Autonomous Robotic Exploration by Incremental Road Map Construction

Chaoqun Wang, *Student Member, IEEE*, Wenzheng Chi, *Member, IEEE*, Yuxiang Sun, *Student Member, IEEE*, and Max Q.-H. Meng, *Fellow, IEEE*

***Abstract*— In this paper, we propose a novel path planning framework for autonomous exploration in unknown environments using a mobile robot. A graph structure is incrementally constructed along with the exploration process. The structure is the road map that represents the topology of the explored environment. To construct the road map, we design a sampling strategy to get random points in the explored environment uniformly. A global path from the current location of the robot to the target area can be found on this road map efficiently. We utilize a lazy collision checking method that only checks the feasibility of the generated global path to improve the planning efficiency. The feasible global path is further optimized with our proposed trajectory optimization method considering the motion constraints of the robot. This mechanism can facilitate the path cost evaluation for the next best view selection. In order to select the next best target region, we propose a utility function that takes into account both the path cost and the information gain of a candidate target region. Moreover, we present a target reselection mechanism to evaluate the target region and reduce the extra path cost. The efficiency and effectiveness of our approach are demonstrated using a mobile robot in both simulation and real experimental studies.**

***Note to Practitioners*—This paper is motivated by the efficient exploration problem, which plays a key role in various areas such as the information gathering and environment monitoring. In these applications, the mission length and executing time are often restricted by the battery capacity of the robot. In this paper, an efficient path planning framework is introduced to reduce the path length and exploration time. The robot keeps a road map of the environment to facilitate the path planning. The proposed target selection mechanism helps the robot determine the next best target to explore. Furthermore, the proposed trajectory optimization algorithm helps in reducing the path cost. Overall, this framework enables efficiently and autonomously exploration with a novel path planning framework.**

***Index Terms*— Autonomous exploration, road map, target selection, trajectory optimization.**

Manuscript received April 10, 2018; revised October 12, 2018; accepted January 14, 2019. Date of publication February 20, 2019; date of current version October 4, 2019. This paper was recommended for publication by Associate Editor R. Fierro and Editor K. Saitou upon evaluation of the reviewers' comments. The work of M. Q.-H. Meng was supported in part by the Hong Kong Research Grants Council General Research Fund under Grant 14205914, in part by Innovation and Technology Commission Innovation and Technology Fund under Grant ITS/236/15, and in part by the Shenzhen Science and Technology Innovation Project under Grant SZJCYJ20170413161616163. *(Corresponding author: Max Q.-H. Meng.)*

C. Wang is with the Department of Electrical Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: cqwang@ee.cuhk.edu.hk).

W. Chi is with the Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215021, China (e-mail: wzchi@suda.edu.cn).

Y. Sun is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong (e-mail: yxsun@link.cuhk.edu.hk; sun.yuxiang@outlook.com).

M. Q.-H. Meng is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, The Chinese University of Hong Kong, Shenzhen 518172, China (e-mail: max.meng@ieee.org).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author. It contains a video that gives the whole process of the proposed algorithm. A tree structure is incrementally built along with the exploration process, as can be seen in the video. Both the simulation and real experiments demonstrate the reliable performance of the proposed method. The total file size is 15.9 MB.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2019.2894748

I. Introduction

AUTONOMOUS robotic exploration is gaining increasing attention in recent years. Typically, it involves information collection of an environment using a robot without the intervention of the human pilot. Researchers could get a better understanding of the environment state through the environment exploration. At present, autonomous exploration is becoming one of the central tasks in various applications such as disaster relief [1], monitoring of water quality [2], and agricultural monitoring [3].

Decision-making and planning are of great importance for a robot to achieve high-level autonomy and flexibility in the exploration process. Decision-making module helps a robot select the next best view (NBV) according to the evaluation metric. One canonical evaluation metric for determining NBV is the distance from the current location of the robot to the frontier in the occupancy grid map [4]. Robot always selects the frontier that is closest to itself for further exploration. Another popular evaluation metric is the localization quality. Various work has been devoted to reducing the localization uncertainty during the exploration [5]. Oftentimes, the robot needs to plan a longer path to reduce its localization uncertainty because robust localization ability is a prerequisite for autonomous robotic exploration. In addition, there are also various evaluation metrics such as the amount of useful information collected during the exploration [6]. However, current research tends to focus on one evaluation metric rather than multiple ones in the NBV selection.

Although a considerable amount of research on decision-making has been done in the autonomous robotics exploration, there is still a need for efficient path planning approaches. For NBV selection involving path cost, it is simple and straightforward to use Euclidean distance for evaluations. This approach is efficient while it is inadequate in complex environments. Conventional planners such as search-based [7] and sample-based [8] methods are frequently used to generate

a trajectory for the NBV evaluation and robot execution. Once the robot determines a target region, it replans a path to execute, which is already generated during the previous evaluation process. Moreover, the choice of NBV leaves several candidate regions unexplored. When the robot is ready for exploring these unexplored regions, it needs to replan a path over the whole map although the robot already knows a path to that area. This replanning mechanism would suffer from a high computational complexity when the robot is exploring large or high-dimensional environments.

In this paper, we present an efficient path planning framework for autonomous robotic exploration in unknown environments. Our main contributions are in threefold.

1) We propose a graph-structured road map that is constructed along with the exploration process, which can facilitate the evaluation of the path cost to every unexplored candidate regions.
2) We propose a trajectory optimization scheme that can optimize the generated global path efficiently, in which the motion constraints of the vehicle are taken into consideration.
3) We develop a utility function that devotes to evaluating the candidate regions according to the path cost to the target area and the information that the robot can collect at the area.

This paper is structured as follows. In Section II, we review some related work on robot exploration. We formulate the exploration problem in Section III. The overall framework of our proposed method is presented in Section IV. In Section V, we introduce our proposed method in detail. Experimental studies are presented in Section VI. We summarize our work and outline future work in Section VII.

II. Related Work

Autonomous robotic exploration has been widely investigated and increasingly used in a number of applications in various fields. Hollinger *et al.* [9] use an underwater vehicle to inspect underwater object autonomously. Plonski *et al.* [10] develop exploration algorithm for habitat monitoring. Krainin *et al.* [11] use a variant of the NBV planning based on the information gain for 3-D object reconstruction using a robot manipulator. The manipulator moves the object to expose the most uncertain surface in order to build a complete object model. Moreover, autonomous exploration is applied for object search in [12]. Uncertainty semantics is utilized to build the relationship among objects to facilitate exploration. In the simultaneous localization and mapping (SLAM) [13], [14] research community, autonomous exploration often involves generating a path that reduces the localization and map uncertainty, which is widely known as the active SLAM [15].

A variety of methods have been proposed to tackle the autonomous exploration problem. The seminal work in [4] propose to use the nearest frontier as the evaluation metric to guide the exploration. The work in [16] adopts the similar strategy that utilizes the nearest neighbor approach for environment exploration. Stachniss *et al.* [6] use Rao–Blackwellized particle filter to build a map and utilized the information gain at different frontier regions to determine the next exploration action. Furthermore, Makarenko *et al.* [17] propose a utility function that incorporated the information gain, navigation, and localization quality for guiding the exploration. The three evaluation metrics are combined using a weighted sum. Lauri and Ritala [18] formulate the exploration as the partially observable Markov decision process. The ray-casting method [19] is applied for evaluating the information gain of candidate regions. Charrow *et al.* [20] present a trajectory planning framework for autonomous exploration. An original objective function based on Cauchy–Schwarz quadratic mutual information is proposed to determine the goal to explore optimally. The generated global path is further optimized to maximize the objective function and satisfy the motion constraints.

Recently, Dornhege and Kleiner [21] propose a frontier-void-based exploration approach that takes the void volumes in 3-D environments into consideration. In order to reduce the localization uncertainty in visual SLAM, Costante *et al.* [5] move the robot through a feature-rich area when it moves to the goal point. The work is dedicated to reducing the uncertainty in robot pose estimation, even the path in the exploration may not be the shortest. Bai *et al.* [22] propose to use the Gaussian process (GP) to model the unknown environment. The GP is then optimized using the Bayesian optimization method to predict the information gain. A star discovery exploration strategy is described in [23] based on the large scale direct-SLAM framework. The vehicle flies with a star pattern to generate the motion parallax for the SLAM module to achieve environment exploration.

Research focusing on balancing the path cost and the information gain during exploration is becoming increasingly important, especially when we want to build a precise map of the environment using a robot with energy constraint. Shnaps and Rimon [24] present an online algorithm for environment coverage using a mobile robot limited by the battery capacity. The work in [25] proposes a utility function to evaluate the path cost and the information gain of every candidate goal regions. Similarly, Heng *et al.* [26] use this function to evaluate the candidate target regions. Both the exploration and the coverage are considered in this paper. The work in [27] presents an original utility function considering the information and the movement cost in a normalized way. Velez *et al.* [28] propose an anytime planning framework for exploration, which considers the motion cost and the information gain at every motion primitive.

Efficient path planning during exploration is vital in large or high-dimensional environments. A sampling-based planning algorithm is proposed for information gathering [2]. The method aims to generalize an optimized trajectory, along which the robot can collect more information. Likewise, a receding horizon path planning method is proposed in [29], where the random tree is computed online and is extended to the area that contains more information. Wallar *et al.* [30] present a reactive motion planning strategy for multiple robots to achieve surveillance task. Robots utilize a riskmap to determine the next best region to survey. Shade and Newman [31] propose to use an artificial potential field (APF) [32] for path planning during the exploration. A harmonic function is

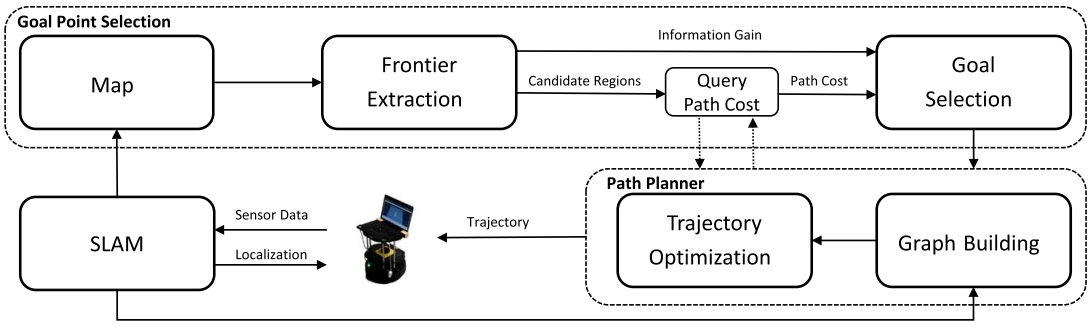

Fig. 1. High-level system architecture of exploration. The SLAM module is responsible for generating the map and localizing the robot simultaneously. The Frontier Extraction module extracts the information gain of the required regions for the following modules. The Goal Selection module generates the goal point for the path planner. The Path Planner module consists of an incremental constructed road map and following trajectory optimization module.

employed to avoid the local minimum problem caused by the APF method. The work in [33] also uses the APF planner for the 3-D reconstruction of the environment. Oriolo *et al.* [34] propose a sensor-based exploration strategy called SRT that can build the road map of the environment. A random sampling tree is expanded with the sensor coverage.

III. Problem Formulation

We model the environment using the occupancy grid map $\mathcal{M}$. Occupancy grids are cells in which the values encode the probability of being occupied. We denote the set of grid as $\{s_1, s_2, \ldots, s_k, \ldots, s_N\}$ and the whole 2-D space as $\mathcal{S} \subset \mathbb{R}^2$. The space is supposed to be unknown initially. During the exploration, the map grids can be divided into three types, the occupied cells $\mathcal{S}_{\text{occupied}}$, free cells $\mathcal{S}_{\text{free}}$, and unknown cells $\mathcal{S}_{\text{unknown}}$. Both occupied cells and free cells belong to cells that are known, i.e., $\mathcal{S}_{\text{known}} = \{\mathcal{S}_{\text{occupied}}, \mathcal{S}_{\text{free}}\}$. The objective of the exploration is to increase the number of cells that belong to $\mathcal{S}_{\text{known}}$ as many as possible.

We apply the entropy H to describe the uncertainty of the map [35]

$$H(\mathcal{S}) = -\sum_{k=1}^{N} p(s_k) log(p(s_k)) \tag{1}$$

where $p(s_k)$ is the occupancy probability of grid s_k in the occupancy grid map, and N represents the total number of grids in the map. We propose to reduce the map uncertainty H with less time and path cost during the exploration. We use the mutual information I to represent the reduced uncertainty after the robot executing a certain path $l_i \in \mathcal{L}$ and the path cost of l_i is c_i. The mutual information is defined as

$$I(\mathcal{S}; l_i) = H(\mathcal{S}) - H(\mathcal{S}|l_i) \tag{2}$$

where $H(S)$ is the current map entropy and $H(\mathcal{S}|l_i)$ is the entropy after executing a certain path l_i. For several candidate regions that lie in the frontier area, namely, $\mathcal{F} = \{f_1, f_2, \ldots, f_n\}$, they have two attributes, the path cost $c_i = L(l_i) \in \mathcal{C}$ to get there, and the information gain $I_i = I(\mathcal{S}; l_i) \in \mathcal{I}$ that it contains. Every time the robot chooses an exploration target to explore that satisfies

$$\arg\min_{f_i \in \mathcal{F}} \; U(f_i\{I_i, c_i\}). \tag{3}$$

We aim to find the target that is more informative while it is also near the current location of the robot. The function $U(\cdot)$ in (3) takes as arguments the information gain I_i and path cost c_i of region f_i and returns an evaluation result of that candidate region. The region that minimizes U will be selected as the target region to be explored. We propose a novel utility function for balanced the information gain and path cost, which is detailed in Section V.

The goal of the whole exploration process is

$$\begin{aligned} &\min \; H(P) \\ &\text{s.t. } \text{cost}(P) \le C \end{aligned} \tag{4}$$

where P represents the total path generated for exploration, i.e., $P = \{l_1, l_2, \ldots, l_{\text{end}}\}$, the function cost$(\cdot)$ is used for getting the path cost, and C is the budget path cost for the exploration. During the exploration, we aim to reduce the map entropy as much as possible, which means generating a map with high resolution without exceeding the cost budget.

IV. Method Overview

We propose an efficient path planning framework for autonomous exploration in 2-D environments. As indicated in Fig. 1, the `Frontier Extraction` module extracts the frontiers in the map built by the `SLAM` module to provide the information gain. Instead of singly considering information gain in the frontier area, we consider both the information gain and the path cost in our work. The `Goal Point Selection` module takes as inputs both the path cost from the current location of the robot to the goal region and the information gain of that region. The prerequisite for this module is an efficient path planner that provides an exact path cost efficiently. We propose an efficient path planner that incrementally constructs the road map of the environment during the exploration process. The `Path planner` module maintains a graph-structured road map that provides a coarse global path from point to point. Then, the generated global path will be optimized by the trajectory optimizer to generate a smooth trajectory for the robot to execute.

As indicated in Algorithm 1, $CalculateFrontiers(\cdot)$ is responsible for extracting the frontier clusters in the map. In our algorithm, the frontier clusters are grouped into several regions in order to reduce the complexity using

Algorithm 1: Autonomous Exploration Algorithm

Input: sensor information: $laserMsg$
Output: whole map of the environment

1 $map \leftarrow Sensing$;
2 $P_{visual}.empty$;
3 **while** *ExplorationFlag* **do**
4 $P_{clusters} \leftarrow CalculateFrontiers(map)$;
5 $P_{centroid} \leftarrow GroupIntoRegions(P_{clusters})$;
6 $roadMap \leftarrow BuildGraph(laserMsg)$;
7 **if** *target reselection flag is true* **then**
8 $[GoalRegion, path] \leftarrow TargetSelection(P_{centroid}, roadMap)$;
9 **end**
10 **if** $LazyCollisionCheck(path)$ *is failed* **then**
11 $roadMap \leftarrow Prune(path, roadMap)$;
12 find another path on the road map ;
13 go to line 10 ;
14 **else**
15 $path_{opt} \leftarrow Optimizer(path)$;
16 **end**
17 **while** *robot.pose-target* $<$ *distThreshold* **do**
18 execute the robot to reach the target along the generated $path_{opt}$;
19 $map \leftarrow Sensing$;
20 **end**
21 **end**

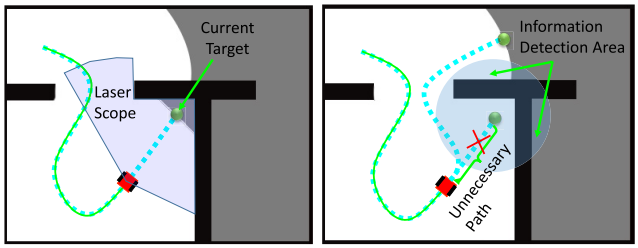

Fig. 2. Change the target at a proper time. (a) Robot is heading toward a chosen target. (b) Ray casting method is employed at the target location to detect the information at the target area.

$GroupIntoRegions(\cdot)$. The function $BuildGraph(\cdot)$ constructs the road map of the environment incrementally with the exploration process. Once the target is found, we can query a path on the road map, as Line 8 shows. We present a target reselection mechanism to reduce the extra path cost. In order to improve the planning efficiency, only if the path is found will it be checked by $LazyCollisionCheck(\cdot)$ function to determine its feasibility. If the path is not feasible, the edges that meet collisions will be deleted on the road map. The robot will find another path on the road map to achieve the exploration. Once the path is found, it will be optimized by the $Optimizer(\cdot)$ to generate an optimal path $Path_{\text{opt}}$for the robot. The robot then moves along this path to the goal until a new target is assigned, as indicated in Lines 17–20.

Algorithm 2: Target Selection

Input: $goals$,Graph $\mathcal{G}$,$robotPose$
Output: goal,path

1 value=0;
2 **for** $goal_c \in goals$ **do**
3 $info \leftarrow Info(goal_c)$;
4 $globalPath \leftarrow FindPath(\mathcal{G}, robotPose, goal_c)$;
5 $path_{opt} \leftarrow Optimizer(globalPath)$;
6 $cost \leftarrow CheckCost(path_{opt})$;
7 $value \leftarrow Evaluate(info, cost)$;
8 **if** $value > value$ **then**
9 $goal \leftarrow goal_c$;
10 $path \leftarrow globalPath$;
11 **end**
12 **end**

V. Proposed Method

A. Target Selection

We use coefficient of variation (CV) c_v to describe the degree of variation of variables, which is defined as follows.

Definition 1 (Coefficient of Variation [36]): Let σ be the standard deviation and μ be the mean value of a sequence of variables, CV is then defined as $c_v = \sigma/\mu$.

c_v is useful in comparing the degree of variation of two variables with different units. In the target selection process, the information gain and the path cost are two input evaluation metrics for the decision maker. The output of the decision maker is the target region to be explored. For these two input decision variables, we define the dominant variable as follows.

Definition 2 (Dominant Variable): For two variables v_1 and v_2 in the decision-making problem, if $c_v(v_1) \gg c_v(v_2)$, variable v_1 is defined as dominate variable.

The dominant variable is the one that imposes a greater impact on the final decision. For example, if the path cost to one target is much smaller than that of the other targets, and the candidate target regions have the similar amount of information gain, we can then choose the path cost as the main variable for the final decision. Formally speaking, the decision variable that has bigger c_v predominates over the other variables in the decision-making process. To encode this idea into our exploration method, we propose a novel utility function to evaluate the candidate goal regions

$$U = c_{v1}\frac{\lambda_1 I_i}{\sum_{i=1}^{N} I_i} - c_{v2}\frac{\lambda_2 c_i}{\sum_{i=1}^{N} c_i} \tag{5}$$

where c_{v1} and c_{v2} are CVs of two corresponding decision variables. Here, we normalize the information gain I and path cost c to evaluate them together. We assume that the two variables are independent, which means that there is no relationship between I and c. We calculate I at target regions using the ray-casting method [32]. Details about how to obtain the information gain are specified in the Appendix. We use our proposed path planning method to get the path cost to a target area. In (5), the information gain I_i and the path cost c_i are normalized to remove their unit. Coefficient c_{vi} will influence the weight of the corresponding decision variable in

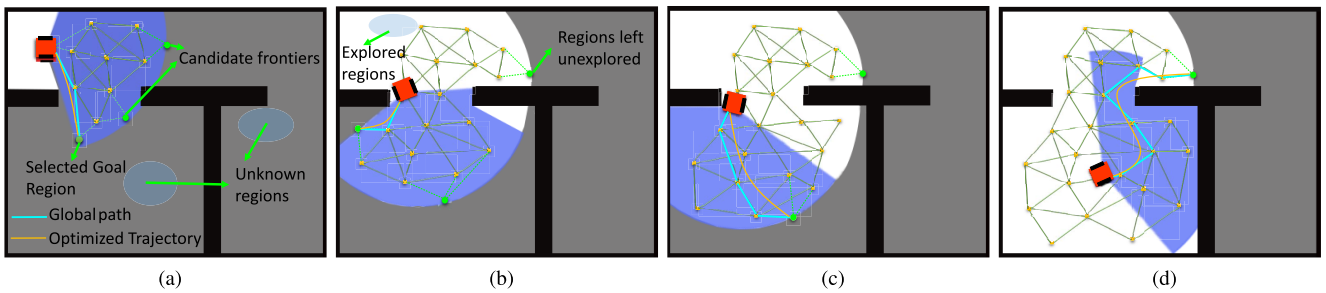

Fig. 3. Conceptual illustration of our proposed path planning framework. (a) Edges of the graph consists of sampling points generated using our proposed sampling method and the centroids of frontier clusters. (b) Graph structure is constructed along with the exploration process. (c) and (d) Robot chooses the most valuable target region to explore.

the utility function. We tune the constants λ_1 and λ_2 manually in different environments.

The objective of target selection is used to find the target that has the minimum value of utility function, as indicated in (3). Algorithm 2 shows the details of goal point selection from the candidate target regions. First, we compute the mutual information of candidate regions and the path cost. We assume that the function $FindPath(\cdot)$ can find a path that is collision free on the road map $\mathcal{G}$. Then, we use the function $Evaluate(\cdot)$ to choose the goal by minimizing the utility function in (5). Algorithm 2 outputs the selected goal region and the global path for reaching there.

B. Target Reselection

Once the goal point is selected, the robot executes a series of control actions to reach the target. Normally, the target reselection is enabled at a fixed frequency. However, if the frequency is high, the robot could not find a long-term goal to execute. If the frequency is low, the robot would go to one selected target area no matter how much information there exists, which will cause extra energy cost.

There is a case that before the robot reaches the target region, that area may already be sensed by the sensor. In our implementation, the frontier regions are swept by the laser range finder. As shown in Fig. 2(a), the target point marked as green point is far away from the robot. Before the vehicle reaches the area, the unclear regions are already swept by the sensor, as shown in Fig. 2(b). Hence, the vehicle does not need to move that much. Based on this observation, we propose a novel method to perform target reselection.

As shown in Fig. 2, when the target area is chosen to explore, we utilize the ray-casting method at a fixed frequency to calculate the information gain at the target area. With this forward simulation method, the robot will determine the amount of information at the target area. Note that when the robot is pretty confident about the state of the gird, i.e., $p(s_k) = 1$ or $p(s_k) = +0$, the information that the grid s_k can provide is zero. We use (1) to describe the information around the target region (see the Appendix for more details). The condition that triggers the target reselection mechanism is: $H(\mathcal{S}_c) \leq Threshold$, where $\mathcal{S}_c$ is the set of grids under a circular region using the ray-casting method. If the total collected information is less than a predefined threshold, which means there is no more information at the target area, the robot does not need to explore the region anymore and will reselect a new target area to explore, as shown Fig. 2(b). By avoiding going to those unnecessary areas, our proposed target reselection mechanism helps to reduce the extra energy and improve the efficiency of the exploration.

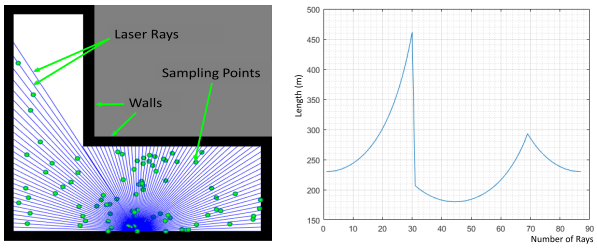

Fig. 4. Proposed sampling strategy. (a) Green points: generated sampling points under the coverage scope of laser sensor. (b) Corresponding function of laser ray length.

C. Incremental Road Map Construction

As Fig. 3 shows, we maintain a road map that is constructed incrementally with the exploration process. The road map is structured as a graph $\mathcal{G} = (V, E)$. The robot can query a path on the generated road map directly. We will introduce how to generate the road map in detail in the following texts.

1) Sampling Points Generation: As Fig. 3(a) shows, when the robot begins the exploration, random points can be generated in the $\mathcal{S}_{\text{free}}$ space. Instead of generating the points randomly in the whole map $\mathcal{M}$ and checking their validity, we generate the required sample points in the free space covered by the laser range finder scope, which is denoted as $\mathcal{S}_{\text{sensorFree}}$. This is known as the sensor-based planning [34]. The points generated in this way do not need the collision checking process; hence, this operation will save a lot of time because the collision checking is relatively a time-consuming part in the process of constructing a graph. The connections between those points are also inclined to be collision free.

In Fig. 4(a), the blue lines represent the laser rays of a lidar. To generate these points, one simple and straightforward strategy is to choose a random point in every ray. However, the generated points are not uniformly distributed in $\mathcal{S}_{\text{sensorFree}}$. In some areas where the laser rays are shorter, the generated sampling points would be dense. Fig. 4(b) reveals the distribution of the length of the laser rays in $\mathcal{S}_{\text{sensorFree}}$.

We, hence, use this distribution to generate the sampling points in $\mathcal{S}_{\text{sensorFree}}$. In some areas that sensor rays are longer, which is corresponding to larger free areas, more sample points are generated. The model we adopted is

$$\frac{1}{\pi R^2} = \frac{N}{\pi r^2} \tag{6}$$

where R represents the maximum length of the length ray, r represents the selected ray, and N represents the number of points generated on the corresponding ray. Hence, the number of points is determined by: $N = r^2/R^2$. The green points in Fig. 4(a) indicate the generated sampling points $V_{rand} = \{v_1, \ldots, v_n\}$. These points are then utilized to construct the road map of the environment. Based on the observation that newly generated cluster centroids are often generated near the explored frontier centroids, we also use the generated frontier cluster centroids V_{centro} to construct the graph. Hence, the vertices V of the graph $\mathcal{G}$ consist the random sampling points V_{rand} and the frontier cluster centroids V_{centro}.

2) Road Map Construction: The road map is constructed incrementally along with the exploration process. Different from the way of constructing the graph in the rapidly exploring random tree (RRT) [8] and the probabilistic road map [37] method, our method is original. In our previous work, we have shown that the efficiency of planning will be improved by making full use of every collision-checking and releasing the restriction of the size of the edge in the graph [38]. In this paper, the generated edges E are connected directly without collision-checking. Only if a path is found, the edges on the path will be checked to determine its feasibility.

When a new point $v' \in V_{\text{new}}$ is generated, the function $N(\mathcal{G}(V, E), v')$ is used to determine the closest vertex v in the existing graph $\mathcal{G}(V, E)$ to v', that is,

$$N(\mathcal{G}(V, E), v') = \arg\min_{v \in V} ||v' - v||. \tag{7}$$

The new edge e is then determined by v and v'. Note that our proposed method is to find a global path first and then optimize the generated path. We can simplify the graph as long as the path between the location of the robot and the target region is found. Hence, in the construction of the graph, instead of connecting all the vertexes that are visible with each other, the generated sampling points will not be connected to other nodes on the graph once it has been added to the tree. We use the function $CheckValidity(e, \mathcal{M})$ to check the feasibility of the edge e. If $CheckValidity$ returns true, the feasible edge will be added to the existing graph. While if the edge e is not collision free, then the generated point v' will continue until a feasible edge is found. In this way, the graph built is more like the tree built by RRT. Different from the tree structure that is used for the single-query purpose, we design the tree as a multiquery structure to facilitate the path-finding.

The whole process of road map construction is introduced in Algorithm 3. Lines 3–6 indicate the process of generating the sampling points in $\mathcal{S}_{\text{sensorFree}}$. The sampling strategy *Random* outputs random points in $\mathcal{S}_{\text{sensorFree}}$. The vertex v_{rand} represents the node generated by our proposed sampling algorithm. Lines 8–19 show the graph construction in which all the generated sampling points and the frontier centroids are added to the existing graph. Line 10 shows the function $rank(\cdot)$ that sorts V in an ascending order according to the distance between $V_{\text{cand}}[i]$ and $v \in V$. $V_{\text{cand}}[i]$ will try to connect to the vertices on the graph from near to far. The function $Line(\cdot)$ generates the edge connecting the nodes $V_{\text{cand}}[i]$ and $V[j]$. As indicated in Lines 13–17, if the edge is collision free, the edge and node are then added to the existing graph. The *break* at Line 16 is for simplifying the graph. In contrast with the traditional method that connects the newly generated point $V_{\text{cand}}[i]$ to all the feasible vertices on the graph, $V_{\text{cand}}[i]$ will not connect to other vertices as long as it has connected to a vertex on the graph. This simplification can facilitate the following path-finding on the road map.

Algorithm 3: Road Map Construction

```
   Input: laserMsg, map M
   Output: road map G
1  V.pushback(start_node) ;
2  //generate candidate vertices ;
3  for every n_th ray of LaserMsg do
4  |  v_rand = Random[0, laserMsg.dist(n_th)] ;
5  |  V_rand.pushback(v_rand) ;
6  end
7  V_cand = {V_rand, V_cebtro} ;
8  //check validity of edges ;
9  for i ∈ I do
10 |  rank(V_cand[i], V, ascend) ;
11 |  for j ∈ J do
12 |  |  edge = Line(V_cand[i], V[j]) ;
13 |  |  if CheckValidity(edge,M) then
14 |  |  |  E.pushback(edge) ;
15 |  |  |  V.pushback(V_cand[i]) ;
16 |  |  |  break ; // to simplify the graph
17 |  |  end
18 |  end
19 end
```

3) Path-Finding on the Road Map: We use A* [39] to search for a global path on the existing graph $\mathcal{G}$. The complexity of the graph search is $\mathcal{O}(|V| + |E|)$. The final global path consists of several edges on the graph. Some generated edges in the graph may not be feasible for constructing the final path. As Fig. 5(a) shows, one candidate edge is constructed. However, the generated edge is demonstrated to be infeasible along with the exploration, as Fig. 5(b) indicates. In our proposed method, we utilize the idea of the lazy collision checking [40], [41]. When a new path is generated, we check the feasibility of the path using function $CheckValidity(e, \mathcal{M})$. If the generated path is not feasible, i.e., one or more edges for constructing the graph are not collision free, we delete those edges on the graph and restart the path-finding process.

D. Trajectory Optimization

Our planning method consists of a global planner and a trajectory optimizer. The generated global plan is just a reference for the robot since it is not optimal. When the global plan is generated, the trajectory optimizer proceeds to optimize the global plan and to make the plan executable for the robot.

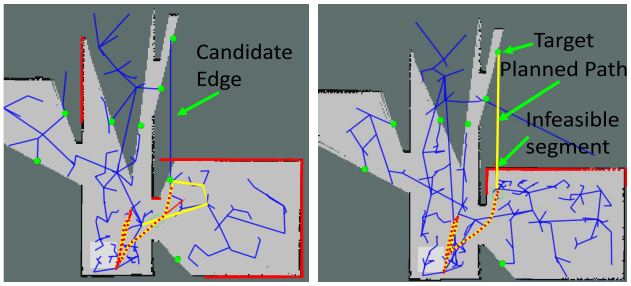

Fig. 5. Lazy collision checking in finding the global path. (a) Edges will be generated in unknown regions. (b) When the final generated path encounters collision, the edge will be deleted and replanning process will be triggered.

The local planner optimizes the global path by using the global path as a heuristic and plans the motion of the robot under the inherent constraints of the robot platform. As shown in Fig. 6(e), we denote the global path σ_G with a series of vertices and $\sigma_G = \{v_0, v_1, \ldots, v_n\}$. Here, the local motion planning problem is to find a trajectory $\sigma : [0, T] \mapsto \mathcal{X}$, coming up with a corresponding series of control inputs $u : [0, T] \mapsto \mathcal{U}$, such that $\sigma(0) = v_0$, $\sigma(T) = v_n$ and $\forall t \in [0, T]$

$$\begin{cases} \sigma(t) \in \mathcal{X}_{\text{free}}, \\ \sigma(t) = f'(\sigma(t-1), u(t)) \\ u(t) \in \mathcal{U}'(u(t-1), \boldsymbol{\varepsilon}) \end{cases} \tag{8}$$

where f' denotes the kinematic functions of the robot and $\mathcal{U}'$ represents the set of the possible controls with respect to the control input $u(t-1)$ of the last node and the inherent control constraints of the robot $\boldsymbol{\varepsilon}$.

In this paper, the differential-driven wheeled robot is utilized as the robot platform. The parameters $\boldsymbol{\varepsilon}$ of the robot utilized here include the maximum linear velocity, the maximum angular velocity, the maximum linear acceleration, the maximum angular acceleration, and the size of the robot, namely, $\boldsymbol{\varepsilon} = \{v_{\text{MAX}}, \omega_{\text{MAX}}, a_{\text{MAX}}, \alpha_{\text{MAX}}, L, W\}$. With the last control input $u(t-1)$ and the constraints of the robot $\boldsymbol{\varepsilon}$, the possible control inputs $u(t)$ are then bounded and all the possible control inputs are recorded in $\mathcal{S}$. Each control input in $\mathcal{U}'$ corresponds to one robot pose at time t, which can be calculated by using the kinematic function f'.

The line-of-sight (LOS) algorithm proposed in [42] is adopted here to do the trajectory optimization, which is used to check whether two nodes on the global path can be connected directly under the nonholonomic constraints. An illustration of the LOS checking algorithm is shown in Fig. 7. If the LOS checking between node P and node Q succeeds without collision, a trajectory $traj_{\text{los}}$ is then generated by the LOS checking algorithm and it is proven to be the optimal trajectory from P to Q under the current constraints of the nonholonomic robot.

On the basis of the LOS checking algorithm, a checking-discrete-checking scheme (CDCS) is proposed to optimize the global trajectory, as shown in Fig. 6. When a global path is obtained, denoted with the blue thick line in Fig. 6, then we start the LOS checking from the current robot state to the nodes on the global path. Fig. 6(a) demonstrates that the LOS

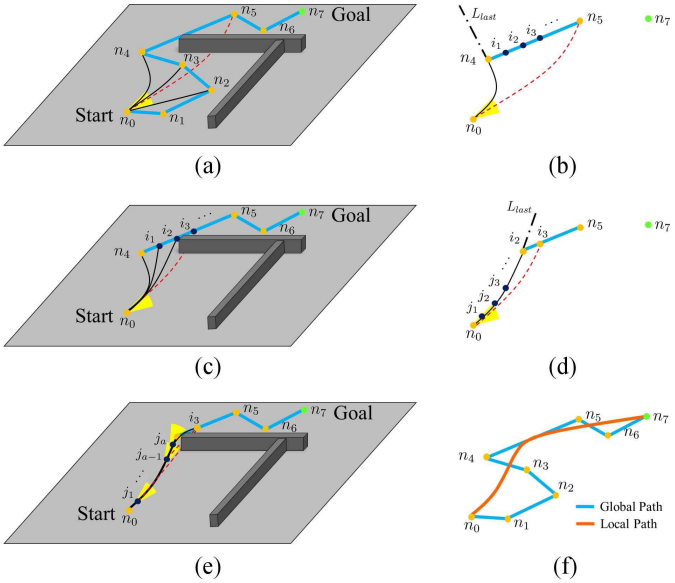

Fig. 6. CDCS trajectory optimization with the global path as a heuristic. (a) Checking: LOS checking from the current robot state to the nodes on the global path. (b) Discrete: discrete the global path from n_4 to n_5. (c) Checking: LOS checking from n_0 to the discretized nodes on the global path. (d) Sampling: sample nodes on the LOS checking trajectory. (e) Checking: LOS checking from the new sampled nodes on the LOS checking trajectory to the node on the discretized nodes on the global path. (f) Global path and the optimized local path.

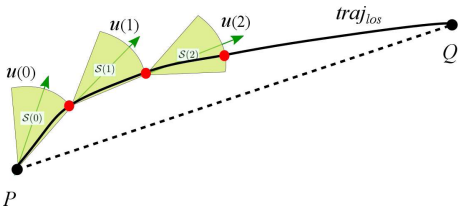

Fig. 7. LOS checking from P to Q under the nonholonomic constraints. Light green arc: possible robot states under the constraints of the last control input u and the inherent nonholonomic constraints.

checking from n_0 (current robot state) to nodes n_2, n_3, and n_4 has already succeeded and the trajectories generated by the LOS checking are denoted with black fine lines. Afterward, the LOS checking from n_0 to n_5 collides with the obstacles and we denote the checking trajectory with a red dashed line.

When encountering collisions, as shown in Fig. 6(b), a judgment is made to make whether the end node of the failure LOS checking, namely, n_5, and the target node, namely, n_7, are on the same side of the last passable LOS checking trajectory L_{last}. As stated in [42], if n_5 and n_7 are not on the same side of L_{last}, the end node of the LOS checking moves to the next node on the global path. If n_5 and n_7 are on the same side of L_{last}, as shown in Fig. 6(b), the global path from the last LOS successful node n_4 to the failure node n_5 is then discretized into a series of adjacent nodes, namely, i_1, i_2, i_3,.... .

Then, the LOS checking from n_0 to the discretized nodes on the global path is further carried out. Fig. 6(c) shows that the LOS checking from n_0 to i_1 and i_2 has succeeded while the LOS checking from n_0 to i_3 collides with obstacles. Since i_2 and i_3 are adjacent nodes, then no discretization will be done

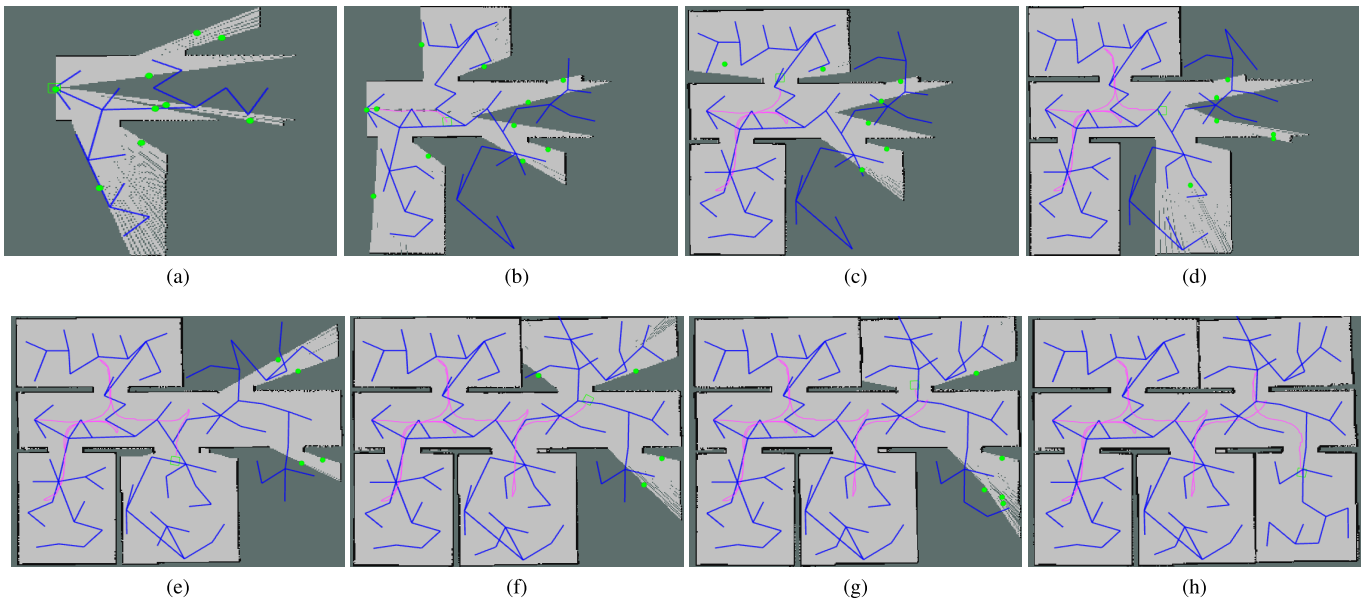

Fig. 8. Autonomous exploration using our proposed framework in typical indoor environments. Green dots: frontier centroids. Blue lines: graph structure built along with exploration. Red line: trajectory of the robot during exploration. (a) $T = 5$ s. (b) $T = 15$ s. (c) $T = 42$ s. (d) $T = 68$ s. (e) $T = 89$ s. (f) $T = 112$ s. (g) $T = 117$ s. (h) $T = 145$ s.

between them and instead we sample nodes on the last passable trajectory, as denoted with j_1, j_2, j_3, ... in Fig. 6(d).

The LOS checking is then conducted from the sampled nodes on the last passable LOS trajectory j_1, j_2, j_3, ... to i_3 until the LOS checking succeeds. Fig. 6(e) shows that the LOS checking from j_a to i_3 succeeds. Then, the start node of the LOS checking moves to j_a. The CDCS process is repeated until the end node of the LOS checking reaches the goal. An inverse search is then applied from the goal to the current robot state and then the local path planning is finished. The optimized path and the global path are shown in Fig. 6(f) and the optimized path is proven to be optimal in the homotopy class of the global path [42]. The robot then executes the optimized path to explore the environment.

VI. Experiments and Results

A. Experiment Setup

We use the turtlebot robot as the mobile robot platform. A Rplidar laser range finder is mounted on the lower base. The laser range finder has 360° field of view (FOV) and adjustable sensor range. The software system is implemented in the robot operation system indigo release on an Ubuntu 14.04 LTS operating system. The simulation environment is built using the Gazebo simulator; the simulated robot is the same with our real robot except that the laser range finder in the simulation has 180° FOV and 30-m sensor range.

B. Simulation Experiments

The autonomous exploration process in a typical indoor simulation environment is shown in Fig. 8. The environment is an office environment that consists of several rooms. Green points in those figures are the centroids of candidate target regions needed to be explored. The robot heads toward to the target regions guided by these green dots. The centroids are added to the road map for further exploration. The blue segments show the road map that is the topological representation of the environment. The pink line represents the trajectory of the robot during exploration. Fig. 8(a)–(h) shows the graph structure constructed along with the exploration. Our proposed graph structure can represent the space topology of the environment appropriately. As Fig. 8 shows, the structure can be extended to every part of the environment. The robot can find a path on the graph structure from its current location to every candidate goal region represented by the green point, which facilitates the process of target evaluation. Our proposed graph structure is more efficient because of the simplification of the whole environment. We can see in Fig. 8 that every green point can be connected to the road map without meeting collision, which means that the path to every candidate goal region can be queried on this graph. Qualitatively speaking, our planning method is more efficient than conventional methods like RRT* and A*. On the one hand, the constructed graph structure simplifies the environment so the robot can query a path on this simplified structure instead of on the whole environment, which is another kind of complex graph when it is represented as the occupancy grid map. On the other hand, our proposed method does not need to replan the path because the path to every goal region has already been computed in the evaluation process.

We tested our proposed method in four different kinds of environment scenarios. As shown in the first row of Fig. 9, the four environment scenarios are different with each other. We compare our proposed method with the nearest frontier method and the random walk method. Both the nearest frontier method and the random walk method use RRT* as the path planner. Fig. 9(a)–(d) shows the performance of different exploration methods corresponding to the environments above. In all the environment scenarios, our proposed method outperforms the other methods in the rate of the

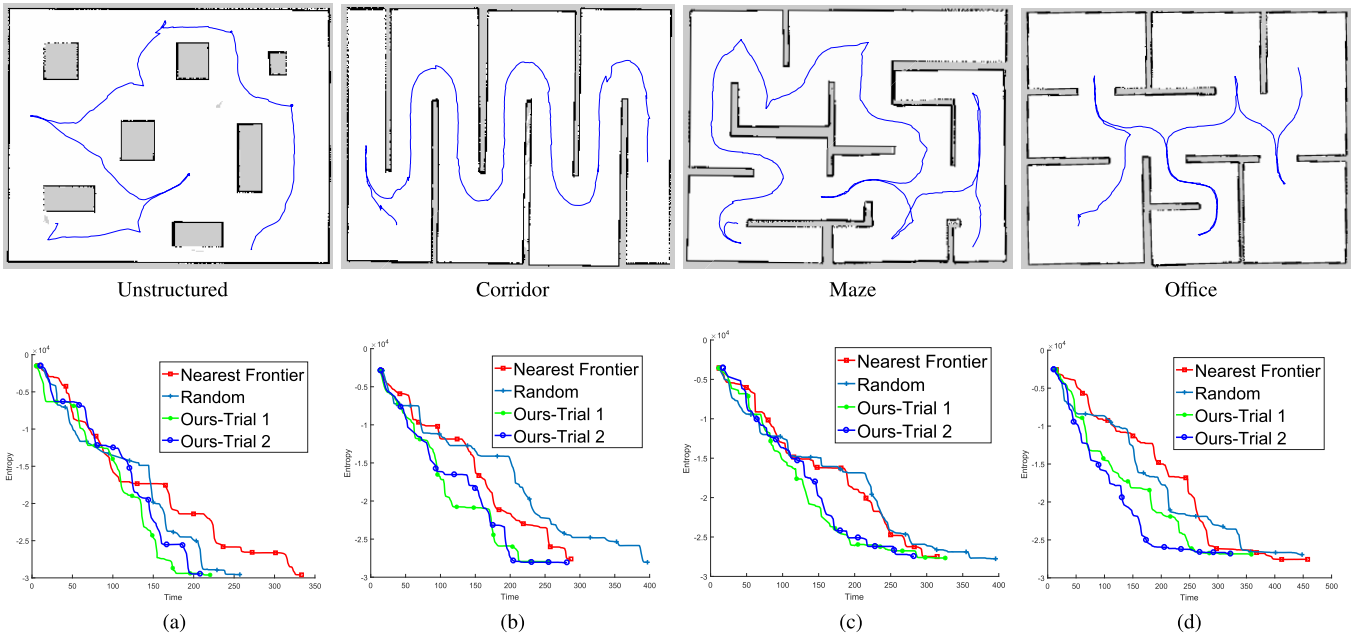

Fig. 9. Map entropy reduction in the experimental studies. The proposed algorithm is tested in four environments. We compare our proposed framework with the nearest frontier method and the random walk method. The blue lines in the first rows show the trajectory during exploration using our proposed method. The charts below show the comparison among our proposed method and other methods in the upper corresponding environments.

entropy reduction. We test our method with different parameters λ_1 and λ_2. Compared with the other methods, our method can reduce the uncertainty of the map with relatively shorter time. As Fig. 9(a) indicates, in the unstructured environment, the performance of our proposed method is similar to the random walk method. While our method is better with different parameter settings, the performance of our proposed method resembles the nearest frontier method in Fig. 9(b). The blue line in the corridor environment map above Fig. 9(b) shows the exploration trajectory of our proposed method. Typically, there is only one unexplored frontier in this special environment during the exploration. Hence, our method is similar to the nearest frontier method, while the efficient path planner and the target reselection mechanism in our proposed method help to reduce the exploration time. As Fig. 9(d) shows, our method is better and more stable than the other two methods in the office environment. For the frontier-based method and the random walk method, their performance is not stable in different environments. The nearest frontier method is better than the random walk in three different kinds of environments. While in the unstructured environment, the random walk method is better than the nearest frontier method. Hence, the performance of those two methods is more environment dependent, which cannot provide reliable performance in different environments.

Fig. 10 shows the statistics of the time and the path costs of different kinds of exploration methods in a typical office environment. Compared with other methods, the random method costs the most energy. The dispersion degree of these two statistics of the random algorithm is also larger than our method, which means the random walk may not be reliable in office environments. The frontier-based method costs less path cost than random walk method and it is more stable in reducing

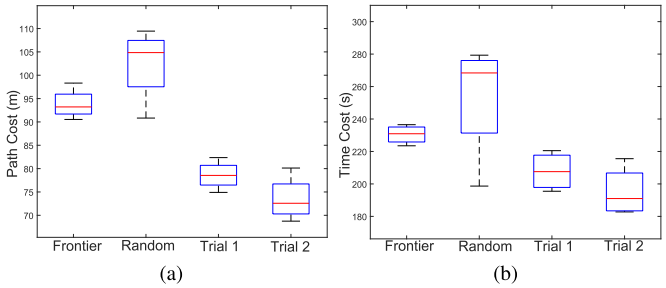

Fig. 10. Office environment exploration with our method, the nearest frontier method, and the random walk method. (a) Exploration path cost. (b) Exploration time. Frontier-based method is better than the random walk method while our proposed method achieves the best performance.

the time cost. In both trials of our methods, our proposed method is superior to the others in reducing the time cost and the path cost. Not only the proposed path planner method helps reduce the planning and evaluation time but also the proposed utility function helps to determine the next best target. The effect of our proposed utility function is shown in Fig. 10. We set different λ in these experiments. For Trial 1, we set $\lambda_1 = 1$ and $\lambda_2 = 2$. For Trial 2, we set $\lambda_1 = 1$ and $\lambda_2 = 1$. As Fig. 10 shows, the performance of Trial 1 is better than that of Trial 2 in five experiments.

We test our proposed exploration method in eight simulated environments with different structures and sizes. In simulated environments, we use simulated Hokuyo laser range finder with 180° FOV. The detection range of the sensor is set according to the environment size. We test our algorithm in four different kinds of environments, including the corridor, the office, the unstructured environment, and the maze environment. We also test our method in the same type of environment with different sizes. As shown in Table I, our

TABLE I
STATISTICS OF OUR ROBOT EXPLORATION METHOD IN SIMULATION AND REAL EXPERIMENTS

Environment	λ_1	λ_2	Map Size / Resolution	Laser Scope & Range	Execution Time	Path Length	Find Target Rate	Nodes	Speed (*average*)
Simulated									
Corridor1	1	1	28m × 15m / 0.05	180°,5m	200 ± 6s	64.24m	2 / 354	2083	0.32m/s
Corridor2	1	1	10m × 10m / 0.02	180°,5m	20 ± 2s	16.65m	2 / 16	2069	0.83m/s
Office1	1	2	25m × 18m / 0.05	180°,5m	300 ± 10s	118.29m	19 / 578	1549	0.39m/s
Office2	1	2	15m × 10m / 0.03	180°,5m	160 ± 6s	55.83 m	15 / 304	1074	0.35m/s
Unstructured1	1	1	28m × 15m / 0.05	180°,5m	400 ± 10s	200.19m	145 / 757	2185	0.50m/s
Unstructured2	1	1	15m × 10m / 0.03	180°,5m	160 ± 10s	61.03m	199 / 522	1403	0.38m/s
Maze1	1	2	28m × 15m / 0.05	180°,5m	280 ± 6s	123.08m	31 / 484	1966	0.44m/s
Maze2	1	2	10m × 10m / 0.02	180°,5m	100 ± 6s	38.71m	25 / 166	1341	0.39m/s
Real-world									
1	3	1	25 m^2 / 0.02	270°,2m	120 ± 20s	15.20m	1/10	110	0.13m/s
2	3	1	120 m^2 / 0.03	270°,6m	200 ± 20s	33.86m	3/23	763	0.17m/s
3	1	1	162.24 m^2 / 0.05	270°,6m	230 ± 28s	20.63m	5/57	697	0.09m/s

proposed method can finish the exploration task in different environment scenarios efficiently. In the corridor environment with the size of 28 × 15 m^2, the robot can finish the task within 206 s with an average speed of 0.32 m/s. In the meanwhile, 2083 nodes of the road map are generated. Comparing with the original map that has over 8000 nodes, our proposed graph simplified the exploration environment. In the same kind of environments with different map sizes, bigger map means more nodes and longer exploration time. We use the Find Target Rate to record the target centroids that failed to connect to the road map, which we cannot evaluate with our proposed graph structure and utility function. We can see the failure rate is very low. In our experiments, the areas that fail to connect to the graph will finally be scanned by the sensor. According to Table I, the Find Target Rate is little higher in the unstructured environment than that of other environments. The reason for this is that in the unstructured environment that contains more scattered objects, the segment that connects the target centroid and the graph is prone to intersect with obstacles. In different scenarios, the more complex the environment, the longer the path length and the higher the time cost.

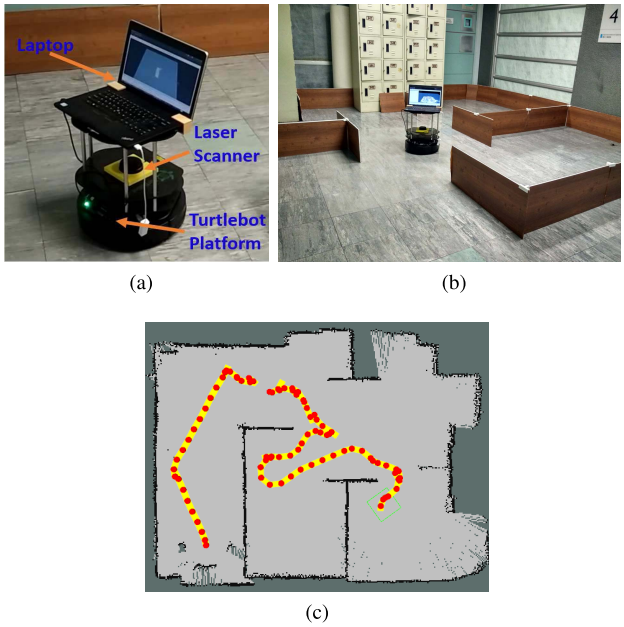

Fig. 11. Experiment in real environment. (a) and (b) Setup of robot and real environment. (c) Corresponding occupancy map. Yellow stripes with red dots indicate the trajectory of robot exploration.

C. Real-World Experiments

The results of our proposed method in real experiments are shown in Fig. 11. Fig. 11(a) and (b) shows the setup of our robot and the structured environment. The environment is set to be an officelike one. As shown in Fig. 11(c), our proposed method can explore the environment with smooth trajectory without walking back and forth. We also test our method in the real environment with different sizes, as shown in Table. I. In the real implementation, the laser range finder with 360° is remolded mechanically to generate the laser range finder with 270°. We increased the map resolution in the small environment for the purpose of generating more frontier grids in the occupancy map. In the real experiment, the average speed for exploration is relatively slower than that in the simulated environments. The Find Target Rate in our real experiments are little larger than that in the office environment of our simulation experiments. This may be caused by that there is much uncertainty in the real environment than the simulated environment.

D. Discussion

We conduct experiments in both simulated and real-world experiments. The qualitative and quantitative analysis in the experimental studies demonstrates the efficiency and efficacy of our framework. However, the limitation is that our graph structure cannot connect to all candidate regions in complex environments. This is because in complex environments, the edges that connect the target regions and the existing graph are sometimes not collision free. Note that many centroids of target candidate regions that fail to connect to the graph structure are generated during the interval of two target selection

process. Hence, these centroids will not be involved the final target selection. However, there still may exist possible failure in more complex and uncertain environments. This could be solved by using other indicators of the frontier clusters to guide the exploration.

VII. Conclusion

In this paper, we propose an efficient framework for autonomous robotic exploration in 2-D unknown environments, which is different from other conventional pipelines in both the planning method and decision-making method. We use frontiers as the candidate target regions to explore. Our method first uses a novel utility function to evaluate target regions, in which the utility function considers both the information gain of the frontier region and the path cost to go there. A target reselection mechanism is presented to reduce the extra exploration effort. We propose a road map that is constructed along with the exploration process. The robot can find a path on this structure efficiently. Hence, we do not need to consider the whole environment every time when we want to find a path from the current location of the robot to the target region. The generated global path is then optimized with our proposed trajectory optimization method. Both simulated and real experimental studies show that our proposed method can build a precise map of the environment autonomously. Compared with other methods, our method cost less path cost.

As we can see from the experiments, there exist some target points that cannot be connected to the graph structure. We will further improve our method in the future. Despite the frontier centroid, other indicators of the frontier, such as the nearest vertex on the existing graph to the frontier cluster, will be tested for guiding the exploration.

Appendix

We calculate the information gain $I(\mathcal{S};\, f_i)$ at target regions using the ray-casting method. $\mathcal{S}_r$ is the set of grids within a circle centered at the frontier area f_i. Since we only consider the information gain at target regions, the path l_i is reduced to the target point f_i. Let $\mathcal{S}_o = \mathcal{S} \setminus \mathcal{S}_r$, then (2) becomes

$$\begin{aligned} I(\mathcal{S};\, f_i) &= H(\mathcal{S}) - H(\mathcal{S}|f_i) \\ &= H(\mathcal{S}_r \cup \mathcal{S}_o) - H(\mathcal{S}_r \cup \mathcal{S}_o|f_i) \\ &= H(\mathcal{S}_r) + H(\mathcal{S}_o) - H(\mathcal{S}_r|f_i) - H(\mathcal{S}_o|f_i). \quad (9) \end{aligned}$$

The operation of ray-casting will not affect the value of the grids in $\mathcal{S}_o$; hence, $H(\mathcal{S}_o|f_i) = H(\mathcal{S}_o)$. In this forward simulation model, we assume the sensor is accurate and can obtain all the information of a grid exactly without any noise. In other words, performing ray-casting around f_i will result in $p'(s_k) \simeq 1$ or $p'(s_k) \simeq +0$, $\forall s_k \in \mathcal{S}_r$, where $p'(s_k)$ is the occupancy probability of grid s_k after performing ray-casting. Then

$$H(\mathcal{S}_r|f_i) = -\sum_{k=1}^{R} p'(s_k)\log(p'(s_k)) = 0 \quad (10)$$

where R is the number of grid $s_k \in \mathcal{S}_r$. Hence, we get

$$\begin{aligned} I(\mathcal{S};\, f_i) &= H(\mathcal{S}_r) \\ &= -\sum_{k=1}^{R} p(s_k)\log(p(s_k)). \quad (11) \end{aligned}$$

Then, we get the information gain at the target area f_i.

References

[1] J. Delmerico, E. Mueggler, J. Nitsch, and D. Scaramuzza, "Active autonomous aerial exploration for ground robot path planning," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 664–671, Apr. 2017.

[2] G. A. Hollinger and G. S. Sukhatme, "Sampling-based motion planning for robotic information gathering," *Robot., Sci. Syst.*, vol. 3, no. 5, pp. 51–58, 2013.

[3] M. P. Christiansen, M. S. Laursen, R. N. Jørgensen, S. Skovsen, and R. Gislum, "Designing and testing a uav mapping system for agricultural field surveying," *Sensors*, vol. 17, no. 12, p. 2703, 2017.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. Comput. Intell. Robot. Automat.*, 1997, pp. 146–151.

[5] G. Costante, J. Delmerico, M. Werlberger, P. Valigi, and D. Scaramuzza, "Exploiting photometric information for planning under uncertainty," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 107–124.

[6] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-blackwellized particle filters," *Robot. Sci. Syst.*, vol. 2, pp. 65–72, Jun. 2005.

[7] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. Robot. IEEE Int. Conf. Automat. (ICRA)*, May 1994, pp. 3310–3317.

[8] S. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Comput. Sci. Dept., Iowa State Univ., Ames, IA, USA, Res. Rep. 9811, 1998.

[9] G. Hollinger, B. Englot, F. Hover, U. Mitra, and G. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *Int. J. Robot. Res.*, vol. 32, no. 1, pp. 3–18, Nov. 2013.

[10] P. A. Plonski, J. Vander Hook, C. Peng, N. Noori, and V. Isler, "Environment exploration in sensing automation for habitat monitoring," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 25–38, Jan. 2017.

[11] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 5031–5037.

[12] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 986–1002, Aug. 2013.

[13] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[14] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable RGB-D SLAM in dynamic environments," *Robot. Auto. Syst.*, vol. 108, pp. 115–128, Oct. 2018.

[15] A. Kim and R. M. Eustice, "Active visual SLAM for robotic area coverage: Theory and experiment," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 457–475, 2015.

[16] P. Quin, G. Paul, and D. Liu, "Experimental evaluation of nearest neighbor exploration approach in field environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 869–880, Apr. 2017.

[17] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *Proc. IROS*, 2002, pp. 534–539.

[18] M. Lauri and R. Ritala, "Planning for robotic exploration based on forward simulation," *Robot. Auto. Syst.*, vol. 83, pp. 15–31, Sep. 2016.

[19] S. D. Roth, "Ray casting for modeling solids," *Comput. Graph. Image Process.*, vol. 18, no. 2, pp. 109–144, 1982.

[20] B. Charrow *et al.*, "Information-theoretic planning with trajectory optimization for dense 3D mapping," *Robot., Sci. Syst.*, pp. 3–12, Jul. 2015.

[21] C. Dornhege and A. Kleiner, "A frontier-void-based approach for autonomous exploration in 3D," *Adv. Robot.*, vol. 27, no. 6, pp. 459–468, 2013.

[22] S. Bai, J. Wang, K. Doherty, and B. Englot, "Inference-enabled information-theoretic exploration of continuous action spaces," in *Robotics Research*. Cham, Switzerland: Springer, 2018, pp. 419–433.

[23] L. von Stumberg, V. Usenko, J. Engel, J. Stückler, and D. Cremers, "From monocular slam to autonomous drone exploration," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2017, pp. 1–8.

[24] I. Shnaps and E. Rimon, "Online coverage of planar environments by a battery powered autonomous mobile robot," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 425–436, Apr. 2016.
[25] H. H. González-Baños and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, nos. 10–11, pp. 829–848, 2002.
[26] L. Heng, A. Gotovos, A. Krause, and M. Pollefeys, "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1071–1078.
[27] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3477–3484.
[28] J. Velez, G. Hemann, A. S. Huang, I. Posner, and N. Roy, "Planning to perceive: Exploiting mobility for robust object detection," in *Proc. ICAPS*, 2011, pp. 266–273.
[29] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon 'next-best-view' planner for 3D exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1462–1468.
[30] A. Wallar, E. Plaku, and D. A. Sofge, "Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 969–980, Jul. 2015.
[31] R. Shade and P. Newman, "Choosing where to go: Complete 3D exploration with stereo," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2011, pp. 2806–2811.
[32] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous Robot Vehicles*. New York, NY, USA: Springer, 1986, pp. 396–404.
[33] C. Wang, L. Meng, T. Li, C. W. De Silva, and M. Q.-H. Meng, "Towards autonomous exploration with information potential field in 3D environments," in *Proc. 18th Int. Conf. Adv. Robot. (ICAR)*, Jul. 2017, pp. 340–345.
[34] G. Oriolo, M. Vendittelli, L. Freda, and G. Troso, "The srt method: Randomized strategies for exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 5, Apr. 2004, pp. 4688–4694.
[35] C. Stachniss, *Robotic Mapping and Exploration*. Berlin, Germany: Springer, 2009, vol. 55.
[36] B. Everitt and A. Skrondal, *The Cambridge Dictionary of Statistics*, vol. 106. Cambridge, U.K.: Cambridge Univ. Press, 2002.
[37] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
[38] C. Wang and M. Q.-H. Meng, "Variant step size RRT: An efficient path planner for UAV in complex environments," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Jun. 2016, pp. 555–560.
[39] S. Bandi and D. Thalmann, "Path finding for human motion in virtual environments," *Comput. Geometry*, vol. 15, nos. 1–3, pp. 103–127, 2000.
[40] G. Sánchez and J.-C. Latombe, "On delaying collision checking in PRM planning: Application to multi-robot coordination," *Int. J. Robot. Res.*, vol. 21, no. 1, pp. 5–26, 2002.
[41] K. Hauser, "Lazy collision checking in asymptotically-optimal motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 2951–2957.
[42] W. Chi, C. Wang, J. Wang, and M. Q.-H. Meng, "Risk-DTRRT-based optimal motion planning algorithm for mobile robots," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–18, Nov. 2018.

Chaoqun Wang (S'14) received the B.E. degree in automation from Shandong University, Jinan, China, in 2014. He is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong.

He was a Visiting Scholar with The University of British Columbia, Vancouver, BC, Canada, for six months. His current research interests include autonomous exploration, active perception, and path planning.

Wenzheng Chi (M'17) received the B.E. degree in automation from Shandong University, Jinan, China, in 2013, and the Ph.D. degree in biomedical engineering from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, in 2017.

She was a Visiting Scholar with The University of Tokyo, Tokyo, Japan, for six months. From 2017 to 2018, she was a Post-Doctoral Fellow with the Department of Electronic Engineering, The Chinese University of Hong Kong. She is currently an Associate Professor with the Robotics and Microsystems Center, School of Mechanical and Electric Engineering, Soochow University, Suzhou, China. Her research interests include human recognition, human–robot interaction, and human-friendly motion planning for mobile service robots.

Yuxiang Sun (S'15) received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, in 2017.

He is currently a Research Associate with the Robotics Institute, Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. His current research interests include mobile robots, autonomous vehicles, simultaneous localization and mapping and navigation, motion detection, deep learning, and so on.

Dr. Sun was a recipient of the Best Student Paper Finalist Award at the IEEE ROBIO 2015.

Max Q.-H. Meng (M'72–F'00) received the Ph.D. degree in electrical and computer engineering from the University of Victoria, Victoria, BC, Canada, in 1992.

He joined The Chinese University of Hong Kong, Hong Kong, in 2001, where he is currently a Professor and the Chairman of the Department of Electronic Engineering. He was with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada, where he was the Director of the Advanced Robotics and Teleoperation Laboratory, an Assistant Professor in 1994, an Associate Professor in 1998, and a Professor in 2000. He is currently with the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology. He is the Honorary Dean of the School of Control Science and Engineering, Shandong University, Jinan, China. He has published 600 journal and conference papers and led more than 50 funded research projects to completion as PI. His research interests include robotics, medical robotics and devices, perception, and scenario intelligence.

Dr. Meng is a fellow of the Canadian Academy of Engineering and HKIE. He is an elected member of the Administrative Committee (AdCom) of the IEEE Robotics and Automation Society. He was a recipient of the IEEE Millennium Medal. He has served as the General and Program Chair of many conferences including the General Chair of IROS 2005 and the General Chair of ICRA 2021 to be held in Xi'an, China. He has served as an editor for several journals.