Learning Multiresolution Maps in Dynamic Worlds

Rui Araújo, David Lourenço, and Aníbal T. de Almeida

ISR - Institute for Systems and Robotics, and
Electrical Engineering Department, University of Coimbra,
Pólo II, P-3030-290 Coimbra, Portugal
{rui,david,adealmeida}@isr.uc.pt

Abstract

Mobile robots require the ability to build their own maps to navigate in unknown environments. This paper introduces a new method for learning multiresolution maps for navigation in changing worlds. It extends our navigation architecture [1] that integrates a multiresolution and fuzzy ART based feature world models. In a companion paper [3] we introduce a new method for updating the fuzzy ART model in non-static worlds. We describe our new navigation architecture that, by integrating this new capability, is able to dynamically not only increase but also decrease local resolution according to variations in the local cluttering and complexity of the world. The paper presents experimental results obtained with a Nomad 200 mobile robot that demonstrate the effectiveness of the proposed method.

1 Introduction

Maps are an essential component to enable mobile robot navigation in complex environments. They are needed for path-planning, self-localisation, and human-robot interaction. While it is possible to pre-install maps in a robot, to navigate in unknown environments, robots must be able to build their own models of the world.

A widespread approach for mobile robot navigation is based on the occupancy grid representation of the environment [7]. Occupancy grids represent the world as a two dimensional array of evenly-spaced cells, with each cell holding a value which represents the confidence in whether it is occupied space or free space. Although grid-based models are easy to build and maintain, they impose a constant resolution structure onto the environment without any selectivity concerning the nature and clutter of the world. A very localized feature of the world may impose a very high (constant-)resolution grid over the entire state-space. This implies high data requirements, and induces excessive detail on world modeling and updating, on reasoning (high computational costs), and on the paths that result from such a model. Also, the difficulties on the direct application of grid-based models on localization have been pointed out in [9]. An alternative for overcoming the space and time complexities of grid-based methods is to use a variable resolution state-space partition (e.g. [10]). Local resolution is usually only high enough to capture the important local detail of the world. This enables a lower number of cells (space) and thus lower search effort (time). Another alternative to the costs of grid-based models is to use a set of geometric primitives (or features) for representing objects in the world (e.g. [5], [9]). Geometric primitive representations, have been difficult to build, but are significantly more compact, less complex, and fully applicable to high- and low-level motion planning (e.g. [1]) and localization approaches (e.g. [9]). With higher dimensions the geometric model data requirements become exponentially smaller than the requirements of constant-resolution cellular models.

In this paper we introduce a new method for learning multiresolution maps on dynamic worlds. The new method works in conjunction with the feature-based method proposed in the companion paper [3]. With the method proposed in this paper, the system is able to locally adapt the multiresolution model used in our architecture, to dynamic increases and decreases of local clutter and complexity in the world. Our new navigation architecture (Fig. 1) integrates the new method, and extends our previous work [1] as described in this paper.

The paper is organized as follows. Section 2 presents an overview of our navigation architecture. Section 3 presents a discussion on dynamic worlds and how our learning architecture relate to them. Section 4 introduces a new method to dynamically update a multiresolution grid model in response to changing worlds. Section 5 presents experimental results. Finally, in Section 6 we give some concluding remarks.

2 Navigation Architecture

For completeness, in this section we present an overview of our current navigation architecture. Please see [1] for further details – lack of space prevents a more detailed description here.

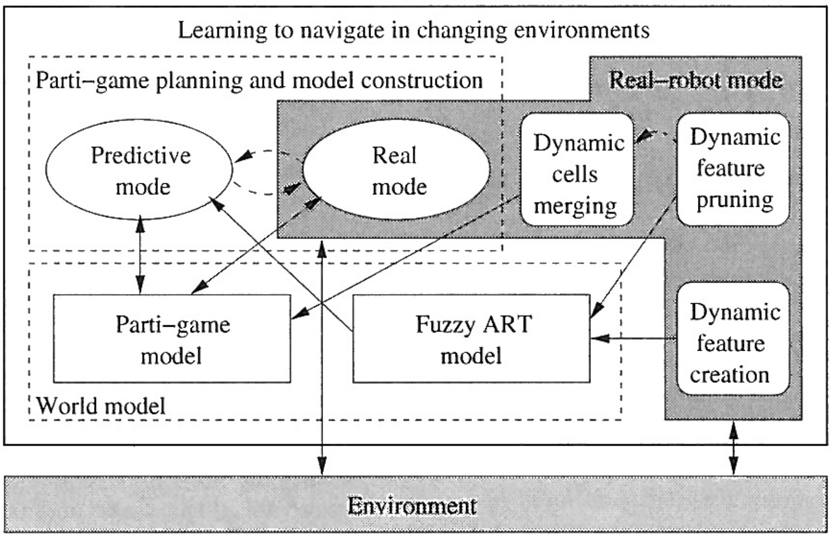

Figure 1: Architecture of predictive on-line trajectory filtering on a dynamically updated model for navigation in changing environments.

2.1 Core of the Learning Architecture

Figure 1 illustrates our current navigation architecture. The original core from which the architecture was developed is the parti-game learning approach [6], [1]. The system can simultaneously, learn a model of its environment, and learn to navigate to a goal region in an unknown world, having the predefined abilities of doing straight-line motion to a specified position in the world, and obstacle detection (not avoidance). The learning approach is based on a selective and iterative partitioning of the state-space. It is a multiresolution approach, beginning with a large partition, P, and then increasing resolution by subdividing the state-space (e.g. see Figs. 2(a), 4) in areas where the learner predicts that a higher resolution is needed. Cells are organised in a kd-tree, for fast state-to-cell mapping [1]. For each cell i there is a set, NEIGHS(i), of (cell-) neighbours of i. In order to reach the goal, the mobile robot path is planned to traverse a sequence of cells. The ability of straight-line motion is used as a greedy controller to move from one cell to the next cell on the path. This request to move to the next cell on the path (which is a neighbouring cell) may fail – usually due to an unexpected obstacle that is detected to be obstructing the robot path. A database, D, that includes the cell-outcomes observed when the system aims at a new cell, is memorised and maintained, accumulating experience in real-time. D includes a collection of OUTCOMES(i,j) sets. OUTCOMES(i,j) is the set of cells that were previously observed to be attained when the system was on cell i and aimed cell at j. In the absence of observed experience an optimistic assumption is taken [1]. The combination of database D, and partition P constitutes one internal learned *world model* – the *parti-game (world) model* (Fig. 1).

Database D is in turn used to plan the sequence of cells to reach the goal cell, using a game-like minimax shortest path approach. The next cell on the path is chosen taking into account a worst case assumption, i.e. we imagine that for each cell we may aim, an imaginary adversary is able to force the worst next-cell outcome. In this way we always aim at the neighbouring cell with the best worst-outcome. For this purpose, the minimax shortest distance from cell i to goal, $J_{WC}(i)$, is computed [1] using Dynamic Programming methods. For choosing spatial resolution, cells are split when the robot is caught on a losing cell - a cell for which the distance to the goal cell is ∞, i.e. for the current resolution, the game of arriving at the goal cell is lost. In these situations, as explained in [1], cells in the border between losing and non-losing cells are split. Cells which have just been split must be subject to forgetting of accumulated cell-outcome experience. This induces further local exploration in places the system had difficulties to navigate [1].

2.2 Learning a Feature Map of the World

In our previous work [1] we have introduced and demonstrated the effectiveness of a new approach for learning a map of the world, that is based on the fuzzy ART neural architecture [4]. The method was integrated into our navigation architecture for improving its world model, by making better use of sensor information received from sensors. The method has several desirable characteristics [1]: it is self-organising and multifunctional, has small data requirements and low computational complexity, has the significant advantage of being capable of incremental on-line operation according to the flow of sensor data reception, and is easy to extend to higher dimensions. With the approach the system incrementally extracts and updates a collection of rectangular geometric primitives, whose union represents occupied space, where sensor data points associated with objects have been perceived - a kind of unsupervised clustering. Familiar inputs are directly associated to their rectangular categories, while novel exemplars continue to trigger the generation of new categories. This method corresponds to the "Dynamic feature creation" module of Fig. 1. The extracted rectangles form what we define as the *fuzzy ART (world) model* [1]. The composite contribution of the parti-game and fuzzy ART models forms an/the improved *(overall) world model* (Fig. 1).

2.3 Improving Learning by Predictive On-line Trajectory Filtering (POTF)

The parti-game learning approach was extended by the introduction of a method for Predictive On-line Trajectory Filtering (POTF) [1]. Figure 1 presents the overall architecture illustrating the main ideas of the POTF navigation method. A distinction between a predictive mode and a real mode is established. One of the main ideas of the method, is to reduce real-robot exploration by giving priority to predictive exploration, by taking advantage of the learned fuzzy ART world model, and allowing a very significant reduction on the time-consuming exploration effort that is associated with searching the world with a real robot. In both modes, path planning is performed using the parti-game approach, with the parti-game model being incrementally updated, according to the results of both predictive and real exploration. However, only in real mode is the fuzzy ART model incrementally updated, because only in this mode is real sensor data available for this purpose. In [2] we have presented quantitative results demonstrating: (1) the benefits of the POTF method, and (2) that the world model and navigation method is general purpose for learning multiple and different navigation paths.

3 Dynamic Worlds

An important general aspect of a map building method is its ability to cope with non static worlds. A changing robot world can be seen as a union of one or more changes, each belonging to one, out of two possible classes [1]. On class 1, a new object is created on a previous free-space location. Changes of class 2 correspond to the opposite, i.e. an object is removed creating a free area on the state-space.

As discussed in [3], [1], the fuzzy ART based map building method is clearly able to cope with changes of class 1. In fact a new object will lead to new sensor perception points, which will generate new, or update existing, fuzzy ART categories and corresponding rectangular geometric primitives on the map. However, the method is not able to appropriately cope with changes of class 2. For that purpose in [3] we introduce the Prune-Able Fuzzy ART neural architecture, and extend the map learning method, complementing it with the ability to remove (or possibly update) geometric primitives on the map, in response to the possible removal of objects in the world. This corresponds to the "Dynamic feature pruning" module of Fig. 1.

As discussed in [1] the core of the navigation architecture (the parti-game learning approach) is able to deal with changes of class 1. On the other hand, changes of class 2 do not prevent navigation to the goal. However, after having converged to a stable start-goal path, the system is not able to take advantage of a possible better path that could have become possible after the removal an obstacle (a change of class 2). The opportunity to explore the new better path will only arise when a new obstacle obstructs the previous solution path. However, in both cases, the execution of new exploration in response to changing worlds is dependent of the important operation of forgetting accumulated experience – that is associated with cell splitting (Sec. 2.1). Further, if we permit that new exploration will always be performed at the cost of (using only) additional increases of partition resolution, then the multiresolution model may no longer have the capability to adapt its resolution to the local clutter and complexity in the world. In this way we would lose the advantage when comparing to constant resolution cellular models (Sec. 1). As noted in [1], this motivates the introduction of methods such that the system is able to tackle changing worlds in a more general way. This will be done in the next section.

4 Dynamic Selective Cells Merging

From the motivation of Sec. 3 we have developed a new method, Dynamic Selective Cells Merging (DSCM), for the dynamic simplification of the/a multiresolution world model (in particular that of our navigation architecture). This also uses our companion work [3]. Whenever a feature is pruned from the fuzzy ART model, this is a sign that some obstacle in the world has disappeared, making the world less cluttered and complex locally. Our system clearly identifies this as an opportunity for simplifying the partition model by lowering local resolution through the merging (e.g. Fig. 2(b)) of selected cells. Thus, whenever a feature is pruned from the world representation, Algorithm 1 (Fig. 3) is called. The objective of this algorithm is to provide a simplification of the partition model of the world by cells merging, whenever it finds an opportunity. In step 1 the algorithm verifies if the changes that have occurred constitute an opportunity for model simplification by cells merging. In our work cells merging was triggered when one of the following two criteria occurred:

Criteria 1. A pruned fuzzy ART rectangle intersected the border between two or more cells (e.g. Fig. 2(c)).

Criteria 2. The model changes (rectangle pruning or cells merging) cause the occurrence of a situation where two "brother-cells" no longer have a fuzzy ART feature inside of both their areas.

When criteria 1 is met the kd-tree is searched until the node that originated the intercepted border is

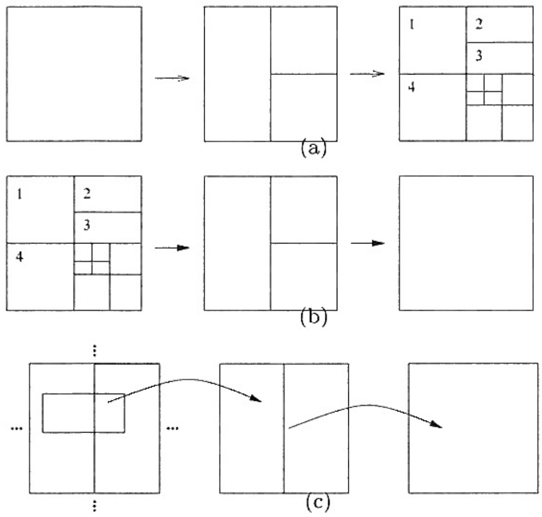

Figure 2: (a) Cells subdivision; (b) Cells merging; (c) Example with feature pruning and merging two cells.

ALGORITHM 1 (SELECTIVE CELLS MERGING)

1. IF fuzzy ART model changes (features pruned) do not allow cells merging (partition simplification) **THEN** RETURN.

2. ELSE WHILE World changes allow cells merging

- **2.1** Let C_1 and C_2 be the cells to be merged.
- **2.2 IF** C_1 and C_2 are "brother-cells" on the kd-tree. **THEN**
  - **2.2.1** Merge C_1 and C_2 and let the resulting cell be referenced as C_1.
  - **2.2.2** Update the new C_1 coordinates in the world.
  - **2.2.3** Remove C_1 and C_2 from the kd-tree. It remains the "parent-cell" called called C_1.
  - **2.2.4** Transfer the contents of **NEIGHS**(C_2) to the **NEIGHS**(C_1) set, i.e. **NEIGHS**(C_1) := **NEIGHS**(C_1) + **NEIGHS**(C_2) − $\{C_1, C_2\}$
  - **2.2.5** Substitute in all sets **NEIGHS**(i) and **OUTCOMES**(i,j) all the references to C_2 by C_1.
  - **2.2.6** Delete in all the sets **NEIGHS**(i) and **OUTCOMES**(i,j) all the repeated (redundant) references to C_2 that were originated by the substitution of C_2 by C_1.

3. END OF THE WHILE CYCLE

4. MODEL-CHANGED:=TRUE.

5. RETURN.

Figure 3: Algorithm 1: selective cells merging.

found. All the descendant nodes from this node are removed from the tree, and give rise to only one cell that results from the merging(s).

In the situation when two "brother-cells" no longer include any features inside (criteria 2), then the two cells are merged, the corresponding leaf nodes are deleted from the tree, and the corresponding "father-node" proceeds by representing the new unique/merged cell, and the system tests criteria 2 with the new cell.

Whenever one of these criteria is met, we start the process of merging "brother-cells" (step 2) which leads to a new cell with the characteristics of the corresponding "father-cell" cell that existed before splitting. During the merging process the NEIGHS(i) set is updated (step 2.2.4) with i being the cell resulting from merging. The NEIGHS(i) set will contain all the cells which are neighbours of both merged cells, except the merged cells themselves. In the sets NEIGHS(i) and OUTCOMES(i,j) all the references to the merged cells are substituted by references to the new cell (step 2.2.5), with redundant references being deleted (step 2.2.6).

It can be easily seen that it may occur a case where the merging of cells propagates to higher levels in the kd-tree, which means that we come back to nodes that represent cells that already existed in very early stages of the partition (lower resolution). This means that the experience accumulated in the OUTCOMES(i,j) sets of the merged cells will be lost. To mitigate this behaviour the approach can be complemented with a mechanism to limit the number of chain-mergings up along the kd-tree. However, the loss of information resulting from the merging of cells poses no problem since the system also has the fuzzy ART model representing obstacles; And the system can make an efficient predictive exploration in predictive mode (POTF - Sec. 2.3) to split the cells again, as needed to arrive at the goal, but now according to the new distribution of obstacles in the world.

Thus the method proposed in this section introduces the following advantageous characteristics. Increase or decrease the partition resolution to better adapt the model according to variations on the spatial distribution of local clutter and complexity of the world. This leads to a forgetting of information which in turn induces new exploration (mostly done in predictive mode). Further: the additional exploration will enable the system to take advantage of better navigation paths that may have become available after a removal of obstacles – thus overcoming the limitation of the system in response to world changes of class 2 (discussion on Sec. 3). The method proposed in this section corresponds to the "Dynamic cells merging" module of Fig. 1.

5 Experimental Results and Discussion

The experiments presented here were conducted using a Nomad 200 robot [8]. The robot is equipped with a Laser range sensor, sonars and infrared range sensors around its turret. In the specific experiments presented in this paper, the infrared sensors were used to create fuzzy ART features, and the Laser was used for the obstacle removal perception mechanism [3]. The robot includes two motors on its base that are used to control its translational and rotational movement. The experiments were organised as a sequence of trials to navigate to a goal. Only the first trial starts with an empty world model, but after that the model is continuously updated during the sequence of trials. To perform robot localisation, we have simply used accumulation of encoder information, with location accumulators being reset at the beginning of each trial. Even though this simple approach induces errors, it was sufficient to experimentally validate the effectiveness of the proposed method.

Figure 4 presents the results of two similar simulation experiments composed of three trials each. All obstacles were removed from the world on the second trials of both experiments. However, Experiment 1 (Figs. 4(a),(b)) did not use DSCM and feature pruning [3], but Experiment 2 (Fig. 4(c)-(e)) used both these methods. In Fig. 4 all fuzzy ART rectangles (FARs) are represented with a security border gap [1]. As expected, without DSCM, the system was not able to discover a shorter path to the goal on Trial 2 of Experiment 1 (Fig. 4(b)). Experiment 2 clearly demonstrates the benefits of the DSCM method. In fact, using DSCM, the system was able to obtain a better/shorter path to the goal (Figs. 4(d),(e)). Ideally, in this situation the system should have merged all the cells, thus enabling the robot to move directly from start to goal. However, due to the perceptual range limitation of the Laser sensor, not all FARs were removed from the model by the method of [3]. The robot only detects the removal of obstacles that can be perceived when the robot traverses the defined path. In this way all the obstacles that disappeared outside the perceptual range of the robot continued to be represented as FARs in the fuzzy ART model, thus limiting the opportunities for the DSCM method to work.

Figure 5 presents Experiment 3 which was performed with the real robot and demonstrates the operation of the system when using DSCM and the pruning method of [3]. Again, these proposed methods enabled the system to correctly update the parti-game model, in particular the experience database, D (Sec. 2.1). Also, because the fuzzy ART model was used in Experiment 3, without using these new methods, the system would have been blocked from arriving at the goal by a closed barrier of obstacle-related fuzzy ART rectangles that would have been temporally accumulated, as early as in trial 2. Experiment 3 is similar to an Experiment presented in [1] except that in the later, only the parti-game part of the navigation system was used, thus not taking advantage of the benefits of POTF.

Compared to other methods in the field of dynamic map building/update, our navigation architecture, based on the multiresolution partition and the fuzzy ART features method, inherits the benefits that were discussed in Sec. 1, and [1], in the context of worlds not necessarily dynamic. In comparison to [9] our method takes advantage of a broader set of sensor data situations to update the fuzzy ART model [3]; Also we integrate the multiresolution model which significantly strengthens the navigation architecture. In the multiresolution method of [10] the world model is updated to reflect the presence of a freshly sensed obstacle when it obstructs the robot path - a change of class 1 (Sec. 3). It is not able to take advantage of sensor information to deal with changes of class 2. It updates a free space confidence of a region from the knowledge that the robot body trajectory has just swept the region, but as natural the system does not chose paths through known obstacles to search for possible class 2 model updates. Our method differs in that it continuously integrates sensor data into the model. Also, it is able to tackle changes of class 2 in the world; being able to use perceived sensor information to selectively and continuously, not only increase, but also decrease partition resolution in response to variations in the spatial distribution of obstacles.

6 Conclusions

In this paper we have introduced a new method for selective cells merging in multiresolution grid world models. The new method enables the system to increase or decrease its local resolution in response to variations on the local clutter of the world, enabling the system to make a better representation of the complexity each region of the environment. The method was integrated into our navigation architecture extending it in order to improve its behaviour in changing worlds. Experimental results were presented demonstrating the effectiveness of the proposed method.

Acknowledgments

This work was partially supported by the Portuguese Science and Technology Foundation (FCT/MCT) by project CTR2001, contract POSI/33594/SRI/2000.

References

[1] R. Araújo and A. T. de Almeida. "Learning sensor-based navigation of a real mobile robot in unknown worlds." *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 29-B, no. 2, pp. 164–178. April 1999.

[2] R. Araújo and A. T. de Almeida. "Learning both a world model and navigation paths in an unknown environment." *Int. Journal of Intelligent Automation and Soft Computing*, vol. 6, no. 2, pp. 159–170. 2000.

[3] R. Araújo, D. Lourenço, and G. Ferreira. "Integrating laser and infrared sensors for learning dynamic self-organizing world maps." *Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI2001)*. Baden-Baden, Germany. 2001.

[4] G. A. Carpenter, S. Grossberg, and D. B. Rosen. "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system." *Neural Networks*, vol. 4, no. 6, pp. 759–771. 1991.

[5] J. A. Janét, S. M. Scoggins, M. W. White, I. J. C. Sutton, E. Grant, and W. E. Snyder. "Self-organising geometric certainty maps: A compact and multifunctional approach to map building, place recognition and motion planning." *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'97)*. Albuquerque, New Mexico, USA, pp. 3421–3426. April 1997.

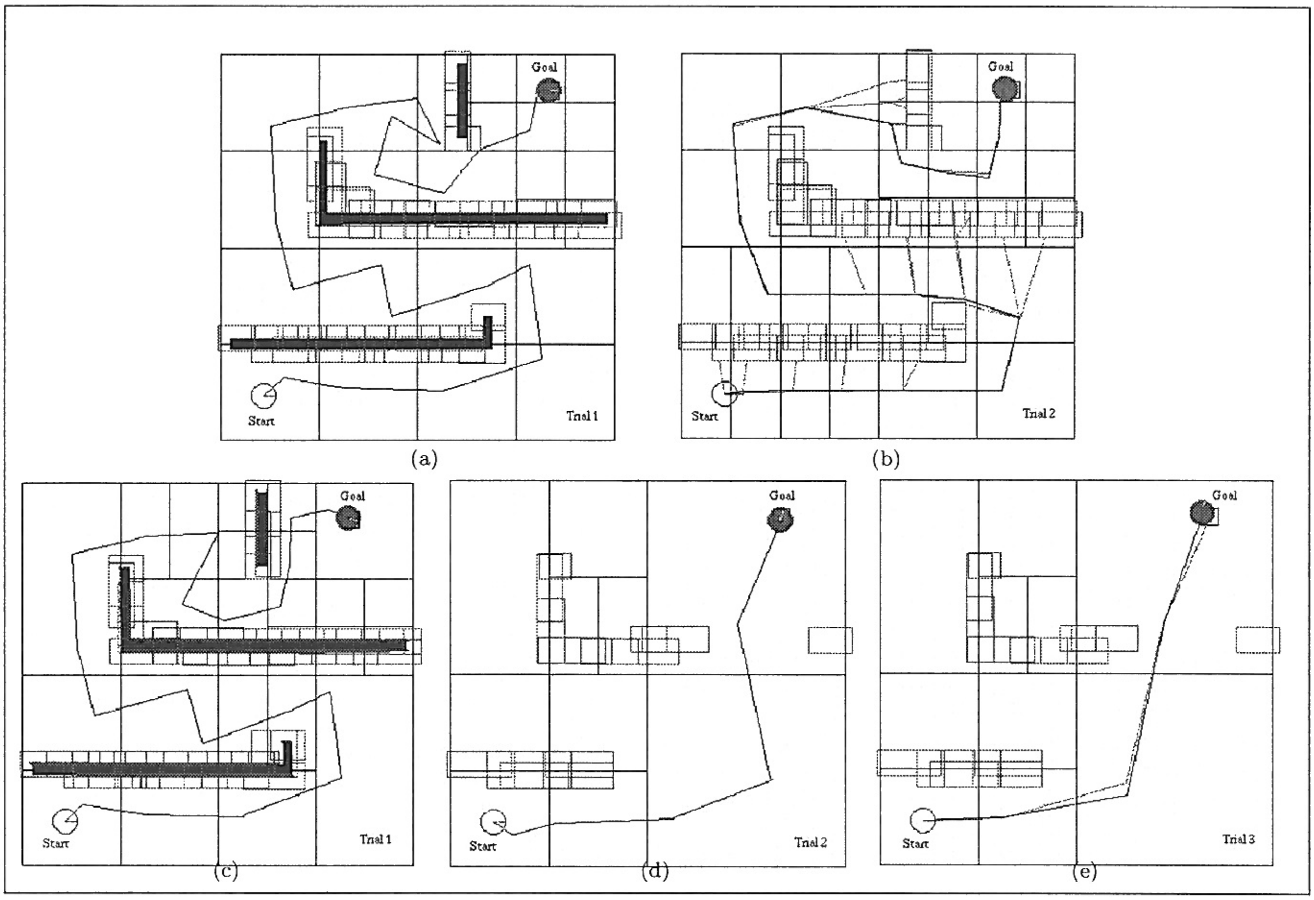

Figure 4: Experiment 1 (not using DSCM): Trial 1 (a), Trial 2 (b); Experiment 2 (using DSCM): Trial 1 (c), Trial 2 (d), Trial 3 (e); All fuzzy ART rectangles include a security border gap [1]. (c) includes infrared sensor data points. (b) and (e) also presents POTF trajectories at the beginning of the trial.

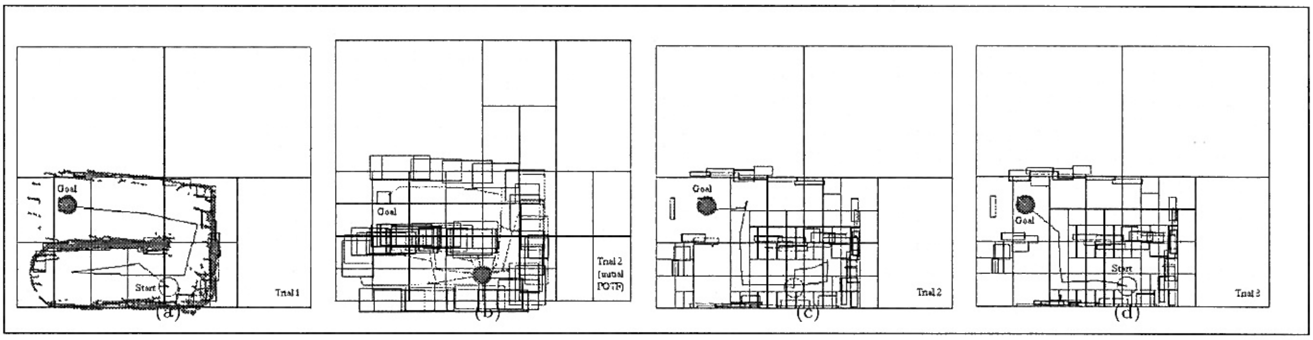

Figure 5: Experiment 3: using the real-robot and the methods proposed in this paper and [3]. The FARs include a border gap [1] only in Fig. 5(b). In this figure the FARs represent the presence of obstacles.

[6] A. W. Moore and C. G. Atkeson. "The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces." *Machine Learning*, vol. 21, no. 3, pp. 199–233. Dec. 1995.

[7] H. P. Moravec and A. Elfes. "High resolution maps from wide angle sonar." *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA'85)*. St. Louis, Missouri, USA, pp. 116–121. 1985.

[8] Nomadic Technologies, Inc. *[Nomad 200] User's Manual*. Mountain View, CA, January 1996.

[9] J. Vandorpe, H. V. Brussel, J. D. Shutter, H. Xu, and R. Moreas. "Positioning of the mobile robot lias with line segments extracted from 2d range finder using total least squares." *Prep. Fifth Int. Symp. on Experimental Robotics (ISER'97)*. Barcelona, Spain, pp. 309–320. June 1997.

[10] A. Zelinsky. "A mobile robot exploration algorithm." *IEEE Trans. Robotics and Automat.*, vol. 8, no. 6, pp. 707–717. December 1992.