



Improving Autonomous Exploration Using Reduced Approximated Generalized Voronoi Graphs

Lin Li^{1,2} · Xinkai Zuo¹ · Huixiang Peng³ · Fan Yang¹ · Haihong Zhu¹ · Dalin Li^{1,4} · Jun Liu¹ · Fei Su¹ · Yifan Liang¹ · Gang Zhou¹

Received: 8 August 2019 / Accepted: 12 November 2019

© Springer Nature B.V. 2020

Abstract

Autonomous robotic exploration has been extensively applied in many tasks, such as mobile mapping and indoor searching. One of the most challenging issues is to locate the Next-Best-View and to guide robots through a previously unknown environment. Existing methods based on generalized Voronoi graphs (GVGs) have presented feasible solutions but require excessive computation to construct GVGs from metric maps, and the GVGs are usually redundant. This paper proposes an improving method based on reduced approximated GVG (RAGVG), which provides a topological representation of the explored space with a smaller graph. Additionally, a fast and robust image thinning algorithm for constructing RAGVGs from metric maps is presented, and an autonomous robotic exploration framework using RAGVGs is designed. The proposed method is validated with three known common data sets and two simulations of autonomous exploration tasks. The experimental results show that the proposed algorithm is efficient in constructing RAGVGs, and the simulations indicate that the mobile robot controlled by the RAGVG-based exploration method reduced the total time by approximately 20% for the given tasks.

Keywords Autonomous robotic exploration · Image thinning algorithm-reduced approximated GVG

1 Introduction

One of the most challenging issues in mobile robotics is the ability to autonomously explore previously unknown spaces [1]. Because of the complexity and structure of an interior space, mobile robotics can be widely used in applications related to building inspections, such as indoor surveying and

mapping and urban searching and rescuing. An exploration strategy is employed to guide mobile robots to traverse in an unknown space and discover unknown features using information collected by various sensors [2]. With the help of an exploration strategy, mobile robots can autonomously decide where to go and build an environment map of a previously unknown space to accomplish a given task, such as map

✉ Haihong Zhu
hhzhu@whu.edu.cn

Lin Li
lilin@whu.edu.cn

Xinkai Zuo
zuoxinkai2012@whu.edu.cn

Huixiang Peng
18603318694@163.com

Fan Yang
yhlx125@whu.edu.cn

Dalin Li
li.dl@cidi.ai

Jun Liu
2011301130092@whu.edu.cn

Fei Su
sftx016@whu.edu.cn

Yifan Liang
lyf0312@whu.edu.cn

Gang Zhou
2014301130059@whu.edu.cn

¹ School of Resource and Environmental Sciences, Wuhan University, 129 Luoyu Rd., Wuhan, China

² Collaborative Innovation Centre of Geospatial Technology, Wuhan University, 129 Luoyu Road, Wuhan 430079, China

³ The 54th Research Institute of CETC, 589 Zhongshan West Road, Shijiazhuang 050000, China

⁴ Changsha Intelligent Driving Institute Co., Ltd, 336 Xueshi Road, Changsha 410006, China

building [3], search and rescue [4, 5], and 3D model building [6, 7]. Hence, a good exploration strategy will enable a mobile robot to cover a space completely in an acceptable amount of time [8] and to detect loop-closure to improve the quality of map models [2].

According to existing research, the main difficulties in autonomous robotics exploration can be reduced to four topics: (I) how to model the environment, (II) how to decide where the robot should go next, (III) how to find global paths between any two positions in the explored space, and (IV) how to control the robot to trace the given path and to avoid obstacles in real time. Fortunately, many clever methods have been proposed and successfully implemented in simulators and in real robots, and the most common solutions are based on greedily visiting the Next-Best-View (NBV) framework [3–7, 9–18], as depicted in Fig. 1.

Backend: The Simultaneous Localization and Mapping (SLAM) algorithms provide a backend module to model the environment and estimate the pose of the robot using the observations from a laser range finder in real time;

Step 1: A number of candidate points (CPs) are generated from the latest environment map;

Step 2: The NBV is selected by evaluating the CPs according to some features, and exploration will stop when no qualified NBV can be found;

Step 3: A global path from the current location of the robot to the location of the NBV is generated by the path-planning algorithm;

Step 4: The robot moves to the NBV by tracing the global path and avoiding obstacles;

Step 5: Go back to step 1 when the robot reaches the NBV, or exploration will stop if some limits (e.g., time or coverage area) are reached.

The well-known frontier-based methods [4, 5, 10–18] are a class of typical solutions using the NBV framework. CPs are generated from the frontiers that locate at the boundary of

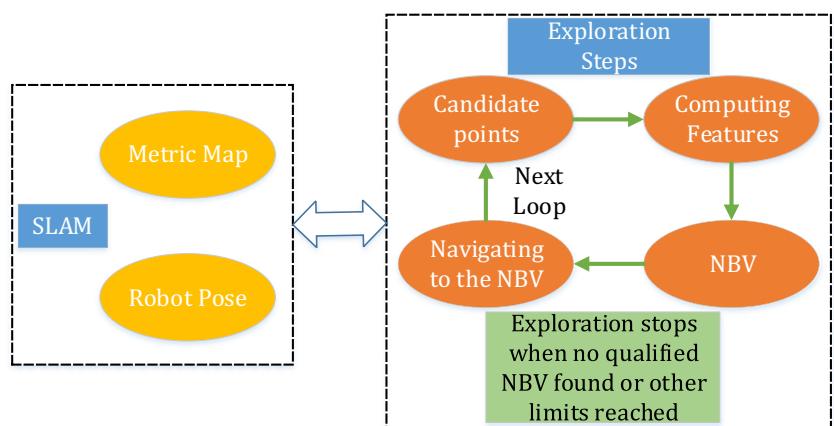
explored and unexplored space according to the latest environment map. Nevertheless, due to the complexity of metric maps, frontier-based methods suffer from low efficiency in evaluating CPs [19] and planning a global path [20]. To avoid the complexity of metric maps, topological maps are employed to represent the environment. The generalized Voronoi graph (GVG) is a kind of topological map, and it performs well as a basis for sensor-based path planning in an unknown static environment [21]. However, the task remains difficult because the existing algorithms for constructing GVGs from metric maps are usually complex and unstable [21–26].

1.1 Paper Contributions

In this paper, we are interested in exploration strategies for discovering the physical structure of environments that are previously unknown, which means we do not have any prior knowledge about the environment. The study is based on the NBV framework and focuses on topic II, which involves fast generating the CPs (step 1) and selecting the NBV (step 2) as mentioned above. Aiming at improving the efficiency and robustness of sensor-based exploration, we propose a straightforward and stable method for constructing a reduced approximated GVG (RAGVG) from an occupancy grid map (OGM) to represent the indoor environments and design an autonomous robotic exploration system using RAGVGs to accelerate the decision-making process.

To distinguish our approach from the existing methods for constructing GVG, the RAGVGs constructed by the proposed method are guaranteed to be exhaustive, non-redundant and non-interrupted. Therefore, the improvements attained are that (I) the resulting RAGVGs can be directly regarded as the topological maps of the explored environment, without any outlier elimination or other post-processing steps, and that (II) the RAGVGs constructed by the proposed method can cover almost every corner of the explored spaces with relatively small graphs. In addition, to accelerate the proposed

Fig. 1 The common solution based on Next-Best-View framework



skeletonization algorithm, a parallel version of the proposed skeletonization algorithm is presented.

With RAGVGs, the autonomous exploration system presented in this paper transforms the problem representation from a Euclidean metric map $\mathcal{M} \subset \mathbb{R}^2$ into a topological graph space $\mathcal{G} = \{E, V\}$. This system is expected to make improvements that (III) CPs generated from an RAGVG are distinctly small in size, which leads to much less time consumption for decision making. In addition, (IV) graph-based path planning is extremely fast, and the global paths generated from RAGVGs are guaranteed to be collision-free. In summary, the proposed method for the autonomous exploration of mobile robotics improves the efficiency of selecting the NBV and planning a global path so that the total time consumption of autonomous exploration is reduced. Note that our research is for single robot exploration; however, the proposed method can be expanded to multi-robot collaboration conditions.

1.2 Paper Structure

The remaining content of this paper is structured as follows. Section 2 reviews previous research and existing methods regarding autonomous exploration for mobile robotics and construction of GVGs. Section 3 provides a detailed explanation of RAGVGs and the proposed algorithms for constructing RAGVGs from OGMs and its parallel version. Section 4 introduces the framework and the other necessary modules employed in our autonomous exploration system. Section 5 presents the experimental design and discusses the results of the simulation of autonomous exploration aimed at a task for capturing indoor point clouds and a task for search and rescue, what's more, the results of capturing indoor points clouds in a real-world museum is presented. In Section 6, we draw some conclusions about our research and discuss the prospects of autonomous robotic exploration.

2 Related Works

In this section, we provide an overview of several existing methods for autonomous robotic exploration. The literature on this topic is diverse, dating back more than 30 years. Using Simultaneous Localization and Mapping (SLAM) [8] as a fundamental module to obtain environment map models, the mainstream approach regards exploration as an incremental NBV process, i.e., a repeated greedy selection of the next best observation location. We focus on recent approaches based on frontiers and GVGs, respectively, since these have been the most successful and are most closely related to the proposed method. In addition, the common algorithms for constructing GVGs are also discussed for the purpose of presenting one of the main contributions of this paper.

2.1 Frontier-Based Exploration

The frontier-based methods involve extracting frontiers between free space and unexplored space, and the NBV is selected from the CPs that are located on the centroids of the frontiers [10–12]. The main challenge of the frontier-based method is how to select the NBV from the CPs. In the earliest versions [11, 12], a global utility function that linearly combines the distance from the robot location and the estimation of information gain was proposed to evaluate each CP. Stachniss et al. [13] presented a decision-theoretic framework that considers the uncertainty and expected information gain to evaluate the cost of executing an action. Then, improved methods were proposed in the next few years. Basilico and Amigoni [5] extended the global utility function using a fuzzy measure approach that linearly combined several criteria (e.g., distance from the robot location, potential information gain and battery). Vallvé and Andrade-Cetto [14] proposed a novel method that evaluates the reduction in joint path and map entropy and computes a potential information field in the explored space using joint entropy reduction estimates. Carrillo and colleagues [15, 16] designed a utility function based on Shannon and Rényi entropy theory according to the estimation of potential information gain.

The frontier-based methods enjoy the advantage of easy implementation but suffer from low efficiency in evaluating CPs and low extensibility to various applications, especially in a large-scale space [20]. First, frontier-based methods usually generate a great number of CPs from many frontiers that locate at the boundaries between the explored area and unexplored area and are easily affected by sensor errors and some empirical parameters. As a result, most CPs are too meaningless and redundant so that a substantial amount of time is wasted on evaluating them [5]. Second, to provide an important feature for evaluating CPs, all the global paths from the current location of the robot to each CP must be found. The heuristic algorithms (such as A*) can be helpful to find a path, but they are too inefficient to satisfy real-time requirements and cannot guarantee success when there are many scattered pixel blocks in the OGM due to environments that are cluttered with a variety of obstacles [27, 28].

2.2 GVG-Based Navigation and Exploration

In recent years, topological graphs have been the focus of robotic navigation studies. In particular, a few studies focus on developing navigation algorithms and autonomous robotics exploration frameworks on the basis of generalized Voronoi diagrams (GVDs) and GVGs. A GVD is a type of skeleton of a Euclidean metric map, and it provides a holistic structural representation of a given environment, such that a GVG is the topological graph constructed from GVD. Takahashi and Schilling [29] presented a novel method for

mobile robotics motion planning in a plane using GVD. Valero-Gomez et al. [28] provided a comprehensive view of the Fast Marching algorithm and presented an efficient method for planning safer mobile robot trajectories using GVD. Tsardoulias et al. [27] demonstrated that the path planning algorithms based on the GVD are faster and have a higher success rate, and the resulting paths are guaranteed to be collision-free. Choset and his team [21, 23], by extending the GVD from one dimension to multi-dimension, proposed the GVG and the hierarchical GVG (HGVG) and proved them sufficient for motion planning.

In regard to autonomous robotics exploration in previously unknown environments, Nagatani and Choset [22] proposed an algorithm to reduce the unnecessary edges and nodes of GVGs and presented an exploration method using a reduced GVGs (RGVGs), which is a subset of the GVG and sufficient for motion planning. Kim and Zhang [24] presented a provably complete exploration strategy by constructing GVD and confirmed the algorithms in an unknown environment of a simple structure. Tsardoulias et al. [19] compared three methods of target selection for full exploration of an unknown space based on approximated GVGs (AGVGs), and the experiments showed that the AGVGs can perform well in autonomous robotic exploration.

The most crucial step for constructing a GVG from an OGM is extracting GVD by means of image thinning algorithms, e.g., the brushfire algorithm [30], the Zhang-Suen algorithm [31] and morphology-based algorithms [32, 33]. Saeed et al. [33] reviewed a number of mainstream image thinning algorithms and compared them by extracting the skeleton of some handwriting images. Lau and colleagues [25] presented a dynamic variant of the brushfire algorithm for updating OGMs and extracting a GVD. Tsardoulias et al. [26] proposed a valid method, based on brushfire algorithm, for producing minimized topological graphs by creating approximated GVDs and eliminating redundant nodes and edges in the graphs.

The GVG-based autonomous exploration methods have the advantages of high efficiency in decision making and better extensibility due to the topological representation of the explored space. Nevertheless, the complex steps for constructing GVG can be simplified and improved. The main problem of the GVG-based methods is that the most of existing algorithms for constructing GVGs from OGMs are usually complex and unstable [22, 25, 26]. Because the GVDs extracted by existing image thinning algorithms are often cracked, redundant and disordered, some post-processing steps have to be conducted to eliminate unnecessary nodes and edges [22, 26], which potentially leads to information loss. In addition, the autonomous exploration system requires relatively efficient algorithms to construct GVGs to satisfy real-time decision making. Thus, a fast and stable method for extracting qualified GVDs from OGMs is needed.

3 Construction of Reduced Approximated GVGs

As illustrated in Fig. 2, the proposed method for constructing RAGVGs from OGMs contains (1) a few image pre-processing steps for extracting a smooth free area map, (2) the proposed corner rounding method and image thinning algorithm for generating an RAGVD and (3) a flood-fill algorithm for constructing the topological graph. Among these, the main improvement is the proposed corner rounding method and image thinning algorithm, since this combination can directly generate RAGVDs without any post-processing steps.

3.1 Pre-Processing

Due to the uncertainties from the sensors and the SLAM algorithm, the OGM is a map of probabilities, so that it cannot be directly used for indoor exploration and navigation and a map with more explicit information is required. The smooth free area map (SFAM) represents the absolutely passable regions for mobile robots in the explored environment, e.g., at least 30 cm away from obstacles. Thus, the SFAM should be built from the OGM.

To build SFAM, a series of pre-processing steps, mainly including threshold operations, morphology operations, smooth filtering and connected component analyses, are conducted on the OGM. An example of extracting the SFAM from an OGM is shown in Fig. 3. First, a grey value threshold $x \geq t_1$ is applied on the OGM; then, a morphological closing operation is employed to fill the possible small gaps, and the small outliers are removed through a connected component analysis; finally, a buffer of the obstacle area, which is extracted by a grey value threshold $x \leq t_2$, is subtracted.

3.2 Eliminating the Weak Edges

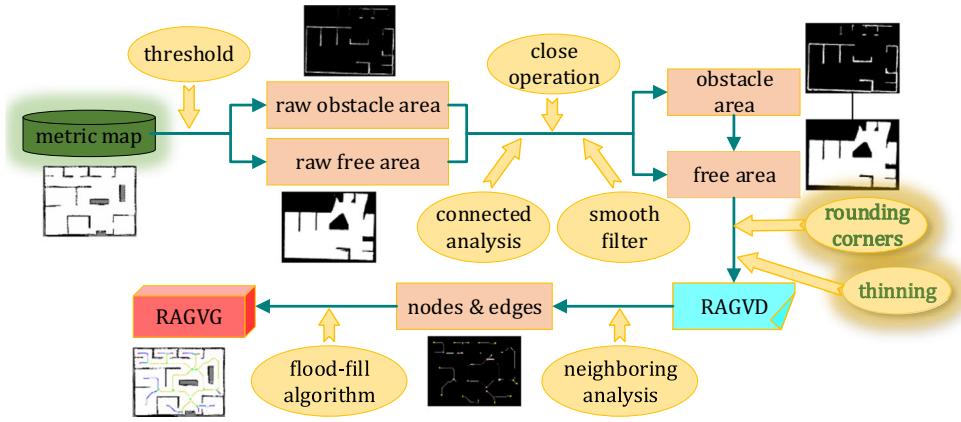
3.2.1 Preliminary

Choset et al. [21] defined the GVD and the GVG by means of a distance function $d_i(x)$ and a distance gradient $\nabla d_i(x)$. Let two obstacles in the metric map $\mathcal{M} \subset \mathbb{R}^2$ be point sets C_i and C_j , and the Two-Equidistant Face \mathcal{F}_{ij} and the Two-Voronoi Set \mathcal{F}^2 can be termed as Eqs. (1) and (2), respectively.

$$\mathcal{F}_{ij} = \{x \in \mathbb{R}^2 | d_i(x) = d_j(x), \nabla d_i(x) \neq \nabla d_j(x), \forall k \neq i, j : d_i(x) \neq d_k(x)\}, \quad (1)$$

where $d_i(x)$ is the minimum distances among point x and all points in obstacle set C_i , and gradient $\nabla d_i(x)$ is a unit vector in the direction from point c_0 to x , where c_0 is the nearest point to x in C_i .

Fig. 2 Workflow of constructing RAGVGs



$$\mathcal{F}^2 = \bigcup_{i=1}^{n-1} \bigcup_{j=i+1}^n \mathcal{F}_{ij} \quad (2)$$

As is shown in Fig. 4, \mathcal{F}_{ij} is the set of points equidistant to C_i and C_j , and \mathcal{F}^2 is the set of points equidistant to two or more obstacles. Hence, \mathcal{F}^2 can be regarded as the GVD of the 2D space that consists of obstacles set $\{C_i\}$.

The GVG is a topological graph constructed from the GVD. To define the GVG, the Three-Equidistant Face \mathcal{F}_{ijk} and Three Voronoi Set \mathcal{F}^3 is defined as Eqs. (3) and (4), respectively. In Fig. 4, \mathcal{F}_{ijk} is the point equidistant to C_i , C_j and C_k , which is also the joint of \mathcal{F}_{ij} , \mathcal{F}_{jk} and \mathcal{F}_{ik} ; then, \mathcal{F}^3 is the set of points equidistant to three or more obstacles. Hence, \mathcal{F}^3 can be regarded as the joints of the GVD.

$$\mathcal{F}_{ijk} = \mathcal{F}_{ij} \cap \mathcal{F}_{ik} \cap \mathcal{F}_{jk} = \mathcal{F}_{ij} \cap \mathcal{F}_{ik}. \quad (3)$$

$$\mathcal{F}^3 = \bigcup_{i=1}^{n-2} \bigcup_{j=i+1}^{n-1} \bigcup_{k=j+1}^n \mathcal{F}_{ijk}. \quad (4)$$

With these definitions above, in a 2D space, the GVG can be defined as Eq. (5), where \mathcal{F}^2 denotes the set of Generalized Voronoi Edges and \mathcal{F}^3 represents the set of Generalized Voronoi Vertices. Having edges and vertexes, the topological

graph GVG can be constructed after computing its connectivity matrix and distance matrix.

$$\text{GVG} = \{\mathcal{F}^2, \mathcal{F}^3\}. \quad (5)$$

However, because of the irregularities of the OGM and the capability for preserving details of most of the existing image thinning algorithms, the GVGs are usually redundant [22] because weak edges exist at some concave walls (e.g., corners), as is depicted in Fig. 5a. These weak edges are unnecessary for robotic exploration and navigation, and they increase the size of the topological graph with information of low value. The RAGVG proposed in this paper is a topological graph constructed from a reduced approximation of GVD, as depicted in Fig. 5b. Without any weak edges, the RAGVGs consist of pivotal nodes and edges that are of higher value. Moreover, it is confirmed in our research that the RAGVGs can also preserve almost the same connectivity and coverage information as the original GVGs for robotic navigation tasks.

3.2.2 Corner Rounding Method

A few studies [22, 26] noticed these redundant weak edges and designed some post-processing methods to eliminate weak edges from original GVGs constructed by the brushfire algorithm. However, the brushfire

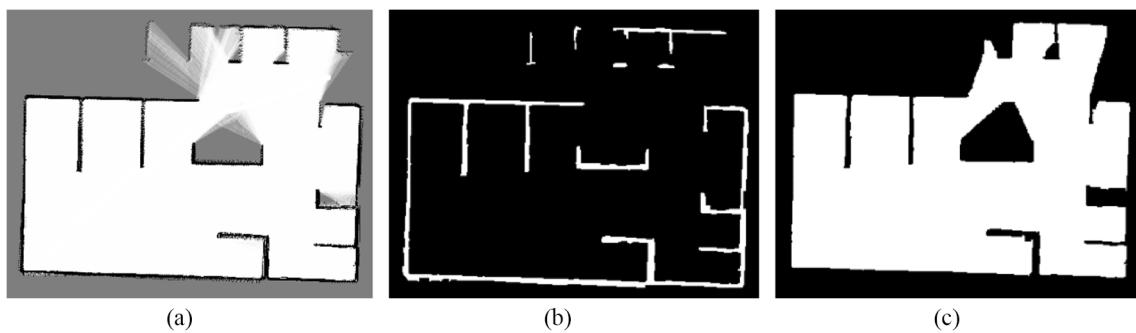


Fig. 3 a The OGM; b The buffer of obstacle area; c The smooth free area map

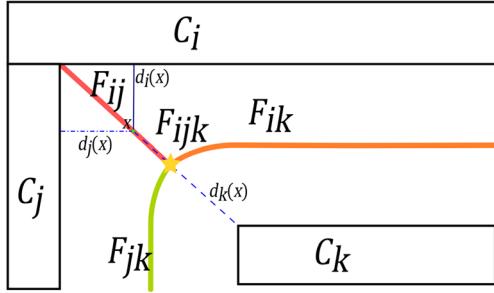


Fig. 4 An example of a GVG in a 2D space, the “yellow star” indicates \mathcal{F}_{ijk}

algorithm based on distance transformation could not guarantee the connectivity of the GVDs, and the post-processing methods were complex and lacked robustness. In this paper, we investigate the reason why the weak edges occur, and propose a novel and robust method for eliminating the weak edges. To the best of our knowledge and as the experiments in this paper show, the proposed method, which combines the proposed corner rounding method and image thinning algorithm, performs well to construct non-interrupted RAGVGs without any weak edges.

By exploring the reason for the occurrence of the weak edges, we find that there are four conditions, as shown in Fig. 6. Two of four conditions are rough wall surfaces that lead to a number of local maximums of distance, and the other two conditions are concave wall corners that lead to one local maximum of distance. In fact, these four conditions can be summarized into one common reason, that is, concave areas where concave angles exist.

To eliminate weak edges, we employ a series of image processing operations to deal with the concave areas. For rough wall surfaces, morphological operations and smoothing filtering are competent to fill the holes and remove the bumps on the rough wall surfaces. For concave wall corners, the local maximum of distance can be eliminated by changing the mitre corner into rounded corner, and an example is shown in Fig. 7.

The rounded corner is built by adding an obstacle $RC_{ij} = EF_{ijk}GH - EF_{ijk}G$, where $EF_{ijk}G$ is a circular sector of a round face RF termed as Eq. (6).

$$RF = \left\{ P \in \mathbb{R}^2 \mid \| \mathcal{F}_{ijk} - P \| \leq d_k(\mathcal{F}_{ijk}) \right\}. \quad (6)$$

It is easy to prove that no two-equidistant faces exist in $EF_{ijk}G$ because the minimum distance $d_i(x)$ from any point $x \in EF_{ijk}G$ to RC_{ij} is unique. Hence, the weak edge \mathcal{F}_{ij} is eliminated. Note that this elimination method can be applied to the concave wall corners with $\angle EHG < \pi$.

3.3 Extracting Reduced Approximated GVDs

3.3.1 The Proposed Image Thinning Algorithm

The proposed method for extracting an RAGVD from the SFAM is an iterative image thinning algorithm. The resulting RAGVDs are “approximated” because the image thinning algorithm produces a kind of skeleton that does not strictly satisfy the rules about the distance transformation of GVD as mentioned above. Nevertheless, the skeletons are very close to GVDs, and this kind of approximation is provably sufficient for robotics navigation [22].

To provide a detailed interpretation of the proposed image thinning algorithm, some notions related to pixel neighbours must be introduced first, with the help of descriptions in Figs. 8 and 9:

- (1) A *boundary pixel* is a foreground pixel whose 8 neighbours have at least one background pixel;
- (2) An *edge pixel* is a boundary pixel whose 4 neighbours have at least one background pixel;
- (3) The weight w of a pixel p is a convolution calculated with a 3×3 convolution kernel K by Eqs.(7) and (8):

Fig. 5 **a** A GVD; **b** A reduced approximated GVD

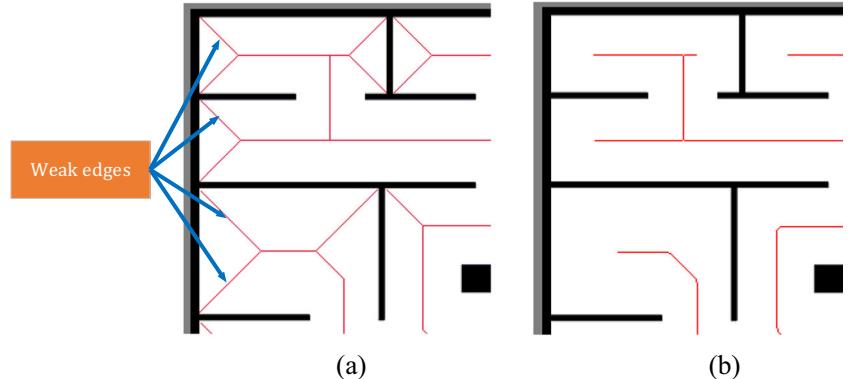
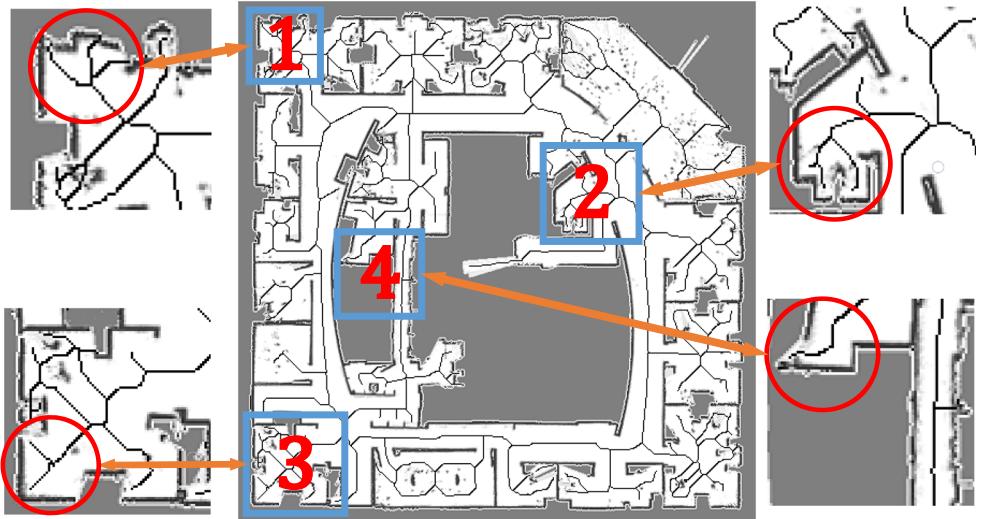


Fig. 6 The reason for the occurrence of the weak edges, where “1” and “2” refer to rough wall surfaces, and “3” and “4” refer to concave wall corners



$$K = \begin{bmatrix} 128 & 1 & 2 \\ 64 & 0 & 4 \\ 32 & 16 & 8 \end{bmatrix}, \quad (7)$$

$$F(p) = \sum_{j=1}^n (u_j(p) - u_{j-1}(p)) \mu(A_j) \quad (18)$$

where I is the binary matrix of the SFAM, and (x_p, y_p) refers to the coordinate of p .

- (4) A *corner pixel* is a boundary pixel whose 8 neighbours have at least one background pixel but 4 neighbours have no background pixel;
- (5) The *contiguous neighbours* of a pixel refer to only the non-interrupted pixel sequence in its 8 neighbours, as shown in Fig. 9, and the weights of pixels that have $m \in \{2, 3, 4\}$ contiguous neighbours are in set S_m , respectively, as the first three in Eq. (9);

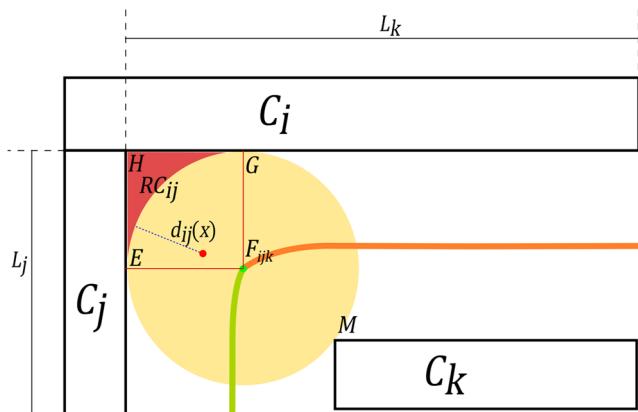


Fig. 7 Changing the mitre corner $\angle EHG$ to a rounded corner by adding RC_{ij} , which is shown by the red pixels

$$\begin{aligned} S_2 &= \{3, 6, 12, 24, 48, 96, 192, 129\} \\ S_3 &= \{7, 14, 28, 56, 112, 224, 193, 131\} \\ S_4 &= \{15, 30, 60, 120, 240, 225, 195, 135\} \\ S_{ns} &= \{3, 5, 7, 12, 13, 14, 15, 20, 21, 22, 23, 28, 29, 30, 31, 48, \\ &52, 53, 54, 55, 56, 60, 61, 62, 63, 65, 67, 69, 71, 77, 79, \\ &80, 81, 83, 84, 85, 86, 87, 88, 89, 91, 92, 93, 94, 95, 97, \\ &99, 101, 103, 109, 111, 112, 113, 115, 116, 117, 118, \\ &119, 120, 121, 123, 124, 125, 126, 127, 131, 133, 135, \\ &141, 143, 149, 151, 157, 159, 181, 183, 189, 191, 192, \\ &193, 195, 197, 199, 205, 207, 208, 209, 211, 212, 213, \\ &214, 215, 216, 217, 219, 220, 221, 222, 223, 224, 225, \\ &227, 229, 231, 237, 239, 240, 241, 243, 244, 245, 246, \\ &247, 248, 249, 251, 252, 253, 254, 255\} \end{aligned} \quad (9)$$

- (6) A non-skeleton pixel, which is a term in image thinning, is a pixel that definitely does not belong

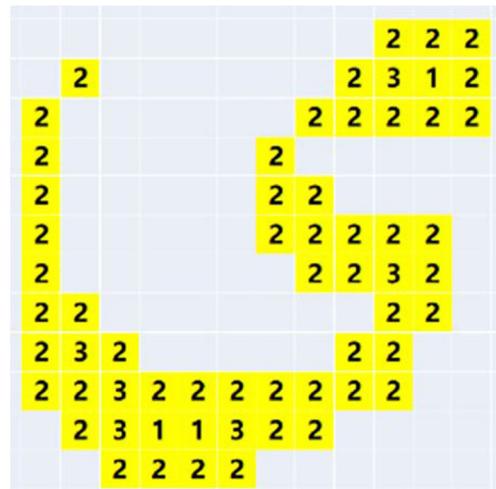
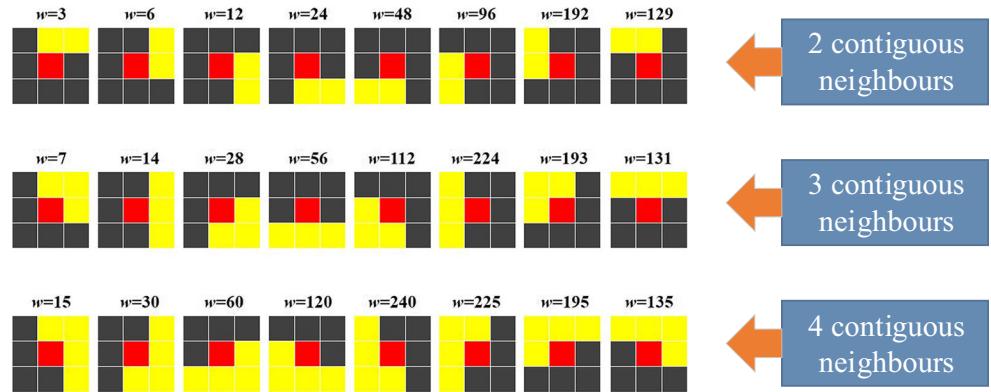


Fig. 8 Types of pixels (yellow-foreground and white-background). Pixels with ‘1’ refer to non-contour pixels, ‘2’ refers to boundary pixels, and ‘3’ refers to corner pixels

Fig. 9 The red pixel has 2, 3 or 4 contiguous neighbours in yellow



to the skeleton and should be removed, and S_{ns} in Eq. (9) provides all possible weights of non-skeleton pixels.

Given the knowledge presented above, the pseudo-algorithm of the proposed image thinning algorithm is presented in Algorithm 1.

Algorithm 1 :Image Thinning

Input: M : the input FAM
Initiate: B : the set of boundary pixels
 $IsDeleted = \text{true}$
 P_e : the set of edge pixels
 P_c : the set of corner pixels
 P_n : the set of pixels have 2,3 or 4 contiguous neighbours
 B : the set of boundary pixels
 B_{next} : the set of boundary pixels for next iteration
Change background pixels into '0' and foreground pixels '1'
State Traverse M and insert all boundary pixels into B

```

1: while  $IsDeleted = \text{true}$  do
2:    $IsDeleted = \text{false}$ 
3:   // one iteration for deleting non-skeleton boundary pixels
4:   for each  $p$  in  $B$  do
5:     if  $w(p)$  is edge pixel then
6:       insert  $p$  into  $P_e$ 
7:     else if  $w(p)$  is corner pixel then
8:       insert  $p$  into  $P_c$ 
9:     else if  $w(p)$  is in  $S_2 \cup S_3 \cup S_4$  then
10:      insert  $p$  into  $P_n$ ,  $IsDeleted = \text{true}$ 
11:    end if
12:   end for
13:   update  $M$  : update all  $p$  in  $P_n$  to '0'
14:   for  $S = P_e, P_c$  do
15:     for each  $p$  in  $S$  do
16:       if  $w(p)$  is in  $S_{ns}$  then
17:         update  $M$  : update  $p$  to '0',  $IsDeleted = \text{true}$ 
18:       else
19:         update  $M$  : update  $p$  to '1'
20:       end if
21:     end for
22:   end for
23:   // update boundary pixels set
24:   for each  $p$  in  $B$  do
25:     for each  $np$  in 8 neighbours of  $p$  do
26:       if  $np=1$  &  $np$  not in  $B_{next}$  then insert  $np$  into  $B_{next}$ 
27:       end if
28:     end for
29:   end for
30:   Update boundary pixels set:  $B \leftarrow B_{next}$ 
31: end while

```

3.3.2 Parallelization

In every iteration, the same tasks are repeatedly conducted on all contour pixels and only the 8 neighbours of each pixel are observed. Hence, the proposed image thinning algorithm can be parallelized using a multi-core processor. The parallelization approach is straightforward, and a simple example is exhibited in Fig. 10, in which the full image is equivalently divided into 4 regions. Then, retrieval and update operations on pixels in the i -th region R_i will be assigned to the i -th processor.

It is necessary to read the pixels of neighbour regions to update the pixels that are on the region edges. For instance, to update the black pixel in the region edge of R_1 , the orange pixels in R_2 have to be read, which means it is necessary to read the green pixels to update R_1 . However, it is possible that green pixels are updated by processors related to other regions. Thus, this border-crossing retrieval would likely cause concurrent-access anomalies if one processor is updating a pixel while the other processor is reading the same pixel.

To avoid updating and reading data on the same grid, an independent copy of all pixels on the region edges is employed to provide all concurrent threads with safe data. All processors can simultaneously read data from this copy but cannot update it in any case. Although some minor differences may exist in the pixel elimination step, our experiments show that no perceptible differences can be seen between the results of the parallel version and the original version.

3.4 Constructing RAGVGs

To represent connectivity and accessibility, an RAGVD should be transformed into an RAGVG that maintains topological information. An example of an RAGVG in a 2D space is shown in Fig. 11. This process should obey the following rules:

- a free endpoint should be transformed into an *end*, the degree of which is 1,

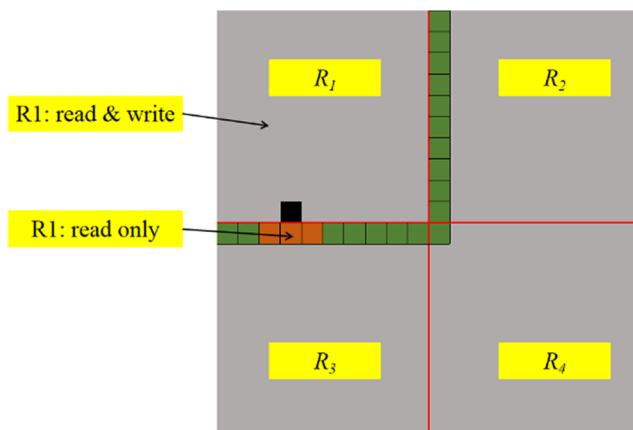


Fig. 10 An example of a parallelization approach

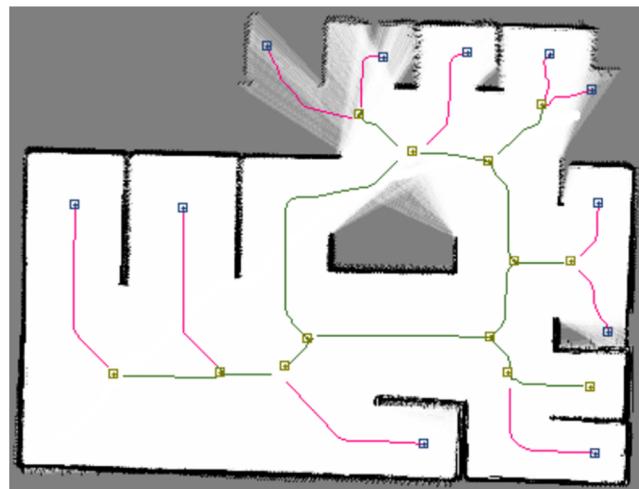


Fig. 11 Elements of an RAGVG (blue rectangles-ends, green rectangles-joints, red lines-branches and green lines-links)

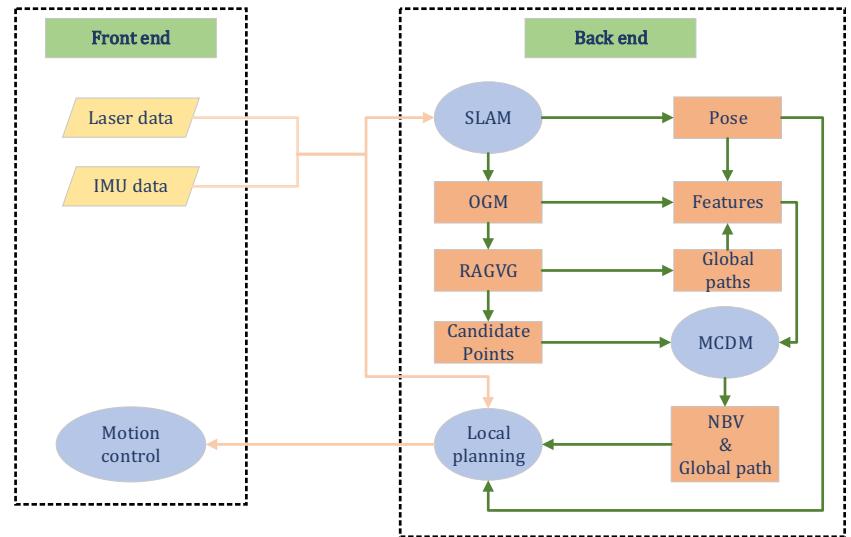
- an intersection should be transformed into a *joint*, the degree of which is larger than 1,
- an edge connecting an end with a related joint should be transformed into a *branch*,
- an edge connecting two joints should be transformed into a *link*.

The approach of extracting four elements from the RAGVD is modified from the flood-fill algorithm, which is easy to determine and implement. Each end is bound to connect with one specific joint through a specific branch, and all branches can roughly represent the primary structure, while all links can play a relatively unimportant role. Taking the RAGVG in Fig. 11 as an example, there are 13 ends and 14 joints in this graph; thus, if all the adjacency relationships are described by one matrix, a 27×27 matrix ADJ must be addressed in the path-finding process.

Because the adjacency matrix ADJ is sparse and its size will increase to hundreds by hundreds when a robot is exploring a large building, a great deal of memory and computation would be wasted if no improvements are made. Moreover, the Dijkstra algorithm is bound to pay little attention to ends and branches because there are no adjacency relationships among the ends, while the ends and branches are distributed in the peripheral part of the RAGVG. Hence, it is reasonable to reduce the space and time complexity in the path-finding process by dividing the RAGVG into two parts and describing the adjacency relationships with two matrices as Eqs.(10) and (11). In addition, the distance matrices D_S and D_M according to the lengths of the branches and links are constructed as Eq. (12).

The set of all branches and their related ends and joints is regarded as the *surrounding network*, and the adjacency relationships of all pairs of ends and their related joints are described by matrix ADJ_S .

Fig. 12 Mobile robot exploration framework



$$\left\{ \begin{array}{l} ADJ_S^{i,j} = 1, \text{ if } i\text{-th end is directly connected with } j\text{-th joint} \\ ADJ_S^{i,j} = 0, \text{ if } i\text{-th end is not directly connected with } j\text{-th joint} \end{array} \right.; \quad (10)$$

The sub-graph consisting of all links and their related joints is regarded as the *main network*, and the adjacency relationships among the joints are described by matrix ADJ_M .

$$\left\{ \begin{array}{l} ADJ_M^{i,j} = ADJ_M^{j,i} = 1, \text{ if } i\text{-th joint directly connects with } j\text{-th joint} \\ ADJ_M^{i,j} = ADJ_M^{j,i} = 0, \text{ else} \end{array} \right.. \quad (11)$$

$$\left\{ \begin{array}{l} D_S^{i,j} = length_{i,j}, \text{ if } ADJ_S^{i,j} = 1 \\ D_S^{i,j} = -1, \text{ else} \end{array} \right., \left\{ \begin{array}{l} D_M^{i,j} = D_M^{j,i} = length_{i,j}, \text{ if } ADJ_M^{i,j} = 1 \\ D_M^{i,j} = D_M^{j,i} = -1, \text{ else} \end{array} \right., \quad (12)$$

where $length_{i,j}$ is a measure of how long the corresponding edge is, e.g., the number of pixel points.

4 Full-Coverage Exploration Using RAGVGs

The whole workflow of the proposed exploration strategy is presented as Fig. 12, and the steps are presented as follows:

- (I) While the robot is moving, an OGM of the explored area is being updated in real time;
- (II) An RAGVG is constructed for the current OGM;
- (III) CPs are extracted from the RAGVG and roughly filtered with a few simple features;
- (IV) The CPs are evaluated by applying a Multi-Criteria Decision Making (MCDM) approach on some features to select the NBV;
- (V) The robot is navigated to the NBV and the exploration strategy restarts from (I) until there is no valid CP left after the filtering step in (III).

In the previous section, the solutions of step (1) and step (2) have already been presented. This section will illustrate the employed methods for the remaining steps, including how to select the NBV from CPs using the MCDM approach and how to quickly find a global path from the location of the robot to the NBV.

4.1 Features of CPs

To determine the NBV, each CP $p_i \in S$ has to be evaluated according to some features, such as the distance $Dist(r, p_i)$ from the location of p_i to the current position of the robot r and the potential information gain $PIG(p_i)$ when the robot stands at the location of p_i . All the features taken into consideration constitute the feature set F . The utility $u_j(p_i)$ is calculated with respect to the j -th feature of p_i according to certain rules, assuming that $u_j(p_i) \in [0, 1]$ and that the larger the utility

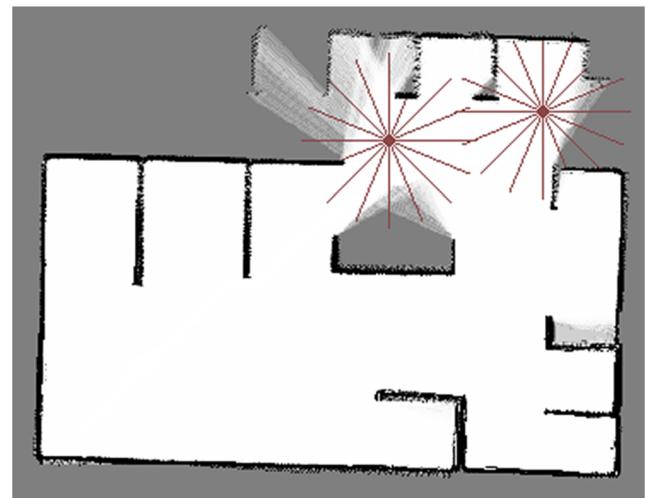


Fig. 13 Sixteen-ray casting approach

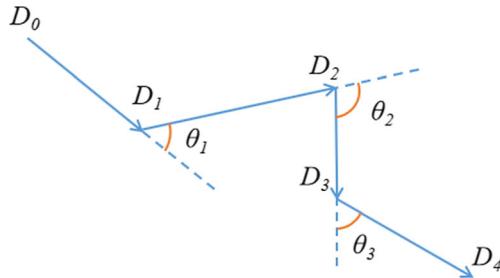


Fig. 14 Part of global path

is, the better the CP. Thus, naturally, if there are n features in F , p_i can be measured with a utility vector as Eq. (13).

$$U(p_i) = (u_1(p_i), u_2(p_i), \dots, u_n(p_i)). \quad (13)$$

In the proposed exploration strategy, a CP p is evaluated according to the following features:

- $A(p)$: the estimation of the unexplored grids that would be sensed from p ;
- $D(p)$: the length of the global path from the location of the robot to p ;
- $T(p)$: the sum of the turning angles of the global path;
- $C(p)$: the coverage condition of p .

Specifically, $A(p)$ is computed using a 16-ray casting approach from within five metres, as shown in Fig. 13. This approach simulates casting a 5-m single laser ray at intervals of $\pi/8$ from p , and estimates the potential information gain by counting the sum of unexplored grids that can be sensed by these 16 laser rays.

As shown in Fig. 14, the global path is represented by a series of sampling points. $D(p)$ is the length of the global path; it is calculated by the sum of Euclidean distances of the sampling points of the global path as Eq. (14). The other feature $T(p)$ describes the complexity of the global path with the sum of the turning angles. By assuming that there are M sampling points SP that are ordered from the starting point to the target

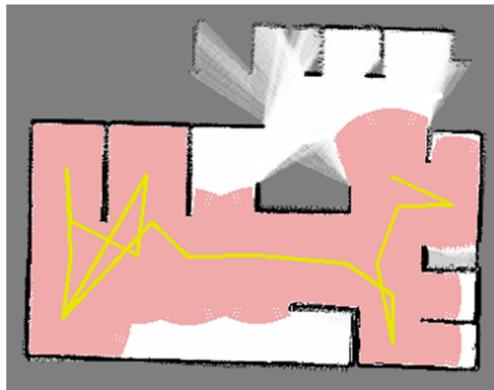


Fig. 15 Coverage occupancy map

point and setting θ_i as the turning angle at related $D_i \in SP$, then $T(p)$ can be calculated as Eq. (15).

$$D(p) = \sum_{i=0}^{M-1} \|D_i D_{i+1}\|. \quad (14)$$

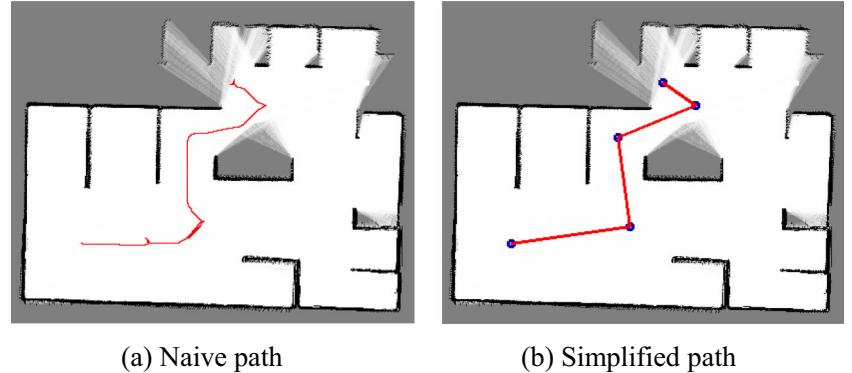
$$\theta_i = \pi - \angle D_{i-1} D_i D_{i+1} \\ T(p) = \sum_{i=1}^{M-2} \theta_i \quad (15)$$

$C(p)$ is used to determine whether a p has been sensed by certain sensors on a robot, which is important information for avoiding omission and repetition. The estimation of $C(p)$ depends on the specialty of the external sensor, which determines the updating on the coverage occupancy map (COM). A COM estimates the group of grids that have been detected by specific sensors (e.g., a biological detector and 3D laser rangefinder (LRF)), and it is built by a constraint region growing algorithm applied at sampling points of the robot trajectory. Figure 15 shows a COM updated by a 2D LRF whose field of view is 2π . With a given COM, if UC is established as the set of uncovered grids and CG is established as the set of covered grids, then $C(p)$ can be computed according to a distance function as Eq. (16):

$$x_{uc} = \operatorname{argmin}_{x \in UC} Dist(p, x) \\ x_c = \operatorname{argmin}_{x \in C} Dist(p, x) \\ C(p) = \begin{cases} -Dist(p, x_{uc}), & \text{if } p \text{ is covered} \\ Dist(p, x_u), & \text{if } p \text{ is uncovered} \end{cases} \quad (16)$$

These four features work together to determine the “best” CP as the NBV to guide the mobile robot to traverse the unknown environment. Moreover, each feature has its own effects while evaluating the CPs. Intuitively, larger values of $A(p)$ and $C(p)$ indicate a better p because the mobile robot must discover unexplored space and detect uncovered space to complete full-coverage exploration. In contrast, larger values of $D(p)$ and $T(p)$ indicate a worse p because the mobile robot would waste time and energy if it ignored what is near and sought what is far away. Hence, $A(p)$ and $C(p)$ are regarded as positive features, while $D(p)$ and $T(p)$ are negative features. The utility of each of these features is computed and linearly normalized to $[0, 1]$ as Eq. (17).

$$u_A(p) = \frac{A(p) - \min_{q \in CP} A(q)}{\max_{q \in CP} A(q) - \min_{q \in CP} A(q)}, \\ u_D(p) = 1 - \frac{D(p) - \min_{q \in CP} D(q)}{\max_{q \in CP} D(q) - \min_{q \in CP} D(q)}, \\ u_T(p) = 1 - \frac{T(p) - \min_{q \in CP} T(q)}{\max_{q \in CP} T(q) - \min_{q \in CP} T(q)}, \\ u_C(p) = \frac{C(p) - \min_{q \in CP} C(q)}{\max_{q \in CP} C(q) - \min_{q \in CP} C(q)}. \quad (17)$$

Fig. 16 Global path

4.2 MCDM Approach

An evaluation system must be developed to assign each CP $p \in S$ a score, and then choose the CP with the highest score as the NBV. Nevertheless, a multi-dimensional utility vector $U(p)$ cannot be directly used to rank CPs. Hence, the MCDM approach [5] is employed to evaluate CPs, which provides an aggregation function to define a global utility using a Choquet integral, thus yielding a single value score based on $U(p)$ and the features set F .

In addition, all features in F are not always independent because two or more features may exist that are very similar or closely related to each other. For example, because the sum of turning angles is likely to increase with the growth of the global path, a redundant relationship exists between D and T . In contrast, a synergistic relationship exists between D and A because a place near the robot that holds considerable potential

information is obviously a prior exploration target. To address redundant and synergistic relationships among features, MCDM is designed to avoid underestimating or overestimating as much as possible by taking correlativity of features into consideration through a normalized fuzzy measure function.

4.2.1 A Simple Filter on CPs

The ends of an RAGVG are located in three conditions: around the corners of a wall, in separated rooms or near frontiers between explored and unexplored areas (see Fig. 11). Hence, the ends can be regarded as a point set S that indicates the critical region and margin of the explored area. In other words, point set S is a superset of CPs. Two types of absolutely unbefitting points $p \in S$ should be first excluded from S to obtain a set CP filled with competitive CPs:

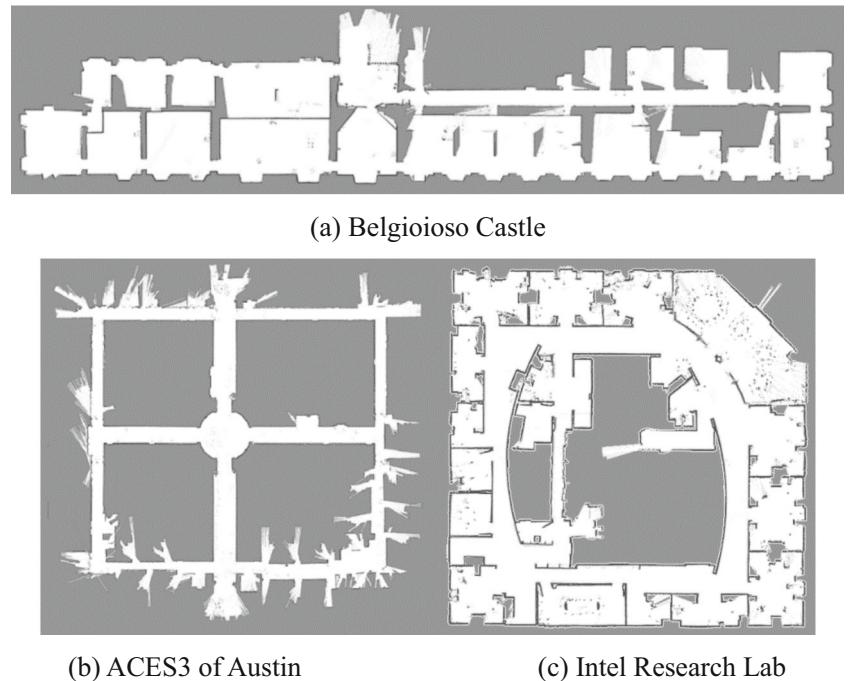
Fig. 17 Famous SLAM data sets

Table 1. Results for Belgioioso Castle.

Algorithm	Version	<i>TT</i> / ms	<i>TG</i> / ms	SNG	<i>TL</i> / p	<i>IR</i> / %
Brushfire	Serial	254	88	211	8233	99.5
	Parallel	223				
Zhang-Suen	Serial	407	64	183	7162	95.3
	Parallel	127				
Proposed	Serial	209	16	66	4092	99.5¹
	Parallel	58				

The entries in boldface indicate the best performance of these comparative algorithms.

- There may exist $q \in S$ having $u_j(q) \leq th$ for a certain j -th feature, where th is an empirical threshold value. In this case, q is quite “bad” with respect to the j -th feature, and thus it should be removed from S ;
- There may exist $p, q \in S$ satisfying $u_j(p) \geq u_j(q)$ for all features. In this case, p has better utility than q with respect to all features; thus, q should be removed from S .

4.2.2 A Fuzzy Measure Function

Given CP , the utility vector $U(p)$ of each $p \in CP$ can be calculated as mentioned above, and $U(p)$ is sorted in ascending order having $0 = u_0(p) \leq u_1(p) \leq u_2(p) \leq \dots \leq u_n(p) \leq 1$. Then, the global utility for evaluating a CP $p \in CP$ is computed by an aggregation function using a Choquet integral as Eq. (18):

$$F(p) = \sum_{j=1}^n (u_j(p) - u_{j-1}(p))\mu(A_j) \quad (18)$$

where the features subset $A_j \subset F$ is defined as $A_j = \{F_m, m \in N \mid u_j(p) \leq u_m(p) \leq u_n(p)\}$, and μ is a normalized fuzzy measure function to define the weights of a group of features.

The normalized fuzzy measure function μ is defined for a group of features $G \subseteq F$, and this function is employed to measure the overall contribution of the group of features to the

Table 2. Results for ACES3 of Austin.

Algorithm	Version	<i>TT</i> / ms	<i>TG</i> / ms	SNG	<i>TL</i> / p	<i>IR</i> / %
Brushfire	Serial	366	60	149	7246	99.8
	Parallel	327				
Zhang-Suen	Serial	459	48	108	5964	93.6
	Parallel	151				
Proposed	Serial	321	31	76	4509	99.7
	Parallel	130				

The entries in boldface indicate the best performance of these comparative algorithms.

Table 3. Results for the Intel Research Lab.

Algorithm	Version	<i>TT</i> / ms	<i>TG</i> / ms	SNG	<i>TL</i> / p	<i>IR</i> / %
Brushfire	Serial	1127	396	468	22,781	99.8
	Parallel	912				
Zhang-Suen	Serial	1943	324	383	17,309	91.0
	Parallel	989				
Proposed	Serial	910	96	169	12,543	99.8
	Parallel	539				

The entries in boldface indicate the best performance of these comparative algorithms.

aggregation function according to their dependency relationships. Specifically, the overall contribution of G should be less than the simple sum of the utilities of G that have redundant relationships and should be larger when synergistic relationships exist.

Diverse tasks require different exploration strategies that vary in features and feature weights. More specifically, there is a chance that a CP p with a high global utility in one strategy has a low global utility in another strategy. In general, MCDM can define various exploration strategies by adjusting the fuzzy measure function μ .

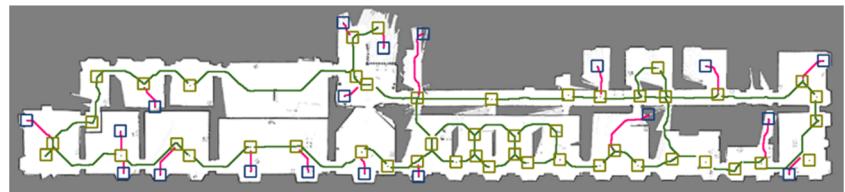
4.3 Global Path Planning

A few methods for finding paths on an OGM through a GVD have been developed, and the main idea is applying the A* algorithm on a GVD space. GVD-based path-finding methods have better performance because a GVD simplifies the explored area into a connected skeleton so that greedy searching algorithms need to address only a small number of grids. The global path generated from a GVD is collision-free but sub-optimal; thus, this global path requires reduction and optimization.

In this paper, the RAGVG mentioned above provides a topological model of the explored area that can be easily employed to find a collision-free path from any starting location L_s to any other destination location L_d . In fact, the proposed solution to path finding is a slight extension of the Dijkstra algorithm on a weighted undirected graph.

However, due to the geometrical characteristics of an RAGVD, the global path that directly consists of a series of continuous pixels is complex and redundant, as shown in Fig. 16a. Curve simplification must be conducted to guarantee less estimation of the time or energy consumption of the resulting global path. The Douglas-Peucker algorithm [34] is employed, in which the possibility of the simplified path is tested by a buffer of the previously mentioned obstacle area. As shown in Fig. 16b, the simplified path is represented by a series of critical path points, which is more conducive to motion planning instead of continuous pixel chains. In addition, it

Fig. 18 RAGVG of Belgioioso Castle constructed by the proposed algorithm



is obvious that the simplified path is more straightforward than the naive path in terms of time and energy consumption, which largely affects the efficiency of autonomous exploration.

5 Experiments

In this section, two categories of experiments will be presented to suggest the validity, practicability and advancement of our method. The first experiment is conducted to evaluate the performance of the proposed method for quickly constructing an RAGVG, and the second experiment focuses on the proposed full-coverage exploration strategy, under the guidance of which a point cloud capturing task and a search and rescue task are completed smoothly and efficiently.

5.1 Experiment on RAGVG Construction

5.1.1 Comparative Algorithms, Data and Metrics

The workflow of constructing a GVG mainly consists of three parts: pre-processing an OGM, extracting a GVD and constructing a topological graph. Among these three parts, the most pivotal step is extracting a GVD by thinning an OGM

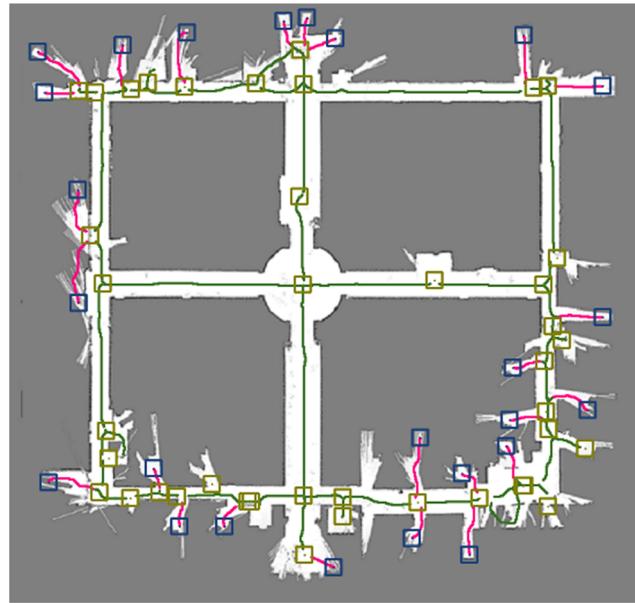


Fig. 19 RAGVG of ACES3 of Austin constructed by the proposed algorithm

because this step is time consuming and the quality of the GVD directly influences the completeness, validity and simplicity of the resulting topological graph. From the literature, we found two typical thinning algorithms, including the brushfire algorithm [30] and the Zhang-Suen algorithm [31]. The brushfire algorithm is based on distance transformation, while the Zhang-Suen algorithm and our proposed algorithm are based on circularly deleting non-skeleton boundary pixels. After some minor modifications, their serial and parallel versions are compared with the proposed method.

Three OGMs (see Fig. 17) are employed to compare the performance of the thinning algorithms mentioned above. All of the OGMs are generated by Gmapping-SLAM developed by Cyril Stachniss et al. (<https://openslam-org.github.io/>) on three common data sets. The first OGM is *Belgioioso Castle*, which contains connecting rooms, rooms on both sides of a long corridor and few clusters. The second OGM is *ACES3 of Austin*, which consists of some vertical and parallel corridors and some parts of rooms on both sides of long corridors. The third OGM is the *Intel Research Lab*, which is the hardest because it is a relatively large indoor space with plentiful clusters that exist in almost every room.

To measure the performance of all the algorithms, some parameters are recorded or computed as metrics, including

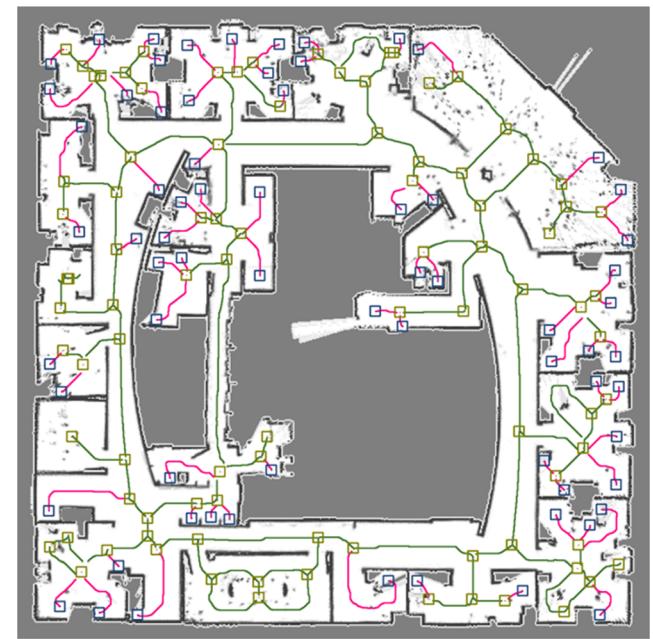
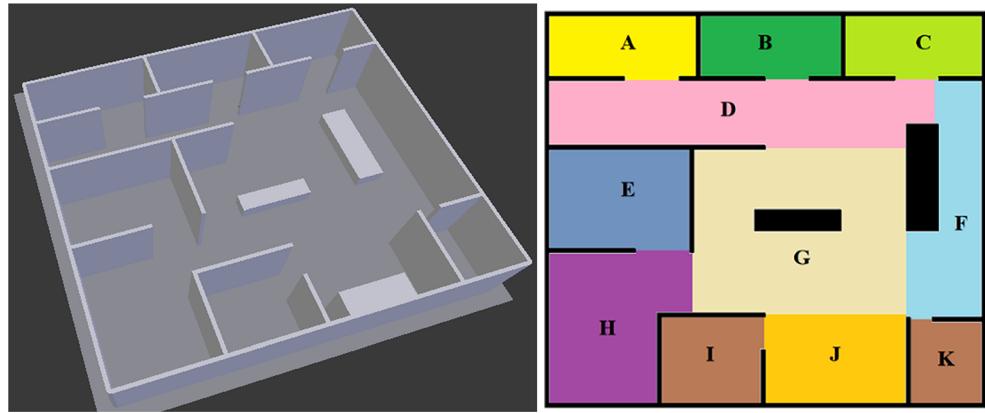


Fig. 20 RAGVG of the Intel Research Lab constructed by the proposed algorithm

Fig. 21 Three-dimensional model and floor plan of the scenario



(1) TT , time consumption of the thinning process; (2) TG , time consumption of constructing the topological graph; (3) SNG , sum of the nodes in the resulting graph; (4) TL , total length of the edges in resulting graph and (5) IR , information retention of the resulting graph. The information retention, which is computed by comparing a constraint buffer of the GVG and the original OGM, indicates the completeness of the topological representation of a GVG because some small pixel blocks would be ignored when thinning the OGM.

5.1.2 Result and Discussion

Ten groups of valid results are collected for each algorithm on each OGM, and the average data values of the corresponding metrics are shown in Tables 1, 2, and 3. The resultant RAGVGs constructed by the proposed method are shown in Figs. 18, 19, and 20.

From the perspective of the time consumption of the thinning process, the proposed algorithm outperforms the other two algorithms for both serial and parallel versions. It must be explained that because brushfire is not a loop algorithm, the parallel version of brushfire is not markedly faster than the serial version. For the processing of constructing GVGs, the GVDs generated by the brushfire algorithm are extremely redundant, and thus it costs more time to construct a larger topological graph. Furthermore, the GVDs generated by the Zhang-

Suen algorithm tend to be fragmented, and thus it costs more time to construct a number of interrupted topological graphs. For information retention, the GVGs constructed by the brushfire and the proposed algorithms covered more than 99% of the area, but the Zhang-Suen algorithm missed a portion of the area due to its fragmented GVDs.

It can be clearly seen that the RAGVG constructed by the proposed method can simplify the free area in an OGM into a topological representation, which covers long corridors, small channels, single rooms and almost all frontiers. Thus, this RAGVG is detailed to navigate the mobile robot with its nodes and edges. In contrast, the GVDs generated by the brushfire algorithm are highly redundant; thus, further edge elimination has to be developed to construct a simpler topological graph, and the GVDs generated by the Zhang-Suen algorithm are too fragmented to be employed to construct a topological graph that describes complete connectivity of the free area.

In summary, on the one hand, the proposed algorithm can address large-scale OGMs with much less time consumption than other algorithms, which satisfies the requirement of real-time decision making for autonomous exploration tasks. On the other hand, the RAGVD generated by the proposed algorithm is much less redundant and non-interrupted, which leads to the resultant RAGVG being smaller and simpler while still covering almost the entire area of the OGM.

Table 4 Values of the fuzzy measure function

Group of features G	$\mu_1(G)$	$\mu_2(G)$	Group of features G	$\mu_1(G)$	$\mu_2(G)$
A	0.25	0.4	C, D	0.8	0.6
C	0.4	0.15	C, T	0.5	0.3
D	0.25	0.35	D, T	0.3	0.4
T	0.1	0.1	A, C, D	0.95	0.95
A, C	0.75	0.6	A, C, T	0.85	0.8
A, D	0.6	0.85	A, D, T	0.6	0.9
A, T	0.4	0.55	C, D, T	0.7	0.7

5.2 Simulation on Full-Coverage Exploration

5.2.1 Comparative Method and Metrics

Two tasks, *point cloud capture* and *search and rescue*, are used to evaluate the performance of the proposed autonomous exploration strategy. In the point cloud capture simulation, the mobile robot autonomously decides where to travel to quickly capture the complete point cloud of an indoor space. In the search and rescue

Fig. 22 Autonomous exploration controlled by MCDM with $\mu_1(G)$

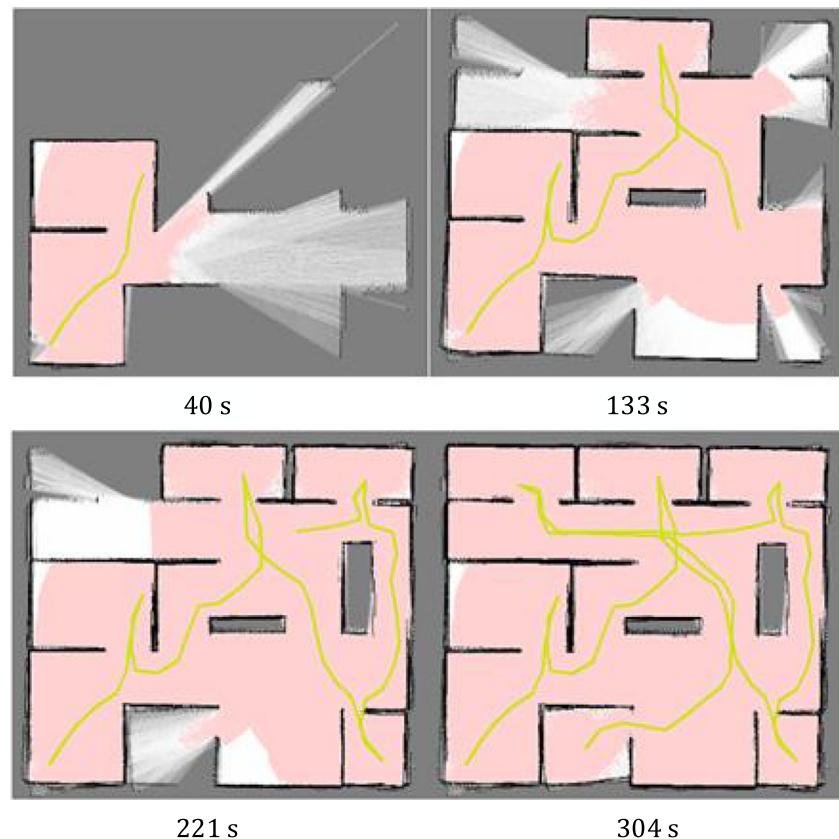


Fig. 23 Autonomous exploration controlled by MCDM with $\mu_2(G)$

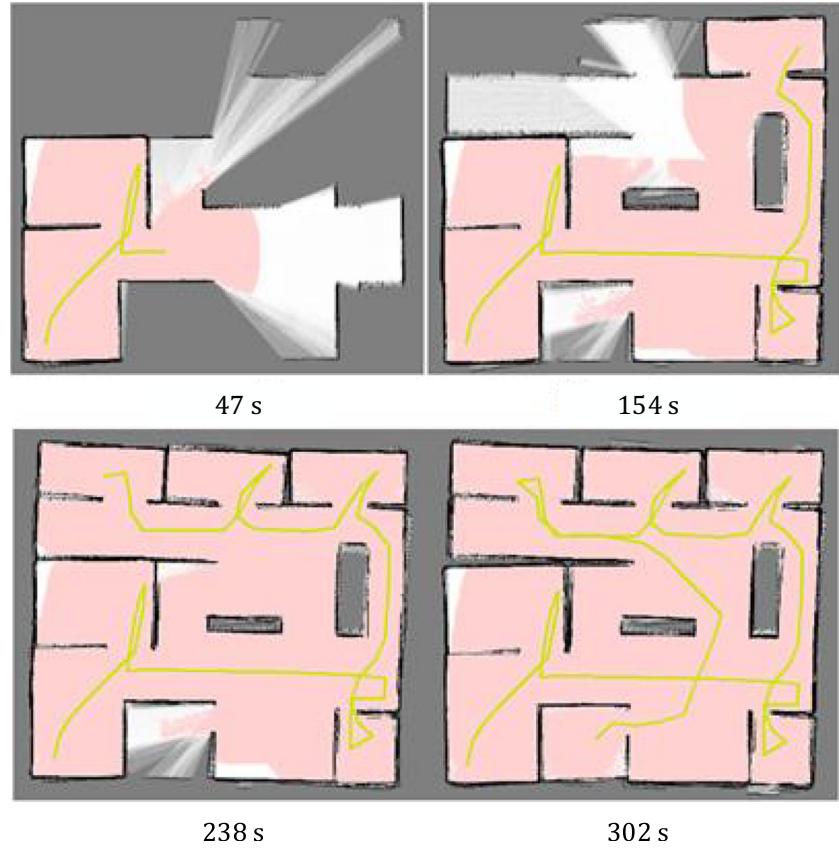
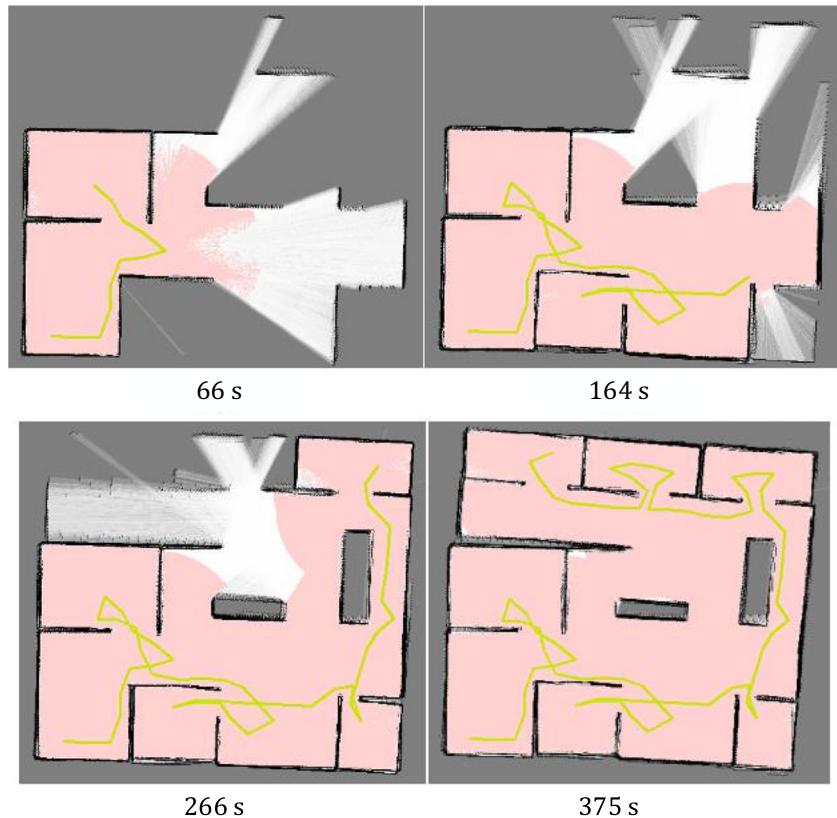


Fig. 24 Autonomous exploration controlled by the frontier-based method



simulation, the mobile robot autonomously decides where to go to search for more trapped people.

In addition, another popular exploration strategy based on greedily traversing frontiers is used to compare our method. The frontier-based method generates CPs from the frontiers that are larger in size than a predefined threshold. In our simulation, the threshold is set to 20 grids, which represents 100 cm, and the NBV for the frontier-based method is chosen according to the features mentioned above as Eq. (19):

$$NBV = \underset{p \in CP}{\operatorname{argmax}} (A(p) \times C(p) \times D(p) \times T(p)). \quad (19)$$

Table 5 TT and TN values for the simulations

Methods		TT / s	TN / ms	SC	PC / %
MCDM- $\mu_1(G)$	Max	332	182	8	79.3
	Average	320	163	5.1	48.1
	Min	304	138	3	24.8
MCDM- $\mu_2(G)$	Max	327	171	9	81.7
	Average	319	162	5.4	46.2
	Min	302	125	3	19.7
Frontier-based method	Max	419	1370	26	38.5
	Average	392	1258	22.0	23.2
	Min	375	943	18	12.1

A number of metrics are used to evaluate the frontier-based method and the proposed method, including (1) CA, sum of the coverage area over time; (2) TT, total time consumption for finishing exploring the entire configuration space; (3) TN, time consumption for selecting the NBV in one single decision-making procedure; (4) SO, sum of the obstacle areas encountered by the robot trajectory; (5) SC, sum of CPs to be evaluated in one single decision-making procedure; and (6) PC, percentage of CPs whose global utility is larger than half of the maximum global utility in one single decision-making procedure. In addition, for the search and rescue task, one more metric is referred to as (7) PF, sum of trapped people found by the robot over time.

For the specific calculations of these metrics, both obstacle and free grids are regarded as the coverage area while counting CA, and SO can be obtained by intersecting the obstacle area with a 30 cm buffer region of the robot trajectory. CA, TT and PF are employed to evaluate the efficiencies of the exploration strategies, while TN, SO and PC for presenting the improvement of our method.

5.2.2 Scenario

An indoor scenario, whose 3D model and floor plan are shown in Fig. 21, is constructed to simulate autonomous exploration. This indoor scenario involves a 20 m × 20 m office environment with small rooms, obstacles and a short corridor. For the convenience of discussing the results, the floor plan is

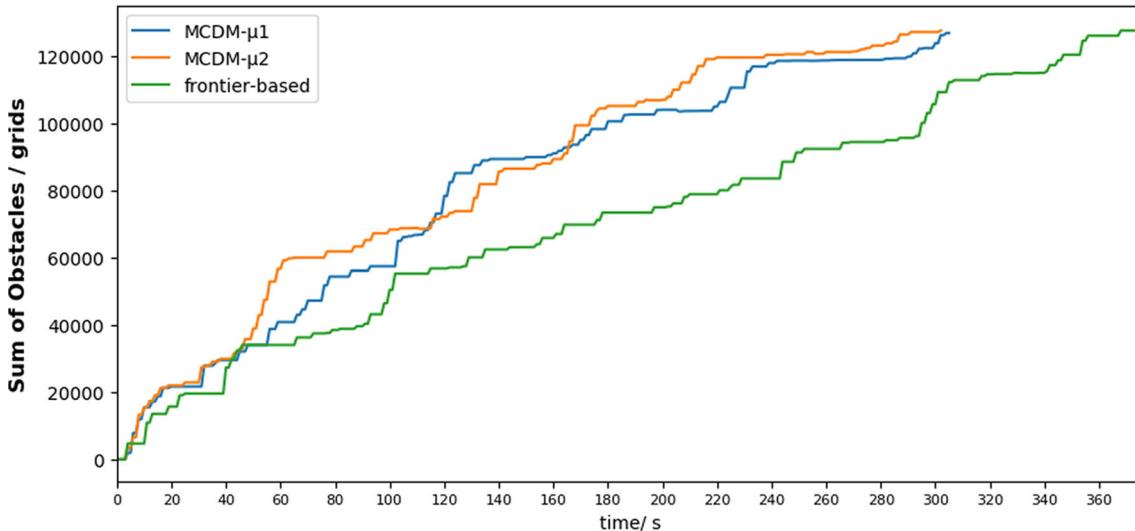


Fig. 25 Sum of obstacle grids encountered by the path

divided into regions labelled with capital letters. For the point cloud capture task in this 3D model, a simulative 3D LRF is employed to collect point clouds. In the search and rescue task, trapped people are randomly distributed in the rooms and can be detected by a simulative biosensor.

5.2.3 Fuzzy Measure Function of MCDM

The specific values of the fuzzy measure function μ for the two tasks are presented in Table 4, in which $\mu_1(G)$ places greater importance on the coverage information and $\mu_2(G)$ places greater importance on the potential information gain and distance.

5.2.4 Result and Discussion

The progression of autonomous exploration controlled by MCDM with $\mu_1(G)$, $\mu_2(G)$ and the frontier-based method

are exhibited in Figs. 22, 23, and 24, in which the pink pixels represent the coverage area and the yellow polyline indicates the trajectory of the mobile robot. Additionally, 5 groups of valid results are collected for each method, and the results corresponding to metrics TT and TN are shown in Table 5, while the results corresponding to metrics SO , CA and PF are shown in Figs. 25, 26, and 27.

The results show that all three methods are qualified to complete full-coverage exploration tasks in this indoor space. The mobile robot starts from room H and decides to go to room E during the first stage. Then, in the next stage, MCDM- $\mu_1(G)$ directs the robot to explore room B by crossing regions G and D, MCDM- $\mu_2(G)$ directs the robot to explore room K by crossing region G and the frontier-based method directs the robot to explore room I by crossing regions G and J. Finally, regardless of the region sequences that the mobile robot explores, almost the entire indoor space is

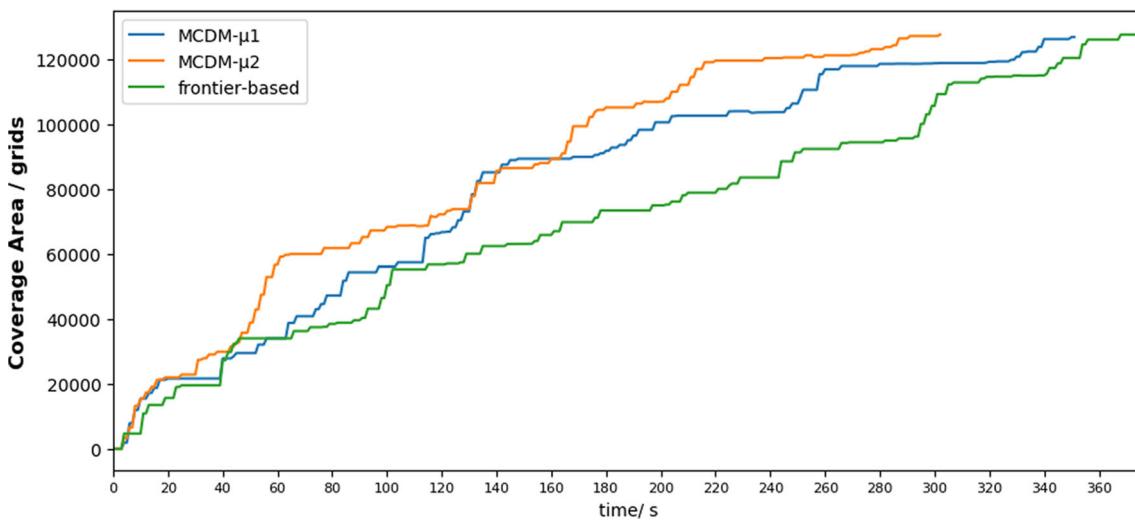
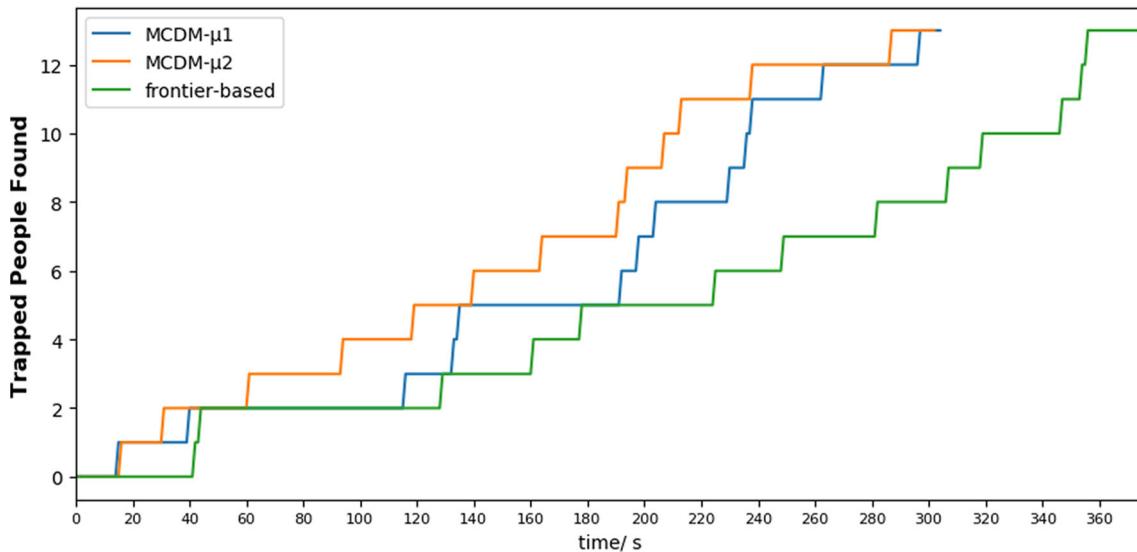


Fig. 26 Coverage

**Fig. 27** Trapped people found

covered. In addition, the robot trajectories of the two proposed methods and the frontier-based method seem different. The vast majority of robot trajectories guided by the global path from the RAGVG are relatively far from obstacles, whereas part of the robot trajectory guided by the global path from the A* algorithm is quite near obstacles. This difference in trajectories stems from the global path and is quantized by metric *SO* in Fig. 25.

Table 5 presents some comparisons of the key components in the three methods. The two proposed methods require approximately 20% less time than the frontier-based method for full-coverage exploration. One of the improvements of the proposed method is that it takes approximately 85% less time than the frontier-based method to select the NBV, which can be partly explained by the *SC* and *PC* metrics. On the one hand, the proposed methods' time costs are mainly in constructing an RAGVG from an OGM, whereas the frontier-based method's time cost is mainly in finding the global path with the A* algorithm. On the other hand, the sum of CPs generated from an RAGVG is substantially less than that

of the frontier-based method, and on average, more than 45% of CPs hold relatively high global utility after simple filtering. In contrast, on average, 77% of CPs generated from frontiers hold low global utility even after the same filtering. Overall, compared with the frontier-based method, although the proposed method requires time to construct RAGVGs, the proposed method saves much more time in path finding and rapidly filtering CPs with potentially low global utility.

Figures 26 and 27 indicate the process of the point cloud capture task and the search and rescue task, respectively. Coverage in Fig. 26 represents the area of the indoor space scanned by 3D LRF, and in Fig. 27, the trapped people found represent how many people the mobile robot found in the simulated building. The results clearly show that the task processes controlled by the two proposed methods are better than that of the frontier-based method. In summary, the proposed method can effectively control a single mobile robot to explore a previously unknown space, and the proposed method outperforms the frontier-based method in terms of time consumption and robot trajectory quality.

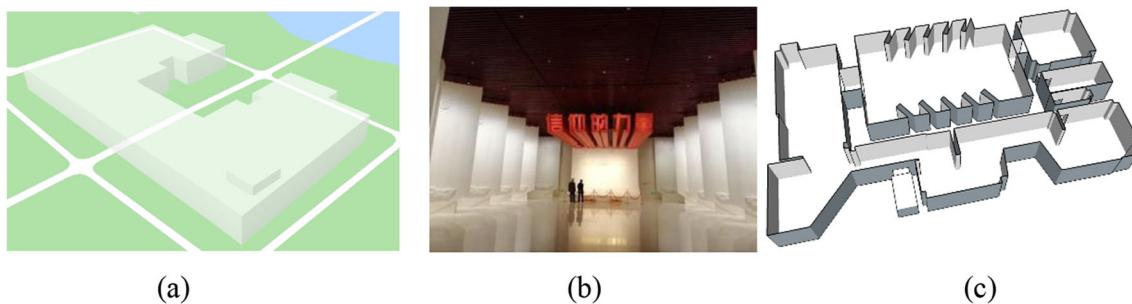
**Fig. 28** **a** 2.5D building model of the museum building; **b** Part of the interior of the museum; **c** 3D indoor model of the museum constructed from point clouds captured

Fig. 29 **a** the design of our mobile robot; **b** our real mobile robot

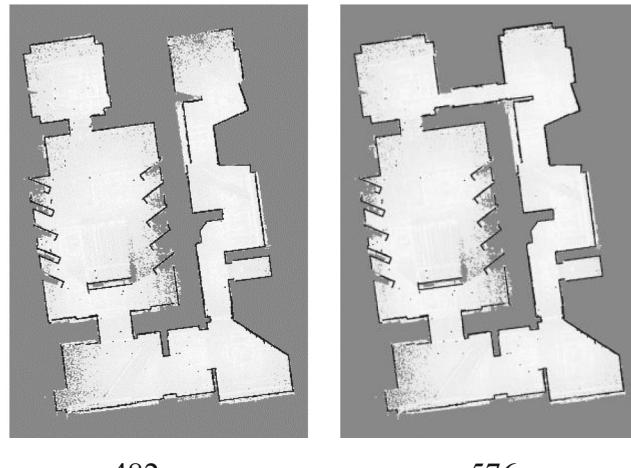
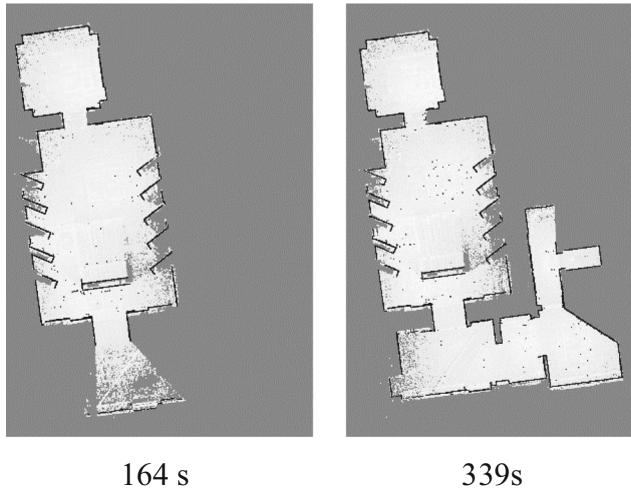
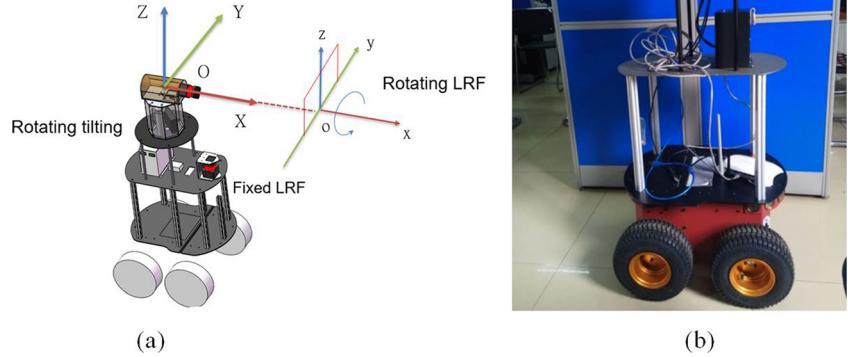


Fig. 30 Autonomous exploration controlled by MCDM with $\mu_2(G)$

5.3 Experiment in a Real-World Museum

The proposed method was also implemented on a real mobile robot, for capturing the point clouds of a real-world museum. What needs illustration is that taking the limited efficiency and computer memory, some real-time logs were closed so that the result is not plentiful as that of simulation experiment.

5.3.1 Scenario

The real-world scenario is a $45 \text{ m} \times 27 \text{ m}$ indoor museum that lies on the left side of the building in Fig. 28a, and this museum can be regarded as a water tight space so that no laser range would leak to exterior space. More intuitively, one part of the interior and a 3D indoor model of the museum is respectively shown in Fig. 28b, c.

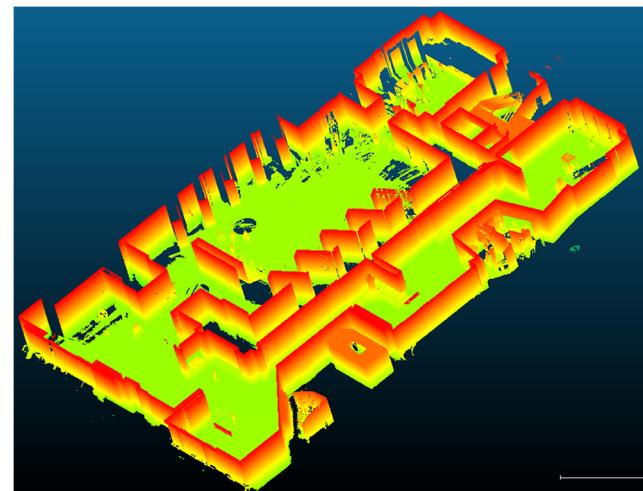


Fig. 31 The resulting point clouds after removing the roof

5.3.2 The Mobile Robot and Scanners

As is shown in Fig. 29, our mobile robot is equipped with two 2D LRFs, the scan range of each of them is 30 m, 270°. The fixed LRF is used for SLAM while the rotating LRF is used for capturing 3D point clouds. At each step, the mobile robot stops at the location of NBV, and the rotating tilting begins to rotate to capture 3D point clouds using the rotating LRF, and it takes 12 s to rotate 360° to finish capturing at this station. What needs illustration is that robot kinematics are beyond our research, we focus on improving performance using better exploration strategy, instead of better robot kinematics.

5.3.3 Result and Discussion

As is shown in Fig. 30, the mobile robot started from the left-up room and travelled across a commodious hall, which has ten projecting walls, then it moved through a narrow corridor and reached the left-bottom room. Next, it went to the right-bottom room and took a left turn, after travelling through some narrow corridors it reached the right-up room, finally it closed a loop at the left-up room across a narrow corridor.

It took 576 s to finish capturing 15 station point clouds of this indoor space, which indicates that the mobile robot stopped more than 180 s. Hence, it only took less than 396 s to finish full-coverage exploration in this museum. The resulting point clouds is shown in Fig. 31, and it can be obviously seen that there barely exists missing part in the integrated point clouds. And it can be claimed that the proposed method also performs well for real-world scenario and real robot.

6 Conclusions

Autonomous exploration is an important ability of mobile robotics, and the main aim of research on this topic is to guide a robot to explore a previously unknown space while consuming less time. This paper presents a novel autonomous exploration system based on an RAGVG, and the discovered improvements are as follows:

- A fast, robust and parallel algorithm was proposed for constructing an RAGVG from an OGM;
- The efficiency of selecting the NBV was markedly improved by generating more competitive CPs and by using fast graph-based path planning;
- The number of local obstacles that must be avoided was reduced by means of a collision-free global path that is rapidly generated from an RAGVG.

Simulation and real robot experiments show that the proposed algorithm can generate an RAGVD quickly and robustly that is one-pixel-wide, non-interrupted and relatively non-redundant, and the RAGVG constructed from this RAGVD can represent almost all the connectivity of an indoor space with fewer edges and nodes. Selecting the NBV based on an RAGVG takes 85% less time than the frontier-based method, and the number of required local obstacle avoidance manoeuvres is decreased by means of a collision-free path. Combining the improvements mentioned above, the total time consumption for a full-coverage exploration by the proposed method is approximately 20% less than that by the frontier-based method.

However, some factors still exist that affect the efficiency of autonomous exploration. From the perspective of graph theory, the edges and nodes of the RAGVG in the coverage area are still redundant and can be further simplified. In terms of selecting the NBV using the MCDM approach, the value of the fuzzy measure function is empirical. These topics should be explored further.

Acknowledgements This study is funded by the National Key R&D Program of China (2017YFB0503701) and the National Natural Science Foundation of China (41871298).

Author Contributions This study was completed by the co-authors, and the major experiments and analyses were undertaken by Xinkai Zuo. Lin Li and Haihong Zhu supervised and guided this study. Dalin Li and Fan Yang designed the proposal for the experiments and conducted the analyses. Jun Liu and Yifan Liang aided in the collection and analysis of the data, and Fei Su contributed to algorithm implementation while Xinkai Zuo wrote the paper. Huixiang Peng and Gang Zhou helped a lot to conduct the experiment of the real-world scenario. All the authors have read and approved the final manuscript.

Compliance with Ethical Standards

Conflict of Interest The authors declare no conflicts of interest.

References

1. Burgard, W., Moors, M., Stachniss, C., et al.: Coordinated multi-robot exploration[J]. IEEE Trans. Robot. **21**(3), 376–386 (2005)
2. Stachniss, C., Hahnel, D., Burgard, W.: Exploration with active loop-closing for FastSLAM[C]. Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on. IEEE, **2**, 1505–1510 (2004)
3. Stachniss, C., Burgard, W.: Mapping and exploration with mobile robots using coverage maps[C]. Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. IEEE, **1**, 467–472 (2003)
4. Calisi, D., Farinelli, A., Iocchi, L., et al.: Autonomous navigation and exploration in a rescue environment[C]. Safety, Security and Rescue Robotics, Workshop, 2005 IEEE International. IEEE, 54–59 (2005)

5. Basilico, N., Amigoni, F.: Exploration strategies based on multi-criteria decision making for searching environments in rescue operations[J]. *Auton. Robot.* **31**(4), 401 (2011)
6. Low, K.L., Lastra, A.: An adaptive hierarchical next-best-view algorithm for 3d reconstruction of indoor scenes[C]. Proceedings of 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006), 1–8 (2006)
7. Quintana, B., Prieto, S.A., Adán, A., et al.: Semantic scan planning for indoor structural elements of buildings[J]. *Adv. Eng. Inform.* **30**(4), 643–659 (2016)
8. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters[J]. *IEEE Trans. Robot.* **23**(1), 34–46 (2007)
9. Kulich, M., Kubalík, J., Přeučil, L.: An integrated approach to goal selection in Mobile robot exploration[J]. *Sensors.* **19**(6), 1400 (2019)
10. Yamauchi, B.: A frontier-based approach for autonomous exploration[C]. Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings 1997 IEEE International Symposium on. IEEE, 146–151 (1997)
11. González-Banos, H.H., Latombe, J.C.: Navigation strategies for exploring indoor environments[J]. *Int. J. Robot. Res.* **21**(10–11), 829–848 (2002)
12. Visser, A., Van Ittersum, M., Jaime, L.A.G., et al.: Beyond Frontier Exploration[C]. Robot Soccer World Cup, pp. 113–123. Springer, Berlin/Heidelberg (2007)
13. Stachniss, C., Grisetti, G., Burgard, W.: Information gain-based exploration using rao-blackwellized particle filters[C]. *Robot. Sci. Syst.* **2**, 65–72 (2005)
14. Vallvé, J., Andrade-Cetto, J.: Potential information fields for mobile robot exploration[J]. *Robot. Auton. Syst.* **69**, 68–79 (2015)
15. Carrillo, H., Dames, P., Kumar, V., et al.: Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi entropy[C]. 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 487–494 (2015)
16. Carrillo, H., Dames, P., Kumar, V., et al.: Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy[J]. *Auton. Robot.* **42**(2), 235–256 (2018)
17. Peng Cheng, S.M. LaValle.: Resolution complete rapidly-exploring random trees[C]// Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on. IEEE (2002)
18. Umari, H., Mukhopadhyay, S.: Autonomous robotic exploration based on multiple rapidly-exploring randomized trees[C]. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 1396–1402 (2017)
19. Tsardoulas, E.G., Iliakopoulou, A., Kargakos, A., et al.: Cost-based target selection techniques towards full space exploration and coverage for USAR applications in a priori unknown environments[J]. *J. Intell. Robot. Syst.* **87**(2), 313–340 (2017)
20. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation[J]. *Artif. Intell.* **99**(1), 21–71 (1998)
21. Howie Choset, J.B.: Sensor based planning, Part I: The generalized Voronoi graph[C]. IEEE International Conference on Robotics & Automation. IEEE (1995)
22. Nagatani, K., Choset, H.: Toward robust sensor-based exploration by constructing reduced generalized Voronoi graph[C]. IEEE/RSJ International Conference on Intelligent Robots & Systems. IEEE (1999)
23. Choset, H.: Sensor-based exploration: the hierarchical generalized Voronoi graph[J]. *Int. J. Robot. Res.* **19**(2), 96–125 (2000)
24. Kim, J., Zhang, F., Egerstedt, M.: A provably complete exploration strategy by constructing Voronoi diagrams[J]. *Auton. Robot.* **29**(3–4), 367–380 (2010)
25. Lau, B., Sprunk, C., Burgard, W.: Improved updating of Euclidean distance maps and Voronoi diagrams[C]. Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 281–286 (2010)
26. Tsardoulas, E.G., Serafi, A.T., Panourgia, M.N., et al.: Construction of minimized topological graphs on occupancy grid maps based on GVD and sensor coverage information[J]. *J. Intell. Robot. Syst.* **75**(3–4), 457–474 (2014)
27. Tsardoulas, E.G., Iliakopoulou, A., Kargakos, A., et al.: A review of global path planning methods for occupancy grid maps regardless of obstacle density[J]. *J. Intell. Robot. Syst.* **84**(1–4), 829–858 (2016)
28. Valero-Gomez, A., Gomez, J.V., Garrido, S., et al.: The path to efficiency: fast marching method for safer, more efficient mobile robot trajectories[J]. *IEEE Robot. Autom. Mag.* **20**(4), 111–120 (2013)
29. Takahashi, O., Schilling, R.: Motion planning in a plane using generalized Voronoi diagrams[J]. *IEEE Trans. Robot. Autom.* **5**(2), 150 (1989)
30. Choset, H., Lynch, K., Hutchinson, S., et al.: Principles of robot motion: theory, algorithms, and implementations. MIT Press, Cambridge, MA[J]. *Proc. Soc. Exper. Biol. Med.* **147**(1), 512–512 (2005)
31. Zhang, T.Y., Suen, C.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. *Commun. ACM.* **27**(3), 236–239 (1984)
32. Saeed, K., Rybník, M., Tabedzki, M.: Implementation and Advanced Results on the Non-interrupted Skeletonization Algorithm[C]. International Conference on Computer Analysis of Images and Patterns, pp. 601–609. Springer, Berlin/Heidelberg (2001)
33. Saeed, K., Tabedzki, M., Rybník, M., et al.: K3M: A universal algorithm for image skeletonization and a review of thinning techniques[J]. *Int. J. Appl. Math. Comput. Sci.* **20**(2), 317–335 (2010)
34. Hershberger, J.E., Snoeyink, J.: Speeding up the Douglas-Peucker Line-Simplification Algorithm[M]. University of British Columbia, Department of Computer Science (1992)

Lin Li has been working as a professor in School of Resource and Environmental Science, Wuhan University. He received the Ph.D. degree from Wuhan University, China, in 1997 and worked for a short time in Joseph Fourier University in France and for about 4 years in the University of Tokyo of Japan. His current research interests focus on 3D modeling and visualization, geographical ontology, 3D cadastre, integration of ubiquitous location information and feature extraction from point cloud data.

Xinkai Zuo received the bachelor's degree from the School of Resource and Environmental Sciences, Wuhan University, where he is currently pursuing the Ph.D. degree of Cartography and Geographical Information Science. His research interests include indoor autonomous robotics, mobile laser mapping, computer vision, and point cloud processing.

Huixiang Peng, senior researcher at the 54th Research Institute of CETC (China Electronics Technology Group Corporation), mainly engaged in remote sensing application, geographic information technology, computer information system development and integration, and won several provincial and ministerial scientific and technological progress awards.

Fan Yang is a Ph.D. student in the School of Resource and Environmental Sciences, Wuhan University, Wuhan, China. He received his bachelor degree in GIS from School of Geography, Nantong University, Nantong, China. He received his master degree in cartography and geographic information system from the College of Geomatics, Xi'an University of Science and Technology, Xi'an, China. His research interests include point cloud processing and 3D indoor mapping.

Haihong Zhu has been working as a professor in School of Resource and Environmental Science, Wuhan University. She received the Ph.D. degree from Wuhan University, China, in 2016. She worked as an associate researcher in University of Tokyo of Japan for about 2 years, and as a visiting scholar in Ohio State University for 1 year. Her current research interests focus on 3D modeling and visualization, geographical ontology, map design, navigation digital map.

Dalin Li received his Ph.D. degree in cartography and geographic information engineering in School of Resource and Environmental Science, Wuhan University, China in 2018. His research interests include machine learning, mobile laser mapping, and point cloud processing.

Jun Liu received his M.S. degree from School of Resource and Environmental Science, Wuhan University in 2018. His research interests include geographical information system, location-based service and SLAM.

Fei Su received the M.S. degree from Yunnan Normal University, Kunming, China, in 2016. And he is currently pursuing the Ph.D. degree with School of Resource and Environmental Sciences, Wuhan University, Wuhan, China. His current research interests include geographic information science, point cloud processing and 3D indoor recognition and reconstruction.

Yifan Liang received the bachelor's degree in surveying and mapping engineering from China University of Mining and Technology, Xuzhou, China, in 2017, where he is currently pursuing the master's degree in Wuhan University. His research interests include indoor mapping and point cloud processing.

Gang Zhou received the bachelor's degree in geographic information science from Wuhan University, Wuhan, China, in 2018, where he is currently pursuing the master's degree in Wuhan University. His research interests include SLAM (simultaneous localization and mapping) and point cloud processing.