3D Indoor Mapping for Micro-UAVs Using Hybrid Range Finders and Multi-Volume Occupancy Grids

Article · January 2010

CITATIONS
30

READS
701

3 authors, including:

Jizhong Xiao
City College of New York
154 PUBLICATIONS **2,333** CITATIONS

SEE PROFILE

3D Indoor Mapping for Micro-UAVs Using Hybrid Range Finders and Multi-Volume Occupancy Grids

William Morris, *Member, IEEE & ASME*
Dept. of Electrical Engineering
The City College of New York
The City University of New York
New York, NY 10031
Email: morris@ee.ccny.cuny.edu

Ivan Dryanovski
Dept. of Computer Science
The Graduate Center
The City University of New York
New York, NY 10016
Email: idryanovski@gc.cuny.edu

Jizhong Xiao, *Senior Member, IEEE*
Dept. of Electrical Engineering
The City College of New York
The City University of New York
New York, NY 10031
Email: jxiao@ccny.cuny.edu

***Abstract*— Autonomous micro-UAV navigation requires techniques that allow for the accurate mapping of unstructured 3D environments such as stairwells, tunnels, and caves. As a step towards that goal, this paper presents a system to build three-dimensional maps of rectilinear environments, where the horizontal cross-section of the world is invariant at different heights. The assumption that the environment is structured makes our approach suitable for mapping of indoor spaces. One of the largest challenges in 3D mapping is correctly estimating the full 6-DOF pose of the vehicle. We present an approach that estimates the pose by fusing the information of an altimeter, an IMU, and a horizontally-mounted laser range-finder. A key step in the estimation is an orthogonal projection of the laser scan data, allowing for accurate scan matching in 2D. We also propose a novel map data structure called a Multi-Volume Occupancy Grid. 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 sensor information. In turn, this enables extracting more reliable probabilistic information about the occupancy of 3D space. Observations are grouped together into continuous vertical volumes, which makes this new data structure considerably more space-efficient than point cloud or voxel-grid representations.**

I. Introduction

The *CityFlyer* (Fig. 1) is a research project at the City College of New York to develop a micro-UAV that is capable of autonomous flight in a variety of three-dimensional environments. These environments impose significant challenges for future research. In this paper, we present our progress towards 6-degree-of-freedom pose estimation and 3D mapping of GPS-denied, indoor environments. Here, we consider one of the less complex environments such as building floors and offices, where one can assume that most obstacles have the same profile at different heights.

The most challenging part of 3D mapping is obtaining the full six degree of freedom pose of the robot between each scan. We can obtain much of this information from odometric sensors. An on-board IMU provides reasonably accurate information about the heading, elevation and bank angles. The altitude is obtained by using a barometric pressure altimeter. This provides a direct means of obtaining measurements of four of the six degrees of freedom.

We present a method to obtain the displacement in the horizontal x and y directions through registration of scans from a laterally-mounted 2D laser scanner. As a preliminary step, we project the laser scans onto the xy-plane, to make them independent of the roll and pitch motion of the vehicle. We then use scan matching to obtain an estimate of the horizontal displacement of the vehicle. This estimation is used as input to a 2D SLAM package [1], and use the output of the SLAM algorithm serves as our final x and y pose component. Once the full 6-DOF pose is obtained, filtered data returned by a Swiss Ranger 4000 depth-camera can then be integrated into our 3D map.

Fig. 1. The *CityFlyer* micro-UAV platform, baser on the Ascending Technologies Pelican quadrotor, currently equipped with a Hokuyo laser range-finder and a Swissranger 4000 depth camera.

For representing maps of 3D environments, we have designed a novel system called multi-volume occupancy grids, or MVOGs. MVOGs store observations about occupied and free space separately. We describe the data structure in detail, and explain how to build MVOG maps from laser readings. The process we describe works for any sensor that can be approximated to a collection of laser beams, where the end of each beam has a very high probability to be occupied, and the space that the beam traversed has a very high probability of being free. Sensors we have used to date to build MVOGs include several laser rangefinders, as well as a Swissranger 4000 camera. We believe that any time-of-flight camera, including RGB-D cameras, will also work. This data structure will provide a means to implement an real-time mapping and navigation system that can fit within the constraints of the

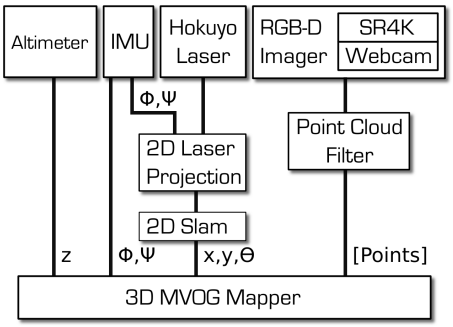

Fig. 2. A system diagram of the entire pose estimation and mapping process.

on-board flight hardware.

A system diagram of the entire pose estimation and mapping process is displayed in Fig 2. This paper is organized as follows: Section II presents related work done on the pose estimation and 3D mapping problems. Section III discusses in detail the process for obtaining the 6-DOF pose of the vehicle, paying particular attention to the laser projection and matching necessary to obtain the x and y components. Section IV discusses the 3D Multi-volume occupancy grid data structure. Section V presents our experimental methods and results, and Section VI discusses our conclusions and future work.

II. Related Work

A. Pose Estimation

Multiple works, including [2], present solutions to the 6-DOF pose estimation problem for ground robots, usually involving scan registration in 3D. However, the implementations are unsuitable for micro-UAVs due to computational and payload restrictions.

Previous works [3] and [4] both use 2D scan matching to perform 2D simultaneous localization and mapping (SLAM) on-board micro-UAV-s. This approach does not address the errors introduced in the scan matching because of varying bank and elevation angles. Moreover, this implementation maintains a single 2D map of the world, and is therefore unsuitable for true 3D path planning.

B. Mapping

3D MVOG maps are similar in concept to 2D occupancy grids in which the value of each cell c_{ij} represents the probability that the corresponding world space is occupied by an obstacle.

A probabilistic method for integrating readings into the map can be found in [5]. 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.

An alternative approach is presented in [6]. Instead of using continuous probability values, integer values are used for each cell, representing the number of times that cell has been reported as an obstacle. Cells are initialized with a value of 0, which is considered free space. Cells with a high count are considered occupied. The higher the count, the more certain we are that the cell is an obstacle. This results in a pseudo-probabilistic representation, which has been shown to produce accurate mappings over time [6]. We adopt a similar incremental pseudo-probabilistic approach in the MVOG.

The direct extension of 2D occupancy grids to 3D are voxel occupancy grids [7]. While they support all the algorithms developed for their 2D counterparts, they are often impractical due to their size and computational requirements.

An effort to create a more compact data structure is presented in [8] and [9] with the introduction of multi-level surface maps, or MLS maps. MLS maps represent 3D structures as in a manner similar to elevation maps, but allow for the storage of vertically overlapping objects. While this is shown to greatly reduce the size of the maps, MLS maps only record positive sensor data, and thus provide no mechanism for decreasing the occupancy value of objects located on the map. As a consequence, any erroneous readings such as false sensor positives are never removed from the map.

III. 6-DOF Pose Estimation

A. Overview

We obtain measurement of the orientation and height of the vehicle using on-board IMU and altimeter sensors. However, there are no proprioceptive sensors available for obtaining the x and y displacement.

Laser scan data is often used to calculate the motion of mobile robots in static environments. Most scan-matching implementations assume that the robot only moves in the xy-plane, and calculate the translation and rotation that best maps sets of points from two consecutive laser scans onto each other.

Attempting to apply this technique directly to robots in 3D presents some key challenges. The vehicle needs to be equipped with a sensor that provides 3D depth information, because the data obtained by statically-mounted planar laser scanners is not sufficient to correctly determine the change in the 3D pose. On many ground robots, a 3D scan can be produced by mounting a high end, fast-scanning laser range-finder on a panning platform, but this approach is impractical on-board micro-UAVs, due to the limited payload capabilities. An alternative is to use a 3D depth camera, such as the Swissranger 4000, to obtain a 3D point cloud for scan matching. However, the additional degrees of freedom (z, bank angle, height angle) make the scan matching search space orders of magnitude larger. This imposes a computational cost currently beyond the real-time capabilities of a micro-UAV's on-board processor.

Therefore, in the this section, we present a method for obtaining the x and y pose components through 2D scan matching.

B. Laser Projection

In most indoor environments, obstacles look the same regardless of the height, due to the rectilinear property. If we know the 3D orientation of the laser scanner, we can reduce the problem of 3D scan matching to 2D by projecting the scans onto the xy-plane. The projection step is fairly simple: from a set of points S in the world reference frame in Cartesian coordinates, we can obtain an orthogonally projected set S^* by discarding the z value of each point in S. Then, we can perform scan matching on two consecutive S^* sets. However, most scan matching algorithms do not work with points in a Cartesian world frame, but with points in a polar laser frame. Thus, we present an algorithm to transform polar-coordinate laser readings taken by a laser L in a frame $laser$, to a laser readings taken by a virtual laser scanner L^*, with a frame $ortho$.

For a laser scanner L, positioned at $[\mathbf{L}]_{world}$ in the $world$ frame, and $[\mathbf{L}]_{laser}$ in its own frame, the transformation $T_{w\to l}$ between the $world$ and $laser$ reference frames is given by:

$$[\mathbf{L}]_{laser} = T_{w\to l}[\mathbf{L}]_{world} \tag{1}$$

where

$$[\mathbf{L}]_{world} = [L_x, L_y, L_z, L_\phi, L_\psi, L_\theta] \tag{2a}$$
$$[\mathbf{L}]_{laser} = [0, 0, 0, 0, 0, 0] \tag{2b}$$

Next, we define a virtual laser L^* with a coordinate system $ortho$, corresponding to the orthogonal projection of L onto the xy- world plane. For L^* positioned at $[\mathbf{L}^*]_{world}$ in the $world$ frame, and $[\mathbf{L}^*]_{ortho}$ in its own frame, the transformation $T_{w\to o}$ between the $world$ and $ortho$ reference frames is given by:

$$[\mathbf{L}^*]_{ortho} = T_{w\to o}[\mathbf{L}]_{world} \tag{3}$$

where

$$[\mathbf{L}^*]_{world} = [L_x, L_y, 0, 0, 0, L_\theta] \tag{4a}$$
$$[\mathbf{L}^*]_{ortho} = [0, 0, 0, 0, 0, 0] \tag{4b}$$

The relationship between the $laser$, $ortho$, and $world$ frame is show in Fig. 3.

The laser reading from L is in the form $\{r_i, \theta_i\}$, where r_i are range readings, and θ_i are angles at which the readings were taken. The points detected by the laser thus have coordinates $[\mathbf{P_i}]_{laser} = [cos(\theta_i)r_i, sin(\theta_i)r_i, 0]^T$. Transforming that into the $world$ reference frame we get

$$[\mathbf{P_i}]_{world} = T^{-1}_{w\to l}[cos(\theta_i)r_i, sin(\theta_i)r_i, 0]^T \tag{5}$$

We then obtain the orthogonally projected points, first in the $world$, then in the $ortho$ frame

$$[\mathbf{P_i^*}]_{world} = [P_{ix}, P_{iy}, 0]^T \tag{6a}$$
$$[\mathbf{P_i^*}]_{ortho} = T_{w\to o}[\mathbf{P_i^*}]_{world} \tag{6b}$$

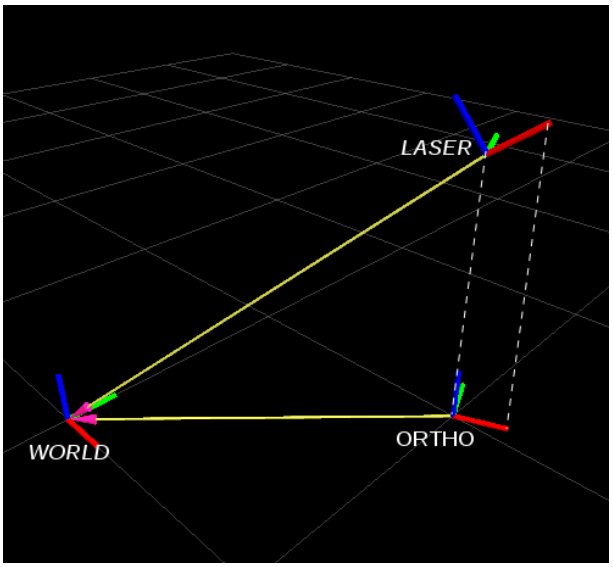

Fig. 3. The $world$, $laser$, and $ortho$ coordinate frames.

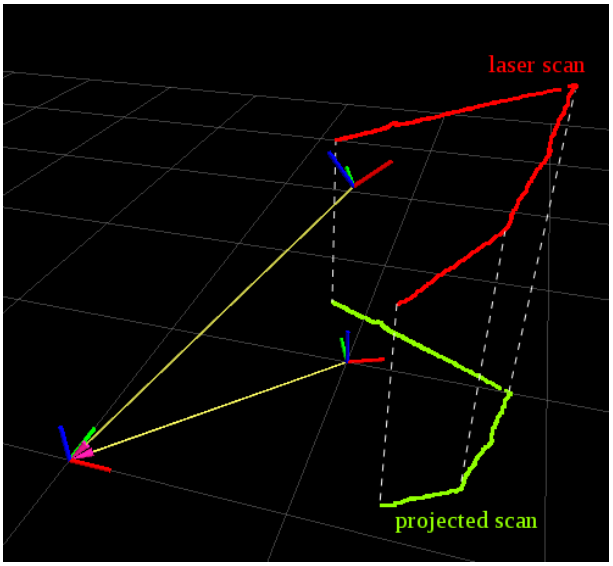

Fig. 4. The original and projected laser scans.

Finally, we have

$$r_i^* = |[\mathbf{P_i^*}]_{ortho}| \tag{7a}$$
$$\theta_i^* = acos\frac{[\mathbf{P_i^*}]_{ortho} \cdot [\mathbf{P_{i+1}^*}]_{ortho}}{|[\mathbf{P_i^*}]_{ortho}||[\mathbf{P_{i+1}^*}]_{ortho}|} \tag{7b}$$

The result of the orthogonal projection process is shown in Fig. 4.

IV. 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 $i \le sp_x < i+1$, $j \le sp_y < j+1$. 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 where $A_{c_{ij}}$ is the unit area of the grid cell.

$$\rho_V = \frac{m_V}{(z_V^{top} - z_V^{bot}) A_{c_{ij}}} \tag{8}$$

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 IV-B.

We impose the following additional restrictions on volume lists:

- 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 is the same as our horizontal one.

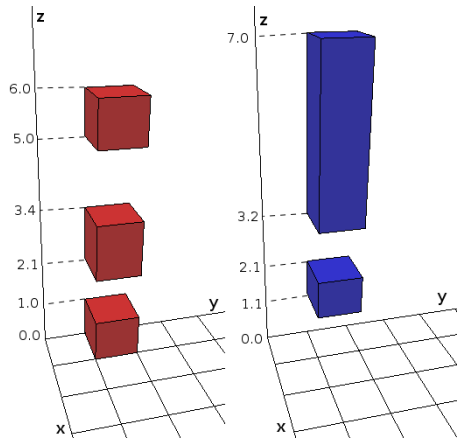

Fig. 5. Positive (red) and negative (blue) volume lists for cell $c_{0,1}$

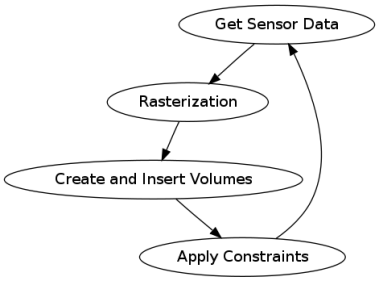

Fig. 6. Process diagram for updating the MVOG map

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. 5 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 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 IV-A. The process diagram of the update process is shown in Fig. 6.

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 $C = \{c_{i_0 j_0}, c_{i_1 j_1} \ldots c_{i_k j_k}\}$ 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 laser ray enters the space above each cell z_{ij}^{enter} and the height where it leaves it, z_{ij}^{exit} [10]. 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 an occupancy density ρ of 1. By knowing the height of the volume and using (8), 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, $c_{i_k j_k}$.

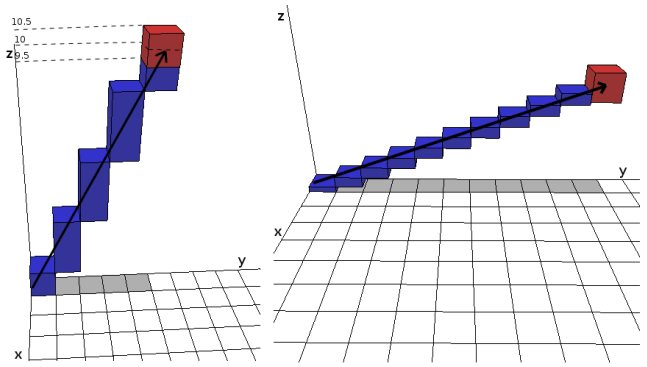

Fig. 7. The two different cases when rasterizing an obstacle reading from a laser. Grid cells in C are marked in gray. The left diagram shows a rasterization of a laser ray from $[0, 0, 0]^T$ reading an obstacle at $[0, 4.5, 10]^T$, requiring a positive and a negative volume to be inserted in the last 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.

$$z_V^{bot} = min(z_{ij}^{enter}, z_{ij}^{exit}) \tag{9a}$$

$$z_V^{top} = max(z_{ij}^{enter}, z_{ij}^{exit}) \tag{9b}$$

For the last cell, we have

$$z_V^{bot} = min(z_{ij}^{enter}, L'_z) \tag{10a}$$

$$z_V^{top} = max(z_{ij}^{enter}, L'_z) \tag{10b}$$

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

$$z_V^{bot} = L'_z - 0.5 \tag{11a}$$

$$z_V^{top} = L'_z + 0.5 \tag{11b}$$

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 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. 7.

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 IV-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:

$$z_{V^*}^{bot} = \frac{1}{2}(z_V^{bot} + z_V^{top}) - 0.5 \tag{12a}$$

$$z_{V^*}^{top} = \frac{1}{2}(z_V^{bot} + z_V^{top}) + 0.5 \tag{12b}$$

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 in new sensor information. Two volumes V^A and V^B overlap if $z_{V^B}^{bot} \in [z_{V^A}^{bot}, z_{V^A}^{top}]$ or $z_{V^B}^{top} \in [z_{V^A}^{bot}, z_{V^A}^{top}]$. 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

$$V^* = V^A \cup V^B \tag{13a}$$

$$m_{V^*} = m_{V^A} + m_{V^B} \tag{13b}$$

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 $z_{V^B}^{bot} - z_{V^A}^{top} \in (0, 1]$. 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

$$V^* = V^A \cup V^{GAP} \cup V^B \tag{14a}$$

$$m_{V^*} = m_{V^A} + m_{V^{GAP}} + m_{V^B} \tag{14b}$$

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 $\rho_{\mathbf{p}}^-$.

$$p(\mathbf{p}) = \begin{cases} \frac{\rho_{\mathbf{p}}^+}{\rho_{\mathbf{p}}^+ + \rho_{\mathbf{p}}^-} & \text{if } \rho_{\mathbf{p}}^+ + \rho_{\mathbf{p}}^- > 0 \\ \text{unknown} & \text{if } \rho_{\mathbf{p}}^+ + \rho_{\mathbf{p}}^- = 0 \end{cases} \tag{15}$$

If both ρ^+ and ρ^- in (15) 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.

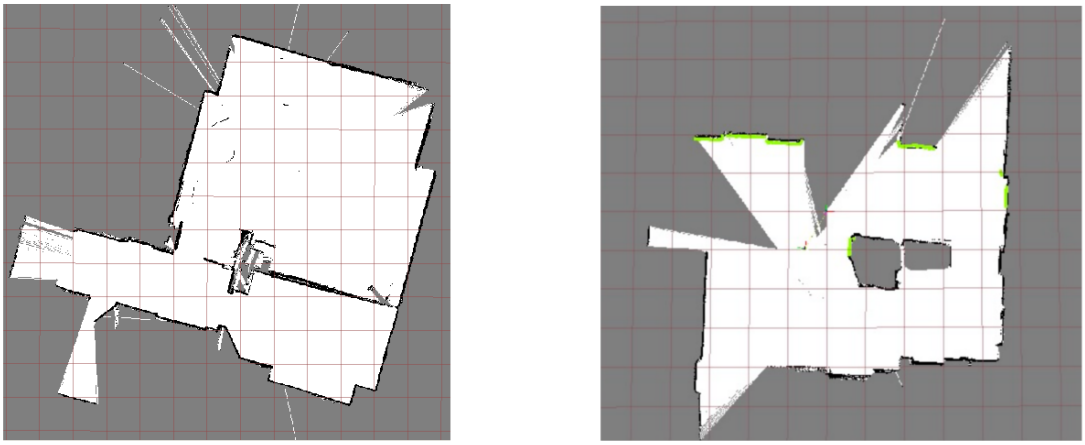

Fig. 8. Two maps built by projected laser data, demonstrating that obtaining the x and y pose components though scan projection and matching is sufficient to produce accurate trajectories with loop closure.

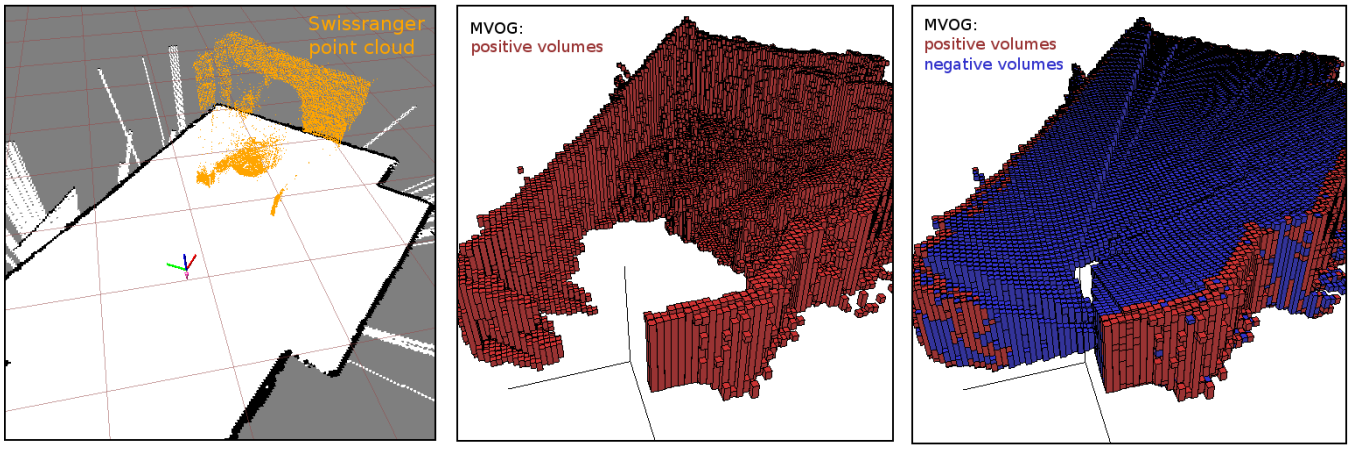

Fig. 9. Stages in the generation of 3D MVOG map an office room. The left image shows the point cloud data from the Swissranger 4000 camera. The center image shows the positive volumes, corresponding to obstacles in the room. The right image shows both the positive and negative volumes, which correspond to free space.

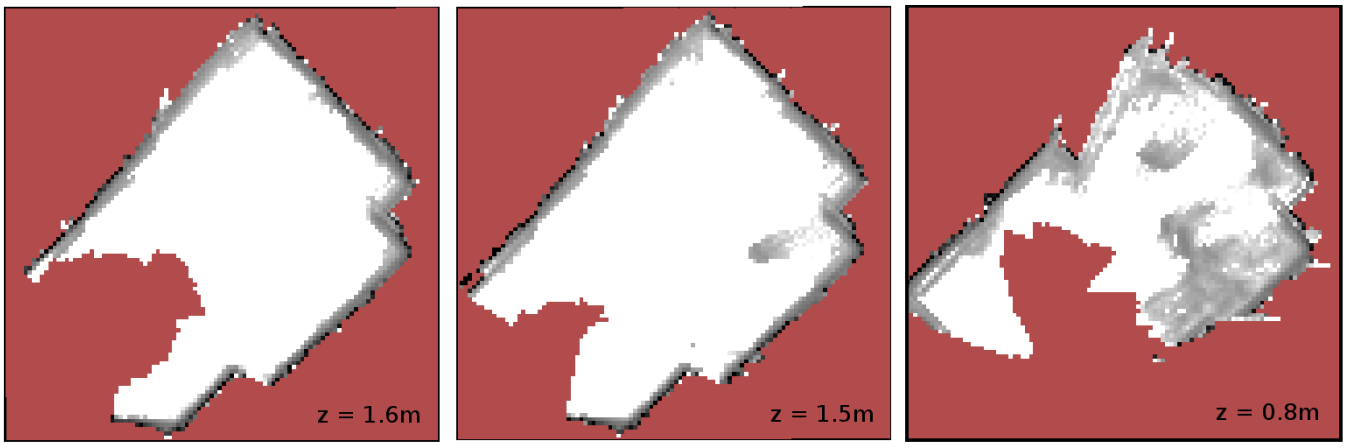

Fig. 10. Occupancy grids obtained from the MVOG by taking a cross-section at different heights. The left image shows a the occupancy at 1.6 meters above the floor. The entire room is navigable. The middle image shows the occupancy at 1.5 meters. The helicopter must navigate around the head of a diligent researcher, visible in the middle of the room. The right image shows the occupancy at 0.8 meters. Most of the room is occupied by the researcher, a desk in front of them, and a chair.

D. Time Complexity Analysis

Since occupancy volumes are stored in the form of a sorted list, 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 $O(m+n)$.

While the linear run time can be improved by storing the lists in a different structure, such as a skip lists, which provides $log(n)$ run time [11], 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.

V. Experimental Methods and Results

Due to payload constraints, we were unable to fly the micro-UAV with the Swissranger 4000 sensor on-board. Therefore experiments were performed by carrying the micro-UAV in-hand. All of the software we developed is open source and available from our website [13]. Logs of the experimental data necessary to duplicate the experiments is available by request.

Fig. 8 shows the results of the 2D SLAM mapping package. They demonstrate that obtaining the x and y pose components though scan projection and matching is sufficient to produce accurate trajectories with loop closure.

Fig. 9 shows stages in the generation of 3D MVOG map an office room. The left image shows the point cloud data from the Swissranger 4000 camera. The center image shows the positive volumes, corresponding to obstacles in the room. The right image shows both the positive and negative volumes, which correspond to free space.

Fig. 10 shows occupancy grids obtained from the MVOG by taking a cross-section at different heights. At 1.6 meters above the floor, the entire room is navigable. At 1.5 meters, the helicopter must navigate around the head of a diligent researcher, visible in the middle of the room. Further down, at 0.8 meters, most of the room is occupied by the researcher, a desk in front of them, and a chair.

Our results show that we are able to produce 3D maps that show a reasonable representation of the environment necessary for basic navigation and obstacle avoidance.

The use of an RGB-Depth camera has enabled us to improve our 3D mapping system so that it provides good experimental results that are robust to various sources of error over time.

VI. Conclusions And Future Works

Our final goal is to achieve 3D SLAM, and autonomous navigation in a variety of unstructured environments. Towards that end, we present a system for pose estimation and 3D mapping that solves the simplified case of rectilinear indoor spaces. We have demonstrated how to fuse readings from an altimeter, an IMU, and a laterally-mounted laser scanner. We have developed an algorithm to project the laser readings orthogonally, in order to make them invariant of the roll and pitch motion of the helicopter. We also presented a novel 3D map structure called a multi-volume occupancy grid, and demonstrate how constructing MVOGs from a Swissranger depth camera can result in maps useful in further research in autonomous 3D navigation.

While our micro-UAV is currently only capable of carrying a 2D laser scanner, we believe that future RGB-D cameras will be significantly lighter and should within our payload limitations. Our future research will focus on computationally-feasible 3D scan-matching techniques for micro-UAVs.

Acknowledgment

This work was supported in part by the U.S. Army Research Office under grant No. W911NF-08-1-0531, W911NF-09-1-0565, and U.S. National Science Foundation under grants No.CNS-0619577 and No. IIS-0644127.

References

[1] Grisetti, G.; Stachniss, C.; Burgard, W.; , "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters," *Robotics, IEEE Transactions on* , vol.23, no.1 pp.34 - 46, Feb 2007

[2] Nuchter, A.; , "3D Robotic Mapping: The Simultaneous Localization and Mapping Problem with Six Degrees of Freedom," *Springer Tracts in Advanced Robotics* , vol.52, 2009.

[3] Bachrach, A.; He, R.;Roy, N.; , "Autonomous Flight in Unknown Indoor Environments," *International Journal of Micro Air Vehicles* , 1(4): 217-228, December 2009.

[4] Grzonka, S.; Grisetti, G.; Burgard, W.; , "Towards a navigation system for autonomous indoor flying," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on* , vol., no., pp.2878-2883, 12-17 May 2009

[5] Thrun, S.; , "Learning occupancy grids with forward models," *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* , vol.3, no., pp.1676-1681 vol.3, 2001

[6] Borenstein, J.; Koren, Y.; , "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on* , vol.7, no.3, pp.278-288, Jun 1991

[7] Duffy, N.; Allan, D.; Herd, J.T.; , "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

[8] Triebel, R.; Pfaff, P.; Burgard, W.; , "Multi-Level Surface Maps for Outdoor Terrain Mapping and Loop Closing," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* , vol., no., pp.2276-2282, 9-15 Oct. 2006

[9] Rivadeneyra, C.; Miller, I.; Schoenberg, J.R.; Campbell, M.; , "Probabilistic estimation of Multi-Level terrain maps," *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on* , vol., no., pp.1643-1648, 12-17 May 2009

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

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

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

[13] "CityFlyer". CCNY Robotics Lab. http://robotics.ccny.cuny.edu/wiki/Robots/CityFlyer. 2010