A theoretical comparison of probabilistic and biomimetic models of mobile robot navigation

Julien Diard, Pierre Bessiere, Emmanuel Mazer

Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

A theoretical comparison of probabilistic and biomimetic models of mobile robot navigation

Julien Diard, Pierre Bessière and Emmanuel Mazer
Laboratoire GRAVIR / IMAG – CNRS
INRIA Rhône-Alpes, 655 avenue de l'Europe
38330 Montbonnot Saint Martin FRANCE
Julien.Diard@free.fr

Abstract— **This paper deals with the domain of space modeling for mobile robotics. It offers a comparison of probabilistic and biomimetic models of navigation. Both approaches are shown to be quite complementary: while the probabilistic methods exploit sound theoretical grounds, they lack the modularity and, as a consequence, flexibility, of their biomimetic counterparts. We propose a new formalism, called the *Bayesian Map* formalism, that attempts to bridge the gap between the two domains: it is based on Bayesian modeling and inference for defining the building blocks, and uses operators for building hierarchies of models.**

I. Introduction

In robotics, modeling the environment that a robot has to face in a navigation task is a crucial problem, that has received a lot of attention in the community. On the one hand, the most successful practical achievements have been obtained using methods based on the probabilistic calculus. Indeed, it offers a sound theoretical basis for handling incomplete models and uncertain information. On the other hand, biomimetic models of navigation, while not as widely applied as their probabilistic counterparts, are centered on the notion of hierarchical and modular models.

However, the state-of-the-art robotic applications are still largely outmatched by animal capabilities, when it comes to dealing with large scale environments and exhibiting complex navigation skills. While the advent of better technological tools (more precise sensors, more on-board computational power) will certainly help progressing toward that goal, we believe a conceptual breakthrough is still needed.

In this paper, we propose a possible marriage between probabilistic and biomimetic models: the *Bayesian Map* formalism. It tries to benefit from both worlds. Being a generalization of the Markov Localization model, it is based on the probabilistic methodology. By defining *operators* for combining Bayesian Maps, it brings modularity and the possibility of incrementally building hierarchies of models of space.

The rest of this paper is organized as follows. Section II presents the Bayesian Robot Programming methodology, whose unique notation we use throughout the rest of the paper. Section III and Section IV briefly present the probabilistic and biomimetic models, respectively, and some of their most salient features. Section V discusses some of their strengths and weaknesses. It also provides the rationale for our Bayesian Map model, which is introduced Section VI.

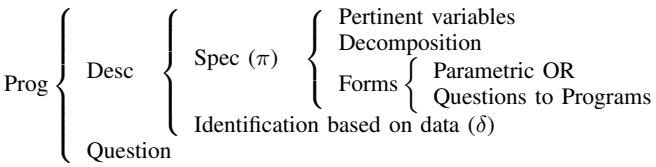

Fig. 1. Structure of a Bayesian Robotic Program.

II. Bayesian Robot Programming

One of the most general probabilistic modeling tool is arguably [1] the Bayesian Robotic Programming methodology. We summarize it here, but still invite the reader to refer to [2] for all the details. In the BRP formalism, a bayesian robotic program is a structure (see Fig. 1) made of two components.

The first is a *declarative* component, where the user defines a **description**. The purpose of a description is to specify a method to compute a joint distribution over a set of relevant variables $\{X_1, X_2, \ldots, X_n\}$, given a set of experimental data δ and preliminary knowledge π. This joint distribution is denoted $P(X_1\ X_2\ \ldots\ X_n \mid \delta\ \pi)$. To specify this distribution, the programmer first lists the pertinent variables (and defines their domains), then decomposes the joint distribution as a product of simpler terms (possibly stating conditional independence hypotheses so as to simplify the model and/or the computations), and finally, assigns forms to each term of the selected product (these forms can be parametric forms, or recursive questions to other bayesian programs). If there are free parameters in the parametric forms, they have to be assessed. They can be given by the programmer (*a priori* programming) or computed on the basis of a learning mechanism defined by the programmer and some experimental data δ.

The second component is of a *procedural* nature, and consists of using the previously defined description with a **question**, *i.e.* computing a probability distribution of the form $P(Searched \mid Known)$. Answering a "question" consists in deciding a value for the variable $Searched$ according to $P(Searched \mid Known)$. It is well known that Bayesian inference is a very difficult problem, which may be practically intractable. But, as this paper is mainly concerned with modeling issues, we will assume that the inference problems are solved and implemented efficiently by the programmer.

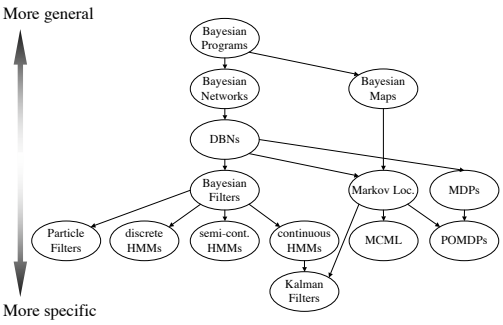

Fig. 2. Some common probabilistic modeling formalisms and their general-to-specific partial ordering (reprinted from [1]).

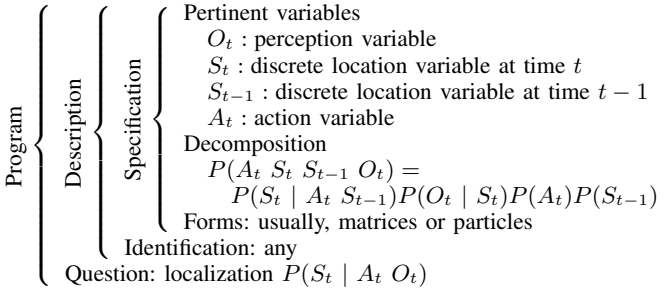

Fig. 3. The Markov Localization stationary model expressed as a BRP.

III. Probabilistic models of navigation

There is currently a wide variety of models in the domain of probabilistic mobile robot programming. These approaches include Kalman Filters (KF, [3]), Markov Localization models (ML, [4]), (Partially or Fully) Observable Markov Decision Processes (POMDP, MDP, [5]), Hidden Markov Models (HMM, [6]), Bayesian Filters (BFs), or even Particle Filters (PFs). The literature covering these models is huge: for references that present several of them at once, giving unifying pictures, see [1], [7], [8], [9]. Some of these papers define taxonomies of these approaches, by proposing some order which helps classifying them into families. One such taxonomy is presented Fig. 2. It is based on a general-to-specific ordering: for example, it shows that the Dynamic Bayesian Networks (DBN) are a specialization of the Bayesian Network formalism, tailored for taking into account time series. We can see Fig. 2 subtrees that correspond to different specialization strategies. In the remainder of this section, we will focus on the Markov Localization subtree, which corresponds to specializing DBNs by the choice of a four variable model.

The ML model is basically a HMM with an additional action variable. Indeed, it seems relevant in the robotic programming domain, since robots obviously can affect their states via motor commands. The stationary model of a HMM is basically the decomposition $P(O_t\ S_t\ S_{t-1}) = P(S_{t-1})P(S_t \mid S_{t-1})P(O_t \mid S_t)$, where O_t is a perception variable, S_t and S_{t-1} are location variables at time t and $t-1$, $P(S_t \mid S_{t-1})$ is commonly called the transition model and $P(O_t \mid S_t)$ is referred to as the observation model. Starting from this structure, the action variable A_t is used to refine the transition model $P(S_t \mid S_{t-1})$ into $P(S_t \mid A_t\ S_{t-1})$, which is called the action model. Due to the generality of the BRP formalism, the model for Markov Localization can be cast into a BRP program. This is shown Fig. 3.

The ML model is mostly used in the literature to answer the question $P(S_t \mid A_t\ O_t)$, which estimates the state of the robot, given the latest motor commands and sensor readings. When this state represents the position of the robot in its environment, this amounts to localization.

The ML model has been successfully applied in a range of robotic applications, the most notable examples being the Rhino ([10], [11]) and Minerva ([12], [13]) robotic guides. The most common application of the ML model is the estimation of the position of a robot in an indoor environment, using a fine-grained metric grid as a representation. In other words, in the model of Fig. 3, the state variable is commonly the pose, *i.e.*, a pair of x, y discrete Cartesian coordinates for the position, and an angle θ for the orientation of the robot. Assuming a grid cell size of $50\ cm$, an environment of $50 \times 50\ m^2$, and a 5° angle resolution entails a state space of 720,000 states.

Using some specialized techniques and assumptions, it is possible to make this model tractable.

For example, the forms of the probabilistic model can be implemented using sets of particles. These approximate the probability distributions involved in Fig. 3, which leads to an efficient position estimation. This specialization is called the Monte Carlo Markov Localization model (MCML, [14]).

Another possibility is to use a Kalman Filter as a specialization of the ML model, in which variables are continuous. The action model $P(S_t \mid A_t\ S_{t-1})$ and the observation model $P(O_t \mid S_t)$ are both specified using Gaussian laws with means that are linear functions of the conditioning variables. Due to these hypotheses, it is possible to analytically solve the inference problem to answer the localization question. This leads to an extremely efficient algorithm that explains the popularity of Kalman Filters.

IV. Biomimetic models of navigation

Very early in their analysis, biomimetic models of navigation dispute the classical view of robotic navigation, which is inherited from marine navigation. In this view, solving a navigation task basically amounts to answering the questions of Levitt and Lawton: "Where am I?", "Where are the other places with respect to me?", and "How do I get to other places from here?" [15], or alternatively, those of Leonard and Durrant-Whyte: "Where am I?", "Where am I going?", and "How should I get there?" [16].

While a valid first decomposition of the navigation task into subtasks, these questions have usually led to models that require a *global model of the environment*, which allows the robot to localize itself (the first questions), to infer spatial relationships between the current recognized location and

other locations (the second questions), and to plan a sequence of actions to move in the environment (the third questions). These skills amount to the first two phases of the "perceive, plan, act" classical model of robotic control.

When studying living beings, the existence of such a unique and global representation that would solve these three questions is very problematic. This seems obvious even for simple animals like bees and ants. For instance, the desert ant *cataglyphis* outdoor navigation capabilities are widely studied, and rely on the use of the polarization patterns of the sky [17]. But it is clear that such a strategy is useless for navigating in their nest; this calls for another navigation strategy, and another internal model. The existence of a unique representation it is also doubtful for human beings. The navigation capabilities of humans are based on internal models of their environment (*cognitive maps*), but their nature, number and complexities are still largely debated (see for instance [18], [19], for entry points into the huge literature associated to this domain).

As a consequence, biomimetic approaches assume, right from the start, the existence of a *hierarchy* of representations.

The hierarchies of models proposed in the biomimetic literature ([20], [21], [22], [23]) have several aspects: they are hierarchies of increasing navigation skills, but also of increasing scale of the represented environment, of increasing time scale of the associated movements, and of increasing complexity of representations. This last aspect means that topologic representations, which are simple, come at a lower level than global metric representations, which are arguably more complex to build and manipulate. This ordering stems from at least two observations. The first one is that animals that are able to use shortcuts and detours (skills that require a metric model) are rather complex animals, like dogs and monkeys. These skills seem to be mostly absent from simpler animals, like invertebrates. The second observation is that children seem to build topological representations earlier than metric ones in their development process.

The resulting proposed hierarchies show a striking resemblance. We present the salient and common features of these hierarchies by summarizing the Spatial Semantic Hierarchy (SSH) proposed by Kuipers [20], [23]. It is, to the best of our knowledge, the only biomimetic approach that was applied to obtain a complete and integrated robotic control model.

The SSH essentially consists of four hierarchical levels: the *control* level, the *causal* level, the *topological* level, and the *metrical* level.

The control level is a set of reactive behaviors, which are control laws deduced from differential equations. These behaviors describe how to move the robot for it to reach an extremum of some gradient measure. This extremum can be zero-dimensional: a point in the environment, called a locally distinctive state. The associated behavior is called a hill-climbing law. The extremum can also be one-dimensional (a line or curve in the environment), in which case the behavior is called a trajectory-following law. Provided that any trajectory-following law guarantees arriving in a place where a hill-climbing law can be applied, then alternating laws of both types displace the robot in a repetitive fashion. This solves the problem of the accumulation of odometry errors. The control level is also referred to as the guidance level [21], [22].

Given the control level, the environment can be structured and summarized by the locations of locally distinctive states and the trajectories used to go from one such state to another. This abstraction takes place in the causal level, which is the second level of the hierarchy of representations. Unlike the control level, it allows the robot to memorize relationships between places that are outside of the current perceptive horizon (what is part of the way-finding capabilities, in other terminologies [21], [22]). To do so, Kuipers abstracts locally distinctive places as views V, the application of lower level behaviors as actions A, and defines *schemas* as tuples $\langle V, A, V' \rangle$ (expressed as first-order logic predicates). The schemas have two meanings. The first is a procedural meaning: "when the robot is in V, it must apply action A." This aspect of the schemas is equivalent to the recognition-triggered response level of the other hierarchies [21], [22], or to the potential field approaches, or to other goal-oriented methods. But the second meaning of schemas is a declarative one, where $\langle V, A, V' \rangle$ stands for: "applying action A from view V brings eventually the robot at view V'." This allows using the schemas for prediction of future events, or in a planning process, for example.

The goal of the topological level is to create a globally consistent representation of the environment, as structured by places, paths and regions. These are extracted from lower level schemas, by an abduction process, which creates the minimum number of places, paths and regions so as to be consistent with the known schemas. Places are zero-dimensional parts of the environment, which can be abstractions of lower level views, or abstractions of regions (for higher-level topological models). Paths are one-dimensional, are oriented, and can be built upon one or more schemas. Finally, regions are two-dimensional subspaces, delimited by paths. It must be noted that, since the accumulation of odometry error problem was dealt with at the control level, building a globally consistent topological representation (*i.e.* solving the global connectivity problem) is much easier. To do so, Kuipers proposes an exploration strategy, the rehearsal procedure, which, unfortunately, relies on a bound on the exploration time, and is not well suited for dynamic environments. The places and paths of the obtained topological representation can be used for solving planning queries using classical graph-searching algorithms.

The last level is the metrical level, in which the topological graph is cast into a unique global reference frame. For reasons outlined above (and detailed in [23]), this level is considered as a possibility, not a prerequisite for solving complex navigation tasks. If the sensors are not good enough to maintain a good estimation of the Cartesian coordinates of the position, for instance, it is still possible to use the topological model for acting in the environment – while shortcuts and detours are not possible. Indeed, few robotics systems implementing biomimetic models include the metrical level [22], [21].

V. THEORETICAL COMPARISON

A. What mathematical formalism?

A major drawback of the biomimetic model presented previously is that it uses a variety of formalisms for expressing knowledge: differential equations and their solutions for the control level, first-order logic and deterministic algorithms for higher-level layers of the hierarchy.

This makes it difficult to theoretically justify the consistency and correctness of the mechanisms for communication between the layers of the hierarchy. In some cases, it even limits and constraints the contents of the layers: for instance, the SSH model *requires* that the behaviors of the control level *guarantee* that the robot reaches the neighborhood of a given locally distinctive state. In our view, this constraint is barely acceptable for any kind of realistic robotic scenario. Consider dynamic environments: how to guarantee that a robot will reach a room, if the door on the way can be closed?

We take as a starting point of our analysis that the best formalism for expressing incomplete knowledge and dealing with uncertain information is the probabilistic formalism [24]. This gives us a clear and rigorous mathematical foundation for our models. The probability distributions are our unique tool for the expression and manipulation of knowledge, and in particular, for the communication between submodels. We will thus argue in favor of hierarchical probabilistic models.

B. Hierarchical probabilistic models

This idea is not a breakthrough: in the domain of probabilistic modeling for robotics, hierarchical solutions are currently flourishing. The more active domain in this regard is decision theoretic planning: one can find variants of MDPs that accommodate hierarchies or that select automatically the partition of the state-space (see for instance [25], [26], or browse through the references in [27]). More exceptionally, one can find hierarchical POMDPs, as in [27]. Some hierarchical approaches outside of the MDP community include Hierarchical HMMs and their variants (see [7] and references therein), which, unfortunately, rely on the notion of final state of the automata. Another class of approaches relies on the extraction of a graph from a probabilistic model, like for example a Markov Localization model [28], or a MDP [29]. Using such deterministic notions is inconvenient in a purely probabilistic approach, as we are pursuing here.

Moreover, the main philosophy used by all the previous approaches is to try to extract, from a very complex but intractable model, a hierarchy of smaller models (structural decomposition, see [27]).

Again, this comes from the classical robotic approach, where the process of perception (in particular, the localization) is assumed to be independent of the processes of planning and action. A model such as the ML model (Fig. 3) is only concerned with localization, not control: therefore its action variable A_t is actually only used as an *input* to the model. In this view, a pivotal representation is used between the perception and planning subproblems. It is classically assumed that the more precise this pivotal model, the better. Unfortunately, when creating integrated robotic applications, dealing with both the building of maps and their use is necessary. Some authors have realized, at this stage, that their global metric maps were too complex to be easily manipulated. Therefore, they have tried to *degrade* their maps, that was so difficult to obtain initially, by extracting graphs from their probabilistic models, for instance [28]. This problem is also the core of the robotic planning domain, where the given description of the environment is assumed to be an infinitely precise geometrical model. The difficulty is to discretize this intractable, continuous model, into a finite model [30], [31], [32], [33] (typically, in the form of a graph).

C. Toward Bayesian Maps

We propose to pursue an alternate route, investigating how, starting from a set of simple models, one can combine them for building more complex models. Such an incremental development approach allows us to depart from the classical "perceive, plan, act" loop, considering instead hierarchies built upon many imbricate models, each of them deeply rooted into lower level *sensory and motor* relationships.

The Bayesian Robotic Programming methodology offers exactly the formal tool that is needed to transfer information from one program to another in a theoretically rigorous fashion. We have seen Section II that, in a Bayesian Robotic Program, a form of a description c^1 could be defined as a probabilistic question to another description c^2. Depending on the way questions are used to link subprograms, several different operators can be created, each with specific semantics: for instance, in the framework of behavior based robotics, Lebeltel has defined behavior combination, hierarchical behavior composition, behavior sequencing, sensor model fusion operators. He has also applied these successfully to realize a complex watchman robot behavior using a control architecture involving four hierarchical levels [2].

This allows to solve a global robotic task problem by first, decomposing it into subproblems, then, writing a Bayesian Robot Program for each subproblem, and finally, combining these subprograms. This method makes robot programming similar to structured computer programming. So far in our work, we let the programmer do this analysis: relevant intermediary representations can be imagined, or copied from living beings. We propose to apply this strategy to the map-based navigation of mobile robots. The submodels can be submaps, in the spatial sense (*i.e.* covering a part of the global environment), or in the subtask sense (*i.e.* modeling knowledge necessary for solving part of the global navigation task), or even in less familiar senses (*e.g.* modeling partial knowledge from part of the sensorimotor apparatus).

Our approach is therefore based on a formalism for building models of the space in which a robot has to navigate, called the *Bayesian Map* model, that allows to build submodels which provide behaviors as resources. We also define *Operators* for combining such maps together in a hierarchical manner.

VI. Bayesian Maps

A. Definition

A Bayesian Map c is a description that defines a joint distribution $P(P\ L_t\ L_{t'}\ A)$, where:

- P is a perception variable (the robot reads its values from physical sensors or lower level variables),
- L_t is a *location* variable at time t,
- $L_{t'}$ is a variable having the same domain as L_t, but at time t' (without loss of generality, let us assume $t' > t$),
- and A is an action variable (the robot writes commands on this variable).

The choice of decomposition is not constrained: any probabilistic dependency structure can therefore be chosen here: see the recent [34] for an example of how this leverage can lead to interesting new models. Finally, the definition of forms and the learning mechanism (if any) are not constrained, either.

We now invite the reader to verify that the Markov Localization model is a special case of the Bayesian Map model by comparing Fig. 3 and Fig. 4. This is also the case for KFs ot MCML, since they are specialization of ML models.

Bayesian Maps can therefore accommodate many different forms, depending on the needs or information at hand: a Bayesian Map can be structured like a real valued Kalman Filter for tracking the angle and distance to some feature when it is available. If that feature is not present, or in cases where the linearity hypotheses fail, we can use another Bayesian Map, which need not be a Kalman Filter (for example, based on a symbolic variable).

A consequence is that hierarchies built of Bayesian Maps can thus be hierarchies of Markov Localization models, hierarchies of Kalman Filters, etc. Moreover, heterogeneous hierarchies of these models can be imagined: ML over KFs, or even n KFs and one ML model, which, in our view, would be a formally satisfying alternative to [35].

For a Bayesian Map to be useable in practice, we need the description to be rich enough to generate *behaviors*. We call *elementary behavior* any question of the form $P(A^i \mid X)$, where A^i is a subset of A, and X a subset of the other variables of the map (*i.e.*, not in A^i). A behavior can be not elementary, for example if it is a sequence of elementary behaviors, or, in more general terms, if it is based on elementary behaviors and some other knowledge (which need not be expressed in terms of maps).

For a Bayesian Map to be interesting, we will also require that it generates *several* behaviors – otherwise, defining just a single behavior instead of a map is enough. Such a map is therefore a resource, based on a location variable relevant enough to solve a class of tasks: this internal model of the world can be reified.

Fig. 4 is a summary of the definition of the Bayesian Map formalism.

B. Putting Bayesian Maps together

In our previous work, we have defined two operators for putting Bayesian Maps together: the *Abstraction operator* [36] and the *Superposition operator* [37].

Program
- Description
  - Specification
    - Pertinent variables:
      - P : perception variable
      - L_t : location variable at time t
      - $L_{t'}$: location variable at time t', $t' > t$
      - A : action variable
    - Decomposition: any
    - Parametric forms: any
  - Identification: any
- Questions:
  - elementary behaviors: $P(A^i \mid X)$, with $A^i \subseteq A$, $X \subseteq \left(\{P, L_t, L_{t'}, A\} \setminus A^i\right)$

Fig. 4. The Bayesian Map model definition expressed in the BRP formalism.

The main idea behind the abstraction operator is to build *a Bayesian Map c whose different locations are other Bayesian Maps $c^1, c^2, \ldots, c^n$*. Intuitively, it can help to imagine maps c^i that cover different parts of the environment, although this is by no means enforced by the mathematical definition of the operator. With this picture in mind, we see that combining such submaps will lead to defining a higher level representation, in the sense that it covers a larger part of the environment that any individual model. In turn, this helps solving larger scale navigation tasks.

The location variable of the abstract map takes n possible symbolic values, one for each underlying map c^i. Recall that Bayesian Maps are designed for generating behaviors. Let us note $a^1, a^2, \ldots, a^k$ the k behaviors defined in the n underlying maps. In the abstract map, these behaviors can be used for linking the locations c^i. The action variable of the abstract map therefore takes k possible symbolic values, one for each behavior of the underlying maps. In order to build an abstract map having n locations, the programmer has to have previously defined n lower level maps, which generate k behaviors. The numbers n and k are therefore small, and so the abstract map deals with a small internal space, having retained of each underlying map only a symbol, and having "forgotten" all their details (the "abstraction" mechanism). The details of the abstraction operator are described in details elsewhere [36].

The main idea behind the superposition operator is that it combines together several maps that describe the same geographical space. Intuitively, it puts together Bayesian Maps by superposing each map's set of possible locations. Therefore the robot localizes itself in all the models simultaneously, giving it a richer "vocabulary" for describing its environment. Building complex models out of simpler ones in this fashion can be useful, for example when different sensory modalities are used by a robot, and have to be mixed.

The Superposition operator exists in several variants. The simpler case is when the combination does not add knowledge to the obtained Bayesian Map, with respect to the initial maps: the probabilistic forms of the obtained map are either taken from underlying maps, or are uniform distributions. While still assuming much independence between the underlying maps, the obtained map has the advantage that it can be built automatically. This initial map can then be refined through

experience: it can be used to navigate the environment and collect experimental data. This data can be used to relax independence assumptions made between the underlying maps, and identify their particular interplay by a learning mechanism. Practical details about this operator are to be found in [37].

VII. Conclusion

We have presented the Bayesian Map formalism: it is a generalization of most probabilistic models of space found in the literature, with an emphasis on hierarchies of models. We have argued that this formalism tries to bridge the gap between the probabilistic and biomimetic models of navigation. The resulting hierarchical models are built upon imbricate sensori-motor relationship that provide behaviors, thus departing from the classical "perceive, plan, act" control loop.

An obvious limitation of our approach is that it currently requires a *programmer* to analyze the navigation task to solve into subproblems. In the study of animal navigation, whether this analysis is of a phylogenetic or ontogenetic nature is an interesting question.

Although we are only at the beginning of this research track, we do believe it is a promising one, and hope it will spark some interest in the community as well.

References

[1] J. Diard, P. Bessière, and E. Mazer, "A survey of probabilistic models, using the bayesian programming methodology as a unifying framework," in *The Second Int. Conf. on Computational Intelligence, Robotics and Autonomous Systems (CIRAS 2003)*, Singapore, December 2003.

[2] O. Lebeltel, P. Bessière, J. Diard, and E. Mazer, "Bayesian robot programming," *Autonomous Robots (in press)*, vol. 16, no. 1, 2004.

[3] J. Leonard, H. Durrant-Whyte, and I. Cox, "Dynamic map-building for an autonomous mobile robot," *The International Journal of Robotics Research*, vol. 11, no. 4, pp. 286–298, 1992.

[4] S. Thrun, "Probabilistic algorithms in robotics," *AI Magazine*, vol. 21, no. 4, pp. 93–109, 2000.

[5] C. Boutilier, T. Dean, and S. Hanks, "Decision theoretic planning: Structural assumptions and computational leverage," *Journal of Artificial Intelligence Research*, vol. 10, pp. 1–94, 1999.

[6] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, New Jersey: Prentice Hall, 1993, ch. Theory and implementation of Hidden Markov Models, pp. 321–389.

[7] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," Ph.D. thesis, University of California, Berkeley, CA, 2002.

[8] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models," *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.

[9] P. Smyth, D. Heckerman, and M. I. Jordan, "Probabilistic independence networks for hidden markov probability models," *Neural Computation*, vol. 9, no. 2, pp. 227–269, 1997.

[10] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schmidt, "Map learning and high-speed navigation in RHINO," in *AI-based Mobile Robots: Case Studies of Successful Robot Systems*, D. Kortenkamp, R. Bonasso, and R. Murphy, Eds. MIT Press, 1998, pp. 580–586.

[11] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schultz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1–2, pp. 3–55, 1999.

[12] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: A second generation mobile tour-guide robot," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[13] ——, "MINERVA: A tour-guide robot that learns," in *Proceedings of the 23rd Annual German Conference on Advances in Artificial Intelligence (KI-99)*, ser. LNAI, W. Burgard, T. Christaller, and A. B. Cremers, Eds., vol. 1701. Berlin: Springer, September, 13–15 1999, pp. 14–26.

[14] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Orlando, FL, 1999.

[15] T. S. Levitt and D. T. Lawton, "Qualitative navigation for mobile robots," *Artificial Intelligence*, vol. 44, no. 3, pp. 305–360, 1990.

[16] J. J. Leonard and H. F. Durrant-Whyte, "Mobile robot localization by tracking geometric beacons," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[17] D. Lambrinos, R. Möller, T. Labhart, R. Pfeifer, and R. Wehner, "A mobile robot employing insect strategies for navigation," *Robotics and Autonomous Systems (Special issue on Biomimetic Robotics)*, vol. 30, pp. 39–64, 2000.

[18] A. Berthoz, *The Brain's Sense of Movement*. Cambridge, MA: Harvard University Press, 2000.

[19] A. D. Redish and D. S. Touretzky, "Cognitive maps beyond the hippocampus," *Hippocampus*, vol. 7, no. 1, pp. 15–35, 1997.

[20] B. J. Kuipers, "A hierarchy of qualitative representations for space," in *Working Papers of the Tenth International Workshop on Qualitative Reasoning (QR-96)*, 1996.

[21] O. Trullier, S. Wiener, A. Berthoz, and J.-A. Meyer, "Biologically-based artificial navigation systems: Review and prospects," *Progress in Neurobiology*, vol. 51, pp. 483–544, 1997.

[22] M. Franz and H. Mallot, "Biomimetic robot navigation," *Robotics and Autonomous Systems*, vol. 30, pp. 133–153, 2000.

[23] B. J. Kuipers, "The spatial semantic hierarchy," *Artificial Intelligence*, vol. 119, no. 1–2, pp. 191–233, 2000.

[24] P. Bessière, E. Dedieu, O. Lebeltel, E. Mazer, and K. Mekhnacha, "Interprétation vs. description I : Proposition pour une théorie probabiliste des systèmes cognitifs sensori-moteurs," *Intellectica*, vol. 26-27, pp. 257–311, 1998.

[25] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, T. Dean, and C. Boutilier, "Hierarchical solution of Markov Decision Processes using Macro-actions," in *Proceedings of the 14th Conf. on Uncertainty in Artificial Intelligence (UAI-98)*, G. F. Cooper and S. Moral, Eds. San Francisco: Morgan Kaufmann, July, 24–26 1998, pp. 220–229.

[26] T. Lane and L. P. Kaelbling, "Toward hierarchical decomposition for planning in uncertain environments," in *Proceedings of the 2001 IJCAI Workshop on Planning under Uncertainty and Incomplete Information*. Seattle, WA: AAAI Press, August 2001.

[27] J. Pineau and S. Thrun, "An integrated approach to hierarchy and abstraction for POMDPs," Carnegie Mellon University, Technical Report CMU-RI-TR-02-21, August 2002.

[28] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[29] T. Lane and L. P. Kaelbling, "Nearly deterministic abstractions of markov decision processes," in *18th Nat. Conf. on Artificial Intelligence*, 2002.

[30] J.-C. Latombe, *Robot Motion Planning*. Boston: Kluwer Academic Publishers, 1991.

[31] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, 1996.

[32] E. Mazer, J.-M. Ahuactzin, and P. Bessière, "The ariadne's clew algorithm," *Journal of Artificial Intelligence Research (JAIR)*, vol. 9, pp. 295–31, 1998.

[33] P. Svestka and M. Overmars, "Probabilistic path planning," in *Lecture Notes in Control and Information Sciences 229*, J.-P. Laumond, Ed. Springer, 1998.

[34] H. Attias, "Planning by probabilistic inference," in *Ninth International Workshop on Artificial Intelligence and Statistics Proceedings*, 2003.

[35] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Hybrid simultaneous localization and map building: a natural integration of topological and metric," *Robotics and Autonomous Systems*, vol. 44, pp. 3–14, 2003.

[36] J. Diard, P. Bessière, and E. Mazer, "Hierarchies of probabilistic models of space for mobile robots: the bayesian map and the abstraction operator," in *Proceedings of Reasoning with Uncertainty in Robotics (IJCAI'03 Workshop)*, Acapulco, Mexico, August 2003.

[37] ——, "Combining probabilistic models of space for mobile robots: the bayesian map and the superposition operator," in *Proceedings of the 3rd IARP Int. Workshop on Service, Assistive and Personal Robots - Technical Challenges and Real World Application Perspectives*, M. Armada and P. González de Santos, Eds. Madrid, Spain: Industrial Automation Institute (IAI-CSIC) Spanish Council for Scientific Research, October 14–16 2003, pp. 65–72.