The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Multi-Volume Occupancy Grids: an Efficient Probabilistic 3D Mapping Model for Micro Aerial Vehicles

Ivan Dryanovski*, William Morris and Jizhong Xiao, *Senior Member, IEEE*

Abstract— **Advancing research into autonomous micro aerial vehicle navigation requires data structures capable of representing indoor and outdoor 3D environments. The vehicle must be able to update the map structure in real time using readings from range-finding sensors when mapping unknown areas; it must also be able to look up occupancy information from the map for the purposes of localization and path-planning. Mapping models that have been used for these tasks include voxel grids, multi-level surface maps, and octrees. In this paper, we suggest a new approach to 3D mapping using a multi-volume occupancy grid, or MVOG. MVOGs explicitly store information about both obstacles and free space. This allows us to correct previous potentially erroneous sensor readings by incrementally fusing in new positive or negative sensor information. In turn, this enables extracting more reliable probabilistic information about the occupancy of 3D space. MVOGs outperform existing probabilistic 3D mapping methods in terms of memory usage, due to the fact that observations are grouped together into continuous vertical volumes to save space. We describe the techniques required for mapping using MVOGs, and analyze their performance using indoor and outdoor experimental data.**

I. INTRODUCTION

One of the fundamental requirements for mobile robots is the ability to perform basic navigation tasks, such as moving from the current location to the desired location while avoiding obstacles. In 2D environments there is a number of well established solutions that produce satisfactory results. However, extending these solutions to 3D often increases computational and memory costs beyond the limits of real-time performance.

In some cases the problem can be simplified to the 2D case by assuming that the robot is in a structured environment where walls are purely vertical and cross sections are constant at various altitudes [1]. Another possible simplification that is often used for outdoor wheeled robots is to consider only the height of the terrain when performing mapping and path planning tasks; however, with this approach, vertically overlapping features such as tunnels and bridges can cause serious problems.

Our research focuses on the development of indoor MAVs which need to navigate unstructured indoor and outdoor environments, requiring the use of true 3D maps. This paper considers a novel technique for efficient 3D probabilistic mapping using a Multi-Volume Occupancy Grid or MVOG. The MVOGs have the following key characteristics. First, they are compact in size, allowing for efficient storage onboard, as well as transfer to other robots under bandwidth constraints. Second, they are updatable in real time by range-finding sensors such as laser scanners and depth cameras. Third, they provide information about free, occupied, and unknown space. Modeling both free and occupied space is important, because it allows the robot to overwrite previous erroneous sensor readings. Modeling unknown space is useful for tasks such as autonomous exploration, as well as safe path-planning.

The paper is organized such that, Section II reviews existing 3D mapping approaches. Next, Section III describes the algorithms for map construction and calculation of occupancy information. Section IV presents the experimental methods and results for creating maps of indoor and outdoor environments. Section V analyses open issues with MVOGs, and their possible solutions. Finally, Section VI summarizes the results of the paper.

II. RELATED WORK

A well established way for creating 2D maps is occupancy grids. A probabilistic method for integrating readings into the map can be found in [2]. All cells take on a continuous value between 0 (free) and 1 (occupied), and are initialized with a value of 0.5. When a sensor reports a certain distance, cells at that distance have their values increased, while cells that lie within the sensor ray area have their values decreased to mark the free space. The probability values are modified according to Bayesian update rules.

A slightly different mapping model for 2D maps is provided by reflection maps [2]. In a reflection map, each cell in the grid keeps track of two counters: one for the number of times a sensor beam was reflected in the cell ($hits$), and one for the number of times a sensor beam passed through the cell ($misses$). The ratio

$$\frac{hits}{hits + misses} \tag{1}$$

provides a probabilistic measure of the likelihood of a sensor reporting that cell as an obstacle. Maps produced under this model are very close to regular occupancy grids [2].

This work is supported in part by U.S. Army Research Office under grant No. W911NF-09-1-0565, and U.S. National Science Foundation under grants No. CNS-0619577 and No. IIS-0644127

*Ivan Dryanovski is with the Dept. of Computer Science, The Graduate Center, The City University of New York (CUNY), 365 Fifth Avenue, New York, NY 10016 (corresponding author, e-mail: idryanovski@gc.cuny.edu; phone: 917-400-2942)

William Morris is with the Electrical Engineering Department, City College of New York, Convent Ave & 140th Street, New York, NY 10031 (e-mail: morris@ee.ccny.cuny.edu; phone: 347-859-6447)

Jizhong Xiao is with the Electrical Engineering Department, City College of New York, Convent Ave & 140th Street, New York, NY 10031 (e-mail: jxiao@ccny.cuny.edu; phone: 212-650-7268)

978-1-4244-6676-4/10/$25.00 ©2010 IEEE

The direct extension of 2D occupancy grids to 3D are voxel occupancy grids. Work with 3D occupancy grids has been presented in [3], [4], and [5]. While voxel grids support all the algorithms developed for their 2D counterparts, they are often impractical due to their large memory requirements.

An alternative approach is to represent the environment using point clouds, where each point is created from a range reading provided by a laser scanner or a stereo-camera [6], [7]. However, this method does not model free and unknown space, and is thus only suitable for sensors that have very high accuracy. Moreover, the size of the map grows linearly and without an upper bound with the number of sensor readings.

An effort to create a more compact data structure is presented in [8] and [9] with their introduction of multi-level surface maps, or MLS maps. Similar to elevation maps ([10], [11]) MLS maps represent 3D structures as height values over a horizontal grid, but allow for the storage of vertically overlapping objects. While this is shown to greatly reduce the memory requirement, MLS maps only record positive sensor data, and provide no mechanism for decreasing the occupancy value of objects located on the map. Thus, any erroneous readings such as false sensor positives are never removed from the map.

Another approach to building 3D maps is by representing the environment using an octree. Work on octree mapping has been done by [12], [13], [14], [15], [16], and [17]. Octomap [18] provides an overview of existing octree approaches, and how they address issues such as updatability, map overconfidence, and compression. Octomap achieves probabilistic, compact 3D maps, but underperforms when compared to MVOGs in terms of size for the data sets we tested.

The MVOG mapping structure that we propose is closely related to multi-level surface maps and reflection maps. Similarly to MLS, MVOGs group readings into continuous vertical volumes, which are placed over a horizontal grid of fixed resolution. Unlike MLS, however, MVOGs record both positive and negative readings, grouping them into distinct positive and negative volumes. The occupancy information is computed similarly to the reflection map model. Note that although we use the reflection model, which does not directly model occupancy, we informally refer to our maps as "occupancy" maps modeling occupancy probabilities, due to the similarity of the two models.

III. MULTI-VOLUME OCCUPANCY GRIDS

A. Volume List Representation

A multi-volume occupancy grid consists of a 2D grid G of square cells c_{ij}, $i, j \in \mathbb{Z}$ lying in the xy-plane. Any point $\mathbf{p} = [p_x, p_y, p_z]^T$, $\mathbf{p} \in \mathbb{R}^3$ projects onto a cell c_{ij} such that $i \leq sp_x < i + 1$, $j \leq sp_y < j + 1$. where s is a constant scaling factor between the world and grid coordinates [8]. Each cell contains two lists of volumes: ${}^+\mathcal{V}_{ij} = \{{}^+V_{ij}^0, {}^+V_{ij}^1 \ldots {}^+V_{ij}^n\}$ and ${}^-\mathcal{V}_{ij} = \{{}^-V_{ij}^0, {}^-V_{ij}^1 \ldots {}^-V_{ij}^m\}$, of sizes n and m respectively. The list ${}^+\mathcal{V}_{ij}$ contains volumes representing positive (obstacle) readings, while ${}^-\mathcal{V}_{ij}$ contains volumes representing negative (free space) readings.

Each volume V is defined using three values: the height of its bottom face $z_V^{bot} \in \mathbb{R}$, the height of its top face $z_V^{top} \in \mathbb{R}$, and occupancy mass $m_V \in \mathbb{R}, m_V \geq 0$. We derive a fourth value, the occupancy density ρ_V. The occupancy mass of a volume corresponds to the amount of sensory information the volume has received. The occupancy density corresponds to the amount of sensory information per unit space.

$$\rho_V = \frac{m_V}{(z_V^{top} - z_V^{bot})A_{c_{ij}}} \tag{2}$$

where $A_{c_{ij}}$ is the unit area of the grid cell.

For positive volumes, the occupancy mass comes from sensory information obtained from obstacle readings. For negative volumes, the mass comes from information from free-space readings. All new volumes start off with a density of 1. The occupancy mass of any volume can only increase over time. For example, if we detect a certain point of space as free, we would not decrease the occupancy mass of a positive volume containing the point, but would rather create a negative volume for that region of space. The exact algorithm for manipulating the size and occupancy densities of volumes over time is described in Section III-B.

We impose the additional restrictions on volume lists that:

- Each volume has a height greater than or equal to 1.
- No two volumes in the same volume list overlap.
- The gap between any two volumes in the same list is greater than 1.

The size restriction on minimum volume and gap size is chosen to be 1, which is the same as the resolution restriction in the x and y directions. In this way, we guarantee that the minimum vertical resolution the same as our horizontal one. However, the vertical resolution is effectively better than the horizontal one, since volumes are allowed to start and end at non-integer z values.

Fig. 1 shows a possible list combination for grid cell $c_{0,1}$, displaying the positive (red) and negative (blue) volumes side by side. Note that volumes from the positive list can overlap

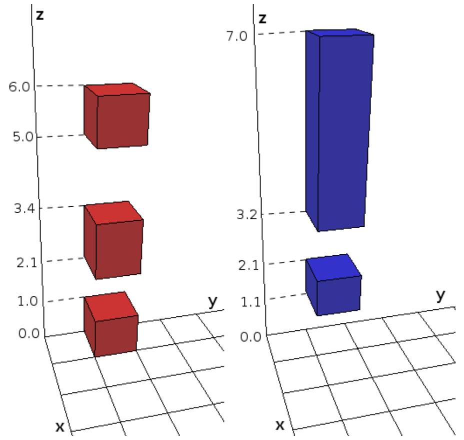

Fig. 1. Positive (red) and negative (blue) volume lists for cell $c_{0,1}$

with volumes from the negative list, corresponding to the situation that the sensors provided contradicting observations for the same region in space.

B. Updating From Laser Data

The process of building a MVOG from range readings provided by a laser scanner consists of three steps. First, we rasterize each individual laser reading to obtain which grid cells it crosses. Next, we create and insert new positive and negative volumes into the volume lists of the corresponding grid cells. Last, we examine the modified volume lists, and apply the constraints defined in Section III-A. The process diagram of the update process is shown in Fig. 2.

1) Rasterization: An individual laser scan can return either an obstacle reading at a distance d or an out-of-range reading, where $d > d_{max}$. From the position $\mathbf{L} \in \mathbb{R}^3$ of the laser and the orientation at the time of the scan, we can calculate the end point of the laser ray $\mathbf{L}' \in \mathbb{R}^3$. In this context, an out-of-range reading means that the space between $\mathbf{L}$ and $\mathbf{L}'$ is free, and an obstacle reading carries the same free-space information, with the additional information that $\mathbf{L}'$ is occupied.

By projecting the ray from $\mathbf{L}$ to $\mathbf{L}'$ onto the xy-plane, we obtain a list cells $C = \{c_{i_0 j_0}, c_{i_1 j_1} \dots c_{i_k j_k}\}$ that the ray crosses, where C is a subset of the grid G, C has a length of k, and $c_{i_k j_k}$ is the grid cell where the laser ray terminates. We can also calculate the heights where the the laser ray enters the space above each cell z_{ij}^{enter} and the height where it leaves it, z_{ij}^{exit} [19]. Note that the laser ray never leaves the last cell $c_{i_k j_k}$, thus we only obtain an entry height value. Instead of an exit height value, we will use the termination height of the laser ray L'_z.

2) Creating new volumes: All new volumes are created with with an occupancy density ρ of 1. By knowing the height of the volume and using (2), we can instantiate all new volumes with the appropriate occupancy mass m.

When adding an out-of-range reading to the map, we create a new volume V for each cell c_{ij} in C, apart from the last cell, $c_{i_k j_k}$.

$$z_V^{bot} = min(z_{ij}^{enter}, z_{ij}^{exit}) \tag{3a}$$

$$z_V^{top} = max(z_{ij}^{enter}, z_{ij}^{exit}) \tag{3b}$$

For the last cell, we have

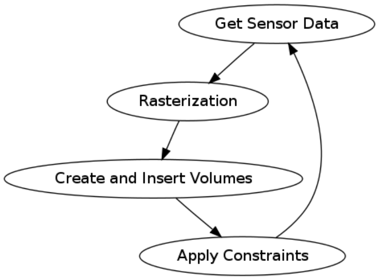

Fig. 2. Process diagram for updating the MVOG map

$$z_V^{bot} = min(z_{ij}^{enter}, L'_z) \tag{4a}$$

$$z_V^{top} = max(z_{ij}^{enter}, L'_z) \tag{4b}$$

Next, we insert the newly created V into the corresponding negative volume list ${}^{-}\mathcal{V}_{ij}$

When adding an obstacle reading, we repeat the above procedure for each cell in C, apart from the last cell, $c_{i_k j_k}$. We need to insert a positive volume of height 1, but depending on the slope of the laser ray, we might need to create an additional negative volume in the same cell. If $abs(z^{enter} - L'_z) \leq 1$, then we create a volume V such that

$$z_V^{bot} = L'_z - 0.5 \tag{5a}$$

$$z_V^{top} = L'_z + 0.5 \tag{5b}$$

and insert it into the positive volume list ${}^{+}\mathcal{V}_{i_k j_k}$. On the other hand, if $abs(z^{enter} - L'_z) > 1$, we create and insert V in the same manner, but also create the additional negative volume volume V' to pad the distance between z^{enter} and V, and insert it into ${}^{-}\mathcal{V}_{i_k j_k}$. The two cases are illustrated in Fig. 3.

3) Constraint application: The last step in the process is to go through each modified volume list and make sure all the volumes satisfy the constraints we defined in Section III-A. The first constraint refers to the minimum volume size. Any volume V such that $z_V^{top} - z_V^{bot} < 1$ is replaced with a volume V^* of height 1 in the following manner:

$$z_{V^*}^{bot} = \frac{1}{2}(z_V^{bot} + z_V^{top}) - 0.5 \tag{6a}$$

$$z_{V^*}^{top} = \frac{1}{2}(z_V^{bot} + z_V^{top}) + 0.5 \tag{6b}$$

Next, we satisfy the constraint that no two volumes in the same volume list overlap by merging them together. Merging two volumes is the main mechanism for incrementally fusing

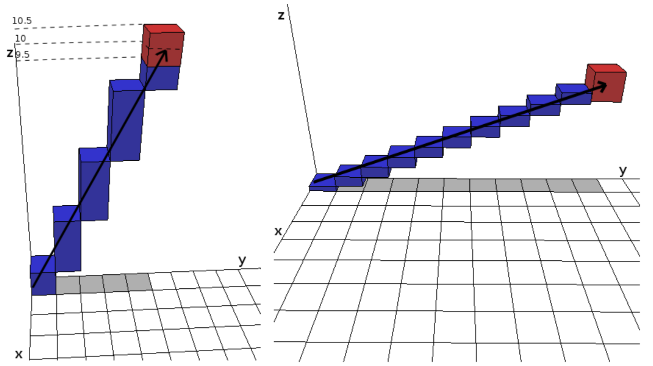

Fig. 3. The two different cases when rasterizing an obstacle reading from a laser. Grid cells in C are marked in gray. The left diagram shows a rasterization of a laser ray from $[0, 0, 0]^T$ reading an obstacle at $[0, 4.5, 10]^T$, requiring a positive and a negative volume to be inserted in the last cell. The right diagram shows a rasterization of a laser ray from $[0, 0, 0]^T$ reading an obstacle at $[0, 10.5, 4]^T$. requiring only a positive volume to be inserted.

in new sensor information. Two volumes V^A and V^B overlap if $z^{bot}_{V^B} \in [z^{bot}_{V^A}, z^{top}_{V^A}]$ or $z^{top}_{V^B} \in [z^{bot}_{V^A}, z^{top}_{V^A}]$. The resulting volume has an occupancy mass equal to the sum of the occupancy masses of the added volumes. Any two overlapping volumes are replaced by a volume V^* such that

$$V^* = V^A \cup V^B \tag{7a}$$

$$m_{V^*} = m_{V^A} + m_{V^B} \tag{7b}$$

Last, we satisfy the constraint that the gap between any two volumes is bigger than 1. Two volumes V^A and V^B are too close if $z^{bot}_{V^B} - z^{top}_{V^A} \in (0, 1]$. We create a new volume V^{GAP} corresponding to the gap between V^A and V^B, initialized with a density of 1. Then we merge all three volumes into one continuous V^* such that

$$V^* = V^A \cup V^{GAP} \cup V^B \tag{8a}$$

$$m_{V^*} = m_{V^A} + m_{V^{GAP}} + m_{V^B} \tag{8b}$$

C. Extracting Probabilistic Occupancy Information

MVOGs allow us to extract probabilistic information about the occupancy of each point in space. The two pieces of information that we maintain for any point are the occupancy densities of the positive and negative volumes containing the point, if such volumes exist.

Having defined the positive and negative density functions, we can define the occupancy probability $p \in [0, 1]$ of a point $\mathbf{p} = [p_x, p_y, p_z]^T$. Let $\rho^+_{\mathbf{p}}$ be the occupancy density of the positive volume containing $\mathbf{p}$ (or 0 if no such volume exists). Similarly, let $\rho^-_{\mathbf{p}}$ be the occupancy density of the negative volume containing $\mathbf{p}$. Then, the probability $p(\mathbf{p})$ is the ratio between $\rho^+_{\mathbf{p}}$ and the sum of $\rho^+_{\mathbf{p}}$ and $\rho^-_{\mathbf{p}}$.

$$p(\mathbf{p}) = \begin{cases} \frac{\rho^+_{\mathbf{p}}}{\rho^+_{\mathbf{p}} + \rho^-_{\mathbf{p}}} & \text{if } \rho^+_{\mathbf{p}} + \rho^-_{\mathbf{p}} > 0 \\ \text{unknown} & \text{if } \rho^+_{\mathbf{p}} + \rho^-_{\mathbf{p}} = 0 \end{cases} \tag{9}$$

If both ρ^+ and ρ^- in (9) are equal to 0, corresponding to the situation when there are neither positive nor negative observations for that point in space, we return an $unknown$ probability.

Using the probability function p, we can construct a standard 2D occupancy grid for any plane in 3D space. The 2D grid can then be used with existing algorithms - for example, localization using a laser that has an arbitrary orientation in space.

D. Time Complexity Analysis

Since occupancy volumes are stored in the form of a sorted array, the insertion and lookup operations for volumes are executed in linear time. For a grid cell with lists $^+\mathcal{V}_{ij}$ and $^-\mathcal{V}_{ij}$ of length n and m respectively, inserting a positive volume has a runtime of $O(n)$, inserting a negative volume has a runtime of $O(m)$, and the calculation of the probability function p has a runtime of $O(m + n)$.

The linear run time can be improved by storing the volume lists in a different structure, such as a skip lists, which provides $O(log(n))$ lookup and constant insertion time [20]. However, we find in practice that the average value of $n+m$ for our test environments remains low. This effectively reduces the runtime of the insertion and lookup operations to constant time.

E. Map Overconfidence

When a section of the map has received a large number of negative readings, it normally takes the same amount of positive readings to raise the probability ratio to 0.5 (and vice versa). Thus, maps can become overconfident. This can lead to serious problems, especially when there are dynamic obstacles in the environment. For example, if the robot has recorded a hallway as free multiple times, and a person suddenly walks in, the robot will not be able to correct the section on the map that the person is occupying until a large number of sensor readings have been collected.

To overcome this issue, occupancy grids that use the Bayesian update model in conjunction with log-odds propose clamping, which effectively sets a limit on the number of readings it takes to change the probability value of a cell in the situation described above [18].

MVOGs overcome the issue of map overconfidence by introducing a decay factor in the map. Both positive and negative volume masses are periodically multiplied by a constant factor k_d, smaller than 1. Decaying the map has the effect that new sensor readings have a higher weight than previous sensor readings. Note that for map areas that do not receive new sensor readings, the occupancy probability remains the same before and after the update step, since both the negative and positive volume masses are scaled down equally.

IV. EXPERIMENTS

A. 3D Maps

We tested the performance of MVOGs using two experiments.

For the first experiment, we use readings from a Swissranger 4000 depth-camera mounted on a quadrotor MAV. Since the depth-camera is too heavy to be carried in flight, we maneuver the MAV by manually carrying it indoors, obtaining multiple views of an office room. The room has two desks and a person sitting on a chair in front of a computer, as well as various clutter. The 6 DoF pose of the UAV is determined by a combination of inertial readings and scan matching [21].

The resulting map is shown in Fig. 4. The left image shows the raw point cloud data. The middle image shows the positive and negative volumes. The right image shows areas of the map where the probability of occupancy is greater than 0.5 (obstacles).

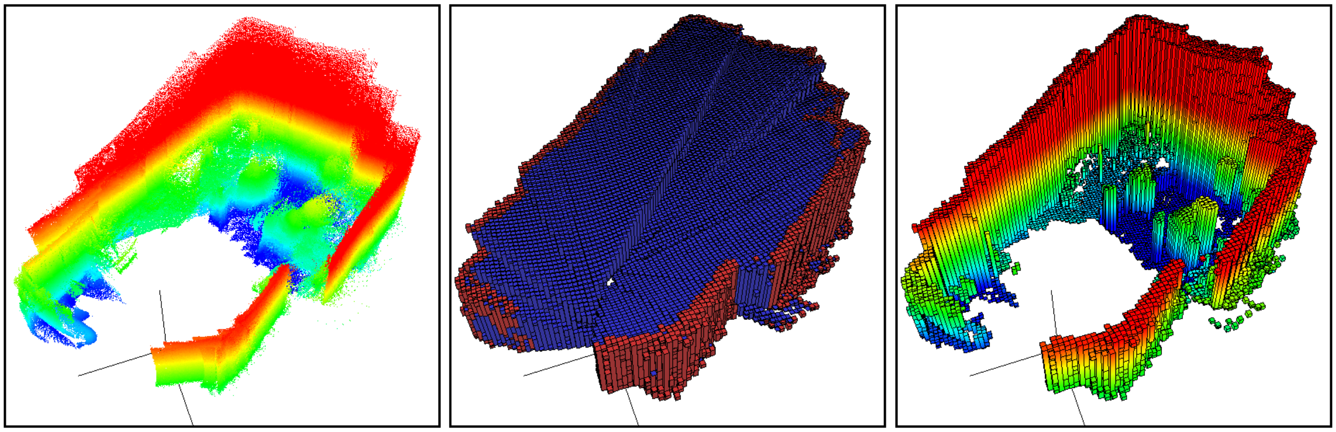

Fig. 4. Stages in the generation of a 3D MVOG map of an office room. The left image shows the raw point cloud data from a Swissranger camera. The middle image shows the positive (red) and negative (blue) volumes. The right image shows areas of the map where the probability of occupancy is greater than 0.5 (obstacles). The volumes in the right image are colored by height.

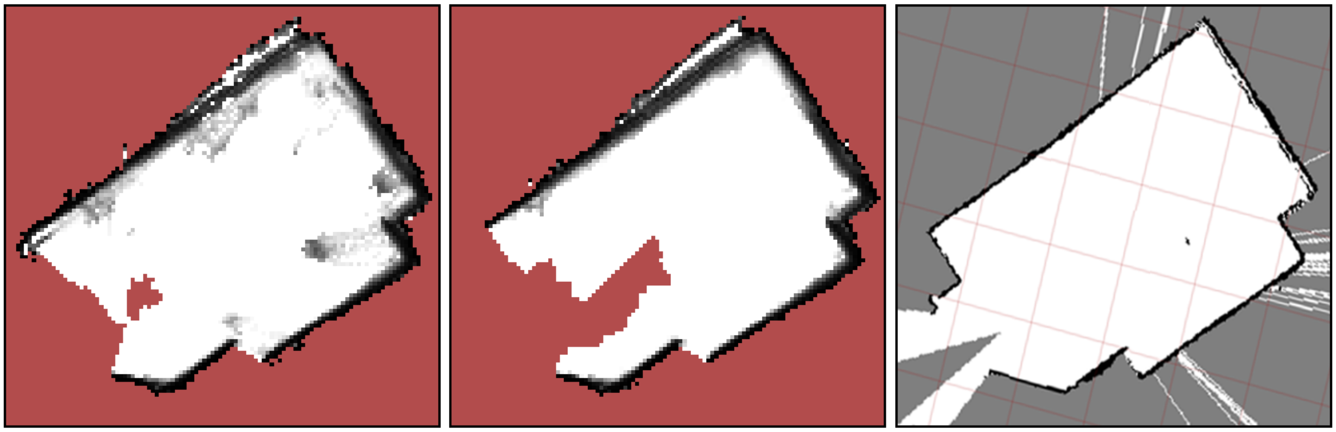

Fig. 5. Occupancy grids obtained from the MVOG in Fig. 4 by taking a cross-section at different heights. The left image shows the occupancy at 1.6 meters above the floor. The entire room is navigable. The middle image shows the occupancy at 1.5 meters, where the top of a person's head and room clutter are visible. The right image provides a high-resolution 2D map of the room for comparison. The 2D map was created by a independent laser data and a particle SLAM algorithm.

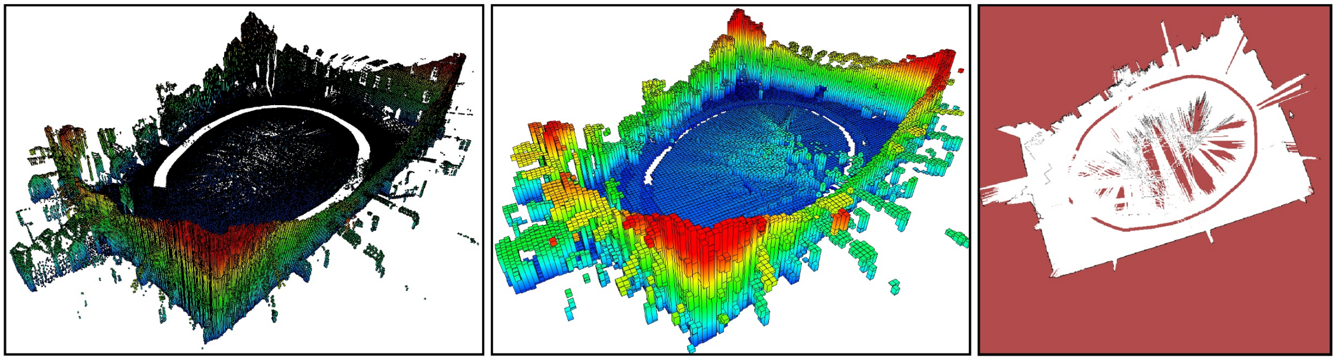

Fig. 6. MVOG of the New College Dataset (Epoch A, one loop). The left image shows $0.10m$ resolution. The middle image shows $0.50m$ resolution. The right image shows the occupancy map obtained by taking a cross-section of the $0.10m$ MVOG, taken at the height of the laser scanners. Note that the double walls and incorrect geometry of the loop are a result of the drift in the odometric pose estimation provided in the dataset.

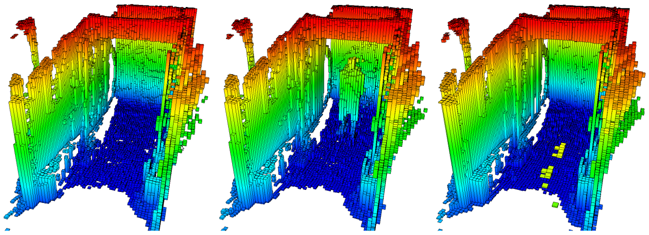

Fig. 7. Experiment demonstrating the addition and removal of dynamic objects to the map. The Swissranger scans an empty hallway (left image), a person walks into the frame (middle image), and the person has walked out of view (right image).

In Fig. 5, we show a cross-section of the map taken at 2 different heights. For comparison, we also show a high-resolution occupancy grid created by a 2D particle-based SLAM algorithm [22] of the same room, using independent data from a laser scanner.

In the second experiment, we build an MVOG using a subset of the New College Dataset [23], collected by a ground robot outdoors. We use the first 17500 laser scan readings, corresponding to the first loop around the Epoch A campus. We use the raw odometry data provided from the robot.

Fig. 6 shows the resulting MVOG map using different resolutions, as well as a cross-section of the map taken at the height of the laser scanners. Note that the double walls and incorrect geometry of the loop are a result of the drift in the odometric pose estimation provided in the dataset.

We perform an additional experiment to demonstrate the updatability of the MVOG in the presence of dynamic obstacles. In that experiment, we collect readings from a Swissranger 4000 depth camera positioned statically. The camera observes a hallway while a person walks in and out of view (Fig. 7).

All experiments were preformed using ROS (Robot Operating System) [24] as an underlying architecture to transmit messages between various components of the system.

B. Size comparison

In this section, we analyze the memory consumption of the MVOGs, compared to raw point cloud, voxel grid, and Octomap representations. We have made at effort at a fair benchmarking, even though we are comparing inherently different data structures. The memory sizes provided for all four data structures assume single-precision floating point representations. For voxel grids, the reported size is for the minimum 3D grid that accommodates all the data. For MVOGs, the reported size is for the minimum 2D grid that accommodates the data, and no restriction on the vertical bounds. For Octomap, we report the size of the pruned (lossless compression) tree needed to represent the data. Out-of-range sensor readings present in the experiments are disregarded in all four representations.

The results from the indoor and outdoor data sets, mapped at different resolutions, are summarized in Table I.

TABLE I

DATA SIZE ANALYSIS

	Point cloud	**Voxel grid**	**Octomap**	**MVOG**
New College				
$1.00m$ res.	27.430 MB	0.422 MB	0.189 MB	0.150 MB
$0.50m$ res.	27.430 MB	3.245 MB	0.785 MB	0.588 MB
$0.10m$ res.	27.430 MB	390.750 MB	56.309 MB	16.438 MB
Office room				
$0.10m$ res.	16.823 MB	0.259 MB	0.273 MB	0.080 MB
$0.05m$ res.	16.823 MB	2.031 MB	1.597 MB	0.328 MB
$0.02m$ res.	16.823 MB	30.175 MB	15.877 MB	2.730 MB

C. Time analysis

The insertion time for the $0.10m$ resolution office room experiment was on average $0.122s$ per Swissranger scan, where each scan contained an average of 24510 range readings. This results in an average insertion rate of 200778 readings per second. The insertion time for the $0.10m$ resolution outdoor dataset was on average $0.008s$ per 2D laser scan, where each scan contained 180 range readings. This results in an average insertion rate of 21886 readings per second. Note that this includes a high number of out-of-range ($50m$) readings in every scan. Timing was carried out on a standard desktop CPU (dual Intel Xeon Processor E5504, 2.0GHz).

V. DISCUSSION AND FUTURE WORK

In this section, we examine the limitations and drawbacks of Multi-Volume Occupancy Grids that have not been addressed in our implementation. Where applicable, a solution is provided that can be implemented in the future.

A. Map Dimensions

As with other grid-based mapping approaches, the size of the grid must be chosen in advance, so that it is big enough to accommodate the map being built. Alternatively, the map can be dynamically expanded when needed, at the cost of copying over previous data. While this approach is usually prohibitive for voxel grid representations due to their large size in memory, the compact nature of MVOGs allows for faster copying.

B. Rasterization errors

When rasterizing a laser beam into positive and negative volumes, one often ends up adding free-space information in incorrect areas of the map, due to rasterization errors. This problem is common to all mapping techniques that model discretized free space, and has been discussed in detail in [18]. The solution provided for Octomap is to treat scans in groups, and only add negative information where it does not conflict with positive information within that scan. This method can be applied to data received from laser scanners on a static pan/tilt platforms, or range-finder cameras. A similar approach is applicable to MVOGs, by discarding any negative volumes that conflict with positive volumes, within each scan group.

C. Incorrect Volume grouping

A more significant drawback of MVOGs is that there exists no mechanism for breaking up existing volumes. Thus, a volume can be erroneously extended in size by a false reading. Alternatively, two correct volumes can be merged together and averaged if a false reading is placed in the vertical space between them. This can lead to corruption of the map data in highly dynamic environments, or when a high amount of sensor noise is present. A technique to resolve this issue could involve breaking up volumes dynamically, or disallowing volumes to be merged until some criteria is met.

VI. CONCLUSIONS

In this paper we presented Multi-Volume Occupancy Grid maps, a novel method for representing 3D environments for the purposes of micro aerial vehicle mapping and navigation. MVOGs explicitly store positive and negative sensor readings and group them together into volume lists attached to a 2D grid. This allows the incremental addition of obstacles, as well as addition of free space, in a model closely related to 2D reflection maps. We described a method for updating the maps with data from rangefinder senors, and a method for extracting probabilistic occupancy information for any point in space.

We also presented MVOGs built using indoor and outdoor data. We showed that the MVOGs can accurately describe occupancy information in 3D environments. They require significantly less memory space than existing mapping methods, and can be updated in real time.

The open-source software developed as part of the research is available at our website `http://robotics.ccny.cuny.edu/`.

REFERENCES

[1] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp.2878-2883, 12-17 May 2009

[2] Sebastian Thrun, Wolfram Burgard and Dieter Fox, *Probabilistic Robotics*, MIT Press, Cambridge, 2005.

[3] N. Duffy, D. Allan, and J. T. Herd, "Real-time collision avoidance system for multiple robots operating in shared work-space," *Computers and Digital Techniques, IEE Proceedings E* , vol.136, no.6, pp.478-484, Nov 1989

[4] Y. Roth-Tabak and R. Jain, Building an environment model using depth information, *Computer*, vol. 22, no. 6, pp. 8590, Jun 1989.

[5] H. Moravec, Robot spatial perception by stereoscopic vision and 3D evidence grids, Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-96-34, September 1996.

[6] D. Cole and P. Newman, Using laser range data for 3D SLAM in outdoor environments, in *Proc. of the IEEE Int. Conf. on Robotics And Automation (ICRA)*, 2006, pp. 15561563.

[7] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, 6D SLAM3D mapping outdoor environments: Research articles, *J. Field Robot.*, vol. 24, no. 8-9, pp. 699722, 2007.

[8] R. Triebel, P. Pfaff, and W. Burgard, Multi-level surface maps for outdoor terrain mapping and loop closing, in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[9] C. Rivadeneyra, I. Miller, J. R. Schoenberg, and M. Campbell, "Probabilistic estimation of Multi-Level terrain maps," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on* , pp.1643-1648, 12-17 May 2009

[10] R. Hadsell, J. A. Bagnell, and M. Hebert, Accurate rough terrain estimation with space-carving kernels, in *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[11] M. Hebert, C. Caillas, E. Krotkov, I. S. Kweon, and T. Kanade, Terrain mapping for a roving planetary explorer, in *Proc. of the IEEE Int. Conf. on Robotics And Automation (ICRA)*, vol. 2, May 1989, pp. 9971002.

[12] D. Meagher, Geometric modeling using octree encoding, *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129147, 1982.

[13] J. Wilhelms and A. Van Gelder, Octrees for faster isosurface generation, *ACM Trans. Graph.*, vol. 11, no. 3, pp. 201227, 1992.

[14] J. Fournier, B. Ricard, and D. Laurendeau, Mapping and exploration of complex environments using persistent 3D model, in *Computer and Robot Vision, 2007*. Fourth Canadian Conf. on, 2007, pp. 403410.

[15] N. Fairfield, G. Kantor, and D. Wettergreen, Real-time SLAM with octree evidence grids for exploration in underwater tunnels, *Journal of Field Robotics*, 2007.

[16] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger, 3D forward sensor modeling and application to occupancy grid based sensor fusion, in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 20592064.

[17] P. Payeur, P. Hebert, D. Laurendeau, and C. Gosselin, Probabilistic octree modeling of a 3-d dynamic environment, in *Proc. of the IEEE Int. Conf. on Robotics And Automation (ICRA)*, 1997.

[18] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: A Probabilistic, Flexible, and Compact 3D Map Representation for Robotic Systems" in *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010. Software available at http://octomap.sf.net/.

[19] R. Yagel, D. Cohen,and A. Kaufman, "Discrete ray tracing," *Computer Graphics and Applications, IEEE* , vol.12, no.5, pp.19-28, Sep 1992

[20] W. Pugh. "Skip lists: a probabilistic alternative to balanced trees". *Communications of the ACM* 33 (6): 668676, June 1990.

[21] W. Morris, I. Dryanovski, and J. Xiao, "3D Indoor Mapping for Micro-UAVs Using Hybrid Range Finders and Multi-Volume Occupancy Grids", in *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[22] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters", *IEEE Transactions on Robotics*, 2006

[23] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, The new college vision and laser data set, *International Journal for Robotics Research (IJRR)*, vol. 28, no. 5, pp. 595599, May 2009.

[24] "ROS - Robot Open Source". Willow Garage. http://www.willowgarage.com/pages/software/ros-platform. 2010