

Autonomous Quadrotor 3D Mapping and Exploration Using Exact Occupancy Probabilities

Evan Kaufman*, Kuya Takami*, Zhuming Ai[†] and Taeyoung Lee*

*Department of Mechanical and Aerospace Engineering

The George Washington University, Washington, DC 20052

Email: {evankaufman, kuya, tylee}@gwu.edu

[†]Information Management & Decision Architectures

U.S. Naval Research Laboratory, Washington, DC 20375

Abstract—This paper deals with the aerial exploration for an unknown three-dimensional environment, where Bayesian probabilistic mapping is integrated with a stochastic motion planning scheme to minimize the map uncertainties in an optimal fashion. We utilize the popular occupancy grid mapping representation, with the goal of determining occupancy probabilities of evenly-spaced grid cells in 3D with sensor fusion from multiple depth sensors with realistic sensor capabilities. The 3D exploration problem is decomposed into 3D mapping and 2D motion planning for efficient real-time implementation. This is achieved by projecting important aspects of the 3D map onto 2D maps, where a predicted level of map uncertainty, known as Shannon's entropy, provides an exploration policy that governs robotic motion. Both mapping and exploration algorithms are demonstrated with both numerical simulations and quadrotor flight experiments.

I. INTRODUCTION

Robotic mapping is the process of generating maps representing the environment surrounding a robot. Mapping serves an important role in tasks such as search-and-rescue, surveillance, and robotic cleaning. Several grid-based map representations have been studied, e.g., [1], [2], [3], because they provide abundant occupancy knowledge about the environment and are computationally tractable.

Most grid-based mapping representations are designed for 2D, while others are extended to capture a full 3D environment. In particular, occupancy probabilities have been assigned to a uniform 3D grid in [4], [5]. Another highly-popular variation of occupancy grid mapping is the Octomap representation [6], [7], which has added features such as extending the mapping limits and changing cell sizes.

However, these approaches are based on an approximate solution of the inverse sensor model, to determine the occupancy probabilities based on the depth sensor measurements. As such, the prior belief of the environment and the stochastic properties of the sensor are not properly accounted. Another popular approach for the inverse sensor model involves a simulated learning technique [8], [9], [10], but this is not suitable for the higher dimensional configurations composed of maps, poses, and measurement due to the large number of required samples. None of these approaches use the sensor properties to directly obtain a Bayesian solution to occupancy probability.

The approximate nature of the occupancy probability motivates an exact Bayesian solution that can be applied in 3D. This paper proposes a computationally-efficient algorithm to obtain the exact solution to the inverse sensor problem to generate probabilities in a 3D environment, following the framework of [11], [12]. The proposed approach integrates all of the stochastic information available for the environment, such as the prior map estimate and multiple sensor measurements, according to the Bayesian framework, to obtain the exact 3D occupancy probabilities. As such, it provides a unified framework for sensor fusion between multiple sensors with various properties, such as sensor accuracy, and fields of view.

The second topic of this paper deals with autonomous exploration, which provides the policies that govern robotic motion. A common approach is known as frontier-based exploration, proposed for 2D environments in [13], [14] and extended to 3D using the Octomap representation in [15], [16], [17], and similarly using a visibility metric in [18]. This approach moves robots toward the boarder between certain and uncertain spaces, though these systematic actions are not based on any consideration of the future probabilistic map uncertainty.

Alternatively, there have been exploration techniques that are based on a quantitative measure of uncertainty known as Shannon's entropy [19], and it is used to acquire expected map information gain in [20], [12]. This paper focusses on extending these concepts to 3D in a numerically efficient manner. The prescribed 3D exact inverse sensor model is critical in computing the expected information gain precisely. There are several challenges with obtaining expected information gain of 3D measurement scans, particularly due to the extended computational load. Considering the typically low variation in floor plans along the vertical direction, this paper proposes a numerically efficient scheme that projects stochastic properties of the 3D probabilistic occupancy grid to 2D spaces for optimal information gain and collision avoidance. This exploits the important characteristics of 3D maps. This process of combining 3D mapping and 2D exploration is completed in a computationally-efficient manner.

Finally, we present both numerical simulations and flight experiments for the proposed exploration schemes. The multiple tasks of sensor data acquisition, mapping, exploration,

flight control, and communications are implemented with Python/C++, to be executed simultaneously in a ROS environment. They are verified with a quadrotor unmanned aerial vehicle operated by the NVidia Jetson computing module with an ASUS Xtion infrared depth sensor and Hokuyo laser scanner.

The paper is organized as follows. The problem is formulated in Section II. Special considerations for mapping and exploration in 3D are established in Section III. A numerical example is demonstrated in Section IV. Experimental results are shown in Section V, followed by conclusions and future work.

II. PROBLEM FORMULATION

The key differences between the approaches presented in this paper and others relate to how map cell occupancy probabilities and measurement ray entropies are calculated directly from a priori cell probabilities and stochastic properties of the sensors involved. These are defined next.

A. Probabilistic Occupancy Grid Mapping

Let a map m be an occupancy grid map decomposed into n evenly-spaced cubic grid cells with fixed edge length α . The i -th grid cell is assigned to a static binary random variable \mathbf{m}_i for $i \in \{1, 2, \dots, n\}$, that is defined as $\mathbf{m}_i = 1$ when occupied, and $\mathbf{m}_i = 0$ when free. The location and size of each grid cell is assumed known, where a smaller cell size better represents a space, but increases computation and memory. Therefore, a map m is defined by $\{\mathbf{m}_1, \dots, \mathbf{m}_n\}$ (2^n possible maps).

Another random variable is defined as $\bar{\mathbf{m}}_i = 1 - \mathbf{m}_i$ for convenience, which is the event that the i -th cell is free, i.e., $P(\bar{\mathbf{m}}_i) = 1 - P(\mathbf{m}_i)$. The random variables composing m are assumed mutually independent,

$$P(m) = P(\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n) = \prod_{i=1}^n P(\mathbf{m}_i). \quad (1)$$

Consider a range sensor that provides scans of the surrounding environment in order to identify the closest occupied space. Let the measurement scan at the t -th time step be Z_t , where $z \in Z_t$ is a single measurement ray. Let the t -th pose be denoted by $X_t = \{x_t, R_t\}$ with location $x_t \in \mathbb{R}^3$ and attitude $R_t \in \text{SO}(3) = \{R \in \mathbb{R}^{3 \times 3} | R^T R = I, \det R = 1\}$, where X_t is assumed to be known. Given the sensor probabilistic properties $p(z|X_t, m)$, commonly referred to as the *forward sensor model*, the goal is finding the *inverse sensor model*, $P(m|z, X_{1:t}, Z_{1:t-1})$. To obtain this probability for all cells, [11], [12] introduced a reduced map $r \subset m$ composed of n_r cells along z , where cells are indexed by increasing distance from x_t , and \mathbf{r}_{k+} refers to the event that the k -th cell of map r is the closest occupied space. Then, the inverse sensor model for the k -th cell is

$$P(\mathbf{r}_k|z, X_{1:t}, Z_{1:t-1}) = \eta \tilde{P}(\mathbf{r}_k|z, X_{1:t}, Z_{1:t-1}), \quad (2)$$

where the unnormalized probability of the inverse sensor model is defined as

$$\begin{aligned} \tilde{P}(\mathbf{r}_k|z, X_{1:t}, Z_{1:t-1}) \\ = \mathbf{P}_k^- \left[\sum_{i=1}^{k-1} \left\{ \prod_{j=0}^{i-1} \bar{\mathbf{P}}_j^- \right\} p(z|\mathbf{r}_{i+}, X_t) \mathbf{P}_k^- \right] \\ + \left\{ \prod_{j=0}^{k-1} \bar{\mathbf{P}}_j^- \right\} p(z|\mathbf{r}_{k+}, X_t) \mathbf{P}_k^-, \end{aligned} \quad (3)$$

where a priori probability is $\mathbf{P}_k^- = P(\mathbf{r}_k|X_{1:t-1}, Z_{1:t-1})$, its complement is $\bar{\mathbf{P}}_k^- = 1 - \mathbf{P}_k^-$, $P(\bar{\mathbf{r}}_0|X_{1:t-1}, Z_{1:t-1}) = P(\mathbf{r}_{n_r+1}|X_{1:t-1}, Z_{1:t-1}) = 1$ for convenience, and $p(z|\mathbf{r}_{(n_r+1)+}, X_t)$ represents the forward sensor model of a maximum sensor reading. The normalizer η is given by

$$\eta = \left[\sum_{i=1}^{n_r+1} \left\{ \prod_{j=0}^{i-1} \bar{\mathbf{P}}_j^- \right\} p(z|\mathbf{r}_{i+}, X_t) \mathbf{P}_k^- \right]^{-1}, \quad (4)$$

and it is independent of the cell index k . The proofs of (2)–(4) are given in [11]. Since the terms of (3) are easily obtained and several are used repeatedly to obtain (4), the computational cost of (2) is trivial and can be applied in real-time. This process is repeated for all measurements composing scan Z_t .

B. Predictive Entropy-Based Autonomous Exploration

Since occupancy grid mapping provides a probabilistic representation of surrounding space, this mapping scheme holds probabilistic information about the uncertainty of the map. Shannon's entropy is commonly used as a measure of uncertainty, such as in [19]. Given the probabilities of each grid cell of map m , Shannon's entropy of an individual cell and the entire map are defined as

$$H(P(\mathbf{m}_i)) = -P(\mathbf{m}_i) \log P(\mathbf{m}_i) - P(\bar{\mathbf{m}}_i) \log P(\bar{\mathbf{m}}_i), \quad (5)$$

$$H(P(m)) = \sum_{i=1}^{n_m} H(P(\mathbf{m}_i)), \quad (6)$$

respectively. The entropy of the i -th grid cell is maximized when $P(\mathbf{m}_i) = 0.5$ (greatest uncertainty) and minimized as $P(\mathbf{m}_i)$ approaches 0 or 1 (smallest uncertainty).

Let an unvisited candidate future pose and its measurement scan be X_c and Z_c , respectively. Since the robot has not visited X_c , any change to the probabilistic map must be predicted. Much like the probabilistic mapping, this is achieved ray-by-ray [20], [12]. Since all grid cell probabilities are conditioned on the history of poses $X_{1:t}$ and measurement scans $Z_{1:t}$, these are removed from the remaining equations for simplicity. For candidate ray z_c , the expected entropy is

$$E[H(P(m|x_c, z_c))] = \sum_{k=1}^{n_r+1} \left\{ H(P(m|x_c, z_{c,k})) P(z_{c,k}|x_c) \right\}, \quad (7)$$

where subscript k refers to the event that the candidate *measurement* captures the k -th cell along the ray, i.e., $z_{c,k}$ is the distance from x_c to the k -th grid cell. The first term of

the summation of (7), namely $H(P(m|x_c, z_{c,k}))$, is obtained with the inverse sensor model (2)–(4) and entropy definitions (5), (6). The second term is found using (4) with

$$P(z_{c,k}|x_c) = \frac{p(z_{c,k}|x_c)}{\sum_{i=1}^{n_r+1} p(z_{c,i}|x_c)} = \frac{\eta_{c,k}^{-1}}{\sum_{i=1}^{n_r+1} \eta_{c,i}^{-1}}, \quad (8)$$

where $\eta_{c,k}$ refers to the normalizer calculated with $z_{c,k}$. The goal of entropy-based autonomous exploration is to select the optimal pose X_c^* that maximizes the expected information gain,

$$X_c^* = \operatorname{argmax}_{X_c} \mathcal{I}(X_c), \quad (9)$$

$$\mathcal{I}(X_c) = H(P(m|X_{1:t}, Z_{1:t})) - E[H(P(m|X_c, Z_c))], \quad (10)$$

where (7) is used repeatedly for several sample rays that compose scan Z_c as shown in [20].

Once an optimal pose X_c^* is selected, a set of collision-free waypoints along the occupancy grid is easily obtained with Dijkstra's algorithm [21]. Then, a constrained least squares optimization provides a trajectory as a tracking command for the robot controller.

III. MAPPING AND EXPLORING IN 3D

A. Mapping in 3D Space

The inverse sensor model from [11], [12] is based on an arbitrary vector spanning 2D space. The distances from the robot to grid cells are obtained geometrically through ray casting. This method is easily extended to 3D space, and the inverse sensor model is applied identically. Since the map is updated ray-by-ray (each measurement ray is considered once by itself), this approach is extended for measurement fusion of any number of sensors; the map is updated by each ray of a scan sequentially, then the map is updated by the rays of the next scan in queue, which might have different characteristics. There are several reasons why a robot might be equipped with multiple sensors, e.g., different fields-of-view or depth sensor properties. For example, a robot may be equipped with a 2D LIDAR with highly accurate measurements, but a limited single-plane field-of-view. Then, the robot can compensate for the LIDAR limitations with an IR depth sensor that provides many more measurement rays with a 3D scan, where each ray has lower accuracy than the LIDAR rays. Since each sensor model is explicitly considered in the cell probability calculations, measurements from both sensors may serve to update the same probabilistic map. However, the impact from a LIDAR measurement with high accuracy on the probabilistic map will be greater than that of the IR depth sensor with lower accuracy. Since sensor properties are considered explicitly, the proposed 3D mapping approach can fuse the impact of measurements on the probabilistic map from different sources with different properties.

However, implementation in 3D requires several careful considerations. First, the number of cell occupancy probabilities tracked with computer memory increases by a factor of the number of cells in the added dimension. In its current

form, the proposed method assumes fixed map limits, so these should be carefully selected with grid cell resolution based on available memory. Second, measurement rays spanning 3D space typically involve more intersections with grid cells. The computational order of ray casting and the inverse sensor model are proportional to the number of grid cells along the ray n_r , so careful consideration must be placed on sensor limits and rays per scan that can be considered. Typically, computers small enough to be carried onboard flying robots lack the processing capabilities to update massive maps quickly, particularly because the ray-by-ray updating method requires many computations in series with large measurement scans. Therefore, an onboard computer can simply stream the sensor data, and a more powerful computer can run the mapping.

Additionally, aerial vehicles are frequently subject to fast movements, and the time stamps for pose estimates and depth sensor readings are not guaranteed to align. For example, the most recent pose estimate and measurement scan may have significantly different time stamps while the robot is rapidly rotating. Using these together violates the assumption that the robot pose is well-known for every measurement scan, i.e., X_t and Z_t are given. Therefore, pose estimates and sensor readings must be paired according to their time stamps. A useful tool that easily satisfies this goal is the Message Filters package with the Robot Operating System (ROS), where stamped poses and measurement scans are synchronized using recently received data.

B. Autonomous Exploration in 3D Space

The goal of the proposed autonomous exploration strategy is moving to poses that maximize the information the robot expects to learn about the map, measured with entropy. To achieve this in real-time, careful consideration is placed on computational costs. In particular, the expected entropy computational order of a ray that intersects n_r cells is $(n_r + 1)^2$. Since most quadrotor flight is close to hover and floor plans exhibit little variation in the vertical direction, the proposed exploration uses 2D occupancy grid maps located at the desired exploration height. These assumptions correspond to rooms and hallways with non-complex geometries. This approach decreases n_r and simplifies the exploration problem with 3D space. These 2D maps project important properties of the 3D map.

1) *Collision and Entropy Maps*: Occupancy grid map cell volume is frequently smaller than the size of the robot taking measurements, particularly with highly-detailed maps. This motivates a method to simply represent locations that risk collision, referred to as a *collision map* C . Let $k_C \geq 1$ be an integer representing the minimum number of 3D grid cells that form a cube that can completely encompass the robot, i.e., the collision map cell edge length is

$$\alpha_C = k_C \alpha. \quad (11)$$

Consider the set $m_{C,k} \subset m$ consisting of the 3D grid cells falling inside the limits of the k -th cell of collision map C ,

namely \mathbf{C}_k . Then the collision probability is

$$P(\mathbf{C}_k) = \max (P(\mathbf{m}_i) \forall i \in m_{C,k}). \quad (12)$$

Then, the robot is only allowed to consider moving into k -th cell of the collision map if

$$P(\mathbf{C}_k) \leq C_{\text{thresh}}, \quad (13)$$

where $C_{\text{thresh}} > 0$ is a maximum acceptable collision threshold, typically chosen slightly greater than 0. In short, this conservative approach limits robotic motion to regions with a low risk of collision while neglecting objects far above or below the exploration plane.

In a similar method, *entropy map* E determines cells based on the entropy metric of (5) over a different subset of cells. Let the E be a 2D map composed of cells with edge length α located at the exploration height. Let the set $m_{E,k} \subset m$ be the cells directly above and below the k -th cell of E , neglecting any 3D cells located lower than the floor height. Selecting this set is advantageous because measurements are assumed not to penetrate occupied cells (edge length α) and measurement rays are typically not closely-aligned with the third axis of the inertial frame (vertically long set). Furthermore, any cells located below the floor, which cannot be reached, should not impact the uncertainty of the space. Then, the occupancy probability of the k -th entropy map cell is selected as

$$P(\mathbf{E}_k) = \underset{P(\mathbf{m}_i)}{\operatorname{argmax}} (H(P(\mathbf{m}_i)) \forall i \in m_{E,k}). \quad (14)$$

After several measurements of the cells belonging to $m_{E,k}$, it is common for $H(P(\mathbf{E}_k)) \approx 0$, which implies that these cells are well-known. However, this does not imply that the captured space is *entirely* free or occupied (e.g. an object on the ground that does not breach the upper map boundary), so applying (14) alone could arbitrarily set $P(\mathbf{E}_k)$ close to 0 or 1. This can have damaging effects on the expected information of cells beyond the k -th cell. Let $P_{\min} > 0$ be a lower bound such that $H(P_{\min}) = H(1 - P_{\min}) \approx 0$. If $H(P(\mathbf{E}_k)) \leq H(P_{\min})$, then this probability is corrected to

$$P(\mathbf{E}_k) = \begin{cases} P_{\min}, & \text{if } \operatorname{mean}(P(\mathbf{m}_i) \forall i \in m_{E,k}) \leq 0.5 \\ 1 - P_{\min}, & \text{otherwise} \end{cases}, \quad (15)$$

where the first case corresponds to mostly open space and the second case corresponds to mostly occupied space (e.g. walls and objects). In short, the candidates are only considered if they can be reached over collision map C via Dijkstra's search, and the information gains of the reachable candidates are predicted using entropy map E .

2) *Collision and Entropy Combination Map*: The second proposed technique combines the advantages of both the collision and entropy maps into a 2D projected occupancy grid map, which can be used for both tasks. Let the *collision and entropy combination map* B be a 2D projected map located at the exploration height with grid cell edges the same as the collision map, namely α_C from (11), using the same subset of grid cells from the collision map, namely m_C .

Considering that the cells begin with uniform probability P_{init} where $0 < P_{\text{init}} < 1$, define a threshold P_{thresh} such that $P_{\text{init}} < P_{\text{thresh}} < 1$. Then, the probability assigned to the k -th grid cell of B is

$$P(\mathbf{B}_k) = \begin{cases} P(\mathbf{C}_k), & \text{if } P(\mathbf{C}_k) \geq P_{\text{thresh}} \\ \min (P(\mathbf{m}_i) \forall i \in m_{C,k}), & \text{otherwise} \end{cases}. \quad (16)$$

This approach has several advantages. First, measurements must enter the vicinity of the k -th cell for $P(\mathbf{B}_k)$ to change from P_{init} . Without prior measurements, capturing $P(\mathbf{B}_k)$ produces a large reward for the expected information gain in (10). Second, the value of P_{thresh} can be modified based on how conservative the collision risk assessment for a robot should be. Additionally, when $P(\mathbf{B}_k) < P_{\text{thresh}}$, this event tends to favor free cells such that expected information gains of cells beyond the k -th cell have greater impact on exploration. Furthermore, a single map is simpler, and since cell sizes satisfy $\alpha_C \geq \alpha$, it follows that n_r can only be decreased with α_C , thereby increasing computational speed.

IV. NUMERICAL RESULTS

A. Software Structure

The Robot Operating System (ROS) provides an excellent framework for several programs, referred to as *nodes*, to communicate easily. These are mostly coded in C++ for speed, except for a few Python scripts for a GUI and trivial message conversions, and launch/yaml scripts for running nodes and setting parameters.

Gazebo is a 3D dynamic simulator [22], with plugins to simulate an Asus Xtion color depth sensor and a Hokuyo LIDAR. Gazebo also provides position and attitude transforms from the world to the quadrotor body, and the ability to set robot model states directly.

Using fixed transforms between the quadrotor body and its onboard sensors, an original mapping node uses a variation of ROS message filters, known as transform message filters, to synchronize the sensor poses with sensor readings. These serve as inputs for ray casting in 3D, which provides the cell probabilities and depths along the ray through geometry. Then, the inverse sensor model serves to update the map ray-by-ray. The node is designed to work for any number of sensors, specified by sensor parameters in a yaml configuration file.

There are two processes for exploration. The first process is projecting the 3D map onto 2D maps; the same code under different conditional parameters produces two nodes, for collision and entropy maps. The second process subscribes to these two 2D maps; the collision map serves as input for candidate pose consideration and Dijkstra's search, and the entropy map serves to predict information gain. This node provides path messages, composed of desired poses 0.01sec apart, displayed with a ROS visualization package, Rviz.

Finally, a path-interpreter node subscribes to the path messages and interpolates the pose at the current ROS time step. This pose is converted into a message that sets the Gazebo quadrotor state.

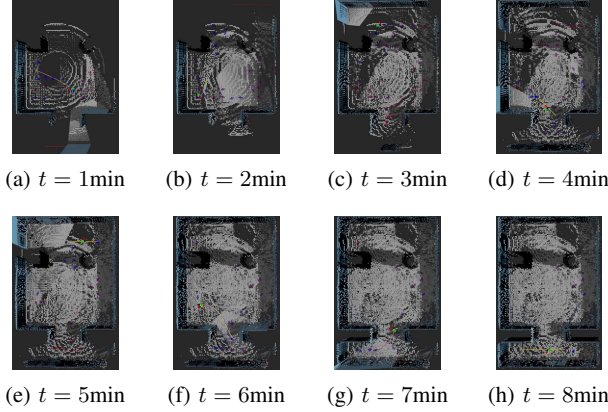


Fig. 1: Simulated 3D Occupancy Grid Map

B. Simulation Results

The 3D mapping and exploration algorithms are simulated with several parameters. In particular, the Xtion and Hokuyo are modeled using normalized Gaussian and uniform distributions for the sensor model, with a Gaussian probability of 0.9 with $\sigma_{\text{Xtion}} = 0.25\text{m}$ and $\sigma_{\text{Hokuyo}} = 0.1\text{m}$, respectively, which are greater than their specified values to account for pose uncertainty and distortion with the Xtion. Since the Xtion tends to distort depths significantly beyond 4m, this maximum reading considered for both the Xtion and Hokuyo, and the uniform distribution (probability 0.1) covers the sensor ranges from 0.5m to 4m. A cell size of $\alpha = 0.075\text{m}$ is selected and the map limits are -4m to 4m in the x-direction, -6m to 6m in the y-direction, and -0.15m to 1.5m in the z-direction. The initial probability of cells is selected at $P_{\text{init}} = 0.1$. For the collision map, the cell edge factor is $k_C = 3$. Additionally, Dijkstra's algorithm produces a cost map composed of distance costs for each collision map cell, denoted d_{cell} , from the current robot position to the candidate pose locations over a collision-free path about the occupancy grid. A normalized variation on the *bump function* [23] with maximum distance $d_{\text{max}} > 0$,

$$\text{bump}(d_{\text{cell}}) = \begin{cases} e^{\frac{1}{1 - (d_{\text{cell}}/d_{\text{max}})^2}}, & \text{if } -d_{\text{max}} < d_{\text{cell}} < d_{\text{max}} \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

is multiplied to (10) when optimizing (9). For this experiment, d_{max} is chosen as the maximum value of d_{cell} on the cost map. This serves to prioritize understanding the local surrounding space before the robot moves across the map to distant regions, while still considering all of the pose candidates.

In the simulation, the robot explores the simulated environment for 8min while taking measurements. The 3D occupancy grid map and true environment are shown in Fig. 1. The Gazebo simulated environment and two 2D projected maps for collision and entropy are shown in Fig. 2. The total map entropy as a function of time is shown in Fig. 3.

The two-map approach exhibits advantages and disadvantages. The exploration algorithm successfully avoided collisions and gave way to a fairly complete 3D occupancy grid.

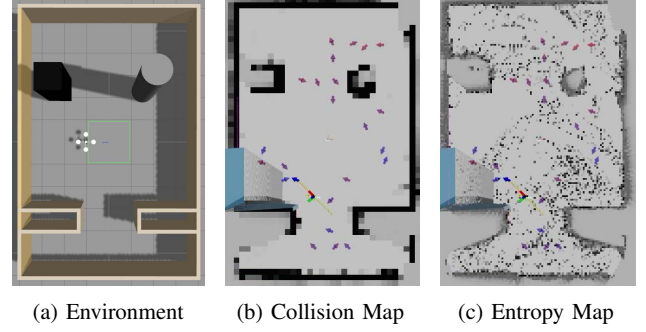


Fig. 2: Simulated Environment and 2D Projected Maps at 4min

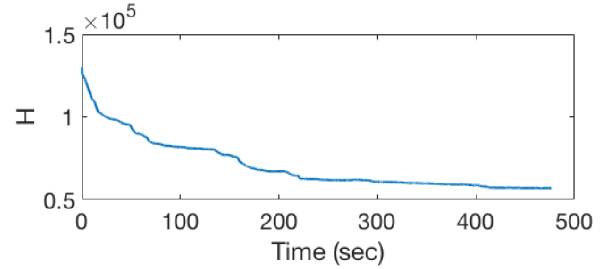


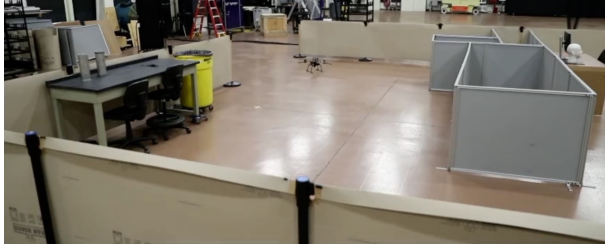
Fig. 3: Simulated Environment Map Entropy

The floor, which is not always visible, is properly estimated in most places. Conversely, this attention to detail has negative consequences that encourage the robot to repeatedly look at the same regions from different vantage points. Since an entropy map cell is composed from many 3D grid cells single-stacked vertically, if just one of these 3D map cells is not captured or hardly captured, the entropy map cell will be recorded as quite uncertain, shown covering numerous locations in Fig. 2(c). The robot may attempt to measure this space again, even when other regions might more strongly warrant observation. These periods of sluggish entropy decrease are shown with flatter sections of Fig. 3. In short, this algorithm successfully explores the space collision-free while producing a complete map, but over-attention to just a few uncertain cells has negative consequences on the exploration speed.

V. EXPERIMENTAL RESULTS

A. Exploration Environment

The Laboratory for Autonomous Systems Research (LASR) at the U.S. Naval Research Laboratory (NRL) provides a large experimental space for testing. An experimental setup resembles a building floor plan, which involves tan cardboard suspended from 1m tall stanchions for a perimeter wall, an internal wall from 80/20 supports with grey sheeting inside, and a floor that reflects some of the florescent lights above. Additionally, there are two wooden table desks with chairs and miscellaneous objects on top, and three plastic trash cans of varying sizes and colors. The map limits are designed to slightly exceed the experimental space: 0.5m to 10.3m in the x-direction (positive east), -8.3m to 3.5m in the y-



(a) North Region



(b) South Region

Fig. 4: 3D Experimental Environment

direction (positive north), and -0.15m to 1.5m in the z -direction (positive vertically up). Pictures of the environment are shown in Fig. 4.

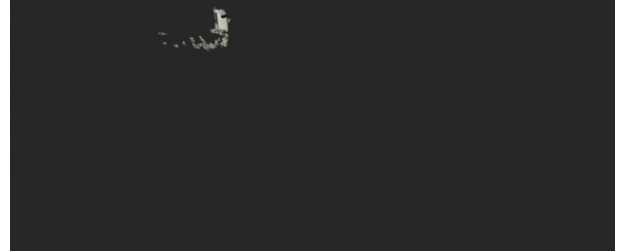
B. Hardware Structure

Instead of using a simulated environment such as Gazebo, where the robot state can be set and measured directly, a Vicon motion capture system produces pose data used in mapping and exploration transforms. An Asus Xtion and Hokuyo LI-DAR are streamed from an onboard NVidia Jetson TX2 via Wifi to a host computer running the mapping and exploration nodes with an Intel Core i7-6800K CPU ($12 \times 3.40\text{GHz}$). The exploration trajectories are sent back to the Jetson TX2 over Wifi, where a nonlinear geometric flight controller [24] is run onboard the Jetson to follow these trajectories. This configuration avoids large processing tasks onboard the robot.

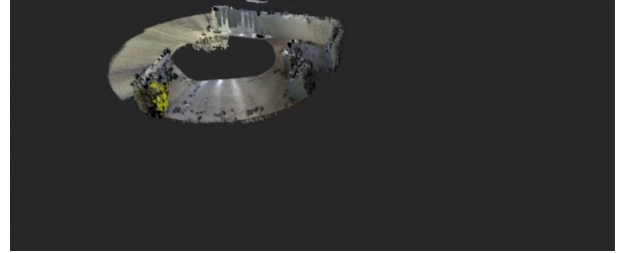
C. Experimental Results

The experimental setup follows similar conditions as the simulated results with a few key differences. Most importantly, the second proposed exploration approach, namely a collision and entropy combination map, replaces the two-map approach used in simulation. The resulting 3D maps at various time steps are shown in Fig. 5. The 2D projected maps (x-direction upward) at corresponding time steps are illustrated in Figure 6, where trajectories are shown with curves to candidate poses are shown with arrows. The total map entropy as a function of time is shown in Figure 7.

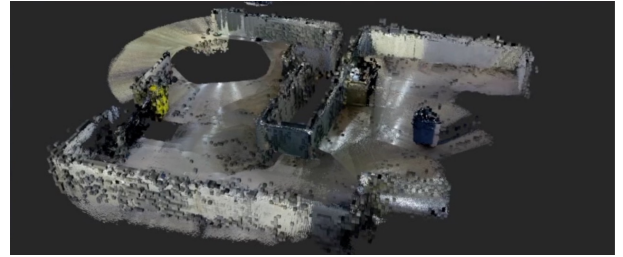
The projected map used for both collision and exploration has several benefits. The robot measures walls and objects with the 3D map, and nicely represents these as occupied spaces with the 2D projected map. Additionally, the robot explores the space much faster than with two separate maps, primarily



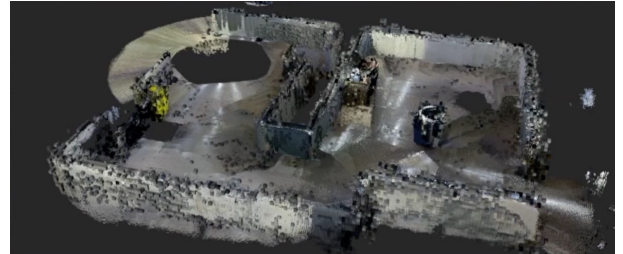
(a) 0min



(b) 1min



(c) 2min



(d) 2min 47sec (end)

Fig. 5: 3D Occupancy Grid Map and Point Clouds

because the robot avoided repeatedly exploring spaces as with the entropy-only map in Section IV. This decision leads to a tradeoff; with the experiment shown here, the robot exploration did not consider low cells, particularly those near the ground. As a result, some floor regions are not captured by the 3D occupancy grid even though cells above the floor are well-known.

The 3D mapping works very well in most scenarios, but exploration requires some tuning. Most importantly, the value of d_{\max} in (17) has a major impact. A value of 5m is selected to prioritize local exploration trajectories first. This value only increases if the maximum value of d_{cell} exceeds d_{\max} , in which

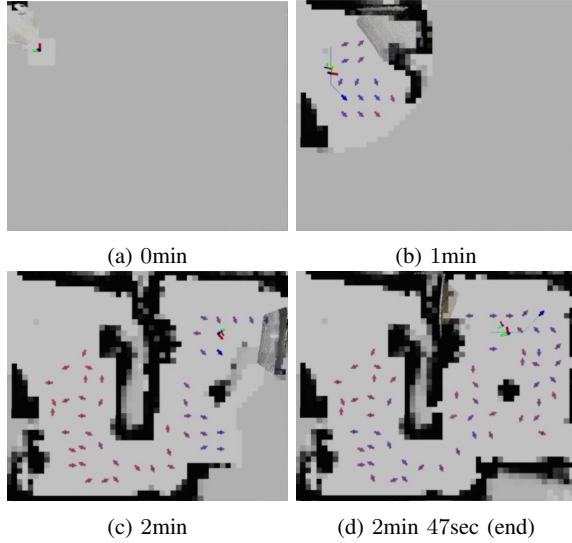


Fig. 6: 2D Collision and Exploration Combination Map

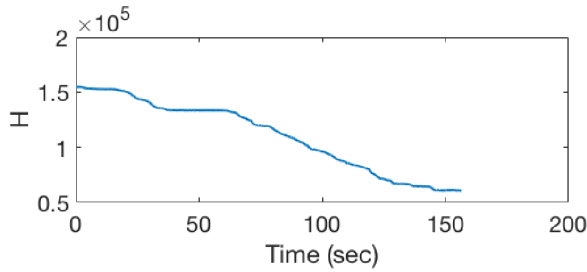


Fig. 7: Experimental Environment Map Entropy

d_{\max} is reassigned to d_{cell} .

VI. CONCLUSIONS AND FUTURE WORK

This paper covers the important aspects of exploring and mapping a 3D environment using exact inverse sensor models and expected entropies. The 3D mapping produces occupancy probabilities with depth sensor fusion from an arbitrary number of sensors. Due to the computational difficulties of exploration in 3D, a 2D version of exploration is combined with 3D mapping. These are demonstrated with numerical simulations and experiments with an aerial vehicle.

There are two important future directions. One direction is a more complete 3D exploration considering the map expected information gain via measurement rays cast in 3D. This approach could better capture cells on the ground or vertically-nonuniform objects, but careful consideration must be placed on computational restrictions. Secondly, the 3D occupancy grid may be used by multiple robots for faster exploration.

REFERENCES

[1] D. Wolf and G. Sukhatme, "Mobile robot simultaneous localization and mapping in dynamic environments," *Autonomous Robots*, vol. 1, no. 19, pp. 53–65, 2005.

[2] W. B. Daniel Meyer-Delius, Maximilian Beinhofer, "Occupancy grid models for robot mapping in changing environments," in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[3] D. W. Georg Tanzmeister, Julian Thomas and M. Buss, "Grid-based mapping and tracking in dynamic environments using a uniform evidential environment representation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[4] F. Andert, "Drawing stereo disparity images into occupancy grids: Measurement model and fast implementation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[5] K. Pirker, M. Ruther, H. Bischof, and G. Schweighofer, "Fast and accurate environment modeling using three-dimensional occupancy grids," in *IEEE International Conference on Computer Vision Workshops*, 2011.

[6] K. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems," in *Proceeding of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[7] M. B. C. S. W. B. Armin Hornung, Kai M. Wurm, "Octomap: an efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[8] S. Thrun, "Learning occupancy grids with forward models," in *Proceedings of the 2001 IEEE/RSU International Conference on Intelligent Robots and Systems*, 2001.

[9] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Massachusetts: Massachusetts Institute of Technology, 2005.

[10] R. M. Anderson A. S. Souza and L. M. G. Gonçalves, *Current Advancements in Stereo Vision, Chapter 9: 3D Probabilistic Occupancy Grid to Robotic Mapping with Stereo Vision*. Intech, 2012.

[11] E. Kaufman, T. Lee, Z. Ai, and I. S. Moskowitz, "Bayesian occupancy grid mapping via an exact inverse sensor model," in *American Control Conference*. IEEE, 2016, pp. 5709–5715.

[12] Z. A. T. L. Evan Kaufman, Kuya Takami, "Autonomous exploration with exact inverse sensor models," *Journal of Intelligent and Robotic Systems*, 2017.

[13] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 1997, pp. 146–151.

[14] —, "Frontier-based exploration using multiple robots," in *Second International Conference on Autonomous Agents*, ACM, Ed., 1998, pp. 47–53.

[15] M. L. Y. W. Cheng Zhu, Rong Ding, "A 3d frontier-based exploration tool for mavs," in *IEEE 27th International Conference on Tools with Artificial Intelligence*, 2015, pp. 348–352.

[16] D. W. P.G.C.N. Senarathne, "Towards autonomous 3d exploration using surface frontiers," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 34–41.

[17] C. D. Alexander Kleiner, "A frontier-void-based approach for autonomous exploration in 3d," *Advanced Robotics*, vol. 27, no. 6, pp. 459–468, 2013.

[18] K. S. Rahul Sawhney, K. Madhava Krishna, "On fast exploration in 2d and 3d terrains with multiple robots," in *International Conference on Autonomous Agents and Multiagent Systems*, 2009, pp. 73–80.

[19] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *In RSS*, 2005, pp. 65–72.

[20] A. T. L. Evan Kaufman, Zhuming, "Autonomous exploration by expected information gain from probabilistic occupancy grid mapping," in *Proceedings of the 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. San Francisco, USA: IEEE, December 13–16 2016, pp. 246–251.

[21] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[22] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[23] S. G. Johnson, "Saddle-point integration of c infinity bump functions," arXiv:1508.04376, August 19 2015.

[24] T. L. Farhad Goodarzi, Daewon Lee, "Geometric nonlinear pid control of a quadrotor uav on se(3)," in *2013 European Control Conference (ECC)*. IEEE, July 17–19 2013, pp. 3845–3850.