Proceedings of the 2017 18th
International Conference on Advanced Robotics (ICAR)
Hong Kong, China, July 2017

Towards Autonomous Exploration with Information Potential Field in 3D Environments

Chaoqun Wang [1], Lili Meng [2], Teng Li [2], Clarence W. De Silva [2] and Max Q. -H. Meng [1]

***Abstract*— Autonomous exploration is one of the key components for flying robots in 3D active perception. Fast and accurate exploration algorithms are essential for aerial vehicles due to their limited flight endurance. In this paper, we address the problem of exploring the environment and acquiring information using aerial vehicles within limited flight endurance. We propose an information potential field based method for autonomous exploration in 3D environments. In contrast to the existing approaches that only consider either the traveled distances or the information collected during exploration, our method takes into account both the traveled cost and information-gain. The next best view point is chosen based on a multi-objective function which considers information of several candidate regions and the traveled path cost. The selected goal attracts the robot while the known obstacles form the repulsive force to repel the robot. These combined force drives the robot to explore the environment. Different from planners that use all acquired global information, our planner only considers the goal selected and the nearby obstacles, which is more efficient in high-dimensional environments. Furthermore, we present a method to help the robot escape when it falls into a trapped area. The experimental results demonstrate the efficiency and efficacy of our proposed method.**

I. INTRODUCTION

Unmanned Aerial Vehicles (UAV) have been widely used in 3D reconstruction, surveillance and agricultural monitoring. Autonomous exploration is a process that combines mapping and planning. The robot chooses where to go and determines path in order to build a high-quality map of the environment. The goal that robot chose could be the area that is close to the robot or the area that helps for better localization. Recently autonomous exploration has gain much attention in UAV research field. Many efforts have been devoted to help robots to collect information without the intervention of human pilots.

It is challenging for robots to autonomously build maps in totally unknown environments. Yamauchi proposed the seminal method using the frontier information to guide the exploration [1]. This method often operates on grid maps and frontier is the area between the free and unknown grids of the map. The robot selects the frontier close to its current location as its exploring region. While this method does not consider the information it will collect in the long run. Makarenko et al. proposed an information gain based method that compared the information gains at different frontiers and chose the one that can reduce more information entropy [2]. Stachniss et al. [3] employed a Rao-Blackwellized Particle Filter to compute the information gain, during which the cost of executing the path and information gain are both considered.

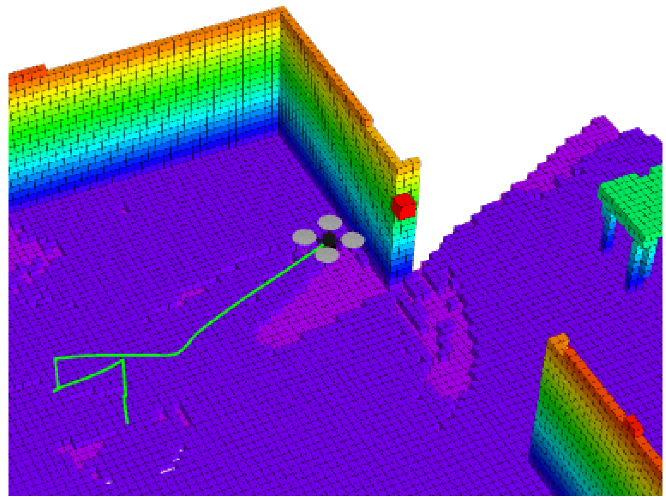


Fig. 1: **UAV exploration in 3D environment.** The colored voxels are 3D OctoMap representation and the green lines show the exploration trajectory.

In this work, we present an autonomous exploration method with information potential filed in 3D environment. As Fig.1 shown, we address the problem of autonomously exploring the 3D environments using aerial vehicles with limited flight time endurance. To minimize both the map entropy and the path cost during exploration at the same time, we proposed a multi-objective reward function to evaluate the potential candidate regions. The region that can satisfy the best reward will be selected as a goal point. The selected goal point together with the obstacles around the robot are used to construct the potential field. Furthermore, our method can handle the local minimum in traditional potential field based methods using virtual obstacles in the trapped area. We demonstrate the efficacy and efficiency of our method through evaluations against baseline method in Gazebo environment which simulates the real physical parameters.

The remaining of our paper is organized as follows. In Sec. II, we introduce related work on robot exploration. We

This project is partially supported by RGC GRF grants CUHK 415512, CUHK 415613 and CUHK 14205914, CRF grant CUHK6/CRF/13G, ITC ITF grant ITS/236/15 and CUHK VC discretional fund # 4930765, awarded to Prof. Max Q.-H. Meng.

[1]Chaoqun Wang and Max Q. -H. Meng are with department of Electrical Engineering, The Chinese University of HongKong, Shatin, HongKong {cqwang,qhmeng}ee.cuhk.edu.hk

[2]Lili Meng, Teng Li, and Clarence W. De Silva are with the Department of Mechanical Engineering, University of British Columbia, Vancouver, BC , Canada {lilimeng,tengli,desilva}mech.ubc.ca

provide the problem formulation in Sec. III.In Sec. IV, we present our proposed method in detail. Experiments will be given in Sec. V. We summarize our work and provide some future directions in Sec. VI.

II. Related Work

There are various efforts towards autonomous exploration. Bircher et al. use a sampling based receding horizon path planning method to assist the robot to explore the 3D environemnt [4]. During exploration, a random tree is computed online and the branch providing more information should be expanded. An efficient path planner that provides probabilistic guarantees for localization and fast exploration was proposed in [5]. Bai et al. [6] presented an exploration algorithm using information gain and mutual information. Bayesian Optimization to predict the information gain is used. It takes some iterations to train before determining the action.

A large body of work has been focusing on the exploration in 2D environments Most of them use 2D occupancy map to represent the environment and calculate the frontiers and corresponding information gain. With the popularity of 3D active perception, exploration in 3D environments has gained much attention in recent years. Christian proposed a frontier-void based approach [7], which considers the void volumes in 3D environments.

In simultaneously localization and mapping (SLAM) community, the exploration component is responsible for providing the path for better localization and perception. Locus et al. proposed a star discovery strategy, which is a local exploration strategy based on LSD-SLAM [8]. Gabriele et al. claimed that the robot should move to the feature-rich areas to reduce its localization uncertainty [9]. The path generated aims at improving the quality of the map, hence the path may not be the shortest. Therefore, it is not optimal for aerial vehicles with limited flight endurance.

Little work has focused on the path optimization during the exploration. For robots with limited flight endurance, path optimization is critical. Charrow et al. proposed a trajectory planning framework for exploration [10]. Firstly a set of candidate points are generated and then a trajectory that maximizes the information gain is chosen. The work in [11] presented an exploration approach with artificial potential field. A harmonic function is used to help with solving the local minimum caused by potential field method. The work [12] is similar to the one employs artificial potential field and an efficient OctoMap [13] is employed to represent the world. However, the information gain and uncertainty during the exploration are not considered. Ramanagopal et al. [14] proposed a potential field based motion planning strategy for mapping the bounded 3D structure. They emphasize the importance of 3D reconstruction using artificial potential field as a local planner, without considering the local minimum caused by different motion planning strategies.

Most methods in exploration focus more on the information collected when choosing the goal. However, the endurance time of robot is seldom considered. For aerial vehicle with limited payload and flight endurance, it is important to acquire more information within limited time. Inspired by the Artificial Potential Field (APF) [15], which offers an efficient and effective way for robot path planning, we employ the concept of information potential field. It is force source that drives the robot to fly towards the unexplored areas while avoiding obstacles. This method only employs the goal region and local information around the robot, which reduces the computation complexity and can work in real time. Our work is also similar to the idea of [16] by Joan Vallvé, in which trajectories considering both map and path uncertainty are computed. They focus on 2D mobile robot exploration with pose SLAM, while our focus is on 3D exploration using aerial vehicles, where high-dimension and limited time endurance pose more challenges. We use an efficient 3D OctoMap to represent the environments and mutual-information to reduce the path and map uncertainty.

III. Problem Formulation

During exploration, robots can actively interact with their surroundings according to the collected information in order to get the entire environment model [17]. The 3D space to be explored is defined as $V \subset R^3$. The entire space V is supposed to be unknown $V_{unknown}$ at the beginning, $V = V_{unknown}$. The space that the robot has already explored is set either V_{free} or V_{obs}, both are regarded as V_{known}. Our goal is to increase the number of voxels that belong to V_{known} while reducing the number of unknown voxels $c \subset V_{unknown}$ as much as possible.

We use entropy H to describe the uncertainty of the map [18]:

$$H(V) = -\sum_{k=1}^{N} p(c_k) log(p(c_k)) \tag{1}$$

here $p(c_k)$ is the occupancy probability of voxel c_k of the OctoMap, N represents the total number of voxels in the OctoMap. One of our goals of the exploration is to reduce the map uncertainty within limited time. The mutual information I is proposed to represent the reduced entropy after the robot takes a certain motion u.

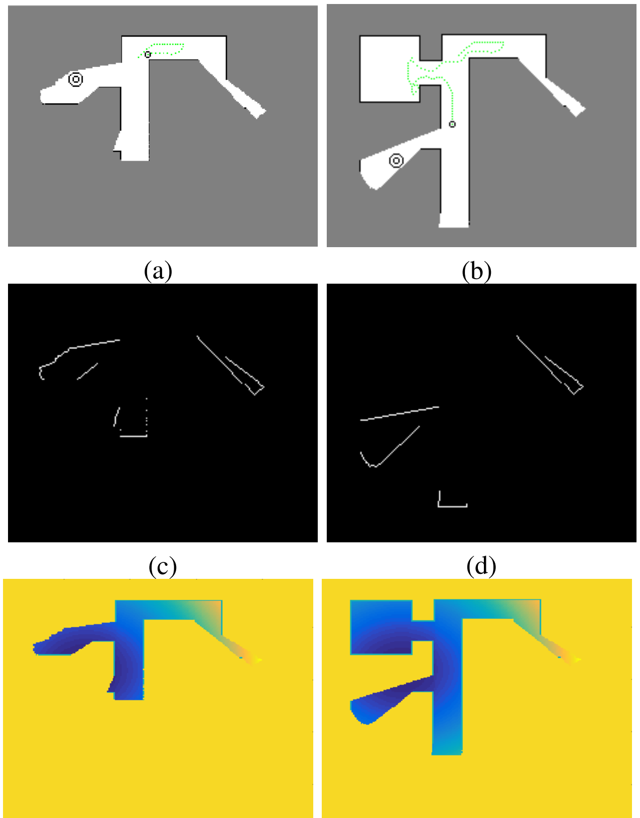
$$I(V) = H(V) - H(V|u) \tag{2}$$

Another goal arises from the use of aerial robot with limited endurance time. We use path length L to represent the path cost of exploration. We aim to reduce both L and H. Thus action u shall be chosen according to the utility function determined by the two requirements mentioned above:

$$u = argmin\ \alpha I \oplus \beta L \tag{3}$$

here α and β are coefficients that balance I and H. Operator $\oplus$ is defined to consider the fundamental units.

The control input $u_i \in U$, where $U = [u_1, u_2...u_n]$ and the corresponding robot position $x_i \in X$ and $X = [x_1, x_2...x_n]$. For potential field guided exploration, one common problem is the local minimum. We use $\Delta = [\delta_1, \delta_2, \delta_j, ...\delta_m]$ to represent the reward of all previous m positions. $\delta_j = ||x_i - x_{i-j}||$, where x_{i-j} represents robot position in j



(a) (b) (c) (d) (c) (d)

Fig. 2: **Information potential fields during the exploration.** (a)–(b) the occupancy map during the exploration. The smaller circle represents the robot position and the bigger circles indicate the selected goal region. (c)–(d) the frontiers extracted from the above occupancy map. (c)–(d) the potential field map that describes the potential field. The darker the color is, the more possible that the robot will explore.

previous timestamp . If $\delta_j = 0$, the robot is in a location that it has visited before. If δ_j is consistently zero for a period of time, it indicates the robot may fall into a local minimum. Our goal is to avoid the local minimum or try to escape from that trapped area when the robot gets stuck.

IV. Information Potential Field Exploration

Information Potential Field (IPF) is inspired by an efficient motion planning algorithm APF [15], which employs the attractive potential to attract the robot and repulsive potential to repel it. Different with our former sampling based planning algorithms for navigation purpose [19], [20], the APF is a local planner that only considers the obstacles within a certain range of the robot. The goal x_{goal} forms an attractive potential to attract the robot,while the obstacles x_{obs} that are around the robot forms a repulsive potential to repel the robot. The robot is guided to move by the sum of these two potentials. Similar to the idea of attractive and repulsive potentials of APF, we employ information potential field, in which the goal region is selected through information evaluation and path cost.

A. Information Potential Field

1) Goal Selection: We aim to explore the regions and reduce the exploring path length at the same time. Candidate regions can provide more information, i.e. the mutual information $I(V)$ is reduced as much as possible after taking an action u_i. As Fig.2 (b)–(c) shown, there are several candidate regions to explore. The selected goal position is the one that is near to robot's current location, and could provide more information relatively as well. The path length that the robot traveled $L(u, t)$ is a function of robot motion u and the time t, (3) becomes:

$$u = argmin\ \alpha I(u) \oplus \beta L(u, t) \tag{4}$$

To consider both the information gain $I(u)$ and path cost $L(u, t)$, we use the weights α and β. In the real implementation, these two values are chosen according to different environments.

2) Potential Field: After the goal is selected, the information potential field is determined by the goal position and the obstacles in the vicinity of the robot. The attractive potential field is determined by:

$$U_{att} = \gamma d^2(x_i, x_{goal}) \tag{5}$$

Here x_g is the selected goal position, $d(x_i, x_{goal})$ represents the distance between the current position x_i and x_{goal}. While the repulsive potential field is determined by two parts: $U_{obstacle}$ that is caused by obstacles around the robot, and U_{local} that is caused by the local minimum.

$$U_{rep} = \zeta U_{obstacle} + \lambda U_{local} \tag{6}$$

where

$$U_{obstacle} = \frac{1}{\sum_k (x_i - c_k)^2} \tag{7}$$

ζ and λ are the weights for these two potential fields. c_k is the same with the one in (1), and we only take into account the voxels around the robot. If the distance between the robot and obstacle exceeds a threshold, the $U_{obstacle}$ becomes zero. U_{lobal} will be explained in Sec. IV-B in detail.

(a) (b)

Fig. 3: **Local minimum avoidance through the local repulsive potential field.** The green line in (a) shows the robot trajectory during the exploration. (b) the corresponding local repulsive potential field.

Algorithm 1: Information Potential Field Exploration

Input: Point Clouds P
Output: Action to the next goal u
Initialization;
while *EntropyFlag* **do**
  $OctoMap \leftarrow OctomapServer(P)$;
  $[H, L] \leftarrow CandidateRegions(OctoMap)$;
  // H is the map entropy, L is the path cost.
  $x_{goal} \leftarrow GoalSelection(H, L)$;
  $U_{att} \leftarrow ComputeAttraction(x_i, x_{goal})$;
  $U_{obstacle} \leftarrow ComputeRepulsion(x_i, x_{goal}, OctoMap)$;
  if *LocalMinimumFlag* **then**
    $U_{local} \leftarrow ComputeLocalRepulsion(x_i, r)$
  end
  $U_{sum} = U_{att} + U_{obstacle} + U_{local}$;
  $F = \bigtriangledown U_{sum}$
end

The total potential filed U_{sum} is the sum of the attractive potential field and repulsion potential field:

$$U_{sum} = U_{att} + U_{rep} \tag{8}$$

The force F implemented on the robot is the gradient of the total potential field $\bigtriangledown U_{sum}$.

B. Local Minimum Avoidance

The robot may fall into traps if just relies on the basic idea of APF, especially for the multi-room environments that have concave obstacles. We propose the local repulsion potential field U_{local} to help the robot escape the local minimum area:

$$U_{local} = \frac{\epsilon(k)}{\sum_k (x_i - c_k)^2} \tag{9}$$

where

$$\epsilon(k) = e^{-t/\tau} \sum_{j=1}^{m} \frac{1}{1 + e^{-||x_i^k - x_{i-j}^k||}} \tag{10}$$

x_i^k and x_{i-j}^k represent the robot locates at voxel c_k at timestamp i and $i - j$ respectively. If a voxel c_k of the OctoMap is visited by the robot more than a certain times within a short time interval, then this voxel is seen as an virtual obstacle. In other words, if the robot is dropped into a local minimum, the voxels in the vicinity of the robot will be revisited multiple times in short time interval. The value $\epsilon(k)$ describes the probability of voxel c_k being occupied due to local minimum. The probability $\epsilon(k)$ is a decreasing function with time t, which begins at the timestamp that c_k is seen as virtual obstacle. As a rule of thumb, the robot may move back to revisit some places, so visiting multiple times at different relatively long time interval shall be treated differently. As Fig.3 shown, the robot falls into a local minimum, then the repulsive potential field will increase, as the yellow regions indicate in Fig.3 (b). The occupancy probability $\epsilon(k)$ of voxel c_k is decreasing with time. We can see in Fig.3 the locations that the robot visited a long time ago would contribute trivially to the repulsive potential.

C. Exploration Process

Algorithm 1 depicts the exploration process. The environment is represented by OctoMap. $OctomapServer$ is responsible for turning the point clouds P generated by RGB-D camera into OctoMap with the help of the localization component. With the goal selection algorithm described above, $GoalSelection$ selects the goal x_{goal} from the candidate regions according to the entropy H and the path cost L. $ComputateAttraction$ and $ComputateRepulsion$ are responsible for calculating the potential field when the robot is located at x_i. Using (10) , we can determine whether the robot is trapped in a local minimum. In our real implementation, the $LocalMinimumFlag$ will become true if $\epsilon(k)$ in (10) exceeds a certain threshold. The $ComputeLocalRepulsion$ here is used for calculating the local potential field that helps the robot to escape the trapped area. The force F that drives the robot is the gradient of the final potential field.

V. Experiment

A. Environment Setup

We use Gazebo simulation environment [21] and Robot Operating System (ROS) [22] running on top of Ubuntu 14.04. Gazebo uses real physical parameters of the robot and the environment, which provides reliable performance. All our experiments run on a ASUS laptop with Intel Core i7-4510U at 2GHz, 16GB memory. The environments are 3D indoor office like environment with walls that may cause local minimums. The flying vehicle is a quadrotor equipped with a Microsoft Kinect RGB-D sensor and a Hokuyo laser. Our quadrotor uses the model developed by Meyer [23], which is a stable robot model that adopts real physical parameters of quadrotor. The RGB-D sensor is responsible for generating point cloud data.

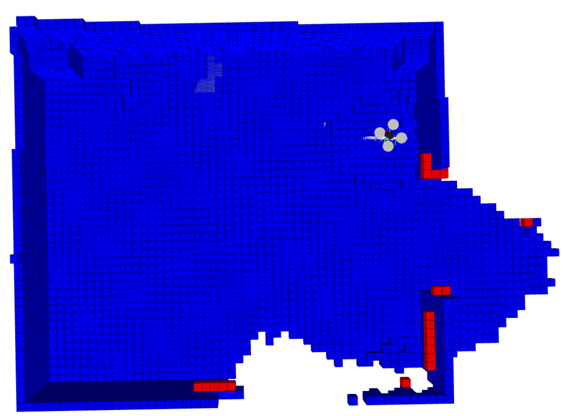
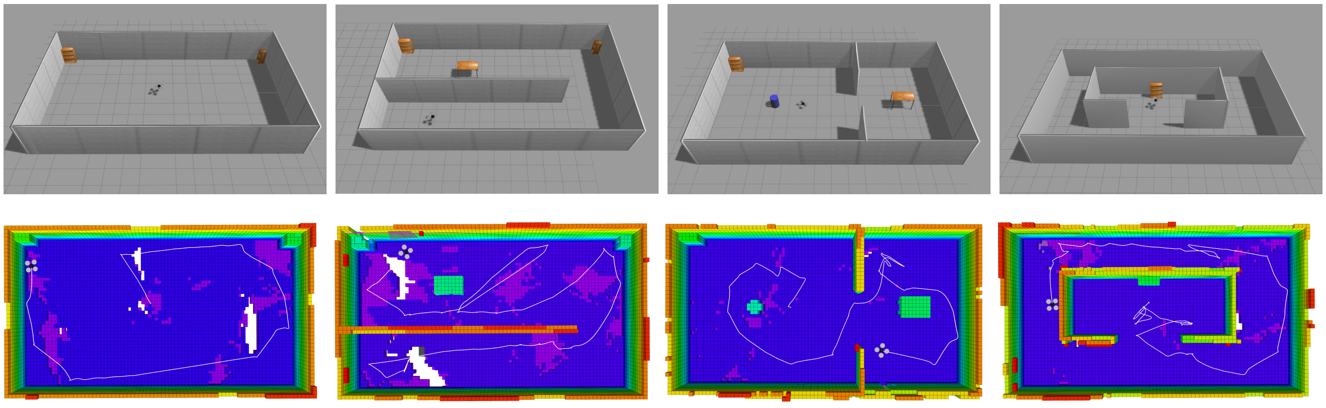


Fig. 4: **Frontier Extraction.** Blue markers indicate occupied voxels and red voxels represent extracted frontiers. The number of red voxels is used to evaluate the importance of reducing entropy.



(a) (b) (c) (d)

Fig. 5: **Autonomous exploration of UAV in four scenarios** The first row shows four environment we developed in Gazebo. The second row is the OctoMap built by our algorithm, the color of the cell indicates its height. The while line describe the trajectory of UAV during exploration.

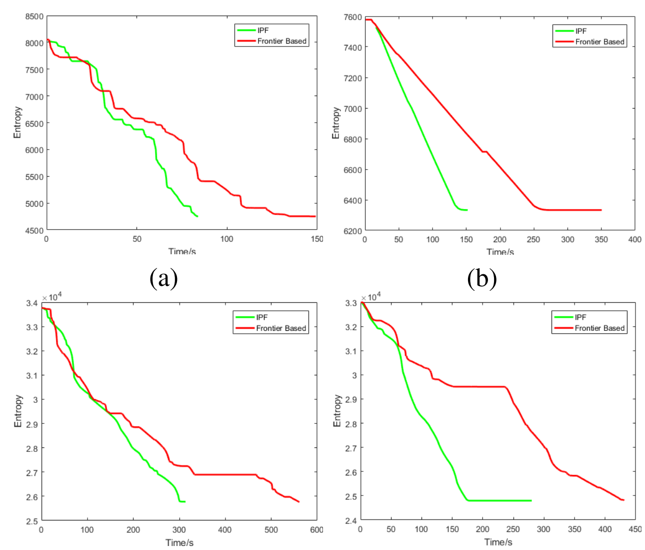
B. Frontier Extraction

We represent our 3D environment using OctoMap, which is an efficient probabilistic representation of free, occupied and unknown space. We can know the occupancy probability of the voxels and their positions, therefore, we can extract the required frontier voxels.

Without losing generality, we use extracted frontiers regions as the candidate regions. However, these candidate regions can also be other regions of interest, for example, the regions that provide the information for loop closure. Different with traditional frontier based method, we select the regions that can reduce the map entropy and path entropy as much as possible according to (4). The tool that we use is similar to the one in [24]. However, the tool in [24] can only generate a small amount of frontiers in large environments. Therefore we use the voxels that exist between free and occupied voxels as frontier. This simplification eliminates setting a bounding box of the whole environment, and is very efficient.

C. Results and Analysis

As Fig.4 shown, the frontiers can be extracted successfully in a 3D environment and then used for exploration. The frontier clusters are the metric for evaluating the entropy reduction in our experiment. The robot can determine the goal region that can reduce both the path cost and map entropy. The quadrotor can fly to the regions while avoiding collisions. When the vehicle falls into the trapped regions, the repulsive potential field will increase until the robot is pushed out of that region. This demonstrates that our algorithm can help with guiding the exploration of the whole environment in a safe and efficient way. We tested our algorithm in four different scenarios. The white lines in the second row of Fig.5 show the robot trajectory during exploration. In the first two scenarios, the environments are open fields with just a few obstacles. The exploration process behaves more like a wall following exploration algorithm. In Fig.5 (c)–(d), the robot operates in environments that have potential concave obstacles. The vehicle performs well



(a) (b) (c) (d)

Fig. 6: **Entropy reducing with time during exploration in four scenarios** The green lines describe our proposed information potential based method. The red lines are the performance of benchmark frontier-based exploration algorithm

in these two environments. When the robot is trapped into a local minimum, as shown in the second row of Fig.5 (d), it successfully escaped from the trapped area using our proposed method.

We also compare our method with the baseline frontier-based exploration method [1]. As Fig.6 shown, our method performs exploration faster than the traditional way of using frontier-based algorithm. In addition, the information gain is decreasing much faster in our method. We test our algorithm in four different environments. In Fig.6 (a)–(d), the complexity of the environments is increasing. The green lines show the performance of our algorithm while the red lines show the performance of baseline. In every scenario, the decreasing rate of the green line is faster than the red one. In complex environments such as Fig.6 (d), the frontier-based method is relatively slow. This may be because the frontier-

	IPF(our method)		Frontier-Based	
	Time cost(s)	Path length(m)	Time cost(s)	Path length(m)
1	83.9	**13.5**	149.1	**47.5**
2	313	**34.25**	560	**24.9**
3	150	**15**	350	**21.6**
4	280	**10**	431	**10.4**

TABLE I: **Cost evaluation in four scenarios.**

based method does not consider the information-gain. The robot always tries to explore the frontier regions without considering the information it would collect and the path cost. Also with the exploration process, the environment area that the global planner needs to consider becomes bigger and the corresponding computational overhead for the global path planner could be heavy. Our proposed method always chooses the goal that can reduce the entropy and path length. Hence our proposed algorithm outperforms the basic frontier-based exploration algorithm in the efficiency of reducing entropy. Our proposed algorithm can collect more information than traditional exploration strategies within certain time constraints. For the aerial vehicle executing the tasks such as environment sensing with limited payload and endurance time, this is especially critical.

Table I provides more detailed information of the path lengths and the time spent using our method and frontier-based method. Almost in all scenarios, the path length and time cost of our method are less than the frontier-based approach. The searching time and path cost along with the reduced entropy are often used to evaluate the efficiency of our path planner. Noted that in the second scenario, the path length of frontier-based method is less than our proposed algorithm, but our algorithm costs less time with the same amount of entropy reduction. Compared with the frontier-based exploration, our method achieves much faster exploration with relatively small path costs.

VI. Conclusion and Future Work

In this work, we present an efficient autonomous exploration method with information potential filed in 3D environment. To minimize both the map entropy and the path cost during exploration at the same time, we proposed a multi-objective reward function to evaluate the potential candidate regions. The region that can satisfy the best reward will be selected as a goal point. The selected goal point together with the obstacles around the robot are used to construct the potential field. Furthermore, our method can handle the local minimum in traditional potential field based methods using virtual obstacles in the trapped area. Experimental results demonstrate the efficacy and efficiency of our method. In the future, we would like to test other potential field functions that can make method more compact. Moreover, we would like to implement information potential field exploration algorithm in real aerial vehicles.

References

[1] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation*. IEEE, 1997.

[2] Alexei A Makarenko, Stefan B Williams, Frederic Bourgault, and Hugh F Durrant-Whyte. An experiment in integrated exploration. In *IROS*. IEEE, 2002.

[3] Cyrill Stachniss, Giorgio Grisetti, and Wolfram Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *RSS*, 2005.

[4] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *ICRA*. IEEE, 2016.

[5] Alexander Ivanov and Mark Campbell. An efficient robotic exploration planner with probabilistic guarantees. In *ICRA*. IEEE, 2016.

[6] Shi Bai, Jinkun Wang, Kevin Doherty, and Brendan Englot. Inference-enabled information-theoretic exploration of continuous action spaces. In *Proceedings of the 17th International Symposium on Robotics Research*, 2015.

[7] Christian Dornhege and Alexander Kleiner. A frontier-void-based approach for autonomous exploration in 3d. *Advanced Robotics*, 2013.

[8] Lukas von Stumberg, Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers. Autonomous exploration with a low-cost quadrocopter using semi-dense monocular slam. *arXiv preprint arXiv:1609.07835*, 2016.

[9] Gabriele Costante, Christian Forster, Jeffrey Delmerico, Paolo Valigi, and Davide Scaramuzza. Perception-aware path planning. *arXiv preprint arXiv:1605.04151*, 2016.

[10] Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. Information-theoretic planning with trajectory optimization for dense 3d mapping. In *Robotics: Science and Systems*, 2015.

[11] Edson Prestes e Silva, Paulo M Engel, Marcelo Trevisan, and Marco AP Idiart. Exploration method using harmonic functions. *Robotics and Autonomous Systems*, 2002.

[12] Robbie Shade and Paul Newman. Choosing where to go: Complete 3D exploration with stereo. In *ICRA*. IEEE, 2011.

[13] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. Octomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, 2013.

[14] Manikandasriram Srinivasan Ramanagopal and Jerome Le Ny. Motion planning strategies for autonomously mapping 3d structures. *arXiv preprint arXiv:1602.06667*, 2016.

[15] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*. Springer, 1986.

[16] Joan Vallvé and Juan Andrade-Cetto. Mobile robot exploration with potential information fields. In *Mobile Robots (ECMR), 2013 European Conference on*. IEEE, 2013.

[17] Peter Whaite and Frank P Ferrie. Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997.

[18] Cyrill Stachniss. *Robotic mapping and exploration*. Springer, 2009.

[19] Wenzheng Chi, Hitoshi Kono, Yusuke Tamura, Atsushi Yamashita, Hajime Asama, and Max Q-H Meng. A human-friendly robot navigation algorithm using the risk-rrt approach. In *RCAR*. IEEE, 2016.

[20] Jiankun Wang, Xintong Li, and Max Q-H Meng. An improved rrt algorithm incorporating obstacle boundary information. In *ROBIO*. IEEE, 2016.

[21] Nathan Koenig and Andrew Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *IROS*. IEEE, 2004.

[22] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*. Kobe, 2009.

[23] Johannes Meyer, Alexander Sendobry, Stefan Kohlbrecher, Uwe Klingauf, and Oskar Von Stryk. Comprehensive simulation of quadrotor uavs using ros and gazebo. In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2012.

[24] Cheng Zhu, Rong Ding, Mengxiang Lin, and Yuanyuan Wu. A 3d frontier-based exploration tool for mavs. In *Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on*. IEEE, 2015.