Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems.
Amsterdam, The Netherlands, March 2004 (to appear).

Some Notes on the Use of Hybrid Maps for Mobile Robots

Pär Buschka and Alessandro Saffiotti
Center for Applied Autonomous Sensor Systems
Dept. of Technology, University of Örebro, Sweden
{par.buschka,alessandro.saffiotti}@aass.oru.se

Abstract. Hybrid maps are quickly becoming popular in the field of mobile robotics. There is, however, little understanding of the general principles that can be used to combine different maps into a hybrid one, and to make these maps to cooperate. In this note, we propose a definition and a classification of hybrid maps, and discuss the synergies that can make a hybrid map something more than the sum of its parts. We illustrate these points with experimental results obtained on a metric-topological map.

1 Introduction

Autonomous robots typically rely on internal representations of the environment, or *maps*, to plan and execute their tasks. Several types of map have been proposed in the literature, and there is general consensus that different types have different advantages and limitations, and that each type is more suited to certain tasks and less to others [5, 14, 22]. For instance, metric maps allow the robot to compute optimal paths and to perform accurate localization. However, metric maps are hard to create and maintain due to the inaccuracies in the robot motion and sensing, involve heavy computational costs, and do not interface well with symbolic problem solvers and humans. By contrast, topological maps scale better to large environments and they interface more naturally with symbolic systems and humans. However, these maps only allow coarse localization and suboptimal path planning, and they make it difficult to distinguish between different places without using some metric information.

Because of these reasons, it is becoming common wisdom in the field of mobile robotics to use *hybrid* spatial representations that integrate several maps, usually of different types. The rationale for this is that by combining maps with different advantages and limitations we can gain the advantages from each component while eliminating some of its drawbacks. For instance, several authors have recently proposed hybrid metric-topological maps with the intent to combine the accuracy of metric maps with the scalability of topological maps [20, 7, 18, 13, 11, 22, 2].

When taking a closer look at the hybrid maps proposed in the literature, one notices that there are almost as many different ways to perform the “hybridization” as there are authors. Some authors propose a hierarchical structure [14, 17] while others rely on a more flat structure [23, 10]. Some build a topological map on the top of a metric one [5, 20, 12] while others do the reverse [11]. Some use an integrated representation including both topological and metric information [7, 10, 2] while others use clearly separated representations [18, 13].

Some let much information flow between different maps [22] while others favor rather isolated maps [1].

In face of this variety, there is unfortunately no systematic analysis of the different ways in which different maps can be combined, and how they can be made to cooperate. This foundational gap makes any hybrid map somehow *ad-hoc*, since there is no clear grounding for the choices made in its construction.

In this note, we report some reflections about hybrid maps with the aim to start filling this gap. In particular, we propose: (i) a definition of hybrid map; (ii) a classification of hybrid maps along the dimensions of heterogeneity, hierarchy, and separability; (iii) a list of the possible synergies that make a hybrid map something more than the sum of its parts; and (iv) a list of the new problems that a designer of a hybrid map should consider. To better illustrate these reflections, we present a specific example of a simple hybrid map developed in our research, and show experiments run on an iRobot Magellan Pro robot in an office environment.

2 Hybrid Maps

In order to study hybrid maps, we first need to fix some terminology. We therefore propose definitions for some basic notions related to robot maps. Our definitions are rather minimalistic. Their aim is to allow us to discuss more precisely the different varieties of hybrid maps and their inner mechanisms, without committing to any specific formalism.

Definition 1. *A* robot map *is any digital representation of the space.*

Definition 2. *A* metric map *is a triple $M = \langle S, X, pos \rangle$ where: S is a metric space equipped with a distance function d, $X = \{o_1, \ldots, o_n\}$ is a set of objects, and pos $: X \to S$ is a function that associates each object in M to an element of S (or to a subset of S).*

Thus, a metric map is simply a collection of objects together with their position (or extent) is some n-dimensional metric space. (See, e.g., [19] for a definition of mathematical metric spaces.) Note that these objects do not need to represent physical objects. For instance, the above definition captures both feature maps [9] and grid maps [15] by letting the objects o_i represent features or cells, respectively.

Definition 3. *A* topological map *is a pair $T = \langle V, E \rangle$ where: $V = \{v_1, \ldots, v_n\}$ is a set of nodes, and $E = \{e_1, \ldots, e_m\} \subseteq V \times V$ is a set of edges.*

This definition does not relate topological maps to topological spaces, as one could expect, but to graphs. This is because most topological maps found in the literature are simply regarded as graphs.

Maps as defined here are mere representations of the space. In order to be used inside mobile robots, they are usually augmented with at least two additional features.[1] First, a *reference system*, that allows the robot to uniquely refer to map elements, e.g., for communicating map information. Typical reference systems include a Cartesian frame in the case of metric maps, and a set of labels in the case of topological maps. Second, a notion of *state*,

[1] In most practical applications, the map is also augmented with a representation of the uncertainty associated with its information. This issue, however, is not relevant to the present discussion.

usually consisting of the current position of the robot in the map: this can be an element of the metric space S in the case of metric maps, or a node in V in the case of topological maps.

It is interesting to better specify the notion of a map being "useful" to a robot for a given task. Consider a robot r, described in terms of its motion and sensing capabilities. In general, we are interested in maps that contain the right information to allow r to localize in them.

Definition 4. *A map M is* localizable *for a robot r if there is a function Loc(t) that estimates the position of r in M at any point in time t.*

This definition is intentionally vague. One could give a more specific definition by choosing a specific formalism to describe the robot and the map, e.g., as a dynamic system, but this would reduce generality. The pre-theoretic definition above shall suffice for our purposes.

Another interesting usability property for a map is that it contains the right information to decide how the robot can move, e.g., information about the geometry of free space.

Definition 5. *A map M is* traversable *for a robot r if there is a predicate Conn(x, y) that, for any x, y elements of M, tells us if there is a sequence of actions for r to go from x to y.*

The Conn(x, y) predicate often computes the connecting path as a side effect.

Two things should be noted here. First, being localizable and traversable are not properties of a map, but rather of the relation between a map and a robot. Intuitively, a localizable map contains the right information to allow a given robot to answer the question "Where am I?", and a traversable map contains the right information to answer the question "How do I go from A to B?". Second, if a map is localizable (or traversable), then it is so on a given spatial *extent* and at a given *resolution*. There is a complexity tradeoff between extent and resolution: for instance, topological maps typically cover a larger extent than metric maps, but at the expense of a lower resolution. As we shall see, hybrid maps offer one way to improve this tradeoff.

We are now in a position to define the notion of a hybrid map.

Definition 6. *A* hybrid map, *or H-map, is a pair $H = \langle \mathcal{M}, \mathcal{C} \rangle$ where: $\mathcal{M} = \{M_1, \ldots, M_n\}$ is a set of maps, and $\mathcal{C} = \{c_1, \ldots, c_p\}$ is a set of links. Each link is a pair $\langle m_i, m_j \rangle$, where m_i is an object of M_i and m_j is an object of M_j, with $i \neq j$.*

Intuitively, a hybrid map is a representation of space that combines two or more distinct maps of different types. We call the constituent maps *components* in order to distinguish them from the H-map itself. We also use the abbreviations M-component and T-component to refer to a metric and a topological component of a H-map, respectively.

The links in a H-map establish the correspondence between its components by binding some of their elements. Without this connectivity information, the H-map would simply be a storage for independent maps, and most of the advantages of the hybridization would be lost.

A simple example of H-map is $H_1 = \langle \{M, T\}, \mathcal{C} \rangle$ where: M is a global metric map, T is a topological map covering the same space, and $\mathcal{C}$ connects each node in T to a point in M. Several mobile robots use a similar setup, e.g., [20, 17]. Typically, the robot plans a path on T, project this path into M by exploiting the links in $\mathcal{C}$, and uses M for path following and self-localization. Another example is $H_2 = \langle \{G, G_1, \ldots, G_n\}, \mathcal{C} \rangle$ where: G is a coarse grid map with n cells, $G_1, \ldots, G_n$ are finer grid maps that constitute a tessellation of the environment, and $\mathcal{C}$ connects each cell in G to a corresponding G_k. This map can be used, for instance, to perform hierarchical self-localization. Similar setups are found, e.g., in [16, 24].

3 Inside a Hybrid Map

A hybrid map as defined above is simply a collection of maps together with the information needed to register them. Multi-resolution mosaic grid maps like H_2, or systems with two independent maps used for two independent tasks like H_1, would all fit this definition. Intuitively, however, we feel that a "real" hybrid map should be more than this. The purpose of this section is to investigate what is in this *more*.

3.1 Dimensions of hybridization

We classify H-maps along three different dimensions, which depend on the types of their components and on how these components are integrated.

Heterogeneity. A H-map is *heterogeneous* if at least two of its components are of essentially different types, and it is *homogeneous* if all its components are of similar types. The H_1 map discussed above is an example of a heterogeneous H-map, while H_2 provides an example of a homogeneous H-map.

Hierarchy. A H-map is *hierarchical* if its components are hierarchically ordered, and it is *flat* otherwise. Two maps are hierarchically ordered if one represents the same environment as the other, wholly or partially, at a lower resolution or a higher abstraction level. An example of hierarchical H-map is the H_2 map above. An example of flat H-map is one consisting of an occupancy grid used for obstacle avoidance, plus a feature map used for self-localization, both covering the same space.

Separability. A H-map is *separable* if each component can be used independently from the other components, and it is *integrated* if each component needs the other ones in order to function. Consider for example the H_1 map above. If the nodes in T have an associated signature, then the T-component can have its own localization mechanism and the H-map is separable. If the nodes in T do not have a signature, then the T-component must rely on the M-component to determine the current node, hence the H-map is not separable.

In practice, each one of these properties is a matter of degrees. Accordingly, we can place any H-map in a 3D continuous space that has these dimensions as axes, as in Fig. 1. For example, the multi-layer maps in [24, 6] consist of a set of maps of the same type but at different resolutions: these are hierarchical and separable, but homogeneous. [10] proposes a topological map enriched with local metric information: this is heterogeneous but it is not fully hierarchical, since the metric and topological components have comparable resolutions, and it is not separable, since the metric component cannot be used as a metric map by itself. The hybrid feature-topological map in [1] is heterogeneous and hierarchical, but it is not separable since the T-component relies entirely on the M-component for node detection. Finally, several authors have proposed hybrid maps consisting of a set of small metric maps connected by a topological map [7, 13, 22, 2]. These are both heterogeneous and hierarchical, and the maps in [13, 22] are also separable since the T-component has its own independent localization and/or planning process. The simple H-map described in Sect. 4 below is also of this type.

In the rest of this note, we shall require that H-maps are heterogeneous, since this property is somehow implicit in calling a map "hybrid". The other two properties are also desirable in

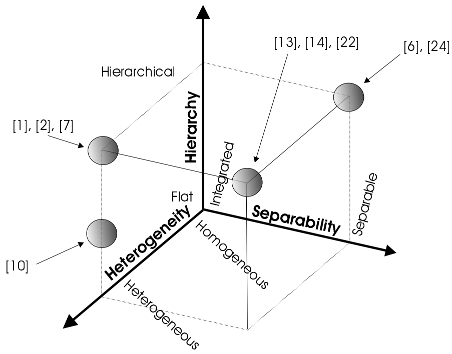

Figure 1: Dimensions of hybridization, indicating the place of some of the H-maps in the literature.

H-maps: hierarchy improves the complementarity between the components, and separability guarantees that each component is able to produce some independent information that can be used by another other one, therefore enabling synergies.

3.2 Synergies

The most obvious reason for using a H-map is the ability to exploit the individual advantages of each one of its component maps. There can be even more to gain, though, by exploiting the interaction between these components. Exchanging and combining information between the components can result in a number of *synergies*, which in turn can increase performance to a level that would be hard or impossible to achieve using one single component. Note that two components can have a synergy between them but still be separable: synergy simply means that a component is able to use information from another one to improve its own performance — although it could still operate without that.

In the following, we discuss some of the synergies that may occur in H-maps. Most of the examples concern topological-metric maps, since these are currently the most popular H-maps, but similar synergies can be exploited between other types of components as well.

Building. During the building of a H-map, one component can provide information used to build another one. Let M_1, M_2 be these components. We distinguish two cases.

Sequential. We first build M_1 in its entirety, and then extract information from it to build M_2. The problem of getting a consistent map must be solved when building M_1. A synergetic effect may occur since we can rely on the correctness of M_1 when building M_2. For example, [11] first build a consistent T-component and then extract a metric map from it. The synergetic gain is that global consistency in the T-component is achieved at a low computational cost using a relaxation algorithm; this consistency is then reflected in the metric component. Other authors first build the M-component, and then extract a T-component from it [20, 12, 17]. The synergetic gain here is that "closing the loop" can be achieved more reliably in metric maps.

Concurrent. M_1 and M_2 are built simultaneously, and each component provides information which is used in building the other one. For instance, in [21] the authors first build a global metric map, then they extract a T-map from it and solve the global consistency problem on this T-map, and finally they fine-tune the M-map to maximize global consistency. The synergetic gain here comes from combining the lower computational complexity of the T-map with the higher resolution of the M-map.

Localizing. If two components M_1, M_2 of the H-map are localizable, then the output of the $Loc()$ function in M_1 can be used as input for the $Loc()$ function in M_2. Usually, this output is seen as a sort of a virtual sensor by the latter $Loc()$ function.

This type of synergy can be observed in several of the metric-topological maps reported in the literature, in which each node in the T-component is associated to an M-component representing a small part of the environment [8, 13, 1]. In these maps, M-localization is used to detect that the robot has passed a "gateway" and hence to signal a node transition to the T-component. In the opposite direction, self-localization in the T-component can be used to select the current M-component, and possibly an entrance point in it [18, 22].

Planning. If a H-map is hierarchical, then task- and path-planning can be made on a component at a higher abstraction level, and the result can be projected, via the $\mathcal{C}$ links, to a component at a lower level for navigation. The advantage is that planning at higher levels of abstraction is less complex, and it can profit from symbolic information. This is the approach taken in most of the metric-topological hybrids in the literature. For example, [13] rely on a knowledge-based planner to generate a sequence of behaviors on a topological map, and use these behaviors as guidelines for reactive navigation using local metric maps. Synergy between different abstraction levels can also be used for exploration planning, like in [17].

Focusing. A general feature of hierarchical H-maps is that we can use the state information (Loc) of a component at a higher abstraction level to focus the attention on one (or some) of the components at a lower level. Focusing can make tasks like localization and map building more computationally tractable. It can also make object recognition and data association less complex, since we can concentrate on the objects which are in the current focus.

An example of focusing can be found in multi-resolution grid maps [6, 17], although these maps are not heterogeneous. In metric-topological H-maps where each node in the T-map is associated to a separate M-map, topological localization allows the robot to focus on a single M-map, and to perform position tracking only on that map [13, 22, 1], greatly improving the scalability of the metric localization algorithms used.

3.3 Challenges of H-maps

While H-maps may offer new advantages in addition to the ones offered by single maps, they also present some new problems that do not arise in single maps. An obvious problem is the increased *complexity* introduced by the synergies. For instance, if the location output from one component is used in another one as a virtual sensor, then we must define a sensor model for this input, which might not be trivial. If there are several flows of location information, care must be taken to avoid circular dependencies.

Another problem is the possibility of unbounded *error propagation* across components. Consider for instance a H-map in which a T-component is used to select one of many local metric maps. If T-localization is one node off, then metric localization is attempted in the wrong patch, which may result in a catastrophic failure, as reported, e.g., by [13, 22].

A related problem is what we call the problem of *location seeding*. Consider a H-map in which different maps (of whatever type) are used to cover different parts of the environment. Suppose that the robot exits one map and enters another one. How should its location (state) in that map be initialized? Depending on how this problem is addressed, the resulting H-map can have different characteristics in terms of robustness and flexibility.

4 A Simple Example

In order to illustrate the notions discussed in these notes, and to show how our framework is connected to a real world scenario, we describe a simple H-map developed in our work, and point out how this H-map realizes some of the above synergies. An extended description of this system is available in [3].

4.1 Representation

Our H-map can be described as a pair $H = \langle \mathcal{M}, \mathcal{C} \rangle$ where:

$\mathcal{M}$ $= \{T, M_1, \ldots, M_n\}$, where T is a T-map, in which nodes represent open spaces in the environment (rooms and corridors) and edges represent passages between these spaces (doors and junctions); and $M_1, \ldots, M_n$ are M-maps, in the form of occupancy grids, each covering the area of one open space.

$\mathcal{C}$ is a set of links that connect each node k in T with one M_k, and each edge (k, h) in T with a pair of positions in M_k and M_h corresponding to a passage between two spaces.

Intuitively, the T-component gives the topological structure of the environment, while the M-components detail the metric structure of each subspace in it. Fig. 2 shows a sample H-map for an office environment. The size of the environment is about 12×36 meters.

4.2 Building

We start with a H-map with an empty T-component T and no M-components. We then add nodes and edges to T. Each time we add a new node, we also add a new M-component to $\mathcal{M}$, together with the corresponding links in $\mathcal{C}$.

The T- and M-components are built concurrently. First, we detect open spaces (rooms and corridors) in the neighborhood of the robot. To do so, we collect sensor data in a small grid centered on the robot, and apply a morphological analysis to extract a local topology from it, as described in [4]. Second, we match this local topology to the current T. As a result, new nodes and edges can be added to T. We associate a signature to these nodes to enable place recognition, consisting in a bounding box that represents the extent of that space. Third, for each new node in T we build an M-component and add it to the H-map. This is an occupancy grid built using the measurements that were taken inside that node. Finally, we extract information about the position of the doors, and add the corresponding links to the H-map. The process is repeated until the entire area is explored. The H-map shown in Fig. 2 was built in this way using an iRobot Magellan Pro equipped with sonar sensors.

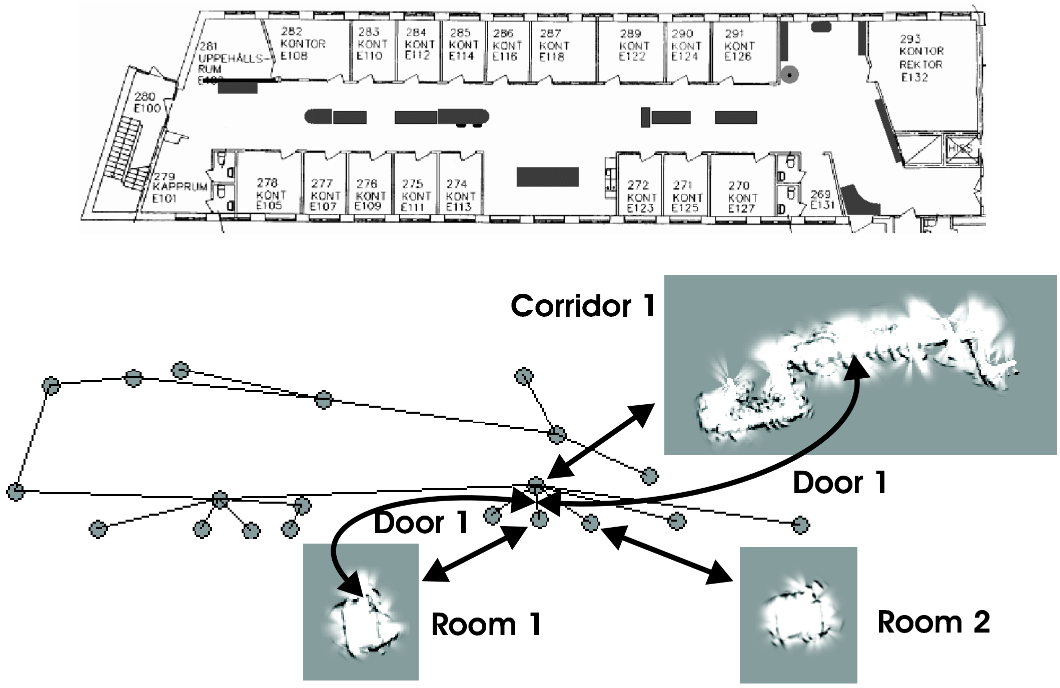

Figure 2: A topological-metric hybrid map built automatically out of data recorded in an ordinary office space. On top is a CAD-like map of the environment. Below is the topological component. Three of the metric components are shown along with some of the links connecting them to the nodes and the door areas to the edges.

4.3 Localizing

The robot's location in our H-map is given by a pair (k, x), where k is a node in T and x is a position in the corresponding M-component M_k. Metric localization inside one M-component is easy since this component is small, and can be done using standard methods like EKF or MCL. In our experiments, we simply rely on odometry for local metric localization.

Topological localization relies on a door transition detector based on the method for node detection described above. When a door transition is detected, we use the metric information in the M-component to find what door is passed. This door is connected to an edge in the T-component, which allows us to determine which node the robot entered. We then switch focus on the corresponding M-component, and restart metric localization in that map. Here we face the location seeding problem. In our implementation we use the observed position and orientation of the door, computed by the topological door detector, to set the robot's pose in the M-component.

4.4 Properties and synergies

With respect to the dimensions defined above, our H-map can be classified as *heterogeneous* since the T- and M-components are fundamentally different: objects in T represent architectural elements, while objects in each M_i represent small cells of space. Our H-map is clearly *hierarchical*, since each M_i gives a fine-grained description of the element represented by a single node in T. Finally, both the T-component and each one of the M-components have their own processes for building and localization, and they could be used as separate maps.

Hence, our H-map is *separable*. This makes our H-map a full-fledged hybrid map.

Our H-map exploits several synergies between the T- and M-components. During building, the node detection process in the T-component provides a decomposition of the space into areas which are suitable for building local metric maps using odometry. It also identifies the measurements needed to build each M-component. Finally, the T-component detects transitions from one local map to the next, and takes care of dealing with small loops.

During localization, location information from the M-component is used to help the Loc() function in T by determining which edge was traversed when the robot exits a node. In exchange, location information from the T-component is used to focus the metric localization on a single M-component, and to seed the metric localization when we detect an edge traversal as discussed above.

Although simple, this example shows in which sense a hybrid map can be more than the mere sum of its parts. Our T- and M-components could be used separately, but they would have serious limitations. The M-components would only have a small *extent*, since our naive mapping and localization algorithms based on odometry would not scale up to larger environments, and the T-component would have coarse *resolution*, since its basic elements are rooms and corridors and we cannot discriminate positions inside these elements. By contrast, the combined H-map can cover the larger extent provided by the T-component at the finer resolution provided by the M-components.

5 Conclusions

Hybrid maps are likely to become the dominant paradigm for representing spatial information in autonomous robots. Still, the term "hybrid map" is used today to denote very different systems, and with little understanding of the principles underlying map combinations. We hope that these notes will help to start filling this important foundational gap.

Although the simple H-map described here was mainly intended for illustration purposes, it incorporates the main advantage of an hybrid map: it covers a larger extent of space, and with better resolution, than what would be individually allowed by any one of its component maps. Current work to improve this H-map include the development of a hybrid SLAM algorithm, and practical testing in mixed indoor/outdoor environments.

Acknowledgements

This work greatly benefit from discussions with Tom Duckett. Financial support was provided in part by the Swedish KK Foundation.

References

[1] T. Bailey and E. Nebot. Localisation in large-scale environments. *Robotics and Autonomous Systems*, 37:261–281, 2001.

[2] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2003.

[3] P. Buschka. An investigation of hybrid maps for robot localization and navigation. Licentiate thesis, Department of Technology, Örebro University, Örebro, Sweden, 2004.

[4] P. Buschka and A. Saffiotti. A virtual sensor for room detection. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 637–642, Lausanne, CH, 2002.

[5] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 138–145, 1985.

[6] D. Chen, R. Szczerba, and J. Uhran. A framed-quadtree approach for determining Eucledian shortest paths in a 2D environment. *IEEE Trans. on Robotics and Automation*, 13(5):668–680, 1997.

[7] K. S. Chong and L. Kleeman. Large scale sonarray mapping using multiple connected local maps. In *Int. Conf. on Field and Service Robotics*, pages 538–545, 1997.

[8] C. Congdon, M. Huber, D. Kortenkamp, K. Konolige, K. Myers, E. H. Ruspini, and A. Saffiotti. Carmel vs. Flakey: A comparison of two winners. *AI Magazine*, 14(1):49–57, Spring 1993.

[9] M. Drumheller. Mobile robot localization using sonar. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:325–331, 1987.

[10] T. Duckett, S. Marsland, and J. Shapiro. Fast, on-line learning of globally consistent maps. *Autonomous Robots*, 12(3):287–300, 2002.

[11] T. Duckett and A. Saffiotti. Building globally consistent gridmaps from topologies. In *Proc. SYROCO'00, 6th Int. IFAC Symposium on Robot Control*. Elsevier, 2000.

[12] E. Fabrizi and A. Saffiotti. Extracting topology-based maps from gridmaps. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2972–2978, San Francisco, CA, 2000.

[13] J. Gasós and A. Saffiotti. Integrating fuzzy geometric maps and topological maps for robot navigation. In *Proc. of the 3rd Int. Symposium on Soft Computing (SOCO)*, Genova, IT, 1999.

[14] B. Kuipers. The spatial semantic hierarchy. *Artificial Intelligence*, 119:191–233, 2000.

[15] H.P. Moravec and A.E. Elfes. High resolution maps from wide angle sonar. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA*, pages 116–121, 1985.

[16] N. J. Nilsson. A mobile automation: An application of artificial intelligence techniques. In *Proc. of the 1st Int. Joint Conf. on Artificial Intelligence*, pages 509–520, Washington, D.C., USA, 1969.

[17] A. Poncela, E.J. Prez, A. Bandera, C. Urdiales, and F. Sandoval. Efficient integration of metric and topological maps for directed exploration of unknow environments. *Robotics and Autonomous Systems*, 41(1):23–41, 2002.

[18] S. Simhon and G. Dudek. A global topological map formed by local metric maps. In *Proc. of the Int. Conf. on Intelligent Robots and Systems*, Victoria, Canada, 1998.

[19] W.A. Sutherland. *Introduction to Metric and Topological Spaces*. Oxford Science Publications, 1975.

[20] S. Thrun and A. Bücken. Integrating grid-based and topological maps for mobile robot navigation. In *Proc. of the AAAI 13th National Conf. on Artificial Intelligence*, Portland, Oregon, 1996.

[21] S. Thrun, J.S. Gutmann, D. Fox, W. Burgard, and B. Kuipers. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In *Proc. of AAAI*, pages 989–995, 1998.

[22] N. Tomatis. *Hybrid, Metric-Topological, Mobile Robot Navigation*. PhD thesis, École Polytechnique Fédérale de Lausanne, Switzerland, 2002.

[23] B. Yamauchi and R. Beer. Spatial learning for navigation in dynamic environments. *IEEE Trans. on Systems, Man, and Cybernetics*, 26(3):496–505, 1996.

[24] A. Zelinsky. A mobile robot exploration algorithm. *IEEE Trans. of Robotics and Automation*, 8(6):707–717, 1992.