Proceedings of 2004 IEEE/RSJ International Conference on
Intelligent Robots and Systems
September 28 - October 2, 2004, Sendai, Japan

Topology Learning and Recognition using Bayesian Programming for Mobile Robot Navigation

Adriana Tapus, Guy Ramel, Luc Dobler and Roland Siegwart
Autonomous Systems Lab
Swiss Federal Institute of Technology, Lausanne (EPFL)
1015 Lausanne, Switzerland
{Adriana.Tapus, Guy.Ramel, Luc.Dobler, Roland.Siegwart}@epfl.ch

Abstract— **This paper proposes an approach allowing topology learning and recognition in indoor environments by using a probabilistic approach called Bayesian Programming. The main goal of this approach is to cope with the uncertainty, imprecision and incompleteness of handled information. The Bayesian Program for topology recognition and door detection is presented. The method has been successfully tested in indoor environments with the BIBA robot, a fully autonomous robot. The experiments address both the topology learning and topology recognition capabilities of the approach.**

Keywords*— *topology recognition, door detection, Bayesian Programming

I. INTRODUCTION

The navigation described by Gallistel in [6], as the capacity to localize itself with respect to a map, is an elementary task that a mobile and autonomous robot must carry out. To navigate reliably in indoor environments a mobile robot must know where it is. For this, the robot needs to construct or to detain a spatial representation of the environment. Reactive navigation is the simplest way to navigate. More complicated navigation approaches require learning and consequently need to memorize information. Stored information is represented as mental maps or cognitive maps – term introduced for the first time in [18] – which permit an encoding of the spatial relations between relevant locations in their environment. More details about cognitive maps can be found in [6, 14]. This has led to the concept of topological representation. The topological map can be viewed as a graph, where at each node the information concerning the visible landmarks and the way to reach the connected places is stored. The topological approach gives a compact representation and allows high-level symbolic reasoning for map building and navigation. The main idea is to learn different types of places and to recognize the situations previously learned.

This paper presents a model for learning the topology and recognizing the learned situations based on a probabilistic approach. Probabilities will be used to express uncertainty and to express knowledge specific to the topological maps, in the context of the Bayesian Programming formalism.

The remainder of this paper is organized as follows. We present in Section II a short review of related work in topological mapping. In Section III, we provide a brief introduction to Bayesian Programming formalism. Section IV is dedicated to topology learning and recognition by using Bayesian Programming. Experimental results are presented in Section V. Section VI concludes the paper with a discussion of the proposed approach and further research directions.

II. RELATED WORK

Many methods have been proposed to represent an environment in the framework of autonomous navigation, from precise geometric maps based on raw data or lines up to purely topological maps using symbolic descriptions. Each of these methods is optimal concerning some characteristics but can be very disappointing with respect to other requirements. Metric maps are suited when it is necessary for the robot to know its location accurately in terms of metric coordinates. However, in office buildings with corridors and rooms, or roads, the topology of important locations and their connections is enough for navigation. Topological maps are less complex and permit more efficient planning than metric maps. Moreover, it is easier to generate and maintain global consistency for topological maps than for metric maps. A full review of navigation systems can be found in [20]. In this section, we focus on topology learning and recognition for mobile robots that have been studied by many researchers.

There are two main approaches to construct topological maps: one is to learn the topological structure directly and the other one is to build the topological map on top of the metric map.

An example of the second method is given by Thrun in [17] who uses the occupancy-grid based maps in order to build the metric map. The topological map is extracted from the grid-based map. Learning the topological representation depends on learning the geometric map, which relies on the odometry abilities of the robot. However, in large environments, it is difficult to maintain the consistency of the metric map, due to the drift in the odometry.

0-7803-8463-6/04/$20.00 ©2004 IEEE

Kortenkamp and Weymouth in [8] have used an approach based on concepts derived from a theory of human cognitive mapping that also involved topological navigation. They have used the data from the sonars combined with vision information in order to achieve a rich sensory place characterization. Their work is an amelioration of Mataric's approach [12]. The main goal of their work was the reduction of the perceptual aliasing problem, improvement obtained by introducing more sensory information for place representation.

A model by Franz, Schölkopf and Mallot [5] was designed to explore open environments within a maze-like structure and to build graph-like representations. Their method has been tested on a real robot equipped with an omni-directional camera. Place recognition was done by comparing the current observation to the stored omni-directional snapshots.

In [7] and [15], the authors used a model based on a self-organizing map which creates a topological representation of the environment while the robot explores it.

The work most similar to ours is by Aycard in [1], who learns places by using the second-order Hidden Markov Models (HMM2). The maximum likelihood estimation criteria, that determine the best model's parameters according to the corpus of observations, are employed, in order to perform the learning. The recognition is carried out using the Viterbi algorithm. For these experiments ultrasonic and infrared sensors were used. Unfortunately these sensors are very sensitive to ambient light, object color, object orientation and surface of reflection.

III. Bayesian Programming Formalism

This section briefly introduces the Bayesian Programming formalism. When programming a robot, the programmer constructs an abstract representation of its environment, which is basically described in geometrical, analytical or symbolic terms. In a way, the programmer imposes on the robot, his or her own abstract conception of the environment. Difficulties appear when the robot needs to link these abstract concepts with the robot's raw signals (obtained either from the robot's sensors or being sent to the robot's actuators). The central origin of these difficulties is the irreducible incompleteness of the models. Controlling the environment is the usual answer to these difficulties, but it may not be desirable or possible when the robot must act in an environment not specifically designed for it, populated, or subject to unexpected and unattended events.

Probabilistic methodologies and techniques offer possible solutions to the incompleteness and uncertainty problems when programming a robot. The basic programming resources are *probability distributions.*

The Bayesian Programming (BP) approach was originally proposed as a tool for robotic programming (see [11]), but nowadays used in a wider scope of applications ([13] shows some examples).

In this approach, a probability distribution is associated with the uncertainty of a logical proposition value. The usual notion of a *Logical Proposition* (true or false) and its operators (conjunction, disjunction and negation) are used when defining a *Discrete Variable*. A *Discrete Variable X* is a set of logical propositions x_i, such that these propositions are mutually exclusive (i.e. for all i,j with $i \neq j$, $x_i \wedge x_j$ is false) and exhaustive (at least one of these propositions x_i is true).

The probability distributions assigned to logical propositions are always defined according to some preliminary knowledge, identified as π. The probability $P(x_i | \pi)$ gives the probability distribution of the variable X having the value x_i, knowing π. Most of the time, probabilities will be manipulated using the Bayes rule. More details about the inference postulates and rules for carrying out probabilistic reasoning in this context can be found in [3, 2, 11].

The Bayesian Programming formalism allows for using a unique notation and provides a structure to describe probabilistic knowledge and its use. The elements of a Bayesian Program are illustrated in Figure 1. A BP is divided in two parts: a description and a question.

$$\text{Prog}\begin{cases}\text{Desc}\begin{cases}\text{Spec }(\pi)\begin{cases}\text{Pertinent variables}\\ \text{Decomposition}\\ \text{Forms}\begin{cases}\text{Parametric OR}\\ \text{Programs}\end{cases}\end{cases}\\ \text{Identification based on Data }(\delta)\end{cases}\\ \text{Question}\end{cases}$$

Figure 1: Structure of a Bayesian Program

A. *Description*

The purpose of a description is to specify an effective method to compute a joint distribution on a set of relevant variables $\{X_1,X_2,\ldots,X_n\}$, given a set of experimental data δ and a priori knowledge π.

In the specification phase of the description, it is necessary to:

- Define a set of relevant variables $\{X_1,X_2,\ldots,X_n\}$, on which the joint distribution shall be defined;
- Decompose the joint distribution into simpler terms, using the conjunction rule. The conditional independence rule can allow further simplification, and such a simplified decomposition of the joint distribution is called decomposition.
- Define the forms for each term in the decomposition; i.e. each term is associated with either a parametric form, as a function, or to another Bayesian Program.

B. *Question*

Given a description $P(X_1,X_2,\ldots,X_n \mid \delta\ \pi)$, a question is obtained by partitioning the variables $\{X_1,X_2,\ldots,X_n\}$ into three sets: *Searched*, *Known* and *Unknown* variables. A question is defined as the distribution $P(Searched \mid Known\ \delta\ \pi)$. In order to answer this question, the following general inference is used:

$$P(Searched \mid Known\delta\pi) = \frac{\Sigma_{Unknown} P(Searched Unknown Known)}{\Sigma_{Unknown, Searched} P(Searched Unknown Known)}$$

Depending on the number of variables (and its discretization) and the decomposition choice, this calculation may need a lot of computational time and turn out to be infeasible. Numerous techniques have already been proposed to achieve an admissible computation time. A brief summary of the approximative approaches used for reducing calculation time can be found in [13]. In [3], one of these approximative methods is described in detail.

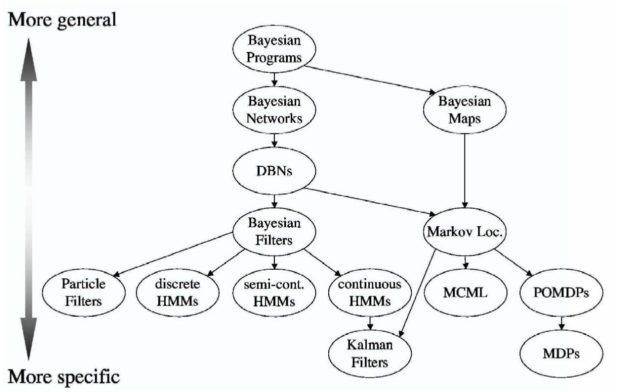


Figure 2: Bayesian Programming and other probabilistic approaches

C. Bayesian Programs and Other Probabilistic Approaches

Bayesian Programs have been shown to be a generalization of most of the other probabilistic approaches [3], as shown in Figure 2. It means that all these probabilistic approaches may be reformulated following the Bayesian Program formalism and thus easily compared with one another. For instance, Bayesian Networks correspond to a description where one and only one variable may appear to the left of each probability distribution appearing in the decomposition. This restriction enables optimized inference algorithms for certain class of questions.

IV. TOPOLOGY LEARNING AND PLACE RECOGNITION WITH BAYESIAN PROGRAMMING

Bayesian Programming can be used to solve typical robotics problems and incorporates the programmer's preliminary knowledge in the specification of the description part (the choice of pertinent variables, the joint distribution decomposition and the parametric forms). Before getting into the details of our BP, a description of the topology situations and doors is shown, as it is very important in the choice of pertinent variables as well as in the decomposition of the joint distribution.

A. The topology situations and doors corpus

A corpus with all the topology situations and doors that the robot must detect during the application phase is constructed as shown in Figures 3 and 4. In Figure 3 the topology situations are illustrated. These are: corridor, X-crossing, T-crossing and L-Intersection. For the T-crossing we have chosen three cases. This decision is justified by the fact that the recognition is only made with probability distributions and if we would have had only one state for the T-crossing, the distribution would not have been sufficient for this state. Figure 4 depicts different types of doors: closed door, right partially-opened door, left partially-opened door, opened door and no door. This implies the assumption that the environment is orthogonal, which is the case for most office buildings including the institute building where the robot operates. The above mentioned limitation is not an inherent loss of generality because it is only a simplification for the current implementation and not a general requirement of the algorithm.

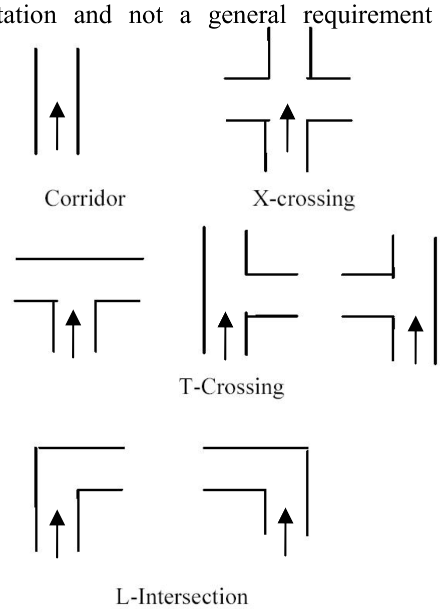


Figure 3: The topology to learn: Corridor, +Intersection, T Intersection, L-Intersection.

The main goal of this work is to determine the state in which the robot may be (for instance that the robot is in a corridor and has a partially-opened door on his right). To solve this problem, two Bayesian Programs will be used, one for the topology recognition and one for the door detection. These two programs are described in more detail in the next section.

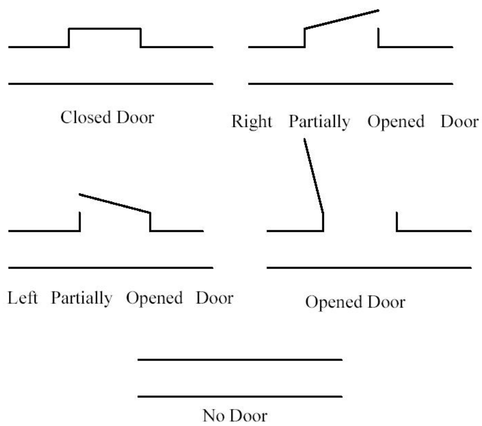


Figure 4: The types of door to learn: closed door, right partially-opened door, left partially-opened door, opened door and no door.

B. The Bayesian Program used for topology learning and place recognition

The new approach presented here is constructed in two steps. The first step is the phase of supervised learning where the robot visits different situations denoted by *State*. In each situation $sit \in State$ the robot takes an observation and stores it along with the name of the situation in a database, denoted by the symbol δ. The second step is the phase of application, when we want the robot to recognize the situation in which it is. To solve this problem, the robot will extract the actual observation and answer the following probabilistic question:

$$sit^* = \underset{sit \in State}{\arg\max} \; P(sit \mid obs\delta\pi)$$

The actual state of the robot may be recovered by comparing the actual observation with the database of known situations and choosing the situation *sit** with the highest probability. Next, we will show how this probabilistic question can be solved by applying the Bayesian Programming technique.

Figure 5 illustrates the Bayesian Program used for the topology recognition and door detection. We consider that the observations $V_1..V_n$ are dependent on the location and these dependencies lead to the decomposition described in the Bayesian Program. In our case, the observation V_i is equal to the maximum distance found with the laser scanner in the corresponding *i* section (see Figure 6). This choice reinforces the robustness of the observation V_i with respect to the robot's orientation (i.e. if the robot's orientation changes with an angle smaller than n/360° degrees, the robot will find similar values) and to the noise in the environment (i.e. a person near the robot will not influence the observation since the robot takes into account only the maximum distances).

From the result of the decomposition formula (see Figure 5) we can distinguish two different kinds of probability distributions:

- Since we have no a priori information about the different topology situations or about the doors, we consider each situation to be equally probable and consequently we express the probability of a state given all the a priori knowledge, as a uniform distribution.
- To determine the probability of one observation V_i, given the topology or door situation and all the a priori knowledge, we use Gaussian laws with means and standard deviations being updated at each new measurement, permitting an incremental learning process.

The two equations in Parametric Form will solve the basic question described in the Bayesian Program (see Figure 5).

The interesting point is that the same Bayesian Program is used for both topology recognition and door detection. The only thing that changes is the domain of the variable *State* and the database used for the learning. For the case of topology recognition, the variable *State* contains the following values: corridor, X-crossing, T-crossing and L-Intersection. The 360° view of the robot is divided in *n* equal parts, as shown in the Figure 6.a). In the case of door detection the variable *State* takes the values: closed door, right partially-opened door, left partially-opened door, opened door and no door.

Program

Description

Specification

Pertinent Variables

State: the state of the robot

$V_1...V_n$: the observations of the robot

Decomposition

$$P(State\ V_1...V_n \mid \pi) =$$

$$P(State \mid \pi) \bullet \prod_{i=1}^{n} P(V_i \mid State\pi)$$

Parametric Forms

$P(State \mid \pi) =$ Uniform

$P(V_i \mid State\pi) =$

$$G\mu_i(State); \sigma_i(State)(V_i)$$

Identification

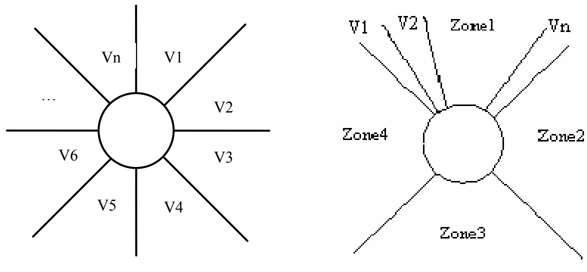
A Priori

Question

$P(State_j \mid V_1..V_n\pi)$

Figure 5: The Bayesian Program used for the topology recognition and door detection, following the unique notation and structure described in section III.

The observation field is portioned into four Zones. The method of splitting the field of view of the robot in zones will permit also the detection of the direction of doors. The Bayesian Program will give an answer for each of these zones in order to detect a door in front, behind, to the left and to the right of the robot. Each of these zones is split in n equal slices, as illustrated in the Figure 6.b). For both cases: topology recognition and door detection, we have fixed n to 8.



a) b)

Figure 6: The observation field of view of the robot: a) for the topology recognition the 360° view of the robot is portioned in *n* parts; b) for doors detection the view of the robot is divided in four Zones and each zone is portioned in *n* parts. In our implementation, for both cases a) and b) n is equal to 8.

Note how flexible this method is with respect to the utilization of the same program for two different tasks: topology recognition and door detection.

V. Experimental Results

The approach has been tested in a 50 m x 25 m portion of our institute building.

For the experiments, the BIBA robot (see Figure 7), a fully autonomous mobile robot, has been used.

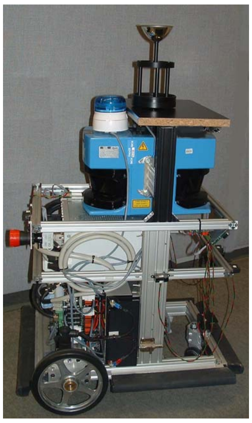


Figure 7: The fully autonomous robot BIBA.

Its controller consists of a VME standard backplane with a PowerPC 750 clocked at 400 MHz running XO/2, a hard real-time operating system and a Pentium III running at 700 MHz, 128 MB RAM on Windows 2000 for all interaction tasks. Both computers can communicate with each other over a 10 Mbit/sec local Ethernet and with a central computer over wireless interfaces to allow for monitoring of the state of the robot for security reasons. Among its peripheral devices, the most important are the wheel encoders, two 180° laser range finders, five infra-red sensors, four ultra-sound sensors and an omni-directional camera. In our application only the two laser range finders are used.

We have constructed the training data and built a model for each of the seven topology situations and for each of the five door types. The robot was placed 25 times in each situation in order to construct a robust training corpus. A simple navigation system has been implemented on the robot, so that the robot will stay in the middle of the corridor (i.e. mid-line following), parallel to the two walls constituting the corridor. In order to complete the training, for each situation and each observation the Gaussian parameters (the mean and the standard deviation) were calculated.

To test our topology and door recognition, we have performed 50 tests for each situation. The results are summarized in Table I and Table II.

In these two tables the results of the topology recognition door detection are presented. Each line corresponds to a situation that the robot observed and each column corresponds to a situation that the robot recognized. In Table I, we can see that for instance, the robot has recognized the corridor almost all the time and the percentage of recognition of a corridor is 98%. For the topology, the percentage of successfully recognition is between 82% and 98%, and at an average of 92.2%. It is important to notice that the falsly recognized situations are always similar to the real topology situations. It would have been more compromising to recognize a situation like ╔ or ╠ where the situation were ╗ and ╣ respectively, because these are opposite topology situations.

TABLE I. The table shows the results of topology recognition. The following notations has been used: ║(corridor), ╗ (left L-Intersection), ╔ (right L-Intersection), ╣ (left T-crossing), ╠ (right T-crossing), ╦ (middle T-crossing), ╬ (X-crossing).

	║	╗	╔	╣	╠	╦	╬
║	98%			2%			
╗		82%				18%	
╔			96%			4%	
╣	8%			86%			6%
╠	6%				90%		4%
╦						98%	2%
╬	4%						96%

In Table II, we can notice that in the case of the "left partially opened door" situation, 36% of responses were false. Instead of detecting the "left partially opened door" situation, the "opened door" is detected. However, this is not very important if the context of recognition is the detection of doors without considering its aperture. A similar false detection can be observed in the case of a "closed door" situation, where there are 20% of "no door" detections. These false detections can still be considered good results knowing that to determine a "closed door" situation, a jump near the frame of the door must be found. The percentage of successful door detection is between 60% and 90%, and at an average of 80.4%.

TABLE II. The table shows the results of door detection. The following notation has been used: nd (no door), cd (closed door), od (opened door), lpod (left partially opened door) and rpod (right partially opened door).

	nd	cd	od	lpop	rpod
nd	90%		4%	6%	
cd	20%	76%			4%
od	6%		90%	4%	
lpod			36%	60%	4%
rpod	4%		10%		86%

The application of our Bayesian Program for door detection shows how well suited our approach is, for this type of recognition. If we regroup the three situations in which the door is partially or completely opened, we have three possible situations "opened or partially opened door", "closed door" and "no door". The results for these three situations are even better than their precedents, the percentage of succesful recognition being of 96%, 76% and 90% respectively. Another interesting statistic was computed in order to detect the percentage of successful door and no door detection. The results are quiet concluding, 90% and 94% respectively.

A combination between the two Bayesian Programs to perform the simultaneous topology recognition and door detection was implemented and was found to produce very promising results. A learning corpus of 50 measurements was constructed and 250 tests (50 tests for topology and 200 tests for door detection) were performed. A topology situation or a door is recognized if the actual observation matches exactly with the real situation. Substitution errors occurred during the tests. We have divided the substitution errors in two types: satisfactory substitutions (applied only for the detection of doors) and false substitutions.

We define them as:

- **Satisfactory Substitution**: The recognized situation is a confusion between the states: "right partially opened door", "left partially opened door", "opened door". For instance, the robot observes a "right partially opened door", when an "opened door" was present in the map.
- **False Substitution**: The confusion of a state with another one, not in the category of satisfactory substitution.

Table III summarized the results obtained for the global recognition.

TABLE III. The table shows the results of global recognition

	Number	%
Tests	250	100%
Recognized	206	82.4%
Satisf. Substituted	18	7.2%
False Substituted	26	10.4%

From the experiments, it can be observed that the different situations (topology and doors) are globally well recognized. The results have given a percentage of successful recognition (classification) of 82.4% and 7.2 % of satisfactory substitution (see Table III). However, the false-substituted situations can still be used in combination with a localization approach such as a Partial Observable Markov Decision Process (POMDP) [4, 19] to give satisfactory results.

VI. Conclusion And Future Work

This paper presents a method for topology learning and place recognition by using the Bayesian Programming methodology. This work took place in the context of a new programming technique based on Bayesian inference, called *Bayesian Programming*. From the experiments, we conclude that the presented approach is practical and very robust. After 250 tests, the Bayesian Program used for the global recognition gave 82.4% of successful classification and 7.2% of satisfactory substitution, which still represents positive results. Even if the correct situation is not always detected, the information given by the false-substituted situations can still be used for localization e.g. by employing a localization approach like POMDP. Future works will focus on the topological map building combined with the fingerprint approach [9, 10 and 16]. This fusion will allow the elimination of the perceptual aliasing problem and the improvement of distinctiveness and uniqueness of places in the environment.

Acknowledgment

The authors wish to thank Nicola Tomatis and Pierre Bessière for their valuable feedback about this work. This work was sponsored by the European project BIBA IST-2001-32115.

References

[1] Aycard, O., Charpillet, F., Fohr, D. and Mari, J.F. (1997). Place learning and recognition using Hidden Markov Models. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems.

[2] Bellot, D., R. Siegwart, P. Bessière, A. Tapus, C. Coué and J. Diard (2003), Bayesian Reasoning for Real World Robotics: Basics, Scaling and Example, Dagstuhl Proceedings on Embodied AI, LNCS (Lecture Note in Computer Science), Springer.

[3] Bessière, P., et al., Survey: Probabilistic Methodology and Techniques for Artefact Conception and Development, Technical Report INRIA, RR-4730, European IST BIBA project, 2003.

[4] Cassandra, A. R., L. P. Kaelbling, et al. (1996). Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. *IEEE International Conference on Robotics and Automation*, Osaka, Japan.

[5] Franz, M.O., Schölkopf, B., Mallot, H.A. and Bülthoff.(1998). Learning view graphs for robot navigation, Autonomous Robots 5 111-125.

[6] Gallistel, R. (1990), The Organization of Learning, MIT Press: Cambridge, MA.

[7] Hafner V.V. (2000), Learning Places in Newly Explored Environments, in Meyer, Berthoz, Floreano, Roitblat and Wilson (Eds.), SAB2000 Proceedings Supplement Book, Publication of the International Society for Adaptive Behavior, Honolulu.

[8] Kortenkamp, D. and Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing, In Proceedings of AAAI-94, Seattle, WA.

[9] Lamon, P., I. Nourbakhsh, B. Jensen and R. Siegwart (2001). Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization. *IEEE International Conference on Robotics and Automation*, Seoul, Korea.

[10] Lamon, P., Tapus A., Glauser E., Tomatis N., Siegwart R. (2003), Environmental Modeling with Fingerprint Sequences for Topological Global Localization, In Proceedings of the IEEE International Conference on Intelligent Robot and Systems, Las Vegas, USA, October 27-30.

[11] Lebeltel, O., P. Bessière, J. Diard and E. Mazer (2003), Bayesian Robot Programming, In Autonomous Robots.

[12] Mataric, M. J. (1991). Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation. In: J.A.Meyer, S.W.Wilson (Eds), From Animals to Animats, MIT Press, Cambridge, MA.

[13] Mekhnacha, K., E. Mazer, and P. Bessière, (2000), A robotic CAD system using a Bayesian framework. In Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems, Takamatsu(JP), Best Paper Award.

[14] O'Keefe, J. and Nadel, L. (1978). The Hippocampus as a Cognitive Map, Clarendon Press:Oxford.

[15] Owen, C. and Nehmzow, U.(1998), Landmark-based navigation for a mobile robot, in : Meyer, Berthoz, Floreano, Roitblat and Wilson (Eds.), From Animals to Animate 5, Proceedings of SAB'98, MIT Press, Cambridge, MA, pp. 240-245.

[16] Tapus A., Tomatis N. and Siegwart R. (2004), Topological Global Localization and Mapping with Fingerprints and Uncertainty, In Proceedings of the International Symposium on Experimental Robotics, Singapore, June 18-21.

[17] Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. In Artificial Intelligence 99(1):21-71.

[18] Tolman, E. C. (1948), Cognitive maps in rats and men, Psychological Review, 55:189-208.

[19] Tomatis, N., I. Nourbakhsh, and R. Siegwart (2003). Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems,* 44:3-14.

[20] Trullier, O., Wiener, S. L., Berthoz, A., Meyer, J. A. (1997). Biologically based artificial navigation systems: review and prospects. Progress Neurobiology, 51:483-544.