The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
15-17 September 2011, Prague, Czech Republic

Hybrid Mapping for Autonomous Mobile Robot Exploration

Matías Nitsche[1] , Pablo de Cristóforis[1] , Miroslav Kulich[2] , Karel Košnar[2]
[1] Departamento de Computacion, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires
mnitsche@dc.uba.ar, pdecris@dc.uba.ar
[2] Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague
kulich@labe.felk.cvut.cz, kosnar@labe.felk.cvut.cz

***Abstract* – Hybrid maps combining several approaches to store robot's interpretation about its working environments are getting popular nowadays. The paper deals with a novel approach to hybrid maps that is based on fixed-size interconnected occupancy grids organized in a topological graph. The presented mapping approach is employed in the exploration scenario, where the map is built from scratch and used for both local and global path-planning, and goal selection. Feasibility of the approach has been validated by a set of experiments in the Player/Stage system [1].**

***Keywords* – Mobile robots, robot exploration, mapping**

I. Introduction

Nowadays mobile robots solve complex tasks in many application areas. In order to provide autonomous behaviour, robots build their own representation about the operating environment, i.e. a map. The type and quality of the map typically depends on the solved task and the used sensors. Many researchers use *metric maps* that are built directly from sensor data and that describe geometric properties of the environment [2]–[4]. These are easy to create, and suitable for tasks in small environments (path planning, collision avoidance, position correction). However, their use in large environments is questionable due to space and computation complexity.

By contrast, *topological maps* are based on recording topological relationships between observed features rather than their absolute positions [5], [6]. The resulting representation takes the form of a graph where the nodes represent the observed features and the edges represent their relationship. While this approach allows to integrate large area maps without suffering from the accumulated position error and can be naturally used for robot-human interaction, it is not suitable for precise localization and exact reconstruction of the environment.

The numerous works combine metric and topological approaches into *hybrid maps* (see [7] for an overview and taxonomy). These approaches differ in the form the particular maps are built, interconnected, and used. For example, the topological layer is built on top of grid-based map by processing Voronoi diagrams of the thresholded grid [8]. In [9], a statistical maximum likelihood approach is introduced, where a topological map solves global position alignment problem and it is consequently used for building a fine-grained map. The map representation in the SLAM framework Atlas [10] consists of a graph of multiple local maps of a limited size. Each vertex represents a local coordinate frame of reference and each edge denotes the transformation between local frames as a Gaussian random variable. Youngblood [11] proposes a place-centric occupancy grid model, where each area or room of the environment is associated with a single occupancy grid. A global graph interconnecting these grids is used to represent the connectivity between rooms. The extent of a room is determined by retrieving a locally confined area followed by gateway extraction.

We present a novel mapping approach that is utilized for exploration of a-priori unknown environment. This approach uses fixed-size interconnected occupancy grids, instead of associating one grid to each room. Since this grid decomposition does not intend to represent the topology of the environment, a separate graph is used for this end. In contrast to Youngblood's approach, our method does not rely on assumptions of the structure of walls and enclosed areas for gateway and topology extraction. Furthermore, with a fixed-size grid approach, large rooms will not generate single large grids, negatively impacting over path-finding exploration.

The exploration algorithm used is goal-biased following Yamauchi's frontier-based approach [12]: the robot is directed into undiscovered areas by performing a search for unexplored nodes over the topological map, and for frontiers (regions between explored and non-explored areas) in a specific occupancy grid. The local path obtained is fed to a motion planner module, which takes care of avoiding obstacles while trying to approach the target.

Path-planning is performed first at the global level using the topological graph. Local grid-based path over each occupancy grid is computed along the way. This *lazy* path-planning approach implies that any changes detected at the metric level, which may have an impact at the topological level, can be handled without problems. Also, in such case, path-planning computation time is reduced

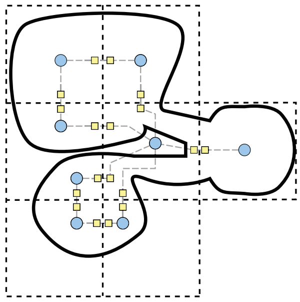


Figure 1: Representation of a three-room unstructured environment, with its corresponding topological map overlaid. Area nodes are represented with circles and gateway nodes with rectangles.

since the full pre-planned metric path would need to be discarded in a *non-lazy* approach.

The rest of the paper is organized as follows. The proposed method is introduced in the next section, experimental results are presented in section III, while section IV concludes the paper.

II. The Method

A. Map Structure Overview

In this hybrid mapping method, the notion of *map* has both a metric and a topological aspect.

On the metric side, the environment is described by a series of interconnected fixed-size occupancy grids. Given that each grid is essentially a matrix of real values representing the occupancy probability over a given cell, a complete map can be imagined as if a single global occupancy grid was used. In this sense, the proposed implementation includes the notion of a *super grid*: a module that maintains and instantiates on demand the necessary occupancy grids in order to map a given portion of the environment. This construct becomes important when applying sensor information obtained from the laser range-finders (this is dealt in sub-section II-B).

On the other hand, the topological aspect takes the form of a graph with two types of nodes: (a) *area nodes*: representing distinct topological areas inside a given occupancy grid and (b) *gateway nodes*: representing the topological view of metric gateways. A metric gateway is defined as a set of contiguous free cells (their occupancy probability is below a certain threshold) that lay over any of the four edges of an occupancy grid (gateway detection and handling is dealt in sub-section II-C).

A conceptual example of a map produced by our method can be seen in Figure 1, for a scenario of three unstructured rooms. Notice that all grids are the same size and one room can hold one or more grids.

B. Sensor Information Application

In order to update the occupancy probabilities of the metric map, the sensor information provided by the laser range-finder is used. This information is applied to a robot-centered window (a grid) first, by using Bayesian filtering, and is applied to the map later on. The purpose of this window is to not be limited by the fixed-size decomposition of the metric map, since the grids could generally be smaller than the maximum laser range. In contrast, the window will be in general several times larger than a single occupancy grid. This implies that applying the values stored in the window onto the map involves updating several occupancy grids.

This update is simplified by using the *super grid* structure previously introduced, by treating the metric map as a single large occupancy grid: the offset of the window is first computed with respect to this global map and then the sensor information is applied cell-by-cell. In this way, the complexity of updating specific map cells of the affected grids is transparently handled by the *super grid* module by translating global cell coordinates to local cell coordinates of the corresponding occupancy grid.

C. Topological Information Extraction

This step consists of updating the global graph, by adding or deleting nodes and edges. These updates are performed when the robot travels between adjacent occupancy grids (both the old and new grids are updated) or when exploration over currently occupied area is finished and another exploration candidate needs to be found.

The first part of this update consists of gateway detection at the metric level. This implies constructing a list of contiguous free cells for each edge of the occupancy grid. A cell is to be considered free if its occupancy probability is below a given threshold. Furthermore, this condition is extended to neighboring cells in the direction perpendicular to the corresponding edge, up to a predefined distance (this involves both cells of the current grid and of the adjacent one). This lookahead (in both directions) is performed for each cell of the grid's edges. The purpose of this search is to only consider an edge-cell as being *free* if there's enough clearance on both sides. This solves a possible problem that may occur when a robot attempts to cross a detected gateway but a wall lays just in the adjacent grid. Since path-planning is performed only on the local grid, this wall would not be considered. This situation wouldn't result in a collision due to the usage of motion planner module, but it will nevertheless be an untraversable gateway that is not modelled correctly.

The second part of the topology extraction consists in associating each metric gateway to a gateway node in the topological graph. If there was no prior topological information for this grid, new nodes will be created. Otherwise, gateway detection is actually invoked by a grid update when new sensory information is gathered (which can reveal new gateways or extend existing ones). In the latter case, existing gateway nodes are updated to reflect new gateway sizes, if necessary.

Third, it is necessary to determine the connectivity between areas and gateway nodes. Gateway nodes must be associated to area nodes to reflect the fact that a gateway detected at the metric level is actually in the same topological area (for example, a room) as the robot currently is. To achieve this, local path-finding is performed between the robot's current position and each detected gateway. When a path exists, a connection between the *current* area node and said gateway is created. Else, a connection is deleted if it previously existed (to correct a wrongly detected connectivity, if a dividing wall wasn't previously seen by local exploration).

Finally, the last step of topological information extraction consists in determining the equivalence of topological areas. Whenever a robot enters a new grid, the corresponding gateway node associated to the entrance used to reach the grid must be connected to the correct area node for accurate topological description of the area. The method used is as follows: (a) a new area node is created, marked as *current* and linked to the gateway node used for entrance, (b) local path-finding is performed between the current robot position and all gateway nodes associated with the grid (c) if one of these gateways is found to be reachable and not connected to any other previously existing area node, a link is created between the current area node and said gateway (d) otherwise, if the gateway is reachable but already linked to other area node, both area nodes are *merged* by producing a single node that includes both edges. An example scenario of a node merging situation is depicted in Figure 2.

It is worth noting that the presented method implies that two gateway nodes will exist for a given passage between adjacent grids, since one gateway node exists for each of the two corresponding edges of the grids. While this may seem redundant, a gateway node not connected to the corresponding gateway node on the adjacent grid serves to model the case of a gateway that may lead to an unexplored area.

In order to reflect the traversability of a passage, the two corresponding gateway nodes should be linked together. To detect if a passage is traversable, it is not practical to require the robot to physically attempt the move. Instead, a heuristic is used: if the a metric gateway is of the same size and location over the edge (withing a threshold) than the metric gateway of the adjacent grid, they are considered traversable and the gateway nodes are linked.

D. Exploration

Exploration in the presented system is divided into two levels: local exploration over the occupancy grid and global exploration over the topological map. Figure 3 shows the Finite State Machine description of the exploration strategy.

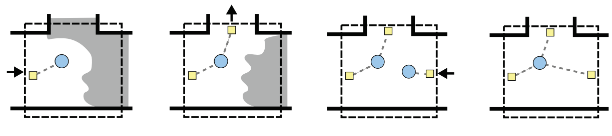


Figure 2: Area node merging example: a) The robot enters a new grid, and performs gateway detection, detecting only the Norh gateway since this grid is not fully explored. Local exploration starts (gray shaded coloring represents unexplored area). b) During local exploration the robot unintentionally exits the grid through the North gateway, leaving the grid not fully explored. While crossing to the adjacent grid, gateways are re-detected and a new gateway node is added and connected to previous area node. c) Later on, the robot returns to the same grid through a different gateway. As in the first step, gateway detection is performed and the corresponded new gateway node is added. A new area node is also added, connected to the gateway and marked as current d) When paths to other existing gateways are searched, a reachable gateway is found that is already connected to an existing area node. Thus, the existing area node is merged with the current one.

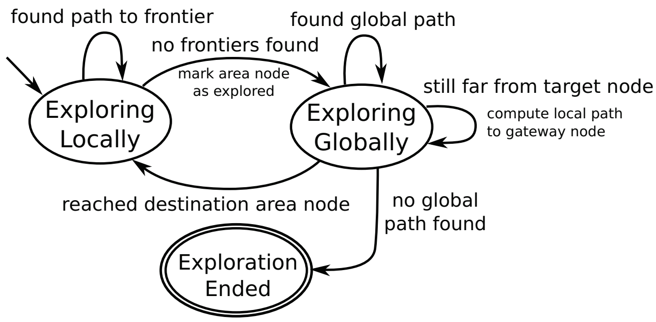


Figure 3: Finite State Machine description of the exploration strategy.

E. Local Exploration

Local exploration consists of searching for frontiers over the current occupancy grid. To determine these frontiers, cells that lay between unoccupied (occupancy probability less than certain threshold) and unexplored cells (a log-odds value of 0) are identified (frontier cells) at first. Then, contiguous frontier-cells are grouped into sets, forming actual frontiers. Finally, from each of these frontiers a small number of representative cells (in our case, best results were obtained with just one representative) are extracted using K-means clustering. Local exploration is then performed by searching a path from the robot current position to these representatives. When no more frontiers are detected, global exploration starts.

In order to optimize the movement of the robot, a cost function is defined that is used by the path planning module to find the most appropriate path. This function penalizes zigzag paths and the presence of occupied cells near the target cell or the proximity to the grid's edges. This results in a mostly straight path that avoids bringing the robot near obstacles or grid edges (minimizes unintentional grid exiting).

F. Global Exploration

During global exploration, the topological graph is used to find a candidate area for exploration. This can be either a non-completely explored area node (where frontiers still exist), or unexplored gateways (not yet associated to an area node, which includes the situation where a new grid is to be discovered). To find this candidate and the path leading to it, the path-finding module is executed over the global graph. To physically follow this global path, a local metric path between gateways is computed for the currently occupied grid along the way.

As with the local exploration method, global path-finding is optimized by giving higher priority to unexplored gateways than to explored ones. Among two explored gateways, the closest one to the robot position is chosen (simply computed as the direct euclidean distance).

Finally, when global exploration finds that no remaining unexplored candidate areas are found, the exploration method as a whole is ended.

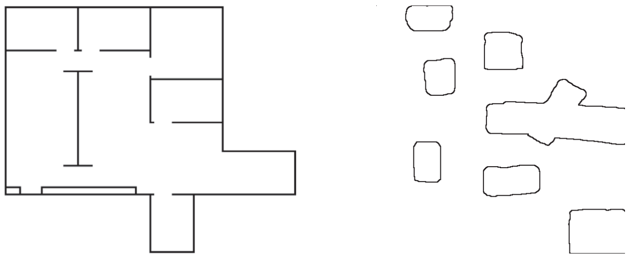
III. Experiments

The method has been implemented in C++ as a client program for the Player/Stage environment (version 3.0). The simulated robot consists of a Pioneer 2DX model equipped with a SICK LMS200 laser sensor (with 180° field of view). The occupancy grids have a cell-size of 7 centimeters, and grids are 49×49 cells in size. The laser sensor range was limited to 4 meters (out of the total 8 meters the sensor is capable of) in order to promote local robot exploration.

For motion planning the VFH+ [13] algorithm was implemented. The VFH+ Player/Stage driver was not used due to stability problems and lack of control of internal parameters. For path-finding, Dijkstra's shortest path algorithm was used, both for local and global exploration (i.e.: over the metric grids and over the global graph).

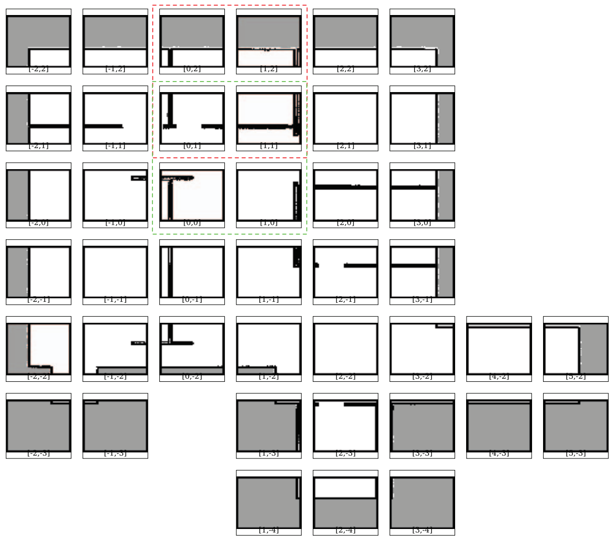
Tests were performed using the Player/Stage robotics simulation software, under two separate simulated environments (Fig. 4): an office-like scenario (with hallways and closed rooms) and an open area with arbitrary shaped obstacles. In both cases the complete environment is of 25×20 meters in size. After the robot completed the exploration, metric and topological maps were obtained. These are presented in Figures 5 and 6.

Due to the complexity of the topological maps, an extract (a subgraph) is presented in Figure 7. These two subgraphs show the main distinction between a metric area (for example: the grid of coordinates $[1, 1]$ of the *autolab* environment) and the topological areas it can contain. In both Figures 7a and 7b a different instance of a node with coordinates $[1, 1]$ is seen. This is due to the

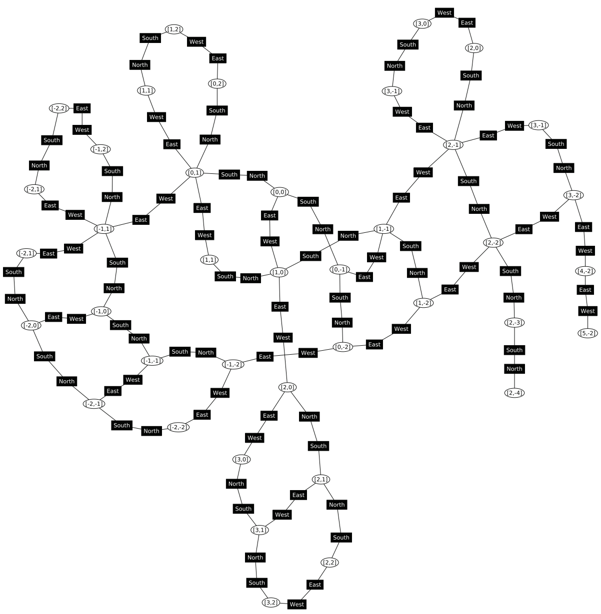


(a) Autolab **(b)** Cave

Figure 4: Testing environments.

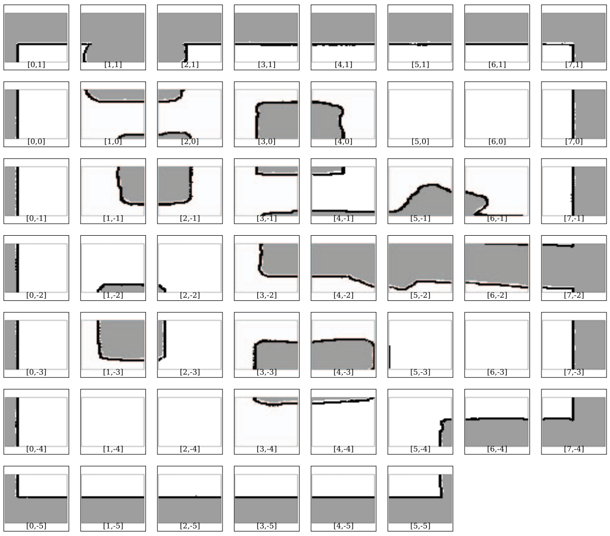


(a) Metric Map

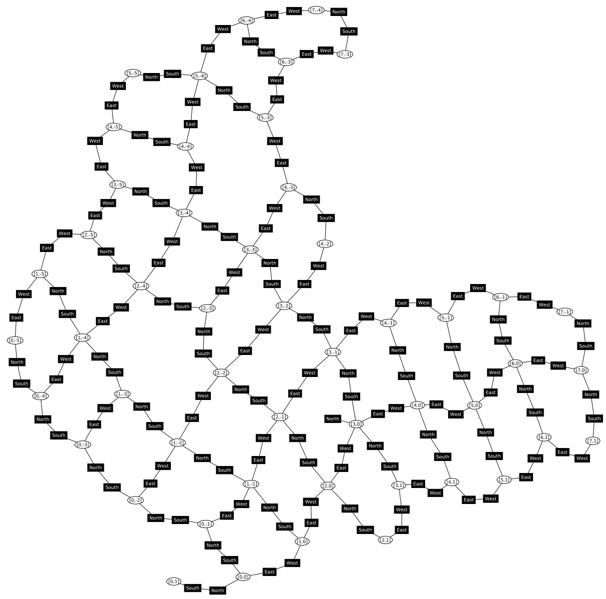


(b) Topological Map

Figure 5: Maps produced by the proposed method for the *autolab* scenario

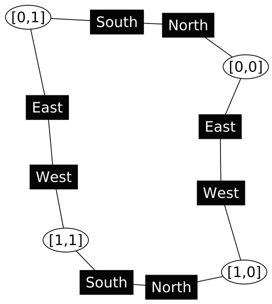


(a) Metric Map

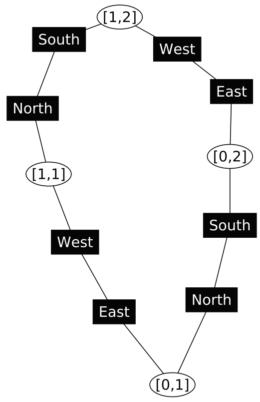


(b) Topological Map

Figure 6: Maps produced by the proposed method for the *caves* scenario



(a) Corresponding topological description of green dashed area from 5



(b) Corresponding topological description of red dashed area from 5

Figure 7: Example of the two distinct topological areas associated to the same metric grid of coordinates $[1, 1]$ from Figure 5 . Edges leading to other nodes have been ommited.

presence of distinct topological areas in the corresponding metric grid. Looking this grid in detail (metric map of figure 5), it can be seen that a first topological area corresponds to the one connected to the adjacent grids of coordinates $[0, 2]$, $[1, 2]$ and $[0, 1]$. On the other hand, a second topological area can be identified, which is connected to the corresponding areas of grids $[0, 1]$, $[1, 0]$ and $[0, 0]$.

IV. Conclusion

The experiments presented show the feasability of the proposed approach. All maps were produced as expected, correctly modelling the environment according to the method.

Having obtained a topological map (associated with a metric map) permits both detailed local and low-cost (in terms of path-finding cost) global navigation. Local grids are only instantiated were metric detail is required. Fixed-size grid provide a bound in terms of path-finding cost, in contrast to a place-centric occupancy grid approach (or even a full global metric map approach). On the other hand, the usage of a global graph for global navigation avoids the need for a global metric path-finding step.

Future work in this subject will include an analysis of impact of the variation of several parameters (being the occupancy grid-size the most significant of these) in the trajectory length. Also a comparison with other existing methods (for example, a full global grid approach) in terms of both computational cost and trajectory length will be included. Finally, a localization method will be considered in order to test this method with real robots.

V. Acknowledgements

This work has been supported by Ministry of Science of the Argentine Republic, under International Cooperation Project RC/09/08 and the Ministry of Education of the Czech Republic, under the Project No. 6840770038.

References

[1] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the 11th international conference on advanced robotics*. Citeseer, 2003, pp. 317–323.

[2] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millenium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.

[3] J. A. Castellanos and J. D. Tardós, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston, MA: Kluwer Academic Publishers, 1999.

[4] A. Elfes, "Occupancy Grids: A stochastic spatial representation for active robot perception," in *Proc. of the Sixth Conf. on Uncertainty in AI*. 2929 Campus Drive, San Mateo, CA 94403: Morgan Kaufmann Publishers, Inc., 1990.

[5] P. Lefevre, A. Prüß, and U. R. Zimmer, "ALICE - topographic exploration, cartography and adaptive navigation on a simple mobile robot," in *TELEMAN Students' Research Project Congress*, Leeuwenhorst, The Netherlands, June 1994.

[6] G. Dudek, P. Freedman, and S. Hadjres, "Using local information in a non-local way for mapping graph-like worlds," in *Proc. of Int. Joint Conf. of Artificial Intelligence*, Chambery, France, 1993.

[7] P. Buschka and A. Saffiotti, "Some notes on the use of hybrid maps for mobile robotics," in *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems (IAS) 547–556*, 2004, pp. 547–556.

[8] 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*. Cambridge, MA, USA: MIT Press, 1998, pp. 21–52.

[9] S. Thrun, J.-S. Gutmann, D. Fox, W. Burgard, and B. J. Kuipers, "Integrating topological and metroc maps for mobile robot navigation: a statistical approach," in *Proc. of the Fifteenth National/Tenth Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, ser. AAAI '98/IAAI '98. Menlo Park, CA, USA: American Association for Artificial Intelligence, 1998, pp. 989–995.

[10] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, "SLAM in Large-scale Cyclic Environments using the Atlas Framework." *The International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113–1139, December 2004.

[11] G. M. Youngblood, L. B. Holder, and D. J. Cook, "A framework for autonomous mobile robot exploration and map learning through the use of place-centric occupancy grids," in *ICML Workshop on Machine Learning of Spatial Knowledge*, 2000.

[12] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of IEEE Int. Symposium on Computational Intelligence in Robotics and Automation. 'Towards New Computational Principles for Robotics and Automation'*. IEEE Comput. Soc. Press, 1997, pp. 146–151.

[13] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," *IEEE Int. Conf. on Robotics and Automation*, pp. 1572–1577, 1998.