Representing a 3-D Environment with a 2½-D Map Structure

Edward H.L. Fong[†*] William Adams[‡] Frederick Crabbe[†] Alan C. Schultz[‡]

[†]Computer Science Department
United States Naval Academy
fong@2002.usna.com
crabbe@usna.edu

[‡]Navy Center for Applied Research
in Artificial Intelligence
Naval Research Laboratory
{adams, schultz}@aic.nrl.navy.mil

***Abstract*— This paper explores the development of a two and one-half dimensional (2½-D) map structure to provide an autonomous mobile robot with a more three-dimensional (3-D) model of its environment than those afforded by current map structures. The 2½-D map structure was created by modifying the widely used evidence grid to store a height, along with a probability value, in each cell location to record the varying elevations of a 3-D environment. Results show that this map structure is capable of providing an autonomous mobile robot with a representation of a limited 3-D environment that will allow it to perform obstacle detection, path planning, and to an extent, localization.**

I. Introduction

An autonomous mobile robot must possess some idea of its surroundings in order to navigate autonomously. This "idea" is usually presented to the robot in the form of a map. In order for the map to be useful, the robot needs to be able to perform four tasks using it: 1) obstacle detection, 2) path planning, 3) localization, and 4) frontier exploration. Currently, robots are able to perform all four tasks as long as they operate in a single plane of motion (a two-dimensional (2-D) environment) [2], [9], [11]. This restricts the robot to operate in environments where the ground is flat and smooth. In a more three-dimensional (3-D) environment, the robot would have to account for additional environmental structures and obstacles such as ramps and cliffs. The map must be capable of storing these features in order to provide the robot with the ability to traverse or avoid these areas. For example, the robot must be able to use its map to identify and move around steep drops in elevations which might damage the robot while identifying and traversing more gentle elevation changes.

Moravec uses 3-D evidence grids to model such environments [5], [6]. This approach requires much memory (16 megabytes to represent an 8 by 8 meter room) and computational power. Thrun produced 3-D models by fitting a low-complexity planar model to collected data [3], [10]. This approach was used to produced a 3-D model of an environment above the ground—the robot could only move around and localize in 2-D. These maps were used to enhance the information obtained from an environment and was not used for navigating the robot.

This paper presents a two and one-half dimensional (2½-D)[1] map structure that is a more compact 3-D model of its environment which is computationally feasible and provides navigational capabilities in such an environment. This map structure was created by modifying a 2-D evidence grid [4] to store a height and probability value pair for each original 2-D cell. Using the 2½-D map structure, a robot was shown to be capable of performing obstacle detection, path planning, and localization.

II. The Evidence Grid

A wide variety of map representation structures (such as Voronoi diagrams, evidence grids, quadtrees, etc.) [1], [4], [8] are currently capable of representing a robot's environment. These maps usually represent the environment in two dimensions. A 2-D map representation is sufficient for robots in many controlled environments, however, as robots are fielded in more rugged, 3-D terrain, these models are insufficient. Instead, we propose an extension to the 2-D evidence grid that enables a mobile robot to model traversable surfaces in three dimensions.

A. The 2-D Evidence Grid

The 2-D evidence grid is one of the most popular map structures for autonomous mobile robots. It is comprised of an array of grid cells where each cell represents the occupancy probability of a location in the environment. The benefits of this map structure include the ability

*Current affiliation: Aeronautics and Astronautics Department, Massachusetts Institute of Technology - usna02@mit.edu

[1]We define a 2½-D map as a structure that is capable of representing partial three-dimensional information. It stores more information than a two-dimensional map but does not provide a complete three-dimensional representation of an environment.

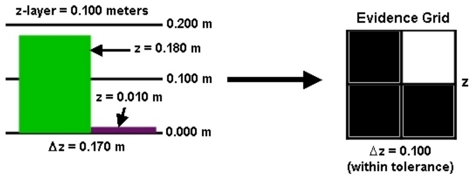


Fig. 1. Hidden Obstacle.

for probabilistic learning in the face of uncertain sensor information and the ability to easily fuse readings from different sensors for the same location. Though its array of grid cells can require more memory than other map structures, the evidence grid allows fast, constant time access to each cell in the map. In addition, numerous algorithms have already been developed for exploration, obstacle detection, path planning, and localization using a 2-D evidence grid.

B. The 3-D Evidence Grid

A 3-D evidence grid is an extension of the 2-D grid to a 3-D array. For a full 3-D evidence grid, the size of the grid increases with the number of z layers required to reliably represent the environment. If a fine resolution is desired, or if the z range is large, the number of z layers would have to be large—adding to the size of the map and to the computations involved in accessing, retrieving, and using that map.

The resolution of the z layer is especially important since it determines whether or not the robot can move from its current cell location to another cell. Many wheeled robots cannot roll over obstacles that are above a certain height. For example, the maximum climbing height of an obstacle that iRobot™'s ATRV-Jr. can roll over is approximately 0.1 meters. Any difference in elevation (from one cell to the next) greater than 0.1 meters must be considered an obstacle. However, using a cell resolution height of 0.1 meters is not fine enough to detect obstacles. In Fig. 1, an obstacle is hidden from the robot since the height used to represent each cell is taken from the center of the cell in world coordinates—the actual height of the object is not recorded (just the probability that there is an obstacle somewhere in that cell). The map would show two adjacent cells to be within the robot's climbing ability despite the obstacle being greater than the climbing height. In order for the 3-D evidence grid to work, each cell height should represent (at most) half of the maximum climbing height of the robot (0.05 meters for the ATRV-Jr.). If each cell were to represent a $0.125m \times 0.125m \times 0.05m$ volume of space, creating a $10m \times 10m \times 1m$ map would require 128,000 cells. To represent a $100m \times 50m \times 50m$ parking structure would require 3.2×10^8 cells. The memory required to store such a grid, and the computational resources and time needed to access and use such a large structure (e.g. for localization), makes it infeasible to implement this map structure for real-time 3-D movements.

III. 2½-D Map Representation

We developed a 2½-D map representation structure to retain the benefits afforded by the evidence grid without the dramatic increase in the map size when representing a 3-D environment. This section describes the structure of this 2½-D map.

The 2½-D map structure consists of a two-dimensional array where each (x, y) cell location contains a pair: the height of the tallest object at that location, and the probability that an object exists *at that height*. All heights are stored as a relative measurement in reference to the robot's initial position as ground zero. Using a 16-bit integer to represent centimeters of height, a map can be created to provide the robot with information in the range of ± 327 meters from the robot's starting altitude.

All of the cells are initialized to $<0,0>$. If the probability value for any cell is 0 (or less than a predetermined threshold) then the robot treats that cell as an unknown area and its z value is not used for any calculations. As the robot moves, its sensors obtain the location of objects in the environment relative to the sensor. Since the robot keeps track of its 3-D pose (x, y, z, $roll$, $pitch$, yaw), knows the position and orientation of the sensor on the robot, and obtains information on where objects are relative to the sensor, the robot can perform a series of transformations to determine the (x, y, z) position of objects in the environment. It would then update the cells of local x and y locations with the measured z value and change the probability value according to its sensor model. The first time an object is detected for a certain cell, the following update is performed on that cell:

$$H_{t(x,y)} = H_s \quad (1)$$

$$P_{t(x,y)} = P(s|O) \quad (2)$$

where $H_{t,(x,y)}$ and $P_{t,(x,y)}$ are the new height and probability value stored in the respective cell for the given (x,y) location after t readings (in this case $t = 1$). H_s is the detected height of the object and $P(s|O)$ is the probability that the sensor returned the value it did given that an object was actually there (obtained from the sensor model).

If an object is later detected for the same (x,y) location, the following method is used to updated the height and probability grid cells. If the height of the new sensor reading, H_s, is greater than the stored height of the previous $t-1$ sensor readings, $H_{t-1(x,y)}$, plus some tolerance, T (which we set to be 0.1 meters), and the probability value, $P(s|O)$ is also above some predetermined limit, then (1) and (2) are used as before. In other words, if $H_s > H_{t-1(x,y)} + T$, then the new height and probability value will replace the old values (as in the case for the first reading at that location). If the $P(s|O)$ value returned does not provide us with a belief strong enough to cause us to replace our current height value, then the reading is ignored. The tolerance value, T, is used to account for sensor noise. If the new reading is within the tolerance of the old reading, then the robot can treat the new height value as the same as the one stored.

If the new height is within the set tolerance of the old recorded value, $H_{t-1(x,y)} - T \leq H_s \leq H_{t-1(x,y)} + T$, then:

$$H_{t(x,y)} = \begin{cases} H_s & \text{if } H_s > H_{t-1(x,y)} \\ H_{t-1(x,y)} & \text{if } H_s \leq H_{t-1(x,y)} \end{cases} \tag{3}$$

$$P_{t(x,y)} = \frac{P(s|O)P_{t-1(x,y)}}{P(s|O)P_{t-1(x,y)} + P(s|\neg O)(1 - P_{t-1(x,y)})} \tag{4}$$

where $P(s|\neg O)$ is the probability that the sensor would have returned the value it did given that there was nothing at that location—also provided by the sensor model. Equation (4) uses Bayes' rule to combine the previous $t-1$ sensor readings with the current one. By increasing the probability for the given height in this manner, we increase our belief that the maximum height of the object at that (x,y) location is indeed the value stored.

If the new height is below and outside the tolerance level of the old reading, $H_s < H_{t-1(x,y)} - T$, then the new sensor reading will be ignored and the $H_{t(x,y)}$ and $P_{t(x,y)}$ values will be the same as the old ones. In this way, multiple detections of an object for an (x,y) location which are below the height range will not affect the probability value for that cell. Detecting a location's height below its recorded value does not significantly increase the belief that our surface profile is correct (it only tells us that the maximum height for that location can be anywhere from that reading upwards) and is therefore ignored. It is also important to note that objects in the robot's environment were assumed to be "complete" from the ground up. For example, structures such as arches did not exists in our testing environment. Though this assumption falls short of modeling a true 3-D environment, it was made to facilitate the initial development of the 2½-D map structure.

When a robot detects an object at a certain (x, y, z) location, a line through empty space can be drawn from the 3-D coordinates of the sensor to the position of the detected object. Each corresponding cell along this line should contain a height below the interpolated height between the sensor and the detected obstacle. If the stored height and probability cell pair indicates a strong probability of a stored height at or higher than the interpolated value (in situations where the first few readings were erroneous, the object moved, or the current reading is false), then the belief probability for that location will be decrease using the value from the sensor model and (4). Meanwhile, the stored height will remain unchanged. If there is no object at that height, then the probability will keep decreasing during subsequent scans until it drops below a threshold—at which point the height at that location is considered unknown. The next sensor reading which indicates an object height (which can include the height of the ground) at that (x,y) location will replace the previous height value and update its probability using (1) and (2). Using this method, the map can be updated to account for a changing environment and erroneous sensor readings.

This storage and updating method was selected because it provided the ability to represent and update only the pertinent information concerning the robot's environment. In a situation where all structures are "complete", the robot is only concerned with the top of the structure at each (x,y) location. In updating the map, the robot only needs to know whether the new height reading is higher, the same, or lower than the value currently stored. If the value is lower, it is ignored since it does not provide any new information about that location—the robot already knows that the space is occupied. If the new reading is similar to the stored one, then the robot has an increased belief that the recorded height exists and will update its probability value accordingly.

The tricky part occurs when the new reading is greater than the stored height (plus the tolerance). The position of most sensors (including the ones we used) are fixed on the robot and cannot provided a full 3-D picture of the environment with a single scan. Therefore, the robot is only able to "see" a portion of its world at a time. As its position and orientation changes during movement, the robot can detect a higher portion of an object than it was able to at its previous position. In this case, the new reading completely replaces the map's previously stored values if the belief probability is high enough. If this reading was erroneous, subsequent scans of empty space below the stored value would correct the map. Since this might take some time depending on the frequency that the area is scanned, an alternative method would be to hold the new reading separate from the stored value and replace it only when it has obtained a strong enough belief. However, this updating method was not implemented after determining that setting a belief threshold was sufficient for the sensor suite that we used.

IV. Evaluation

To evaluate the 2½-D map structure described in the previous section, we examined whether or not the robot can use it to perform obstacle extraction, path planning, and localization. The robot must be able to demonstrate these three capabilities before it can perform frontier exploration.

A. *Obstacle Extraction*

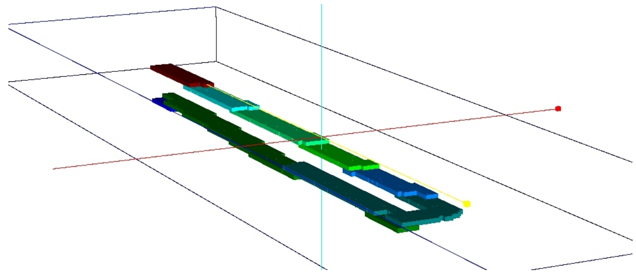
To show that the 2½-D map structure will actually represent changes in elevation that can be used by the robot, two maps were created of a ramp area in back of Building 1 at the Naval Research Laboratory (NRL) in Washington, D.C. (as pictured in Fig. 2(a)). The first map (Fig. 2(b)) is a 3-D evidence grid with an x, y, and z cell size of 0.125 meters. For demonstration purposes, the figure only displays a portion of the map. As can be seen, the 3-D evidence grid represents the smooth ramp as a series of broken intervals separated by a height of 0.125 meters. Because 0.125 meters is greater than the climbing height, this map will not allow the robot to accurately differentiate some ramps from obstacles or even identify many obstacles. Even at this coarse resolution, the map contains 262,144 cells ($256 \times 64 \times 16$).

Fig. 2(c) is a map of the same ramp created using the 2½-D map structure. The data incorporated into the map were obtained from the robot's 2 SICK™ laser sensors as it made a single pass down the ramp. Each cell containing usable data is displayed as shown in the figure (the shadings were added to help distinguish between neighboring cells) with each cell drawn from the recorded height to the height of its lowest neighbor.

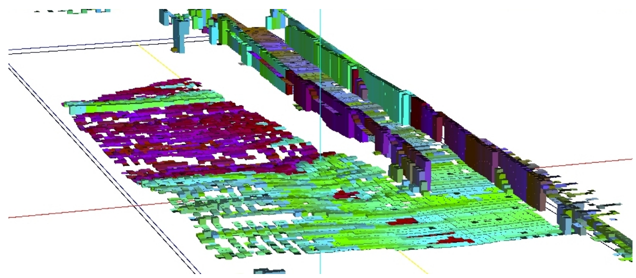
Using the maximum height of the occupied space at each location, the robot can determine where it can and cannot move by converting the map into a form similar to a 2-D evidence grid.

For each cell, if the difference between the height of that cell and the height of one of its neighbors is greater than the robot's maximum climbing distance, the transition is marked as an obstacle. Since each cell represents only a small portion of the world (0.125 meters in this map), it is

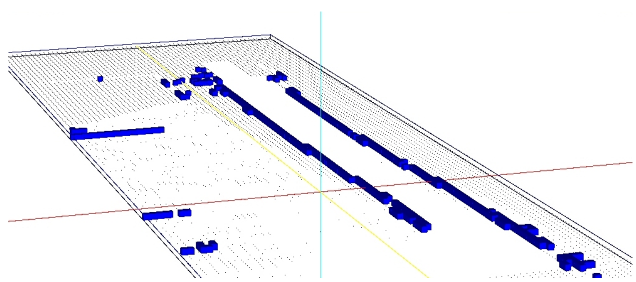
(a) Photograph of ramp.



(b) 3-D Evidence Grid with a trace of the robot's recorded altitude moving down the ramp.



(c) 2½-D map (with SICK readings).



(d) Converted 2½-D map.

Fig. 2. Ramp in back of Building 1 at NRL.

able to mark obstacles without limiting the areas around the obstacle that the robot can maneuver in. This method can also be used to determine negative obstacles such as cliffs. If a drop in elevation is beyond what the robot can handle safely, it will be able to determine that area as a non-traversable boundary.

Fig. 2(d) was created using this method. The black dots represent unknown areas, the white areas represent traversable spaces, and the blocks represent obstacles for the robot. The actual size of the original 2½-D map only contains 32,768 cells ($256 \times 64 \times 2$)—fewer than the 262,144 cells used by the coarse 3-D evidence grid.

B. Path Planning

Using the method discussed in the previous section to determine areas where the robot can and cannot move to, the 2½-D map structure can also be used to generate paths to a goal using a path planning algorithm that works on 2-D evidence grids, such as the TRULLA [7] algorithm. The converted 2½-D map (Fig. 2(d)) provides these algorithms with all the information it would need to perform its path planning.

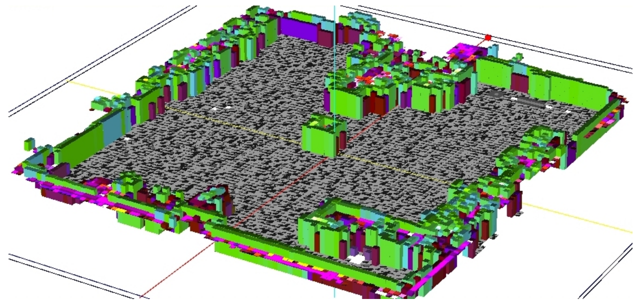
Some algorithms, including TRULLA, allow cells to be weighted to indicate more and less desirable paths. In the converted 2½-D map, the weight of each cell can incorporate the height difference between the current cell and those around it. This weighing system can help TRULLA determine the more desirable path in a rugged environment (with multiple curbs, hills, ramps, etc.).

C. Localization

To test the localization capabilities of the 2½-D map structure, we implemented a modified Continuous Localization [9] algorithm. Since this algorithm requires a long-term map of an environment, we created such a map of Room 105 (in Building 1 at NRL)[2] by continuously providing the robot with its exact location as it moved about the room and recorded its sensor readings. Fig. 3(a) is a panoramic view of the room and Fig. 3(b) is the long-term 2½-D map structure of that room. The maximum error in the generated map was less than 3 cells (0.375 meters). This map representation was created using a $128 \times 128 \times 2$ size grid .

The main modification made to the original Continuous Localization algorithm was in the way the short-term maps are compared to the long-term map—since the actual structure has changed. In a 2-D evidence grid registration process, the difference between the occupancy probabilities are used to align the maps. For the 2½-D model, we use the difference between the stored heights. The calculation of the match score for each map comparison is determined by taking the sum of the absolute difference between *valid* cell heights of the two maps[3]. If multiple registrations are found to produce a good match score, then the weighted average of these (based on the actual score) is taken to determine the *x*, *y*, and *yaw* correction to update the robot's location. Using the *x*, *y* correction, along with the dead reckoned location, the actual *x* and *y* position of the robot can be determined. That (*x*, *y*) cell location in the long-term map will provide the actual *z* location of the robot at the time of registration. The difference between the actual *z* value and the height at which the robot thought it was at will become the *z* correction for the robot. Since the ATRV-Jr. obtains its *roll* and *pitch* values directly from the sensors, they are not subjected to any systematic errors and are not corrected for in this registration.

(a) Panoramic Photograph of Room.



(b) 2½-D Long-Term Map of Room.

Fig. 3. Room 105 in Building 1 at NRL.

The robot was remotely controlled, while running a modified continuous location algorithm, around the room depicted in Fig. 3(a) (using Fig. 3(b) as its long-term map). After a single lap, the robot's actual position in the room was measured and its corrected and uncorrected pose estimates recorded. Comparing these values showed that the estimated position of the robot (with correction) placed it 0.047 meters from its actual position (with a 1° rotation error) while the uncorrected estimate was off by 0.693 meters (with a 6° rotation error). From this test, it is determined that the robot is able to localize itself using a 2½-D map structure within the tolerance range of the map (roughly the size of a cell—since that is the accuracy of the map) in a similar environment (a flat room).

V. Discussion and Future Work

As mentioned earlier, an assumption was made that objects in the environment were "complete" from the ground up—leaving no gaps directly below objects (such

[2] We used the indoor laboratory because it was easier to develop an accurate map to evaluate just the localization capabilities.

[3] Valid cells are those (*x*,*y*) locations which have a strong probability belief of existing as determined by a threshold.

as arches). This assumption was made to facilitate the initial development of the map structure and future developments of the 2½-D map should explore methods to represent "incomplete" objects such as arches.

Similar to the issue just mentioned, another item which should be explored is how to represent a multi-layered environment. In a 3-D environment, structures exists which can allow a robot to occupy a certain (x, y) location but at different altitudes (such as overhead passes). The 2½-D map structure currently does not provide the capability to represent such an environment. One possibility might be to stack multiple 2½-D maps on top of each other. This could create a structure similar to a 3-D evidence grid (which can represent multi-layered environments) but with the resolution capability and lower computational and memory requirements of the 2½-D map.

In addition, the testing of the 2½-D map structure's localization capability was conducted indoors. This facilitated the development of an accurate long-term map but also limits the implications of the results since the ground was relatively flat and smooth (not representative of most 3-D environments). To further explore the 2½-D map's ability to provide localization capabilities to a robot on a more 3-D terrain, more testing would have to be performed in such an environment and for long term navigation.

Once these issues are resolved, the frontier exploration aspect of autonomous mobile robotics would just be an extension of the 2-D algorithm using the converted 2½-D map.

VI. Conclusions

This paper described a 2½-D map structure to represent a 3-D environment for autonomous mobile robots. The 2½-D map structure was created by modifying a 2-D evidence grid to store a height and probability value for each (x, y) cell location. The alternative of extending the 2-D evidence grid to a full 3-D model had proved to be prohibitively expensive (in memory and computational requirements).

The new map structure was capable of storing an environment with varying ground elevations (a ramp was used for testing) to a high degree of precision while its memory requirements was only double that of a 2-D evidence grid. The robot was then able to use the map to perform obstacle detection and path planning in a 3-D environment—implementing existing algorithms created for the 2-D evidence grid. The map structure was also capable of providing the robot with the ability to localize in the original 2-D laboratory environment. With this 2½-D map, we have provided an initial look into a map representation structure that could be used in assisting a robot's transition from a constrained 2-D environment into a more 3-D world.

VII. References

[1] H. Choset and J. Burdick. Sensor-based exploration: The hierarchical generalized voronoi graph. *The International Journal of Robotics Research*, 19(2):96–125, February 2000.

[2] K. Graves. Continuous localization and navigation of mobile robots. Trident Report 247, US Naval Academy, Annapolis, MD, 1997.

[3] D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3D models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 2002.

[4] H. P. Moravec. Sensor fusion in evidence grids for mobile robots. *AI Magazine*, pages 61–74, 1988.

[5] H. P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. Technical Report CMU-RI-TR-96-34, CMU Robotics Institute, September 1996.

[6] H. P. Moravec. Robust navigation by probabilistic volumetric sensing. Technical report, CMU Robotics Institute, February 2002.

[7] R. Murphy, K. Hughes, E. Noll, and A. Marzilli. Integrating explicit path planning with reactive control for mobile robots using trulla. *Robotics and Autonomous Systems*, (27):225–245, 1999.

[8] H. Samet. An overview of quadtrees, octrees, and related hierarchical data structures. *Theoretical Foundations of Computer Graphics and CAD*, 40:51–68, 1988. NATO ASI Series F.

[9] A. C. Schultz and W. Adams. Continuous localization using evidence grids. *IEEE International Conference on Robotics and Automation*, pages 2833–2839, 1998.

[10] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000. IEEE.

[11] B. Yamauchi, A. Schultz, and W. Adams. Integrating exploration and localization for mobile robots. *Adaptive Behavior*, pages 217–229, 1999.