mu_j)(o_i - \mu_j)^T, \quad (9)$$

$$c_j = \frac{1}{\lambda} \sum_{i=1}^{t} w_{ij}. \quad (10)$$

The constraint $\sum_{j=1}^{N} c_j = 1$ (total probability for all possible states of the model) leads to

$$\sum_{j=1}^{N} \left(\frac{1}{\lambda} \sum_{i=1}^{t} w_{ij} \right) = 1,$$

which corresponds to $\lambda = t$ and, consequently, (10) becomes

$$c_j = \frac{1}{t} \sum_{i=1}^{t} w_{ij}.$$

After the maximization process in one step, the algorithm returns to the expectation step with the values of (8),(9) and (10). This procedure repeats until all the w_{ij} parameters have stabilized.

Initialization and Stopping Criteria

The EM algorithm is initialized with

$$c_j = \frac{1}{N}, \quad \mu_j = random, \quad R_j = I, \qquad \forall j = 1, \ldots, N$$

corresponding to the values used in the first expectation step. In particular, the μ initialization is a N - dimensional vector, assuming values in an interval bounded by the physical constraints of the sensors.

The EM algorithm, as an iterative procedure, also requires a stopping criteria. This algorithm belongs to the class of unsupervised learning algorithms and, consequently, the only variable that expresses the representation accuracy is the parameter w_{ij}. As mentioned above, w_{ij} is the probability that the observation o_i belongs to the state s_j and, according to the maximization step of EM, w_{ij} stabilizes after some iterations. The parameter w_{ij} is considered stabilized when the difference between a couple of successive iterations is less or equal to a threshold, for all the observations o_i. In addition, to evaluate the stabilization of w_{ij}, it is necessary to record the value of w_{ij} in each iteration k, $w_{ij}(k)$. The differences between a sequence of L iterations is evaluated for a single state s_j, as:

$$\delta_j = \sum_{i=1}^{t} \sum_{l=1}^{L} |w_{ij}(k) - w_{ij}(k-l)|. \quad (11)$$

To evaluate the overall stabilization of the algorithm it is necessary to define the stabilization for all the states. When δ in (12)

$$\delta = \sum_{j=1}^{N} \delta_j \quad (12)$$

is lower than a given threshold Δ_δ, it is considered that the EM algorithm has stabilized.

After stabilization it is necessary to analyze the quality of the representation obtained by the algorithm. A good representation, s_j, for the observation o_i, corresponds to a high value of w_{ij}. Moreover, a good representation for all the observations o_i, reducing the number of outliers, requires high values w_{ij} for all the states s_j. However, a good representation may not occur, this resulting from the existence of spurious states or even from a small number of states for the representation.

3.2 Dynamic EM

According to the previous subsection the initial number of states, N, which is constant during the EM algorithm, does not necessarily guarantees a good representation of the environment. Even for a good representation at a given time instant, a possible update of the number of states might be required as the robot is always acquiring new measurements. Consequently, it is strictly necessary to re-evaluate the number of states after the stabilization of the EM algorithm as represented in Figure 2.

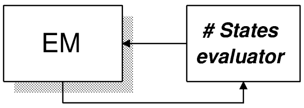

Figure 2: Number of states analyzer

The evaluation of the number of states is explained in the sequel. Starting with an initial estimate of the number of states, the EM algorithm is applied iteratively. As represented in Figure 3, when $\delta < \Delta_\delta$ the EM algorithm converges to an environment representation. The next step assesses the quality of each state to find possible superfluous states. To improve this analysis, it is necessary to evaluate the amount of observations represented by each state, s_j. A natural way to formalize this analysis is to consider H_j, the entropy of state s_j, as referred in [1],

$$H_j\left(w_{ij}\right) = \sum_{i=1}^{t} w_{ij} \log(w_{ij}). \quad (13)$$

If H_j is less or equal than a given threshold, H_{min}, the j-th state is removed and the number of states is decreased by one. If $H_j > H_{min}$ for all $j = 1, \ldots, N$, the set of observations could require a new state to improve the representation. Accordingly, the number of states is increased by one and the new state is initialized by a mean vector μ and a covariance matrix R as described in Section 3.3. The EM algorithm runs again and when the stabilization is reached for this new number of states, a new evaluation takes place, along the steps in Figure 3, to check if the new state improved the representation.

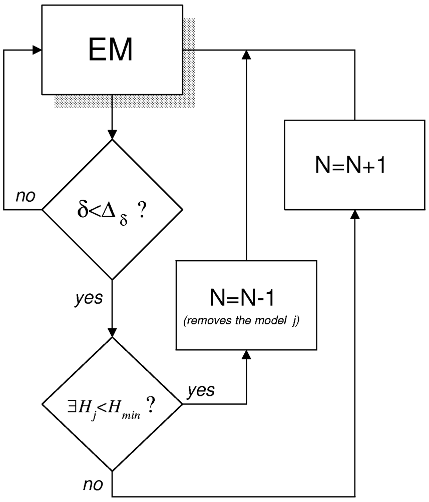

Figure 3: Increasing and decreasing the number of states

The parameter H_j quantifies the accuracy of the representation. Low values for H_{min} lead to a large number of states, that could yield states with similar parameters. Additionally, the parameter Δ_δ defines the level of oscillations during the algorithm.

As represented in Figure 3, the algorithm is always trying to adjust the number of states to update the model. Therefore it is prepared for changes in the environment, adding new states and/or removing useless states. Nevertheless, without any changes in the environment, the algorithm converges and the number of states oscillates around a given value. Whenever these oscillations are above a threshold during a given time interval the algorithm stops.

3.3 Initializations in the Dynamic EM

In the previous subsection, when the number of states increases, the new state has to be initialized. According to the dimension of the observations acquired since the time instant t_0 and the iterative nature of the EM algorithm, it is strictly necessary to optimize the initialization procedure to reduce time consumption. The initialization step establishes the values μ and R for a new state. There are two possible ways to accomplish this issue:

- **Random combinations** of the current states: When more than one state is available, it is possible to generate the mean of the new state as a combination of the means of two or more of the current states. The *selected* states $s_{sel_1}, s_{sel_2}, \ldots, s_{sel_{N_sel}}$ ($N_{sel} = 2, 3, \ldots N - 1$) are randomly selected with uniform distribution.

- **Random values:** In a real application, the values of each component of μ are bounded by an interval imposed by the physical constraints of the sensors. Therefore, the vector μ might be generated by two different pdfs:
  - Uniform inside the interval described above;
  - $1 - P(O \mid S)$, which is equivalent to $1 - \sum_i \sum_j w_{ij}$. This corresponds to include the observations which are not yet duly represented by the current states.

In both cases the covariance matrix is initialized as an identity matrix.

4 Experimental Results

The dynamic EM was tested in a real-world environment using the mobile robot ATRV-Jr in Figure 4. The robot is equipped with a laser range scanner, a ring of ultra-sonic sensors, an inertial measurement unit, GPS, vision and odometric sensors. The experimental results were obtained only with the range sensors (Sick Laser LMS and built-in sonars). From the acquired data four features were extracted: the mean and the variance of the free-area measured by the laser and by the sonars.

The algorithm was tested by tele-operating the robot from a corridor to a room, containing chairs, tables and people moving and recording the sensor data. The left side of Figure 5 displays the measurements acquired by the laser and the sonars. Setting low accuracy to the mapping algorithm the result is a topological map with three states, corresponding to the room, its entrance and the corridor. The states were distinguished based on the differences detected on the free-area and variance: the room is defined as a state

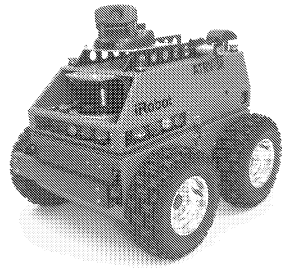

Figure 4: Mobile Robot ATRV-Jr with a Sick Laser LMS and a Pan and Tilt camera Sony EVI-D31

with a large free-area measured by the laser and sonar, the corridor with a large free-area measured only by the laser and the entrance mainly by the noise (high variance). The states do not contain any parametric information and the algorithm that establishes the connections between them is described in [3]. The right side of Figure 5 represents the measurements of the laser and sonar with three different grey-levels, corresponding to each state. The odometry was only used to record the location where each measurement was acquired.

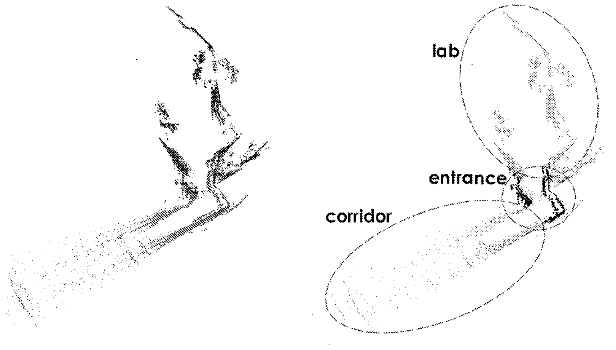

Figure 5: Left: The Laser and sonar measurements acquired in the environment. Right: The Topological map compiled by the algorithm

5 Future Development

In this paper a mapping algorithm was presented aiming at the representation of rough outdoor environments. Preliminary experimental results using a reduced number of features were presented for indoor and structured environments. According to the robot capabilities, the future work includes: i) the test of the algorithm using a larger number of features extracted from the available robot's sensor measurements (e.g., free space, environment sharpness, geometric features, speed, orientation), ii) the extensive testing in outdoor environments, iii) the integration of the Markov Model approach for localization presented in [3] in a simultaneous localization and mapping procedure in unstructured environments aiming at search and rescue operations.

Acknowledgments

Work supported by the FCT project RESCUE - Cooperative Navigation for Rescue Robots (SRI/32546/99-00) and the Programa Operacional Sociedade de Informação (POSI) in the frame of QCAIII. The first author acknowledges the Ph.D. grant SFRH/BD/929/2000 from FCT.

References

[1] M. Law, A. K. Jain, M. Figueiredo. "Feature selection in mixture-based clustering", Neural Information Processing Systems, 2002.

[2] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit. "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem", Proc. of the AAAI National Conference on Artificial Intelligence, Edmonton, Canada, 2002.

[3] A. Vale, M. I. Ribeiro. "A Probabilistic Approach for the Localization of Mobile Robots in Topological Maps", Proc. of the 10th IEEE Mediterranean Conf. on Control and Automation - MED2002, Lisboa, Portugal, 2002.

[4] S. Thrun. "Probabilistic Algorithm in Robotics" Artificial Inteligence Mag., **21(4)**, pp. 93–109, 2000.

[5] E. Nebot, S. Baiker. "Autonomous Navigation and Map building Using Laser Range Sensors in Outdoor Applications", Journal of Robotic Systems, 2000.

[6] S. Thrun, W. Burgard and D. Fox, "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping", Proc. of the IEEE ICRA, San Francisco, CA, 2000.

[7] S. Thrun, A. Bucken. "Integration Grid-Based and Topological Maps for Mobile Robot Navigation", Proc. of the 13th National Conf. on Artificial Intelligence AAAI, Portland, Oregon, 1996.

[8] A. Papoulis. *Probability, Random Variables and Stochastic Processes*, McGraw-Hill, 1991.

[9] L. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications", Proc. of the IEEE, pp. 257–286, 1989.