Mobile Robot Exploration and Map-Building with Continuous Localization

Brian Yamauchi, Alan Schultz, and William Adams

Navy Center for Applied Research in Artificial Intelligence
Naval Research Laboratory
Washington, DC 20375-5337

{yamauchi, schultz, adams}@aic.nrl.navy.mil

Abstract

Our research addresses how to integrate exploration and localization for mobile robots. A robot exploring and mapping an unknown environment needs to know its own location, but it may need a map in order to determine that location. In order to solve this problem, we have developed ARIEL, a mobile robot system that combines frontier-based exploration with continuous localization. ARIEL explores by navigating to frontiers, regions on the boundary between unexplored space and space that is known to be open. ARIEL finds these regions in the occupancy grid map that it builds as it explores the world. ARIEL localizes by matching its recent perceptions with the information stored in the occupancy grid. We have implemented ARIEL on a real mobile robot and tested ARIEL in a real-world office environment. We present quantitative results that demonstrate that ARIEL can localize accurately while exploring, and thereby build accurate maps of its environment.

1.0 Introduction

We have been investigating the problem of how to integrate exploration with localization in mobile robots. A robot needs to know its own location in order to add new information to a map, but a robot may also need a map to determine its own location. Robots often use dead reckoning to estimate their position without a map, but wheels slip, and internal linkages may be imprecise. These errors accumulate over time, and the dead reckoning position estimate becomes increasingly inaccurate.

For a robot exploring an unknown environment, a key question is how to build a map while simultaneously using that map to self-localize. We have addressed this question with ARIEL (Autonomous Robot for Integrated Exploration and Localization). ARIEL combines frontier-based exploration [9] with continuous localization [7] in a mobile robot system that is capable of exploring and mapping an unknown environment while maintaining an accurate estimate of its position at all times.

In this paper, we describe how frontier-based exploration and continuous localization work, and how we integrated these capabilities. ARIEL has been implemented on a real robot and tested in a real-world office environment, and we present quantitative results comparing the performance of exploration with and without localization.

2.0 Frontier-Based Exploration

2.1 Overview

The central question in exploration is: *Given what you know about the world, where should you move to gain as much new information as possible?*

The central idea behind frontier-based exploration is: *To gain the most new information about the world, move to the boundary between open space and uncharted territory.*

Frontiers are regions on the boundary between open space and unexplored space. When a robot moves to a frontier, it can see into unexplored space and add the new information to its map. As a result, the mapped territory expands, pushing back the boundary between the known and the unknown. By moving to successive frontiers, the robot can constantly increase its knowledge of the world. We call this strategy *frontier-based exploration*.

If a robot with a perfect map could navigate to a particular point in space, that point is considered *accessible*. All accessible space is contiguous, since a path must exist from the robot's initial position to every accessible point. Every such path will be at least partially in mapped territory, since the space around the robot's initial location is mapped at the start. Every path that is partially in unknown territory will cross a frontier. When the robot navigates to that frontier, it will incorporate more of the space covered by the path into mapped territory. If the robot does not incorporate the entire path at one time, then a new frontier will always exist further along the path, separating the known and unknown segments and providing a new destination for exploration. In this way, a robot using frontier-based exploration will eventually explore all of the accessible space in the world.

2.2 Perception and Spatial Representation

We use evidence grids [6] as our spatial representation. Evidence grids are Cartesian grids containing cells, and each cell stores the probability that the corresponding region in space is occupied. Evidence grids have the advantage of being able to fuse information from different types of sensors.

We use sonar range sensors in combination with a planar laser rangefinder to build our robot's evidence grid maps. In order to reduce the effect of specular reflections, we have developed a technique we call *laser-limited sonar*. If the laser returns a range reading less than the sonar reading, we update the evidence grid as if the sonar had returned the range indicated by the laser, in addition to marking the cells actually returned by the laser as occupied.

As a result, evidence grids constructed using laser-limited sonar have far fewer errors due to specular reflections, but are still able to incorporate obstacles detected by the sonar below (or above) the plane of the laser. In practice, we have found that laser-limited sonar drastically reduces the number of uncorrected specular reflections from walls and other large obstacles, which tend to be the major sources of errors in evidence grids built using sonar.

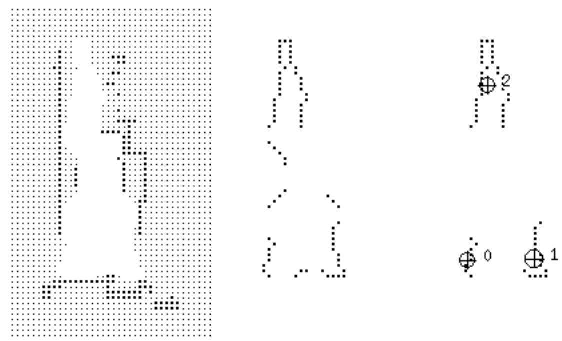
2.3 Frontier Detection

After an evidence grid has been constructed, each cell in the grid is classified by comparing its occupancy probability to the initial (prior) probability assigned to all cells. This algorithm is not particularly sensitive to the specific value of this prior probability. (A value of 0.5 was used in all of the experiments described in this paper.)

Each cell is placed into one of three classes:

open: occupancy probability < prior probability
unknown: occupancy probability = prior probability
occupied: occupancy probability > prior probability

A process analogous to edge detection and region extraction in computer vision is used to find the boundaries between open space and unknown space. Any open cell adjacent to an unknown cell is labeled a frontier edge cell. Adjacent edge cells are grouped into frontier regions. Any frontier region above a certain minimum size (roughly the size of the robot) is considered a frontier.



(a) (b) (c)

Figure 1: Frontier detection: (a) evidence grid, (b) frontier edge segments, (c) frontier regions

Figure 1a shows an evidence grid built by a real robot in a hallway adjacent to two open doors. Figure 1b shows the frontier edge segments detected in the grid. Figure 1c shows the regions that are larger than the minimum frontier size. The centroid of each region is marked by crosshairs. Frontier 0 and frontier 1 correspond to open doorways, while frontier 2 is the unexplored hallway.

2.4 Frontier Navigation

Once frontiers have been detected within a particular evidence grid, the robot attempts to navigate to the nearest accessible, unvisited frontier. The path planner uses a depth-first search on the grid, starting at the robot's current cell and attempting to take the shortest obstacle-free path to the cell containing the goal location. While the robot moves toward its destination, reactive obstacle avoidance behaviors prevent collisions with any obstacles not present while the evidence grid was constructed.

When the robot reaches its destination, it performs a sensor sweep using laser-limited sonar, and adds the new information to the evidence grid. The robot then detects frontiers in the updated grid, and navigates to the nearest remaining accessible, unvisited frontier.

3.0 Continuous Localization

Previous techniques for localization have been based upon learning and recognizing landmarks in the environment. Our localization technique does not rely on the presence of specific landmarks, but instead uses the entire local environment of the robot to determine its location.

An important issue in localization is how often to relocalize. Many existing techniques only relocalize when an error in position is detected or after an unacceptable amount of error has accumulated. With continuous localization, the robot makes frequent small corrections instead of occasional large corrections. The advantage is that the error is known to be small, so fast correction techniques can be used.

Continuous localization builds short-term perception maps of the robot's local environment. These maps typically contain very small amounts of positional or rotational error. These short term maps are used to locate the robot within the global map via a registration process. The results from this process are used to correct the robot's odometry.

In the experiments described in this paper, continuous localization updates the robot's position estimate whenever the robot moves move than 24 inches from the position of its last update (as measured by dead reckoning). In addition, each degree of (cumulative) rotation is treated as equivalent to 0.03 inches of translation.

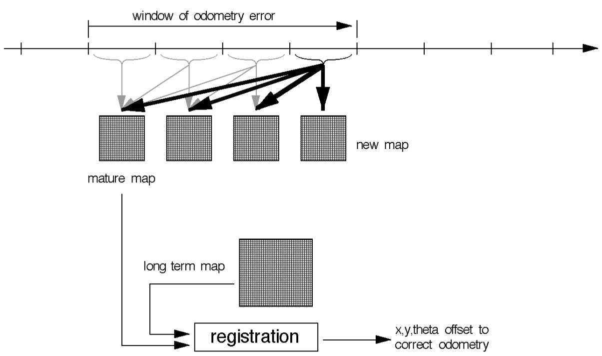


Figure 2: Continuous localization

Figure 2 shows the process of continuous localization. The robot builds a series of short-term perception maps of its immediate environment, each of which is of brief duration and contains only a small amount of dead reckoning error. After several time intervals, the oldest (most "mature") short-term map is used to position the robot within the long-term map via a registration process.

The registration process consists of sampling the possible poses within a small area around the robot's current estimated pose. For each tested pose, the mature short-term map is rotated and translated by the difference in pose (the offset) and a match score is calculated based on agreement between the cell values of the short-term map and the long-term map, summed across all cells. The match scores for all tested poses are then treated as masses and the offsets as distances, and a center of mass calculation is performed to determine the offset that is likely to have the highest match score. This offset is applied to the robot's odometry, placing it at the pose which causes its local perceptions to best match the long-term map. After the registration takes place the most mature map is discarded.

4.0 ARIEL

4.1 System Overview

Frontier-based exploration provides a way to explore and map an unknown environment, given that a robot knows its own location at all times. Continuous localization provides a way for a robot to maintain an accurate estimate of its own position, as long as the environment is mapped in advance. The question of how to combine exploration with localization raises a "chicken-and-egg" problem: the robot needs to know its position in order to build a map, and the robot needs a map in order to determine its position.

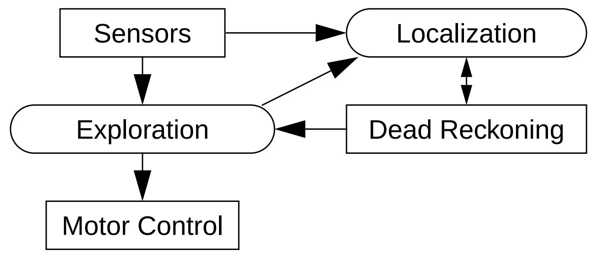


Figure 3: ARIEL system architecture

ARIEL is designed to address this problem. We assume that the robot starts with an accurate initial position estimate, so localization only needs to correct for dead reckoning errors that accumulate while the robot moves through the world. However, these errors can accumulate quickly, so it would not be feasible to map a large environment using dead reckoning alone.

The solution is to use the partial maps constructed by frontier-based exploration These maps are incrementally extended whenever the robot arrives at a new frontier and sweeps its sensors. Even though these maps are incomplete, they describe the spatial structure of the robot's immediate environment, including all of the territory between the robot's current location and all of the detected frontiers. These maps are passed to continuous localization to be used as long-term maps.

As the robot navigates to the next frontier, continuous localization constructs short-term maps that represent the robot's recent perceptions. If dead reckoning error starts to accumulate, these short-term maps will deviate from the long-term map. The registration process will then correct for this error by adjusting the robot's position estimate.

When the robot arrives at the new frontier, its position estimate will be accurate. When frontier-based exploration performs the next sensor sweep, the new information will be integrated at the correct location within the map.

Figure 3 shows the system architecture for ARIEL. Frontier-based exploration and continuous localization run in parallel. Both processes make use of information from the robot's sensors, but only frontier-based exploration sends commands to the robot's motor control system. Frontier-based exploration passes a new map to continuous localization every time the robot arrives at a new frontier. Continuous localization corrects the robot's dead reckoning transparently, so no direct communication is necessary from localization to exploration.

4.2 Implementation

ARIEL is implemented on a Nomad 200 mobile robot equipped with a planar laser rangefinder, sixteen sonar sensors, and sixteen infrared sensors. Frontier-based exploration and continuous localization run on separate Sparcstation 20s that communicate with each other over an ethernet and with the robot over a radio ethernet. A Pentium processor onboard the robot handles low-level sensor processing and motor control.

5.0 Experiments

5.1 Overview

In previous work [9], we have demonstrated that frontier-based exploration can successfully map real-world office environments. In relatively small environments, such as a single office or laboratory, frontier-based exploration was capable of mapping accurately without continuous localization. However, for larger environments, significant amounts of position error can accumulate using dead reckoning, so localization is necessary for building accurate maps.

To measure ARIEL's effectiveness in a larger environment, we have conducted a set of experiments in a hallway environment (70 feet long). This hallway, like many of those in office buildings, is cluttered with obstacles. These obstacles include a printer table that blocks half the width of the hallway, a set of open cabinets containing electrical wiring, switchboxes mounted on the walls, various cardboard boxes, a water fountain, and a water cooler.

In order to measure ARIEL's performance, we initially constructed a ground truth grid by manually positioning the robot at viewpoints throughout the hallway and sweeping the robot's sensors. This ground truth grid is only used to score the grids learned by ARIEL. The ground truth grid is ***not*** used by ARIEL for exploration or localization.

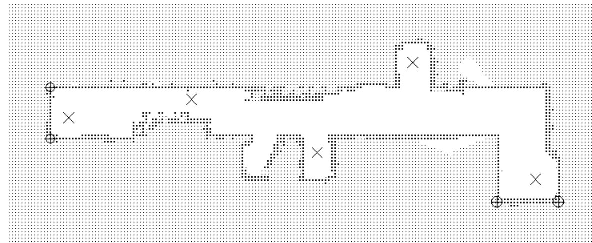


Figure 4: Ground truth evidence grid for hallway

Figure 4 shows the ground truth evidence grid for the hallway environment. Cells representing open space are represented by whitespace. Cells representing occupied space are represented by black circles. Cells representing unknown territory (beyond the hallway walls) are represented by small dots. The five Xs correspond to the robot's starting locations for ARIEL's exploration trials.

The four crosshairs on the map indicate reference points at the corners of the ends of the hallways. Since dead reckoning error accumulates as the robot moves through the world, the points explored last are likely to have the greatest amount of positional error. And since ARIEL always moves to the closest unexplored frontier, one of the ends of the hallways is generally the last place explored. By measuring the difference between the actual position of these hallway corners and the position of these corners in ARIEL's learned maps, the amount of positional error incorporated into the map can be estimated. In these experiments, the maximum error between a reference point and the corresponding feature on the learned grid is used as a bound on the positional error introduced into the map. We refer to this metric as the *reference point error* for the learned grid.

5.2 Exploration Without Localization

Our first set of trials measured the performance of frontier-based exploration without continuous localization. Five exploration trials were conducted, one from each of the starting locations marked on Figure 4. In three of these trials, frontier-based exploration directed the robot to explore the hallway and build a map, but substantial amounts of position error accumulated during each trial. As a result, sensor information was incorporated into the map at the wrong locations, and the magnitude of this error increased over time.

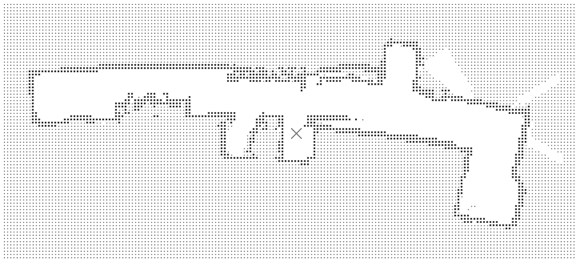


Figure 5: Evidence grid learned without localization

Figure 5 shows a map learned by frontier-based exploration without localization. The robot started at the position marked with the X. Initially, the robot explored the territory on the left side of the map. Then it navigated back to explore the remaining frontiers on the right side of the map. As the robot explored, position error constantly accumulated. As a result, the right half of the map is considerably more distorted than the left. This grid has a reference point error of 7.0 feet.

In two of the trials, the position error was sufficiently large to prevent further exploration. In both of these cases, the robot started in the middle of the hallway, and explored one side of the hallway first, while remembering the frontier location corresponding to the other side of the hall. When the robot went back to explore the other side, the robot's position error was so large that the relative location of the frontier corresponded to a position behind the (real) hallway walls.

Frontier-based exploration without localization was successful at mapping the entire hallway in 60% of the trials. In the successful trials, the average reference point error for the learned maps was 7.9 feet, and the average amount of time required to explore the hallway was 18.4 minutes.

5.3 Exploration With Localization

Our second set of trials measured ARIEL's performance using frontier-based exploration in combination with continuous localization. We used the same hallway environment, the same starting points for the robot, and the same ground truth evidence grid. Frontier-based exploration again directed the robot to explore the environment, but continuous localization also regularly updated the robot's position estimate as the robot explored. Starting from the same five initial positions shown in Figure 4, ARIEL was able to build a complete map of the environment in all five trials.

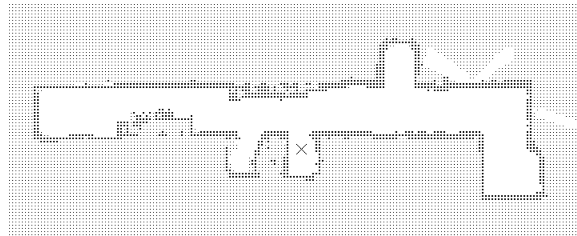


Figure 6: Evidence grid learned with localization

Figure 6 shows the evidence grid learned using localization starting from the position marked with the X (the same initial position as in Figure 5). This grid has a reference point error of only 0.4 feet, which is equal to the width of a single grid cell.

ARIEL was successful at mapping the entire hallway in all of the trials using continuous localization. The average reference point error for the learned maps was 2.1 feet, or roughly one quarter of the error in the maps learned without localization. ARIEL's 100% success rate indicates that this accuracy is sufficient to navigate robustly through this cluttered hallway environment. Reactive obstacle avoidance allows the robot to deal with small errors in the map.

The average amount of time required to explore the entire hallway was 20.7 minutes. This is slightly longer than the average time (18.4 minutes) required without localization, due to the time required for frontier-based exploration to send its learned evidence grids to continuous localization. However, since the localization process runs on a different processor than the exploration system, the computation required for localization does not slow down the exploration process. For further details about these experiments, see [10].

6.0 Related Work

Considerable research has been done in robot map-building, but most of this research has been conducted in simulation [3] or with robots that passively observe the world as they are moved by a human controller [2]. However, a few systems for autonomous exploration have been implemented on real robots.

Mataric [5] has developed Toto, a robot that combines reactive exploration, using wall-following and obstacle-avoidance, with a simple topological path planner. The reactive nature of Toto's exploration limits its ability to map environments where wall-following is insufficient to explore the complex structure of the world.

Lee [4] has implemented Kuipers Spatial Semantic Hierarchy [3] on a real robot. However, this approach assumes that all walls are parallel or perpendicular to each other, and this system has only been tested in a simple environment consisting of a three corridors constructed from cardboard barriers.

Thrun and Bücken [8] have developed an exploration system that builds a spatial representation that combines an evidence grid with a topological map. This system has been able to explore the network of hallways within a large building. While this approach works well within the hallway domain, it also assumes that all walls are either parallel or perpendicular to each other. An implicit assumption is that walls are observable and not obstructed by obstacles. These assumptions make this approach unsuitable for rooms cluttered with obstacles that may be in arbitrary orientations.

Duckett and Nehmzow [1] have developed a mobile robot system that combines exploration and localization. This system uses wall-following for exploration. For localization, this system uses a self-organizing neural network trained using ART. Since this system relies upon dead reckoning to determine the robot's position during exploration, any drift in dead reckoning during exploration will be incorporated into the map. This robot has only been tested in a small enclosed area (6 meters by 4 meters), so it is unclear whether this approach will scale to larger, more complex, environments.

ARIEL has a number of advantages over previous exploration systems. ARIEL can explore efficiently by moving to the locations that are most likely to add new information to the map. ARIEL can explore environments containing both open and cluttered space, where walls and obstacles are in arbitrary orientations. Finally, ARIEL can maintain an accurate estimate of the robot's position even as it moves into unknown territory.

7.0 Conclusion

We have introduced ARIEL, a mobile robot system that combines frontier-based exploration with continuous localization. ARIEL answers the question of how to learn a new map while simultaneously using that map to self-localize. We have tested ARIEL in a cluttered hallway from a real-world office environment. These experiments have shown that ARIEL can explore an unknown environment and build accurate maps that can be used for robust navigation.

8.0 Acknowledgments

This work is supported by the Office of Naval Research.

9.0 References

[1] Tom Duckett and Ulrich Nehmzow, "Experiments in evidence-based localisation for a mobile robot," *Proceedings of the AISB Workshop on Spatial Reasoning in Mobile Robots and Animals*, Manchester, UK, 1997.

[2] David Kortenkamp, *Cognitive Maps for Mobile Robots: A Representation for Mapping and Navigation*, Ph.D. Thesis, Electrical Engineering and Computer Science Department, University of Michigan, 1993.

[3] Benjamin Kuipers and Yung-Tai Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *Journal of Robotics and Autonomous Systems*, 8:47-63, 1991.

[4] Wan Yik Lee, *Spatial Semantic Hierarchy for a Physical Robot*, Ph.D. Thesis, Department of Computer Sciences, The University of Texas at Austin, 1996.

[5] Maja Mataric, "Integration of representation into goal-driven behavior-based robots," *IEEE Transactions on Robotics and Automation*, 8(3):304-312, June 1992.

[6] Hans Moravec and Alberto Elfes, "High resolution maps from wide angle sonar," *Proceedings of the IEEE International Conference on Robotics and Automation*, St. Louis, MO, 1985, pp. 116-121.

[7] Alan Schultz, William Adams, and John Grefenstette, "Continuous localization using evidence grids," NCARAI Technical Report AIC-96-007, Naval Research Laboratory, Washington, DC, 1997.

[8] Sebastian Thrun and Arno Bücken, "Integrating grid-based and topological maps for mobile robot navigation," *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, August 1996, pp. 944-950.

[9] Brian Yamauchi, "A frontier-based approach for autonomous exploration," *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 1997, pp. 146-151.

[10] Brian Yamauchi, Alan Schultz, and William Adams, "Integrating exploration and localization for mobile robots," submitted to *Autonomous Robots*, Special Issue on Learning in Autonomous Robots.