2012 IEEE International Conference on Robotics and Automation
RiverCentre, Saint Paul, Minnesota, USA
May 14-18, 2012

Autonomous Indoor 3D Exploration with a Micro-Aerial Vehicle

Shaojie Shen, Nathan Michael, and Vijay Kumar

Abstract— **In this paper, we propose a stochastic differential equation-based exploration algorithm to enable exploration in three-dimensional indoor environments with a payload constrained micro-aerial vehicle (MAV). We are able to address computation, memory, and sensor limitations by considering only the known occupied space in the current map. We determine regions for further exploration based on the evolution of a stochastic differential equation that simulates the expansion of a system of particles with Newtonian dynamics. The regions of most significant particle expansion correlate to unexplored space. After identifying and processing these regions, the autonomous MAV navigates to these locations to enable fully autonomous exploration. The performance of the approach is demonstrated through numerical simulations and experimental results in single and multi-floor indoor experiments.**

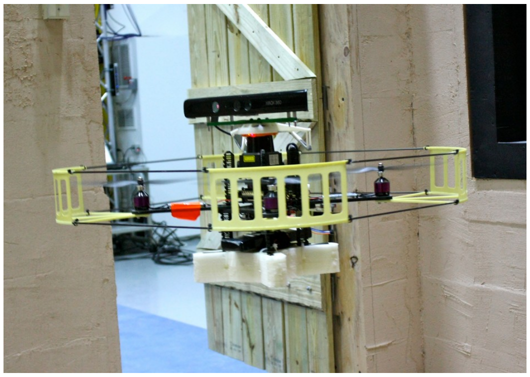


Fig. 1. The experimental platform with onboard computation (1.6 GHz Atom processor) and sensing (laser scanner, Microsoft Kinect, and IMU).

I. Introduction

In this paper, we present a methodology for exploration in three-dimensional indoor environments with a payload constrained micro-aerial vehicle (MAV). We consider the problem of autonomous exploration as consisting of two parts: (1) the definition of regions (frontiers) that, when visited, spatially extend the current environment model, and (2) autonomous navigation to those regions, including mapping, localization, planning, and control. As payload limitations on our MAV restrict the onboard processing and sensing options, we pursue a methodology for identifying regions for further exploration that is amenable to the system limitations.

Exploration is a classic problem in the field of mobile robotics and relevant to applications that require a robot to autonomously navigate through unknown environments. The two-part definition above is consistent with traditional exploration approaches such as entropy-, frontier-, and information gain-based exploration [1]–[5]. These strategies define locations in the map that, if visited by the robot, reduce environment uncertainty and guide the exploration process. Frontier-based exploration approaches generally compute exploration frontiers as the discrete boundary between the certain and uncertain regions of the current environment estimate. Thus, given a dense occupancy grid representation of the world, such a computation requires both known (occupied and unoccupied) and unknown cells in the map. Entropy- and information gain-based methods compute regions that reduce map uncertainty by considering the information currently available in the map paired with the probability of reducing the uncertainty in the map through sensing in unexplored regions.

While these approaches are effective in two dimensions (e.g. [6]–[8]), the naive extension of these methods to three dimensions introduces several challenges when considering systems with payload constraints. The primary challenge is the ability of onboard sensors to faithfully provide information about known and unknown environment regions in three dimensions. Due to limited available power, our MAV (Fig. 1) is only able to carry a restricted set of sensors such as lasers and cameras (mono, stereo, or RGB-D) that fit within the payload capacity of the vehicle. However, scanning lasers provide only partial information about the three-dimensional structure of the surrounding environment. The fact that any three-dimensional information is available from a laser rigidly mounted to a MAV is a consequence of the motion of the MAV during flight. Cameras suffer from similar limitations, where three-dimensional information is dependent on vehicle position and orientation. Hence, at any moment, full information about the surrounding environment is only available as a function of the current vehicle state and prior motions. A secondary challenge resulting from payload constraints is limited onboard processing. While this challenge continues to relax due to technological advancements, it is of pragmatic concern in this work as the use of the sensor data is restricted due to available onboard processing.

The problem of 3D exploration and mapping for ground robots is considered in the literature using simplified 3D polygonal representations [9] and 2.5D elevation maps [10]. These methods are able to acquire 3D models of the environment but for 2D locomotion and are generally unable to handle indoor environments that contain multiple stories. Complete 3D surface exploration is considered in the litera-

S. Shen, N. Michael, and V. Kumar are with the GRASP Laboratory at the University of Pennsylvania, Philadelphia, Pennsylvania.
Email: {shaojie,nmichael,kumar}@grasp.upenn.edu

We gratefully acknowledge the support of ARL Grant W911NF-08-2-0004, ONR Grants N00014-07-1-0829, N00014-08-1-0696 and N00014-09-1-1051, and the NSF Rapid Grant.

978-1-4673-1405-3/12/$31.00 ©2012 IEEE

ture, where depth discontinuities in the workspace are eliminated by using graph search [11] or frontier-void [12] based methods. Three-dimensional extensions of frontier regions are integrated with a vector field based approach in [13] to achieve 3D exploration with a stereo camera. However, these approaches require a high computational budget and operating environments are limited to small workspaces.

In this work, we detail our approach to exploration that addresses these challenges resulting from limited onboard sensing and processing, permitting autonomous exploration with a MAV in complex three-dimensional environments. We begin in Sect. II by describing the motivation for this work, the key concepts of our approach, and place these ideas into context with related literature. The methodology and algorithm details are provided in Sect. III. We present numerical simulations that contrast performance between a similar existing approach and our method in Sect. IV. Experimental results detailing an autonomous MAV exploring complex indoor environments are reported in Sect. V.

II. Motivation

Consider a robot starting in a completely unknown environment. As the robot acquires sensor information, it is able to build a map representing the unoccupied and occupied spaces of the environment. Concurrently, the robot identifies those sensed regions as known and therefore explored. The goal of this work is to identify regions that are presently unknown and guide the robot to those unknown locations, in the process exploring and expanding the map representation of the environment.

The development of our approach results from evaluation of existing methods, such as frontier-based exploration, with naive extensions to three dimensions. We found that sensors providing incomplete information about the surrounding three-dimensional environment often fail to accurately capture the differences between unoccupied and unknown space. We frequently observed cases where unknown and unoccupied regions of space were nearly co-located and sparsely filled regions of the environment that should clearly be identified as unoccupied. Frontier-based exploration performs poorly in this case as it relies on the boundary between unoccupied and unknown space to determine the next exploratory step of the vehicle (Fig. 2); yielding exploration strategies that ultimately drive the system to provide complete local sensor coverage but at the cost of reducing the rate of expansion of the map. Additionally, while in two dimensions, maintaining a dense map that contains a representation of known and unknown regions is computationally tractable, attempting the same in three dimensions quickly becomes intractable for systems with limited memory and computational capabilities due to the density of the information.

Therefore, we pursue an approach that does not require a dense representation of free space. In doing so, we address some of the issues resulting from incomplete sensor information. Fundamental to our approach is the observation (and assumption) that unstructured or uncluttered regions of the map generally correlate to unexplored regions of the indoor environment. Therefore, we wish to identify these regions as locations for further exploration. Following the literature, we call these regions *frontiers* as they serve to differentiate between the known and unknown regions of the environment.

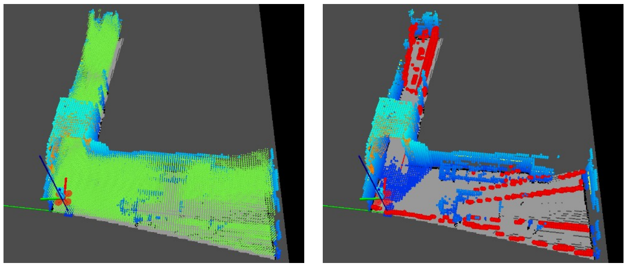

(a) (b)

Fig. 2. Challenges with traditional frontier-based exploration methods in three dimensions. Figure 2(a) depicts experimental data showing the observed free and occupied space given a dense voxel grid representation (where free and occupied space voxels are shown as green points and blue cubes, respectively). Frontier-based exploration methods look toward the boundary between free and unexplored space. However, in three dimensions, free space observations are often incomplete due to sensor field-of-view, resolution, and occlusions; yielding frontiers (red spheres in Fig. 2(b)) that result in poor exploration performance.

III. Identifying Frontiers for Exploration via Stochastic Differential Equations

A. Overview

The exploration algorithm begins when the robot starts sensing the environment and building a map of the occupied space while concurrently initializing particles that represent the environment free space based on sensor observations (Sect. III-C). The particles are dispersed through the known and unknown space with dynamics defined by a stochastic differential equation (SDE) that considers collisions with the known occupied space defined by the current map (Sect.III-D). After the simulated application of particle dynamics, exploration frontiers are identified based on the particle dispersion and sent to the autonomous navigation system in the form of navigation goals (Sect. III-E). The MAV navigates to these locations while incorporating sensor information into the map and defining new particles based on the sensor observations of the free space. After the final frontier is visited by the vehicle, the particle set is resampled based on the local density and current particle set (Sect. III-F), the SDE is re-evaluated and new frontiers are identified and sent to the autonomous navigation subsystem.

We refer to the process of initializing, extracting frontiers, and resampling as the Stochastic Differential Equation-based Exploration algorithm (SDEE). The SDEE algorithm is run repeatedly as required for the duration of the experiment or until the environment is completely explored. Each step of the algorithm is detailed in the remainder of this section.

B. Assumptions

We begin by assuming that the MAV is able to autonomously navigate. Hence, this work builds upon previous

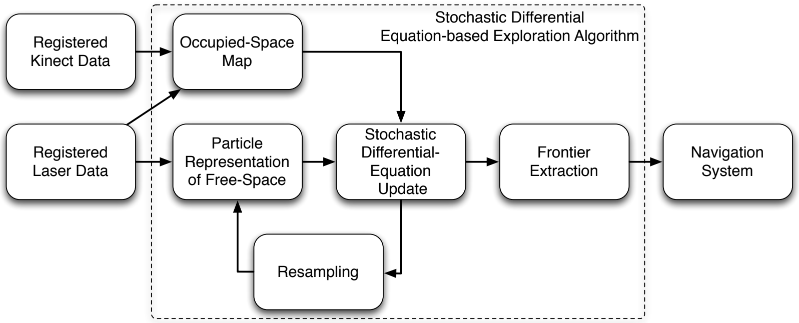


Fig. 3. The autonomous exploration system design.

results [14] where we detail the design of a fully autonomous MAV system able to localize, map, plan, and control onboard the vehicle without requiring external infrastructure or human interaction beyond high-level navigation goals. We define the output of the SDEE algorithm as the equivalent input of the navigation system. Thus, the SDEE algorithm provides a sequence of goal locations in the current map (see Fig. 3).

We assume that the characteristic sensor length, or maximum observable distance of the least capable sensor (in terms of range), is known in advance, and denote this value as D. For this work, we set the maximum observable distance of the laser scanner and RGB-D camera sensor as 30 m and 4 m, respectively. Hence, $D = 4\,\mathrm{m}$ for the experimental results.

C. Particle-based Representation of Free Space

Our approach requires a sparsely sampled representation of the free space. As the robot observes and maps the environment, it generates or emits particles at known free space locations according to the sensors. Therefore, as the robot is equipped with a laser scanner and RGB-D camera sensor, we may readily define free space locations by uniformly sampling over the observed unoccupied space – the space between the robot and the observed occupied space.

Define a set of particles, $\mathbf{X}$, where each particle, x_i, represents a discrete point-mass in free space ($\mathbb{R}^2$ or $\mathbb{R}^3$ for this work). As the robot observes the environment, the number of particles, $N = |\mathbf{X}|$, increases. We require that at least N_{min} particles exist by enforcing this fact through resampling (Sect. III-F). N_{min} represents the minimum number of particles required to represent the volume of the free space in our framework. Additionally, we define N_{max} as the maximum number of particles permitted to exist at any time. N_{max} is selected based on CPU capabilities.

Therefore, excluding the initial phase, $N_{\text{min}} \leq N \leq N_{\text{max}}$. When $N = N_{\text{max}}$ we force a resample of the particles. In doing so, we reduce the number of particles to N_{min} while also possibly identifying new exploration frontiers as navigation goals. During the experiments we also limit the rate of particle emission by only creating new particles after a small translation or rotation. This heuristic is applied to prevent the unnecessary emission of particles while the robot hovers in place and generally observes the same parts of the environment.

Thus far we have only identified the particles as representing point-masses in the free space. We now associate with the particles dynamics that allow us to simulate particle expansion given the current map of the occupied space.

D. SDE-based Particle Dynamics

Consider an enclosed three-dimensional environment that contains a fixed number of gas molecules with dynamics that follow a Langevin equation:

$$m\ddot{x}_i(t) = -\nabla U(x_i(t)) - \gamma \dot{x}_i(t) + \sqrt{2\gamma k_b T}\eta(t) \qquad (1)$$

with m defining the molecule mass, $x_i(t) \in \mathbb{R}^3$ is the position of the i^{th} molecule at time t, $U(x_i(t))$ is the potential due to interactions between molecules, γ is a damping term, k_b is the Boltzmann constant, T is the temperature, and $\eta(t)$ is a δ-correlated stationary zero-mean Gaussian process.

We want the molecules to emulate an *ideal gas*. This means that the molecules do not interact ($U(x_i(t)) = 0$), and the collisions are perfectly elastic and satisfy the assumptions for frictionless impact. The evolution of the state of each molecule, $x_i(t)$, is dictated by (1), with initial conditions $x_i(0)$, $\dot{x}_i(0)$, and the environment. At steady state, the pressure of the gas, P, in an environment volume, V, is given by the ideal gas law:

$$P = \mu \frac{N}{V} \qquad (2)$$

where $\mu = k_b T$ can be considered to be a constant.

As the particles disperse through the environment according to the motion equations in (1), the volume increases and according to (2), the pressure and the density of particles, $\rho = \frac{N}{V}$, decrease. Both of these quantities, P and ρ, can be estimated locally based on the distances between particles. The lowest density regions, which are areas of maximum expansion, correspond to unexplored regions and can be used to guide the identification of frontiers as discussed in Sect. III-E.

1) Time and Length Scales: From (1), the time constant associated with the deterministic, macro-scale Newtonian motion of a particle is given by:

$$\tau = \frac{m}{\gamma}.$$

A choice of the time scale and the initial velocity of a particle leads naturally to a length scale. By solving the differential equation we get an expression of the distance travelled by the particle:

$$x_i(t) - x_i(0) = \tau \dot{x}(0)(1 - e^{\frac{-t}{\tau}}).$$

Thus the length scale, L, associated with the time scale τ is given by:

$$L = \tau \dot{x}_i(0)(1 - e^{-1}) \tag{3}$$

It is natural to choose the length scale based on the maximum sensor measurement range, $L \sim D$. Thus, we can pick either the initial velocities for the particles in the SDE simulation or the time scale τ and determine the other using the equation above.

2) SDE Integration and Initialization: Each run of the SDE simulation is carried out for a randomly chosen initial velocity $\dot{x}_i(0)$ for the time duration τ. However, the integration time step, Δt, and initial velocity must be chosen to ensure that we detect any particle collisions with the environment. Therefore, we choose Δt and define $\dot{x}_i(0)$ as a function of Δt and the resolution of the occupied space with voxels of dimension $\Delta M \times \Delta M \times \Delta M$. To ensure that we are able to detect any possible collisions of the particles with the environment, we require $\|\dot{x}_i(0)\| = \frac{\Delta M}{\Delta t}$. The magnitude is consistent across all particles and we draw the direction of the initial velocity vector randomly from a uniform distribution in azimuth and elevation angles.

In the experiments we found that the exploration rate is improved if we bias the velocity vector directions to favor x-y motion over z motion. This improvement is due to the fact that buildings tend to be larger in length and width as compared to height.

E. Frontier Extraction

Based on the previous statement that we are interested in detecting regions of largest volumetric expansion, we can consider volumetric or density changes as a function of local changes in $\mathbf{X}$ resulting from the SDE simulation. For the sake of brevity, we discuss here only detection based on volumetric changes as in practice this method yields better performance with fewer false positives – frontiers that empirically do not correspond to points for further exploration. However, changes in local density are readily computed and may be approached in a similar manner.

Denote the sets $\mathbf{X}(0)$ and $\mathbf{X}(\tau)$ as $\mathbf{X}_0$ and $\mathbf{X}_\tau$, respectively. We are interested in the particles, $x_{i,\tau} \in \mathbf{X}_\tau$, that are spatially separated from the particles, $x_{i,0} \in \mathbf{X}_0$. To this end, we construct a KD-Tree based on the $\mathbf{X}_0$, compute the closest neighbor to the particle $x_{i,\tau}$ for all $i = 1, \ldots, N$, and set this value to $\hat{d}_i$. Therefore, $\hat{d}_i$ represents:

$$\hat{d}_i = \min_{x_{j,0} \in \mathbf{X}_0} \|x_{i,\tau} - x_{j,0}\|$$

and measures the volumetric change between the two particle sets. We choose the largest distances through thresholding:

$$\hat{d}_i > \alpha D$$

where $\alpha \in (0, 1]$ is a scalar. Decreasing α will yield more possible frontier locations but at the cost of increasing the number of false positive solutions. The selection of this parameter is discussed in Sect. III-H.

The actual goal for autonomous navigation should be in known free space to ensure that the robot is able to continue to localize itself. Therefore, we define the point g_i as the position of the i^{th} particle immediately following its last reflection (assuming an obstacle collision occurs). The orientation of the goal is defined as the orientation of the velocity of the i^{th} particle at g_i: $\frac{\dot{g}_{i,\tau}}{\|\dot{g}_{i,\tau}\|}$. Hence, the exploration goal will always have direct line-of-sight sensing of both explored and unexplored space. These goals are clustered, sorted based on proximity to the robot, and sent to the global planner and controller to achieve full autonomy [14].

F. Resampling

As we are continually emitting particles into the free space representation of the environment, we wish to maintain some bound on the number of particles. We focus the resampling process on preserving the local density in the free space representation. We define the weight w_i of the i^{th} particle using the metric:

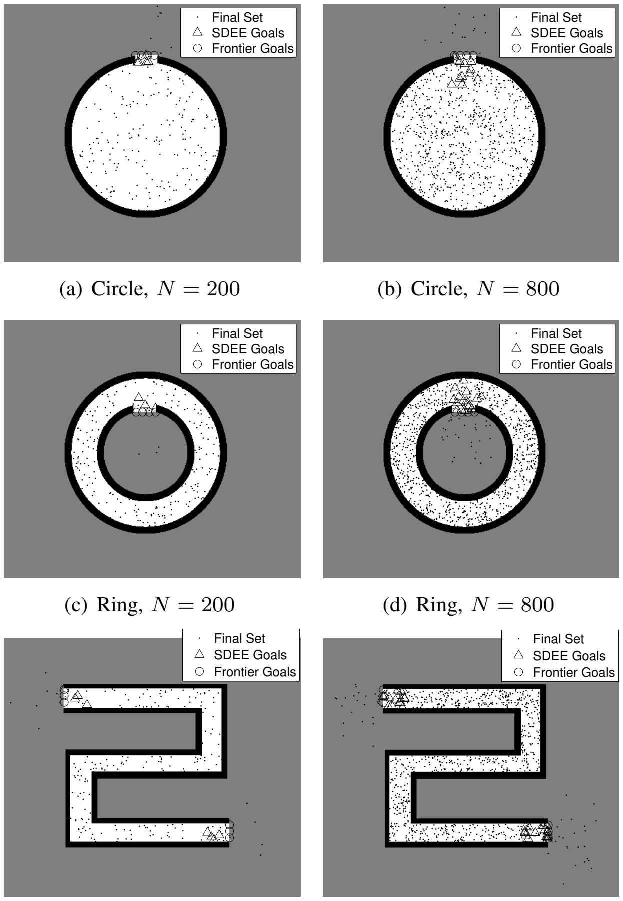
$$d_i = \frac{1}{k_i} \sum_{j \in K_i} \|x_j - x_i\|$$

$$w_i = \frac{d_i}{\sum_{j=1}^{N} d_j}$$

where K_i denoting the neighboring set of the nearest k_i particles (i.e. $|K_i| = k_i$). Note here that w_i is analogous to the density of a mixture medium with uniform mass per particle but with particles in lower density regions having a higher weight, and thus a higher probability of being sampled.

The particles are resampled according to their weights and the particles representing the free space samples after this procedure become the set $\mathbf{X}$. As resampling may result in picking the same particle multiple times, particle degeneracy becomes a concern. However, the SDEE algorithm introduces stochasticity into the particle state via the SDE step, avoiding the issue of particle degeneracy.

A complication in our approach is that we rely on sparse density of particles to provide insight into the locations of frontier regions. However, as the volume size grows in time, maintaining N as the constant resampling size yields an increasingly sparse representations of the environment free space. Therefore, we introduce the step of pruning the set $\mathbf{X}$ based on the difference between the present time and when the particle was created. As particles are emitted in areas of new sensor observations, older particles correspond to regions that were previously observed some time prior. A cost to forgetting particles in regions of previous exploration is that if the robot returns to the location, it will not be able to distinguish that the region was previously explored. However, this concern is easily overcome via the loop-closure methods employed by the autonomous navigation system [14] to determine that the area was previously visited. Note that pruning is only necessary as $N \to N_{\text{max}}$.



(a) Circle, $N = 200$

(b) Circle, $N = 800$

(c) Ring, $N = 200$

(d) Ring, $N = 800$

(e) Tube, $N = 200$

(f) Tube, $N = 800$

Fig. 4. Numerical simulations of the SDE-based frontier identification approach. The output of the SDEE algorithm is shown for two cases, $N = 200$ and $N = 800$, for three different types of environments. We show the resulting frontier goals (g_i) and results from traditional frontier-based exploration.

G. Complexity

The SDEE algorithm consists of three computational steps: simulation of the SDE, selection of the frontiers, and resampling. The complexity of the SDE is $\boldsymbol{O}(TN)$, where $T = \tau/\Delta t$ is the total number of update steps. The complexity of the frontier identification and resampling are both $\boldsymbol{O}(N \log N)$. Therefore, the total complexity of our exploration algorithm is $2\boldsymbol{O}(N \log N) + \boldsymbol{O}(TN)$.

Available onboard memory is most greatly impacted by the occupied and free space storage. For a map of size $M \times M \times M$, the occupied-space requires $\boldsymbol{O}(M^3)$ storage space. This amount may be reduced by leveraging sparse mapping methods [15] such that the required memory becomes $\boldsymbol{O}(mM^2)$, where m is a small number compared to M. The free space particle representation requires $\boldsymbol{O}(N)$ space to store. Note that for a voxel grid free space representation the map will require $\boldsymbol{O}(M^3)$. In most practical applications $N \ll M^3$. Therefore, the proposed algorithm can run using much less memory than a dense voxel grid environment representation.

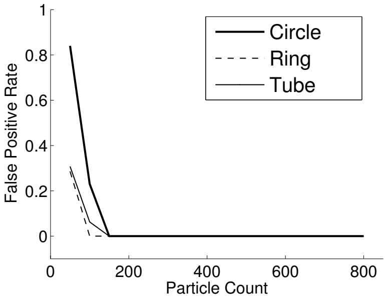


Fig. 5. The SDEE algorithm is sensitive to the setting of N_{min} and α. Here we show the effects of varying N while holding $\alpha = 0.6$ for the three environments in Fig. 4. The figure depicts the average of ten simulated trials for each N for each representative map. The rate of false positives is the percentage of identified frontiers that do not lie in the unexplored space.

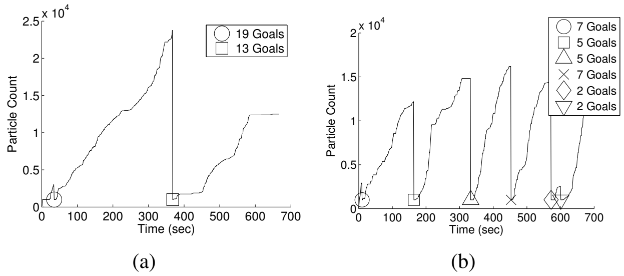
H. Algorithm Parameters

The SDEE algorithm requires selection of two interrelated parameters N_{min} and α. The parameters represent the notion of particle dispersion and differentiate between the uncluttered environment as unexplored space or unoccupied space. Define the minimum density $\rho_{\text{min}} = \frac{N_{\text{min}}}{V_{\text{min}}}$, where V_{min} represents the volume spanned by these particles. Also, define the density associated with the thresholded set of particles with $\hat{d}_i > \alpha D$ as ρ_α. We wish for $\rho_\alpha \ll \rho_{\text{min}}$. In practice, we find a suitable α and N_{min} for a simulation environment with similar spatial characteristics and use these values for all future experiments. In general, these values need only be within the correct order-of-magnitude and are not sensitive to small changes. However, moving to a new environment with very different spatial characteristics will require new parameter selection.

IV. Comparison to Frontier-based Exploration

In this section, we compare the performance of the SDE-based frontier extraction to traditional frontier-based exploration methods [1]. As noted previously (Sect. II) and depicted in Fig. 2, the traditional approach performs poorly in three dimensions. For this reason, we consider the comparison in two dimensions. In Fig. 4 we show three different maps of different geometric characteristics, the output of the SDE-based frontier extraction (Sect. III-E), and the output of the tradional frontier-based approach. We see that the SDE-based approach yields results similar to the tradional approach.

As previously detailed, algorithm parameters, specifically N_{min} and α, play an import role in performance. In Fig. 5, we depict the rate of false positive frontiers (i.e. the percentage of identifed frontiers that do not lie in the unexplored space) while varying N and holding $\alpha = 0.6$. The figure depicts the average of ten simulated trials for each N for each representative map in Fig. 4. Note that for $N < 100$, the false positive rate is high, but when $N > 150$ this number drops to zero and remains at this value. This fact supports the argument in Sect. III-H that one need only select an appropriate N_{min} and α for a simulated environment of simular scale and the algorithm will work well for a variety



(a)

(b)

Fig. 6. The number of particles required during exploration and identified and clustered goals resulting from the SDEE algorithm. The update of the SDEE algorithm is clearly visible as the particles reduce to $N_{\min}$ during resampling. Figs. 6(a) and 6(b) correspond to the results shown in Figs. 7 and 8, respectively.

of different environments. These parameters must be reconsidered only when the environment scale and characteristics change considerably.

V. Experimental Results

A. Experiment Design and Implementation Details

We present two experiments to demonstrate the performance of the proposed algorithm in 3D indoor environments: (1) a single floor exploration in the hallway of a building; (2) a full 3D exploration in an unstructured multi-floor environment. In both experiments, the exploration process is completed when the SDEE algorithm no longer identifies frontiers for further exploration.

The robot platform is sold by Ascending Technologies, GmbH [16] and equipped with an IMU (accelerometer, gyroscope, magnetometer) and pressure sensor. We developed custom firmware to run at the embedded level to address feedback control and estimation requirements. The other computation unit onboard is a 1.6 GHz Atom processor with 1 GB of RAM. The sensors on the robot include a Hokuyo UTM-30LX (laser), and a Microsoft Kinect sensor. A custom 3D printed mount is attached to the laser that houses mirrors and redirects a small number of laser beams upward and downward. Communication with the robot for monitoring experiment progress is via 802.11n networking. Figure 1 shows a picture of our robot platform. All algorithm development is in C++ using ROS [17] as the interfacing robotics middleware. The experimental environment includes two buildings in the School of Engineering and Applied Science at the University of Pennsylvania. In all experiments, the robot starts without any knowledge of the environment and operates fully autonomously without any human interaction. We bound the total size of the environment in order to ensure mission completion within the battery life of the robot.

B. Exploration of a Single Floor Hallway

In this experiment, the robot explores a single floor hallway. Figure 7 shows the intermediate stages of the exploration process. The robot continuously explores and gathers information as it traverses the length of the hallway. This experiment requires the full lifetime of the battery and results in a complete map of the environment, including a dense covering of all vertical walls, floors, and ceilings, as shown in Fig. 7(h). The number of particles and goals as well as the SDEE algorithm updates with resampling associated with this trial are visible in Fig. 6(a).

C. Exploration of a Multi-floor Building

In this experiment, the robot operates in an unstructured lobby of a multi-floor building, where there are several vertical spaces for the robot to explore. Figure 8 shows the intermediate stages of the exploration process. We see the goals that lead the robot to finish the exploration of the first floor (within the boundary) before exploring the vertical direction.

Figure 8(h) shows the full 3D map created by the robot after exploration. Despite the fact that the ceiling height exceeds four meters, the proposed algorithm successfully finds exploration goals that guide the robot to sense the high ceiling area, resulting in full coverage of the ceiling and the second floor. The number of particles and goals as well as the SDEE algorithm updates with resampling associated with this trial are visible in Fig. 6(b).

VI. Conclusion and Future Work

In this paper, we propose a stochastic differential equation-based exploration algorithm to enable exploration in three-dimensional indoor environments with a payload constrained MAV. We are able to address memory and sensor limitations by considering only the known occupied space in the current map (as opposed to also representing known free space and unknown space). We determine regions for further exploration based on the evolution of a stochastic differential equation that simulates the expansion of a particle system with Newtonian dynamics. The regions of most significant particle expansion correlate to unexplored space. After identifying and processing these regions, the autonomous MAV navigates to these locations to enable fully autonomous exploration. The performance of the approach is demonstrated through numerical simulations and experimental results in single and multi-floor indoor experiments.

We are interested in moving forward in two directions. At present, our path generation does not take into account optimal platform motion given energy dissipation or information gain [18]. We are interested in pursuing trajectories that seek to optimize based on these properties. Finally, we are interested in extending our methods to consider autonomous exploration with multiple aerial robots.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005.

[2] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. of Robot.: Sci. and Syst.*, Cambridge, MA, June 2005, pp. 65–72.

[3] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *IEEE Intl. Sym. on Comput. Intell. in Robot. and Autom.*, July 1997, pp. 146 –151.

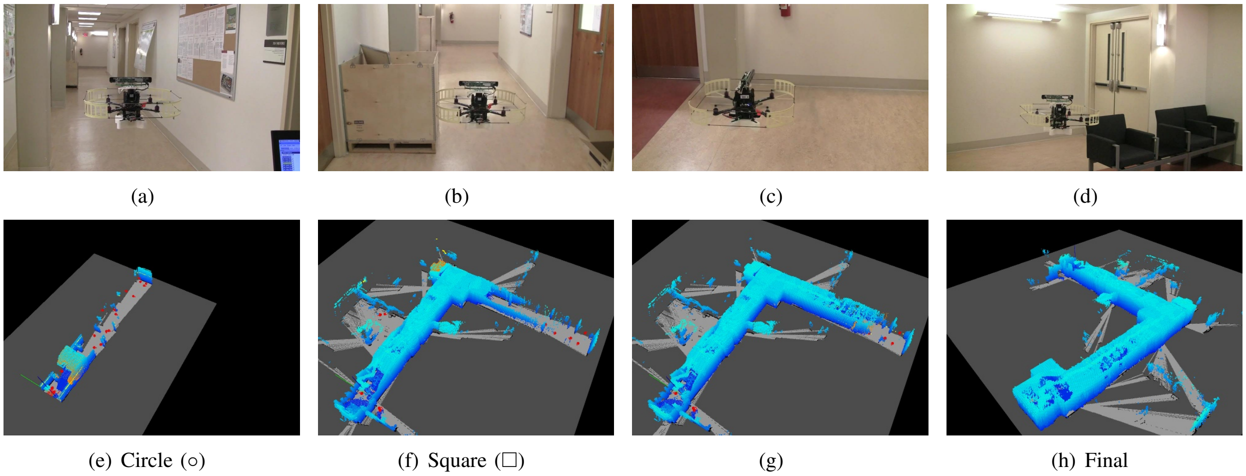


Fig. 7. Exploration of a single floor hallway (Figs. 7(a)-7(d)) and visualization of the map, SDEE goals (red spheres), and sensor information (Figs. 7(e)-7(h)). The circle (○) and square (□) correspond to the same times noted in Fig. 6(a) via the corresponding shapes. Videos of the experiments are available at http://mrsl.grasp.upenn.edu/shaojie/ICRA2012.mov.

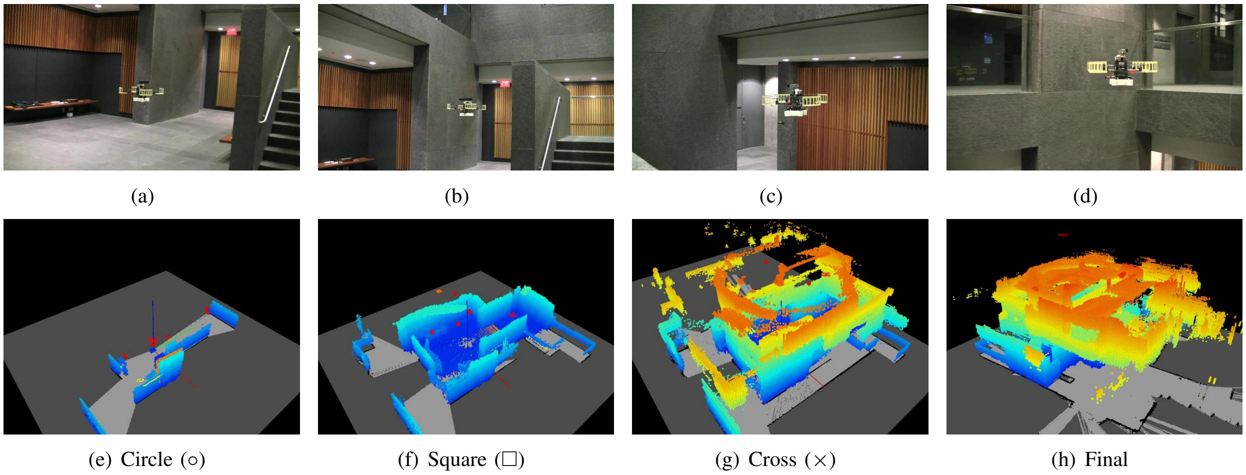


Fig. 8. Exploration of a multi-floor building with online data visualization. The circle (○), square (□), and cross (×) correspond to the same times noted in Fig. 6(b) via the corresponding shapes.

[4] A. Visser and B. A. Slamet, "Balancing the information gain against the movement cost for multi-robot frontier exploration," in *Euro. Robot. Sym.*, ser. Springer Tracts in Advanced Robotics. Springer Berlin, 2008, vol. 44, pp. 43–52.

[5] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unknown indoor environments," *Intl. J. of Micro Air Vehicles*, vol. 1, no. 4, pp. 217–228, December 2009.

[6] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hahnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker, "Autonomous exploration and mapping of abandoned mines," *IEEE Robot. Autom. Mag.*, vol. 11, no. 1, pp. 79–91, Dec. 2004.

[7] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schultz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proc. of the IEEE*, vol. 94, no. 7, pp. 1325–1339, July 2006.

[8] S. Bhattacharya, N. Michael, and V. Kumar, "Distributed converage and exploration in unknown non-convex environments," in *Intl. Sym. on Distributed Auton. Syst.*, Lausanne, Switzerland, Nov. 2010.

[9] A. Nuchter, H. Surmann, and J. Hertzberg, "Planning robot motion for 3D digitalization of indoor environments," in *Proc. of the Intl. Conf. on Adv. Robot.*, Coimbra, Portugal, June 2003, pp. 222–227.

[10] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous exploration for 3D map learning," in *Fachgesprach Autonome Mobile Systeme*, Kaiserslautern, Germany, Oct. 2007.

[11] R. Shade and P. Newman, "Discovering and mapping complete surfaces with stereo," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Anchorage, AK, May 2010, pp. 3910–3915.

[12] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3D," in *Proc. of IEEE Intl. Sym. on Safety, Security, and Rescue Robotics*, Kyoto, Japan, Nov. 2011.

[13] R. Shade and P. Newman, "Choosing where to go: Complete 3D exploration with stereo," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011, pp. 2806–2811.

[14] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Shanghai, China, May 2011.

[15] I. Dryanovski, W. Morris, and X. Jizhong, "Multi-volume occupancy grids: An efficient probabilistic 3d mapping model for micro aerial vehicles," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Taipei, Taiwan, Oct. 2010, pp. 1553–1559.

[16] "Ascending Technologies, GmbH," http://www.asctec.de/.

[17] "Robot Operating System," http://pr.willowgarage.com/wiki/ROS/.

[18] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Intl. J. Robot. Research*, vol. 8, pp. 1448–1465, Dec. 2009.