Learning Variable-Resolution Maps for Navigation in Dynamic Worlds

Rui Araújo, Gonçalo Gouveia, and Aníbal T. de Almeida
Department of Electrical and Computer Engineering,
Institute for Systems and Robotics (ISR), and
University of Coimbra, Pólo II,
P-3030-290 Coimbra, Portugal,
{rui,ggouveia,adealmeida}@isr.uc.pt

Abstract — Map building is an important component to enable autonomous mobile robots to navigate in complex unfamiliar environments. This paper introduces a new method for learning multiresolution maps for navigation in unknown and dynamic worlds. It extends our navigation architecture [1] that integrates a multiresolution grid world model and fuzzy ART based world model. The fuzzy ART world model is composed of a set of rectangular geometric primitives, or features. In a companion paper [2] we introduce a new method for updating the fuzzy ART model in dynamic worlds. In this paper we describe our new navigation architecture that, by integrating the new proposed map building method for dynamic worlds, is able to dynamically not only increase but also decrease local resolution according to variations in the local clutter and complexity of the world. With the new overall navigation architecture the mobile robot is able to cope with, and navigate, in changing worlds. The paper presents experimental results obtained with a Nomad 200 mobile robot that demonstrate the effectiveness of the proposed methods.

I. INTRODUCTION

SUCCESSFUL navigation of autonomous mobile robots in unknown and complex environments require that robots have the ability to construct their own maps. Maps are required for motion-planning, self-localization, and human-robot interaction.

A popular representation paradigm for robot maps is the occupancy grid model [3]. In this approach the world is represented as a two-dimensional matrix of cells with constant spatial resolution, with each cell containing some measure of the certainty that the corresponding area of the environment is occupied space or free space. Grid-based models are easy to build and maintain. However, to accurately represent even the most detailed and particular feature of the world, they impose a very high (constant-) resolution representation to the entire model without any selectivity concerning the local degree of detail and clutter of the real world. 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. On the other hand, as pointed out in [4], the direct application of grid-based models on localization also presents difficulties. The use of a multiresolution state-space partition is an alternative for overcoming the space and time complexities of grid-based methods - e.g. [5]. In such models, local resolution is chosen only high enough to model the important local detail of the world. With this paradigm the model attains a lower number of cells (space) and thus lower search effort (time) in involved. Another alternative to the costs of grid-based models is to use a set of geometric primitives (or features) for representing objects of the environment - e.g. [6], [4]. 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. [4]). When considering higher dimensions the requirements of geometric models become exponentially smaller than the requirements of constant-resolution grid 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 [2]. With the proposed method, the system is able to locally adapt the multiresolution model, to dynamic increases and decreases of local clutter and complexity in the world. The 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 II presents an overview of our navigation architecture. Section III presents a discussion on dynamic worlds and how our learning architecture relate to them. Section IV introduces a new method to dynamically update a multiresolution grid model in response to changing worlds. Section VI presents experimental results. Finally, in Section VII some concluding remarks are given.

II. NAVIGATION ARCHITECTURE

For completeness, in this section we present an overview of our current navigation architecture. Please see [1] for further details.

A. 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 [7]. 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, $\mathcal{P}$, and then increasing resolution by subdividing the state-space (e.g. see Fig. 6) in areas where the learner predicts that a higher resolution is needed. Cells are organized in a kd-tree, for fast state-to-cell mapping. For each cell $i$ there is a set, $\text{NEIGH}(i)$, of (cell-) neighbors 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 neighboring 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 memorized and maintained, accumulating experience in real-time. $D$ includes a collection of $\text{OUTCOMES}(i, j)$ sets.
Several desirable characteristics: it is self-organizing 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 (fuzzy ART rectangles (FARs)), 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). To provide a safety distance to obstacles the architecture expanded the fuzzy ART rectangles with a border gap when performing motion planning.

B. Learning a Feature Map of the World

In previous work [1] it was 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 [8]. The method was integrated into our navigation architecture for improving its world model, by making better use of information received from sensors. The method has several desirable characteristics: it is self-organizing 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 (fuzzy ART rectangles (FARs)), 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). To provide a safety distance to obstacles the architecture expanded the fuzzy ART rectangles with a border gap when performing motion planning.

C. 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 [9] 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.

D. Frequency of Predictive Effort

The “frequency” of predictive effort may be controlled, by configuring the condition that is used to trigger the
transition from real mode to predictive mode. The system always starts in predictive mode, and the following five options (listed in increasing order of predictive frequency) may be used: (1) after cell splitting that takes place when the robot is caught on a losing cell, or (2) at the end of every failed cell-aim, or (3) at the end of every cell-aim, or (4) at the end of the rotation phase of every cell-aim and every cell-aim itself, or (5) at the end of every motion sampling interval. The robot advances from cell to cell using cell aims, which are subdivided in two phases: (1) a pure rotation phase, and (2) a straight-line motion phase. It was noticed that after updating the fuzzy ART model using sensor information received in the rotation phase, and then using the updated model in predictive mode enabled the system to identify that the subsequent straight line motion of the same aim was blocked by objects in the world. Thus this motion is not performed and a different aim is chosen. This lead to the introduction of the new option 4 on the list above. Options 1-3, 5 were proposed in previous work [1].

III. 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. 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 [2], 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 [2] we introduce the Prune-Able Fuzzy ART neural architecture (PAFARTNA), and extend the map learning method, complementing it with the ability to remove (prune), shrink, or split geometric primitives on the map, in response to the possible removal of objects in the world. This corresponds to the “Dynamic feature pruning”, “Dynamic feature shrinking”, and “Dynamic feature splitting” modules of Fig. 1.

As discussed in [1] the core of the navigation architecture (the part-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 of 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. II-A). 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. I). 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 sections.

IV. Dynamic Selective Cells Merging

From the motivation of Sec. III 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 [2]. Whenever a feature is pruned, shrunk, or split on 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 of selected cells (e.g. Fig. 2). Thus, whenever a feature is pruned, shrunk or split on 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

<table>
<thead>
<tr>
<th>ALGORITHM 1 (SELECTIVE CELLS MERGING)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. IF fuzzy ART model changes of type “feature pruned”, or type “feature shrunk”, or “feature split” do not allow cells merging (partition simplification) THEN RETURN.</td>
</tr>
<tr>
<td>2. ELSE WHILE World changes allow cells merging</td>
</tr>
<tr>
<td>2.1 Let $C_1$ and $C_2$ be the cells to be merged.</td>
</tr>
<tr>
<td>2.2 IF $C_1$ and $C_2$ are “brother-cells” on the kd-tree. THEN</td>
</tr>
<tr>
<td>2.2.1 Merge $C_1$ and $C_2$ and let the resulting cell be referenced as $C_1$.</td>
</tr>
<tr>
<td>2.2.2 Update the new $C_1$ coordinates in the world.</td>
</tr>
<tr>
<td>2.2.3 Remove $C_1$ and $C_2$ from the kd-tree. It remains the “parent-cell” called $C_1$.</td>
</tr>
<tr>
<td>2.2.4 Transfer the contents of NEIGHS($C_1$) to the NEIGHS($C_1$) set, i.e. NEIGHS($C_1$) := NEIGHS($C_1$) + NEIGHS($C_2$) - {$C_1$, $C_2$}.</td>
</tr>
<tr>
<td>2.2.5 Substitute in all sets NEIGHS($i$) and OUTCOMES($i$, $j$) all the references to $C_2$ by $C_1$.</td>
</tr>
<tr>
<td>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 $C_1$.</td>
</tr>
<tr>
<td>3. END OF THE WHILE CYCLE</td>
</tr>
<tr>
<td>4. MODEL-CHANGED:=TRUE.</td>
</tr>
<tr>
<td>5. RETURN.</td>
</tr>
</tbody>
</table>

Fig. 2. Example with features pruning (1), shrinking (2), and splitting (3), leading to the merging of two cells (4).

Fig. 3. Algorithm 1: selective cells merging.
merging is triggered when one of the following two criteria occurs:

**Criteria 1.** A pruned fuzzy ART rectangle, or a pruned sub-rectangle in the case of a FAR shrinking, or a FAR splitting operation (see Sec. III), intersected the border between two or more cells (e.g. Fig. 2).

**Criteria 2.** The model changes (rectangle pruning, or shrinking, or splitting; 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 first originated the intersected border is 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” that existed before splitting. During the merging process the \( \text{NEIGHS}(i) \) set is updated (step 2.2.4) with \( i \) being the cell resulting from merging. The \( \text{NEIGHS}(i) \) set will contain all the cells which are neighbors of both merged cells, except the merged cells themselves. In the sets \( \text{NEIGHS}(i) \) and \( \text{OUTCOMES}(i,j) \) 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 \( \text{OUTCOMES}(i,j) \) sets of the merged cells will be lost. To mitigate this behavior 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 if necessary the system can make an efficient predictive exploration in predictive mode (POTF – Sec. II-C) to split the cells again, as needed to arrive at the goal, but now according to the new distribution of obstacles in the world. This subsequent splitting of cells is also subject to forgetting of accumulated cell-outcome experience. Both these types of selective forgetting of experience enable the system to better adapt its internal world model in response to dynamic changes 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 (Sec. III). The method proposed in this section corresponds to the “Dynamic cells merging” module of Fig. 1.

V. FORGETTING WITH FARs REMOVALS

Section IV described a method for selective cells merging that take place when certain changes in the world take place. Cells mergings lead to forgetting of locally observed cell-outcome information. In this section we present another class of relevant situations where it is important to perform local forgetting of outcome information. This constitutes the forgetting with FARs removals method (FWFR).

Figure 4 illustrates a situation that often occurs, but is not handled by the two cells-merging criteria of Section IV. Let us suppose that the FAR totally contained inside cell 5 was removed. Since this FAR does not intersect the border cell of any cell, it does not satisfy criteria 1. The brother of cell 5 is cell 6 that contains FARs: thus the second criteria is also not satisfied. Let us also suppose that the path planned before the removal of the FAR is composed by the following sequence of cells \( \{1, 2, 3, 4\} \). If the system has past aim-fail experience information regarding the aim from cell 1 to cell 5 (this is very likely to occur due to the existence of FARs signaling the possible existence of obstacles), it will not be able to plan a predictive trajectory to the goal that includes this 1-to-5 cell transition. However, if all aim-fail information of real experience associated to cell 5 is forgotten, then a new path composed of the following sequence of cells \( \{1, 5, 3, 4\} \) can be planned: if the topological distance between cells (cells have a graph organization) is made an increasing function of an expected geometric traveling distance between the two cells, then this new sequence of cells will be favored giving a shorter total travelling distance to goal.

Thus, whenever a FAR is removed, besides the procedures introduced in Section IV, the following additional processing is performed: if the removed FAR is completely contained inside a cell, then all the information regarding real cell-outcome experience associated to that cell is removed from the cell-outcome database. Algorithm 2 (Fig. 5) summarizes this forgetting procedure. It will allow the system to try to predictively consider planning a trajectory that passes through that cell. However, even if a path through such a cell would be better (shorter), it is not guar-
and the system returns to its previous state. But if that is the case, then a new better (shorter) path to the goal. As easily understood, without FARs removals this new better path would not be used because the FARs would block a subset of the corresponding POTF trajectories.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments presented here were conducted using a Nomad 200 mobile robot [10]. 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 and Laser sensors were used to create fuzzy ART features, and the Laser was used for the obstacle removal perception mechanism [2]. The robot includes two motors on its base that are used to control its translational and rotational movement. The experiments were organized 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 localization, 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 methods.

Figs. 6(a)-(c) present the results of a simulation experiment. Figs. 6(d)-(h) present a real-robot experiment. All the figures include the FARs and the constructed multiresolution partition model. Only in Fig. 6(a) are the FARs represented without the security border gap (see Sec. II-B, and [1]). Fig. 6(b) also presents the subset of POTF trajectories (see Secs. II-C and II-D, and Ref. [1]) that were performed at the beginning of Trial 2. Figs. 6(c),(d),(h) also present the points perceived by the infrared and Laser sensors.

Experiment 1 (Fig. 6(a)-(c)) used the PAFARTNA (see Secs. III, IV), and the DSCM (Sec. IV) methods. In Trial 1 the system explores the world and constructs a resulting model. The POTF trajectories of Fig. 6(b) give a clear example of how the predictive exploration (POTF method, Secs. II-C, II-D) works and significantly decreases the real-robot exploration effort. Please see [9] for a quantitative analysis of the improvements provided by the introduction of POTF. Shortly after the beginning of Trial 2, two obstacles were removed from the world (Fig. 6(c)). As can be seen, in this Trial 2, the PAFARTNA method (Sec. III, and [2]) enabled the fuzzy ART world model to be correctly updated, by removing FARs (or sub-FARs) in response to these Class 2 changes in the world. It can also be observed that, in response to the removal of FARs, the dynamic cells merging method proposed in Sec. IV, has triggered the merging of a number of cells close to the destination region, thus lowering the local resolution consistently with the decrease in local complexity and clutter of the world. Forgetting of cell-outcome information is induced by the DSC method (Sec. IV), the FWFR method (Sec. V), and the subsequent operation of the other mechanisms of the navigation architecture (Sec. II, and [1]) that also provide exploration and path planning. The interaction of these methods enabled the overall navigation system to explore and discover, new and better (shorter) paths to the goal.

As easily understood, without FARs removals this new better path would not be used because the FARs would block a subset of the corresponding POTF trajectories.

Figs. 6(d)-(h) present the results of Experiment 2 which was performed with the real robot and using the PAFARTNA method (Secs. III, IV), the dynamic cells merging, and FWFR methods proposed in this paper. In Trial 1 (Figs. 6(d)) the robot explored the world and constructed a model in its navigation to goal. In Trial 2 (Fig. 6(e)) there were changes on the distribution of objects in the world: these changes can be observed by comparing the fuzzy ART model (the FARs) of Trials 1 and 2. The solution-path of Trial 1 was blocked by a new object that was placed in the world, but the system was able to learn a new alternate path to the goal that became available as a consequence of the removal of another object. The robot enters a dead-end situation but then escapes from it after all blocking obstacles have entered the perceptual range of the robot sensors, and by this reason have been perceived and integrated into the fuzzy ART world model. In Trial 3 (Fig. 6(f)) there are no changes in the robot world, and the system, attains a better path to navigate to goal, using the previously learned world model. In Trial 4 (Fig. 6(g)) the previous solution-path became blocked by new obstacles placed at two different places (one of them had been present on Trial 1 but was removed on Trial 2). However, the system was able to learn a new solution-path that became available due to the removal of other two obstacles. As in Trial 3, the robot enters a dead-end situation but then escapes from it after all blocking obstacles fall inside the perceptual range of the robot sensors and as a consequence are integrated again into the fuzzy ART model. In Trial 5 (Fig. 6(h)) another object was removed close to the starting position. This did not significantly change the navigation possibilities of the robot, and the system converged to shorter navigation path in this trial. In Trials 4 and 5 some world-obstacles have been removed close to the initial robot location, which has lead to the pruning of some FARs in this area. This in turn triggered cells-mergings in this area, leading to a lower local resolution that is consistent to (and better represents) a world that has became less cluttered in this region of the state-space.

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. I, and [1], in the context of worlds not necessarily dynamic. In comparison to [11] our method takes advantage of a broader set of sensor data situations to update the
fuzzy ART model [2]; Also we integrate the multiresolution model which significantly strengthens the navigation architecture. In the multiresolution method of [5] 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. III). 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 arriving sensor range data (richer than robot-swept-area information) 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.

VII. Conclusions

In this paper we have introduced new methods for selective cells merging in multiresolution grid world models, and selective forgetting of previous navigation experience in dynamic worlds. The new methods enable 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, with advantages also in decreasing the planning complexity due to a lower number of cells. The methods were integrated into our navigation architecture extending it in order to improve its behavior in changing worlds. Experimental results were presented demonstrating the effectiveness of the proposed methods.

ACKNOWLEDGMENTS

This work was partially supported by the Portuguese Science and Technology Foundation (FCT/MCT) by project LRN2002, contract POSI/SRI/42043/2001.

REFERENCES