

Research  
Robotics—Article

# Enhanced Autonomous Exploration and Mapping of an Unknown Environment with the Fusion of Dual RGB-D Sensors



Ningbo Yu <sup>a,b,c,\*</sup>, Shirong Wang <sup>a,b</sup>

<sup>a</sup> Institute of Robotics and Automatic Information Systems, Nankai University, Tianjin 300353, China

<sup>b</sup> Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300353, China

<sup>c</sup> State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

## ARTICLE INFO

### Article history:

Received 4 February 2018

Revised 25 June 2018

Accepted 8 November 2018

Available online 27 December 2018

### Keywords:

Autonomous exploration

Red/green/blue-depth

Sensor fusion

Point cloud

Partial map simulation

Global frontier search

## ABSTRACT

The autonomous exploration and mapping of an unknown environment is useful in a wide range of applications and thus holds great significance. Existing methods mostly use range sensors to generate two-dimensional (2D) grid maps. Red/green/blue-depth (RGB-D) sensors provide both color and depth information on the environment, thereby enabling the generation of a three-dimensional (3D) point cloud map that is intuitive for human perception. In this paper, we present a systematic approach with dual RGB-D sensors to achieve the autonomous exploration and mapping of an unknown indoor environment. With the synchronized and processed RGB-D data, location points were generated and a 3D point cloud map and 2D grid map were incrementally built. Next, the exploration was modeled as a partially observable Markov decision process. Partial map simulation and global frontier search methods were combined for autonomous exploration, and dynamic action constraints were utilized in motion control. In this way, the local optimum can be avoided and the exploration efficacy can be ensured. Experiments with single connected and multi-branched regions demonstrated the high robustness, efficiency, and superiority of the developed system and methods.

© 2019 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Autonomous robots can acquire information on the environment and assist humans in various circumstances and applications, such as rescuing humans from danger, providing service within urban or home environments, and guiding or otherwise aiding people in need [1–6]. Despite the dramatic development of relative technologies and algorithms, major challenges remain. The complexity and variability of the unknown environment make it difficult for human operators to provide prior information to a robot [7]. Thus, it is of great significance to equip a robot with the capabilities of autonomous exploration and mapping, including incremental map construction, localization, path planning, motion control, navigation, and so on, without direct human intervention.

Heuristic and reactive exploration approaches have been studied in the literature. In Ref. [8], Yamauchi proposes a frontier-based heuristic exploration algorithm. In Ref. [9], based on the angular uncertainty of sonar sensor, a heuristic exploration

strategy is proposed that uses a sonar sensor array for perception and mapping. In Ref. [10], a local map-based frontier graph search exploration method is developed, and the environment is represented by a tree structure, making it efficient to determine the next target position even within a large environment. In Ref. [11], Keidar and Kaminka describe a morphological frontier detection method using light detection and ranging (LIDAR) data to accelerate the map frontier point detection process. In Ref. [12], a reactive exploration strategy is proposed that uses current velocity and bearing for rapid frontier selection.

The simulation-based autonomous exploration strategy has been attracting increasing research effort, since it can assist in measuring information and generating the goal pose based on a current robot's status during exploration. In Ref. [13], Carrillo et al. propose a utility function with Shannon and Rényi entropy in order to measure the robot's actions for exploration. In Ref. [14], Lauri and Ritala present a forward simulation algorithm, and formulate the exploration to a partially observable Markov decision process (POMDP). In Ref. [1], with partially known environment information, a red/green/blue-depth (RGB-D) sensor is used for loop closure detection in an autonomous mapping and exploration process. In Ref. [15], Bai et al. propose a Gaussian

\* Corresponding author.

E-mail address: [nyu@nankai.edu.cn](mailto:nyu@nankai.edu.cn) (N. Yu).

process regression-based exploration strategy, and predict mutual information with robot motion samples. In Ref. [16], the autonomous exploration problem is formulated as a partial differential equation; the information is described as a scalar field from the prior known area to the unknown area.

In existing exploration methods, range sensors are mostly used and two-dimensional (2D) grid maps are generated. However, a 2D grid map only contains planar geometric information, and is often insufficient for human perception [17–19]. RGB-D sensors directly provide both color and depth information, and can help to generate a three-dimensional (3D) point cloud map. RGB-D sensor-based simultaneous localization and mapping (SLAM) systems were first proposed independently by Henry et al. [20] and Engelhard et al. [21]. Image features were used to find matches among each frame, and an iterative closest point (ICP) algorithm was applied to estimate the point cloud transformations in the RGB-D SLAM systems. In Ref. [22], Klein and Murray present the parallel tracking and mapping (PTAM) framework and implement tracking and mapping in dual parallel threads. Labbé and Michaud [23] propose the real-time appearance-based mapping (RTABMAP) algorithm, in which loop closure detection thread and a memory management method are added. Online incremental mapping and loop closure detection can be attained, and the mapping efficiency and accuracy can remain consistent over time.

The field of view (FoV) of existing RGB-D sensors is limited [24]. To obtain a greater scope of environment information, multiple RGB-D sensors can be used together. Dual RGB-D sensors have been utilized in visual SLAM for robust pose tracking and mapping [25], and an RGB-D sensor has been fused with an inertial measurement unit (IMU) for indoor wide-range environment mapping [26]. In Ref. [27], Munaro et al. present an RGB-D sensor network, in which multiple RGB-D sensors are deployed in an indoor environment for human tracking. In Ref. [28], multiple RGB-D sensors with non-overlapping FoVs are employed for line detection and tracking, thus providing a 3D surrounding view and morphological information on the environment.

In our previous work, we developed RGB-D-based localization and motion planning and control methods [24], and realized RGB-D-based exploration [17]. Nevertheless, the robustness and efficiency of the exploration process was limited due to the small FoV of the RGB-D sensor. In this work, we used dual RGB-D sensors for the mobile robot. The deployment of dual RGB-D sensors provided a larger FoV, but also introduced technical challenges in interference, data synchronization, and processing. With the synchronized data from the dual RGB-D sensors, location points were generated and a 3D point cloud map and 2D grid map were incrementally constructed. An autonomous exploration strategy was established by combining partial map simulation with global frontier search methods. In this way, the local optimum can be avoided and the exploration efficacy can be ensured. The experimental results demonstrated the high robustness and efficiency of the developed system and methods.

This paper is organized as follows: Section 2 presents the 3D and 2D mapping method with dual RGB-D sensors, and Section 3 describes the autonomous exploration algorithm in detail. Section 4 describes the mobile robot system that validates the proposed mapping and exploration method. Section 5 presents the experiments and results. Finally, Section 6 concludes the paper.

## 2. Mapping with dual RGB-D sensors

With the information provided by dual RGB-D sensors, a 3D point cloud map can be constructed in real time. This process includes location point creation, mapping, and loop closure detection.

### 2.1. Location point generation

Fig. 1 presents the process of creating location points. In order to construct a 3D point cloud map, the sensor data must be processed; this includes the RGB images and depth images from the dual RGB-D sensors, and the wheel odometry data from the mobile chassis. The data should then be synchronized. The pose between the dual RGB-D sensors is known, as it can be determined by calibration after system assembly.

Past information is not included in the current frame of the information queue, including RGB images, depth images from the dual RGB-D sensors, and the wheel odometry information. Fig. 2 depicts the synchronization implemented with the robot operating system (ROS) message filter method.

The synchronized sensor data are processed to create location points. Dual RGB-D color and depth data are simultaneously combined into an integrated data structure that comprises the current sensor data, which can be used for the location point creation process. The oriented fast and rotated brief (ORB) feature points and corresponding descriptors are obtained from the RGB images. The 3D feature points in the robot frame are integrated with the 2D feature points and with corresponding depth data from the depth images, as the pose between the dual RGB-D cameras is known. The fundamental matrix based on the epipolar constraint is calculated in order to determine the inliers, using matched 3D feature points from the last frame and the current frame. With enough inliers, data from the wheel odometry is extracted to provide an initial guess of the transformation between the last frame and the current frame.

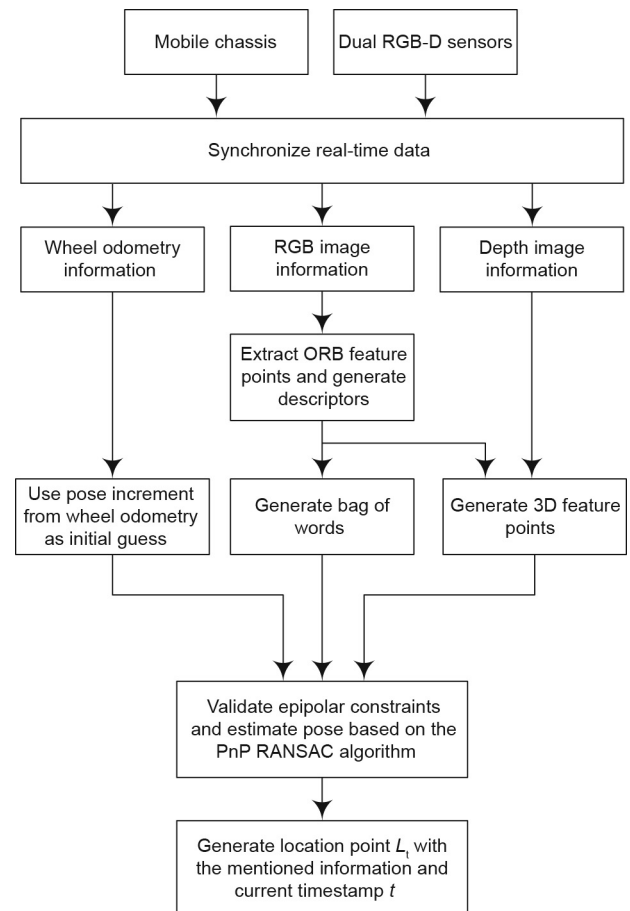


Fig. 1. The process of location points creation.

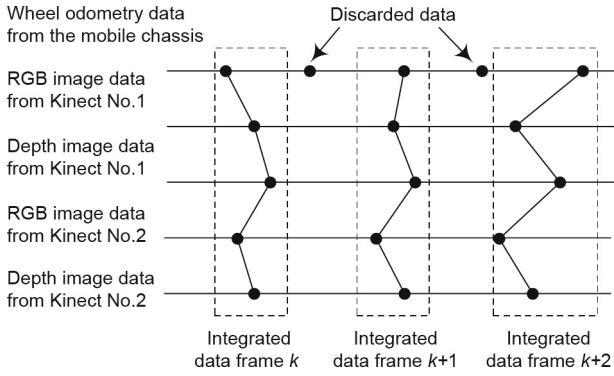


Fig. 2. Sensor data synchronization.

The transformation between the frames is estimated by the random sample consensus (RANSAC) algorithm based on the perspective- $n$ -points (PnP) model, using matched 3D feature points. The bag of words (BoW) of a frame is created by corresponding ORB descriptors, and the environment vocabulary is incrementally constructed with the BoW online, rather than pre-training in the target environment, as doing so is more suitable for autonomous exploration tasks. The location point,  $L_t$ , is created with the above information and the current timestamp,  $t$ , and the edge between the last and current location point is initialized, with its weight set to zero.

Depth errors and feature mismatch errors may exist in an unknown environment, especially in a place with few features and varying illumination. A sudden change in RGB-D localization may occur, which can detrimentally impact the real-time motion control of the mobile robot. Furthermore, although wheel odometry changes smoothly, it contains accumulated error. However, the RGB-D localization and wheel odometry can be fused by means of an extended Kalman filter in order to achieve robust localization.

## 2.2. Point cloud mapping and loop closure detection

The mapping and the loop closure detection process are tightly coupled using graph optimization and memory management methods to ensure stability and real-time performance. The similarity and pose of the location points are optimized in the tree-based network optimizer (TORO) framework [29] to ensure global consistency.

The location points are stored in the working memory (WM) or long-term memory (LTM), as determined by the similarity, as shown in Fig. 3. The location points in the WM are used for real-time loop closure detection, while other location points are stored in the LTM as candidates. The candidates will either be retrieved to the WM or deleted, according to their similarity and storage time in the LTM.

First,  $L_t$  is set as a new vertex of the graph and the similarity of  $L_t$  and the nearest  $M$  location points is calculated by their BoWs, as shown in Eq. (1), where  $K_{\text{matched}}$  represents the matched BoW number between the BoW of  $L_t$  and a BoW from one of the nearest  $N$  location points,  $L_c$ .  $K_t$  and  $K_c$  are the BoW numbers of  $L_t$  and  $L_c$ , respectively. The similarity of  $L_t$  and  $L_c$  is marked as  $\lambda$ . When  $\lambda$  is greater than the threshold,  $\varepsilon$ ,  $L_c$  is merged into  $L_t$  and  $L_t$  is loaded into WM.

$$\lambda(L_t, L_c) = \begin{cases} \frac{K_{\text{matched}}}{K_t}, & K_t > K_c \\ \frac{K_{\text{matched}}}{K_c}, & K_t \leq K_c \end{cases} \quad (1)$$

The conditional probabilities of the loop closure among  $L_t$  and other location points on the graph are updated with  $\lambda$ . The weight of the graph edges among  $L_t$  and the other location points are cor-

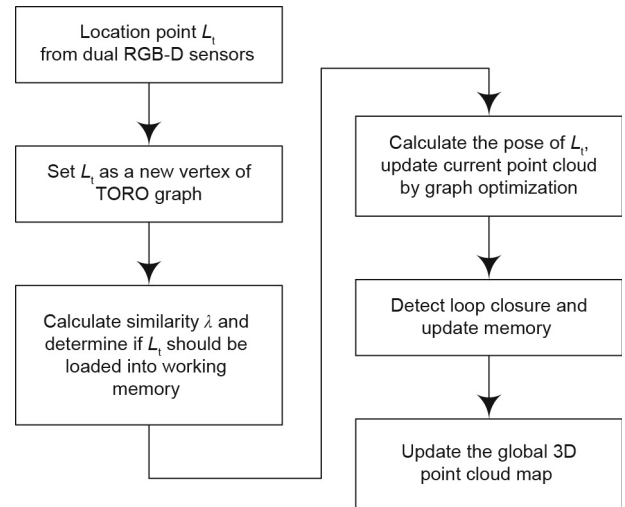


Fig. 3. The process of mapping and loop closure detection.

rected by TORO, including the location points in the WM and LTM. The transformations among  $L_t$  and the location points in the WM are calculated by the ICP algorithm, with an initial guess from the wheel odometry. Loop closure is established if the variation is smaller than the given threshold. With the estimated current transformation, the weights of the edges are updated and the poses of other location points are corrected to ensure the global consistency of the pose from each location point. The location point in the WM with the lowest weight and longest residence time will be transferred to the LTM, and the location point in the LTM with the greatest weight will be retrieved to the WM. The global 3D point cloud map is updated with the corrected information from the graph.

Fig. 4 shows the 3D point cloud segmentation and 2D grid map construction process with dual RGB-D sensors. The point cloud map is voxelized and down-sampled with the voxel filter in the Point Cloud Library (the voxel size is set as 0.05 m). A point set is selected and its normal vector is estimated by the  $k$  nearest neighbor algorithm. The point set size  $k$  is set as 10, which reconciles the computing efficiency with the precision. The normal vector is estimated as shown in Eq. (2).

$$C = \frac{1}{k} \sum_{i=1}^k (p_i - \bar{p})(p_i - \bar{p})^T, C \in R^{3 \times 3} \quad (2)$$

$$C \cdot \mathbf{v}_d = \lambda_d \cdot \mathbf{v}_d, d \in \{1, 2, 3\}, \mathbf{v}_d \in R^{3 \times 1}, \lambda_d \in R^+$$

$$\mathbf{v}_{\text{normal}} = \mathbf{v}_n, \text{ where } \lambda_n = \max\{\lambda_1, \lambda_2, \lambda_3\}$$

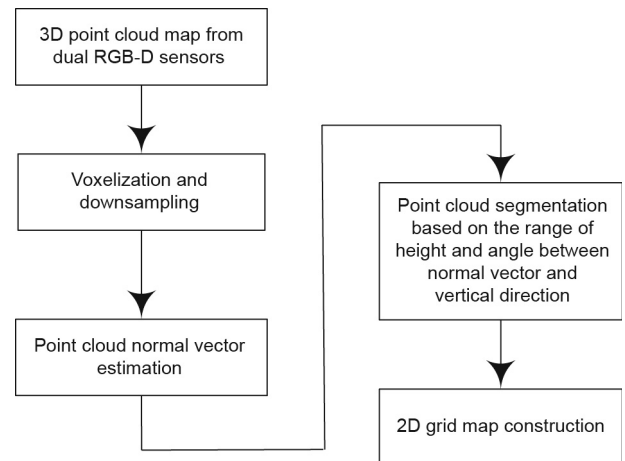


Fig. 4. The process of 3D point cloud segmentation and 2D grid map construction.

Here,  $p_i$  is the coordinate of each point and  $\bar{p}$  is the center coordinate of the corresponding point set.  $C$  is the covariance matrix, and  $\mathbf{v}_d$  and  $\lambda_d$  are the eigenvector and eigenvalue of  $C$ , respectively. The eigenvector with the biggest eigenvalue,  $\mathbf{v}_n$  is selected as the normal vector of the point set,  $\mathbf{v}_{\text{normal}}$  [30]. The point sets in the point cloud can be determined and segmented with the above information. If the point set has a normal vector that deviates from the vertical direction by more than a given threshold, it is set as an obstacle. Otherwise, it is possible to move to the next step. The point set is then processed with the Euclidean clustering extraction method. If the height of its center is greater than a given threshold, the point set will be set as an obstacle. Otherwise, it is taken as free space. The grid map is constructed with the  $(x, y)$  coordinates from the obstacle portion of the point cloud.

A robot may pass by a region with few visual features or with features that are similar to the surroundings, which leads to the failure of loop closure detection [31]. In such circumstances, the mobile robot can execute autonomous motion in the exploration process; this changes the FoV of the dual RGB-D sensors in order to obtain more environment information.

### 3. Autonomous exploration

The real-time updated 2D grid map and localization are set as prior information for the autonomous exploration. The updated map and odometry information are used for the next goal pose generation, so as to expand the map and further explore the environment.

#### 3.1. Autonomous exploration based on the POMDP model

Assuming that the environment is partially known and contains stochastic information, the mobile robot exploration process can be described by a POMDP model  $\langle \tau, \mathcal{S}, \mathcal{U}, \mathcal{Z}, \mathbb{T}_S, \mathbb{O}, \mathcal{R} \rangle$ .

- $\tau = \{0, 1, \dots, k, \dots\}$  represents the decision epoch of the goal generation in the exploration task.
- $\mathcal{S} = \mathcal{X} \times \mathcal{M} = \{[X_0, M_0], [X_1, M_1], \dots, [X_k, M_k], \dots\}$  represents the state space of the exploration task, including the mobile robot state  $\mathcal{X}(X_k = [x_k, y_k, \theta_k]^T)$  and the map state  $\mathcal{M}$ .  $\mathcal{X}$  and  $\mathcal{M}$  are both observable. (The errors from applying a given motion to the robot and from applying the localization data are not considered, so  $\mathcal{X}$  is completely observable and  $\mathcal{M}$  is partially observable.)
- $\mathcal{U} = \{U_0, U_1, \dots, U_k, \dots\}(U_k = [v_k, \omega_k]^T)$  represents the feasible motion according to the robot motion constraint.
- $\mathcal{Z}$  represents the observation space.
- $\mathbb{T}_S = \mathbb{T}_X \times \mathbb{T}_M$  represents the state transfer model, which consists of  $\mathbb{T}_X$  and  $\mathbb{T}_M$ .  $\mathbb{T}_X: \mathcal{X} \times \mathcal{U} \times \mathcal{X} \rightarrow \mathbb{R}^+$  describes a state transfer  $T_k^{k+1}$  from the current mobile robot state  $X_k$  and motion  $U_k$  to the next state,  $X_{k+1}$ .  $\mathbb{T}_M: \mathcal{M} \times \mathcal{U} \times \mathcal{M} \rightarrow [0, 1], [0, 1]$  represents the scope of occupancy grid probabilities.  $\mathbb{T}_M$  is independent of  $\mathcal{U}$ , which means that  $\mathbb{T}_M(M_k, U_i, M_{k+1}) = \mathbb{T}_M(M_k, U_j, M_{k+1})$  for any  $U_i, U_j \in \mathcal{U}$ .
- $\mathbb{O}: \mathcal{Z} \times \mathcal{X} \times \mathcal{M} \times \mathcal{U} \rightarrow [0, 1]$  represents the observation model, as shown in Fig. 5. According to the current mobile robot pose and the 2D grid map, an observation value of the current state is taken by ray tracing the model.
- $\mathcal{R}: \mathcal{B} \times \mathcal{X} \times \mathcal{M} \times \mathcal{U} \rightarrow \mathbb{R}^+$  represents the reward function constructed by the state, observation, and action, as measured by the belief state value. Mutual information function is chosen as the form of the reward function.

Based on the 2D grid map and the robot pose, a series of mobile robot motions are generated for trajectory simulation, from which the current goal pose is selected. The motions are evaluated by the reward function, with the target being to maximize the reward

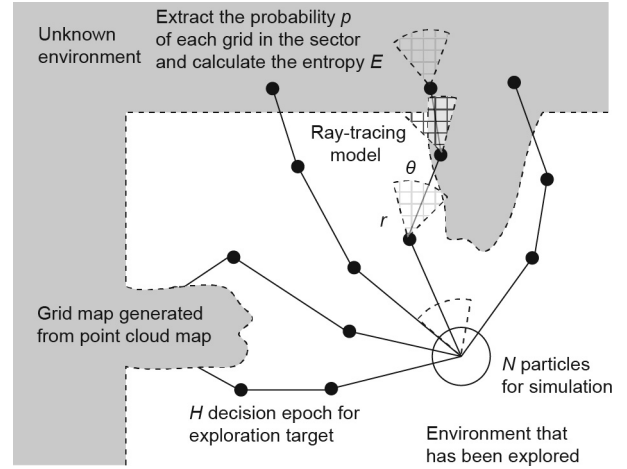


Fig. 5. The exploration process based on the POMDP model.

function in the corresponding decision epoch. In the POMDP model-based exploration process, the reward function is defined as the mutual information function, as in Eq. (3).  $R_k$  represents the reward function value in decision epoch  $k$ , which is determined by the states of  $N$  particles. Particles are simulated from the belief state  $B_k$  and  $(U_k)_N$ , with the robot state  $(X_{k+1})_N$ , and with the observation  $(Z_{k+1})_N$  of the map state  $M_{k+1}$ .

$$R_k = \frac{1}{N} \sum_{n=1}^N \left[ \log \left( \frac{p(X_{n,k+1}|B_k, U_k, Z_{n,k+1})}{p(X_{n,k+1}|B_k, U_k)} \right) + \int_{\mathcal{M}} p(M_{n,k+1}|B_k, U_k, Z_{n,k+1}, X_{n,k+1}) \cdot \log \left( \frac{p(M_{n,k+1}|B_k, U_k, Z_{n,k+1}, X_{n,k+1})}{p(M_{n,k+1}|B_k, U_k)} \right) \right] \quad (3)$$

The first term of Eq. (3) is the expected information gain of the mobile robot state. Considering that the experimental environment is a flat terrain on which the given motion can be executed correctly, we assume that there is no noise in the mobile robot motion model  $X_{k+1} = f(X_k, U_k)$  [32], as shown in Eq. (4). This assumption aligns with the decision to choose the particle state with the maximum likelihood value  $X_{n_m,k+1}$  as the expected result. Therefore, the mutual information gain described by the robot state and observation is set as 0.

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} -\frac{v_k}{\omega_k} \sin \theta_k + \frac{v_k}{\omega_k} \sin(\theta_k + \omega_k \Delta t) \\ \frac{v_k}{\omega_k} \cos \theta_k - \frac{v_k}{\omega_k} \cos(\theta_k + \omega_k \Delta t) \\ \omega_k \Delta t \end{bmatrix} \quad (4)$$

The second term of Eq. (4) is the expected information gain of the map state, which is represented by the mutual information gain of the map state and the observation value. The integral is implemented by summing up the entropy of the selected grids during the simulation. The probability range of the grid is  $[0, 1]$ : 0 for the free state and 1 for the occupied state. The entropy of the binary random variable is defined in Eq. (5).

$$E(p) = -p \log(p) - (1-p) \log(1-p) \quad (5)$$

The entropy summing scope is set based on the ray-tracing observation model of the sensor. The model is described as follows: Define a sector with  $(x_k, y_k)$  as the origin, the maximum observation range  $r$  as the radius, and the observation angle  $\theta$  as the center angle. Select the probability value of the grids in the ray-tracing beam direction of the sector and calculate the corresponding entropy value. For grids with an unknown probability (which means that the grid is in the unknown environment), set the entropy value as 1.



### 3.2. Partial map simulation and global frontier search

The mobile robot autonomous exploration process is described by the POMDP model, in which the goal pose of each epoch is generated by the combined strategies of partial map simulation and global frontier search, as shown in Fig. 6. A goal pose  $X_g$  and its reward function value  $R_g$  is generated by the partial map simulation process with the current 2D grid map and robot pose. This process is listed as Procedure 1, and contains particle sampling, weight calculation, and resampling. The parameters include the iteration number  $I$ , particle number  $N$ , sampling number  $\{\phi_i\}_{i=1}^I$ , sample distribution kernel function  $\zeta_i$ , ray-tracing model  $\chi$ , simulation stage  $H$ , resampling lower bound  $\epsilon_N$ , and discount factor  $\gamma$ .

The partial map simulation process based on the POMDP model is presented in Procedure 1. First, the algorithm iterates over  $i = 1, \dots, I$  and simulates particles over  $n = 1, \dots, N$ . Based on the iteration number, a kernel function  $\zeta_i$  is chosen for particle sampling, generating the motion series  $U_{0:H-1,i}^{(n)}$ . For uniformity of the initial sampling and a better solution for subsequent sampling,  $\zeta_1$  is set as a uniform distribution and  $\zeta_i (i \geq 2)$  is set as a normal distribution with the mean from  $U_{0:H-1,i}^{(n)}$  by initial sampling  $\zeta_1$ . The range of the uniform distribution  $U_{range}$  and the covariance matrix of the normal distribution  $\Sigma$  are set based on the velocity range of the mobile chassis. Current particle poses are updated based on the mobile robot motion model and the generated motions. The map observation is updated from the ray-tracing model and given an observation range, and the reward function value of each particle is also updated. The sampling number  $\{\phi_i\}_{i=1}^I$  is an increasing linear sequence of  $i(\phi_i = ai + b)$ , and is a faster way to converge to the optimal solution of the probability

meaning. The weights of the particles are updated after the simulation stage has occurred  $H$  times. The prior and posterior information are evaluated with observation  $z_{1:H,j}^{(n)}$ , based on ray-tracing model  $\chi$  and the inverse sensor model. In practical applications, the reward function is calculated using prior and posterior information. The log odds of the prior and posterior probability are calculated, with random value  $l_{ISM}$  as the noise of the inverse sensor model. The discount factor  $\gamma$  indicates the relationship between the simulated information and the current information. For example,  $\gamma > 1$  means that further simulated information takes on more weight. If the weights of the particles are overly dispersed,  $1/\sum_{n=1}^N [w_i^{(n)}]^2 < \epsilon_N$ , then the particles should be resampled and new particle poses and map observations should be generated. The goal pose and corresponding reward function value are simulated from the motion series with maximum weight.

The optimization time and simulation range must be limited. An optimal solution of the probability meaning can be reached for the partial map simulation process by means of the Monte Carlo method over a limited simulation range. The mobile robot may continue to pace around, or may get stuck when entering the region with the local information optimum. Thus, a global frontier search strategy is introduced in order to avoid a local optimum.

**Procedure 1.** The partial map simulation process based on the POMDP model.

**Input:**  $X_c, M_c, I, N, \{\phi_i\}_{i=1}^I, \zeta_i, H, \epsilon_N, \gamma$

**Output:**  $X_g, R_g$

```

1:  $X_0 \leftarrow X_c$ 
2: for  $i = 1, \dots, I$  do
3:   for  $n = 1, \dots, N$  do
4:     if  $i = 1$  then
5:        $U_{0:H-1,1}^{(n)} \sim \zeta_1(U_{range})$ 
6:     else
7:        $U_{0:H-1,i}^{(n)} \sim \zeta_i(U_{0:H-1,i-1}^{(n)}, \Sigma)$ 
8:     end if
9:      $X_{1:H,i}^{(n)} \leftarrow f(X_0, U_{0:H-1,i}^{(n)})$ 
10:    for  $j = 1, \dots, \phi_i$  do
11:      for  $k = 0, \dots, H - 1$  do
12:         $l_{prior} = l[z_{k,j}^{(n)}]$ 
13:         $z_{k,j}^{(n)} \sim \chi[z_k | S_k^{(n)}]$ 
14:         $l_{ISM} = \text{rand}[l(p)]$ 
15:         $l_{posterior} = l[z_{k,j}^{(n)}] + l_{ISM}$ 
16:         $R_{0:H-1,j}^{(n)} = E(l_{prior}) - E(l_{posterior})$ 
17:      end for
18:    end for
19:     $w_i^{(n)} \leftarrow w_{i-1}^{(n)} \prod_{j=1}^{\phi_i} \sum_{t=0}^{H-1} [\gamma^{t-H} R_{t,j}^{(n)}]$ 
20:  end for
21:  for  $n = 1, \dots, N$  do
22:     $w_i^{(n)} \leftarrow \frac{w_i^{(n)}}{\sum_{n=1}^N w_i^{(n)}}$ 
23:  end for
24:  if  $\frac{1}{\sum_{n=1}^N [w_i^{(n)}]^2} < \epsilon_N$  then
25:    resample particles,  $w_i^{(n)} \leftarrow \frac{1}{N}$ 
26:  end if
27: end for
28:   $n_m \leftarrow \text{argmax}_n [w_i^{(n)}]$ 
29:   $X_g \leftarrow f[X_0, U_{0:H-1,i}^{(n_m)}]$ 

```

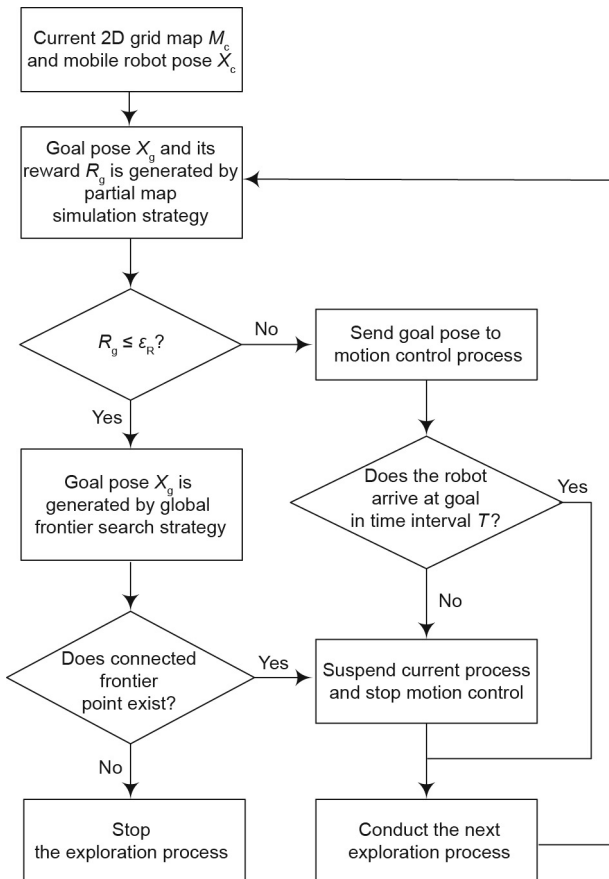


Fig. 6. Diagram of the exploration strategy.

Frontiers between the known region (which is constructed by the exploration process) and the unknown region are detected by means of a global frontier search strategy based on a breadth-first search algorithm. The nearest connected frontier point to the current mobile robot pose is selected, and the connected point is defined as a point with eight existing grids around it. The exploration time and region cannot be ensured, as the observable unknown environment near frontier points cannot be determined by such a heuristic search strategy. The global frontier search strategy can avoid a local optimum using the current known map. Information on the unknown environment can be measured by means of the partial map simulation strategy, thus ensuring new information is obtained within each exploration epoch. The combined strategy is shown in Fig. 6. The goal pose  $X_g$  and its reward function value  $R_g$  are generated by the partial map simulation. If  $R_g > \epsilon_R$ , the goal pose can be sent to the motion control process. Otherwise, the global frontier search strategy is triggered for a new goal pose. The new exploration process begins when the mobile robot arrives at the goal pose within a given time; alternatively, the current exploration and the motion control process will be suspended if the time exceeds the limit.

The lower bounds of the reward, iteration number, and particle number mainly affect the efficiency and robustness of the exploration process. The lower bound ensures the efficiency of the partial map simulation, and a local optimum can be avoided with a global frontier search. More information can be obtained when using increased iterations and particles, but the efficiency may decrease with the resulting heavier computation.

## 4. System implementation

### 4.1. System implementation with dual RGB-D sensors

A mobile robot system was constructed to validate the proposed autonomous exploration and mapping method. The system consisted of a Yujin Kobuki mobile chassis, dual Kinect RGB-D sensors, and an HP Envy 15-j105tx mobile computing unit. The environment color and depth data obtained by the Kinect were used in the method. The resolution of the RGB and depth images was  $640 \times 480$  pixels, with a frequency of 30 Hz. The FoV was  $57^\circ$  horizontal degrees by  $43^\circ$  vertical degrees, and the depth range in the indoor environment was 0.5–5 m [33]. The Yujin Kobuki mobile chassis was differentially driven by the two wheels, with a diameter of 0.46 m, a maximal linear speed of  $70 \text{ cm}\cdot\text{s}^{-1}$ , and an angular speed of  $180^\circ\cdot\text{s}^{-1}$ . The frequency of the wheel odometry had a maximum of 50 Hz. The mobile computing unit was equipped with a CPU i7 4702 M and 8 GB RAM. ROS Indigo, OpenNI RGB-D driver, the Point Cloud Library, and OpenCV were also installed on the mobile computing platform.

Kinect obtains depth data by means of structured light measurement, mainly by projecting an infrared speckle to the measured object and capturing the depth information through the complementary metal-oxide semiconductor (CMOS) infrared sensor. An interference problem occurs in situations when multiple Kinect FoVs overlap [34]. To avoid structured light interference among the sensors and to expand the FoV of the system, dual Kinect RGB-D sensors were installed, as shown in Fig. 7; their pose relative to the mobile chassis is shown in Eq. (6).  $[x, y, z]^T$  and  $[\theta, \phi, \psi]^T$  are the translation and ZYX Euler angles of the Kinect frame relative to the mobile chassis. The transformation among the dual Kinect RGB-D sensors and the mobile chassis is given by Eq. (6).

$$\begin{aligned} [x_1, y_1, z_1, \theta_1, \phi_1, \psi_1]^T &= [-0.118, 0.0, 0.271, 0.0, 0.0, 0.0]^T \\ [x_2, y_2, z_2, \theta_2, \phi_2, \psi_2]^T &= [-0.118, 0.0, 0.481, -90^\circ, 0.0, 0.0]^T \end{aligned} \quad (6)$$

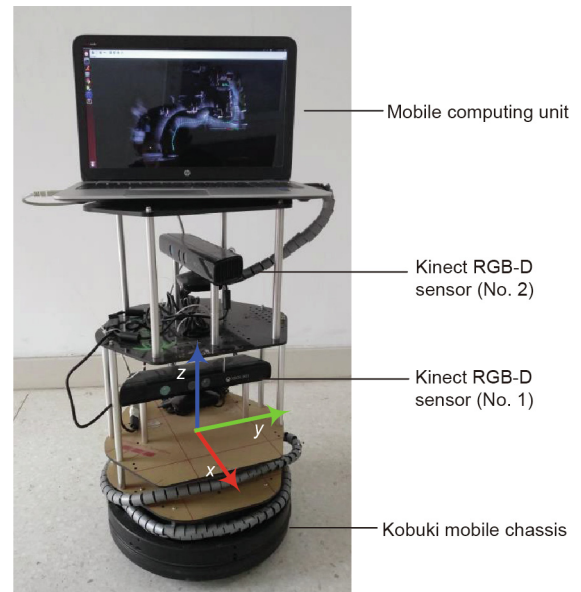


Fig. 7. The mobile robot hardware platform.

### 4.2. System workflow

Fig. 8 shows the mobile robot autonomous exploration and mapping workflow.

(1) **Mapping and localization.** According to the environment RGB-D information that was obtained by the dual Kinect RGB-D sensors and the wheel odometry from the mobile chassis, the RTABMAP algorithm was adopted to estimate localization information. The pose was fused with the RGB-D localization and wheel odometry based on an extended Kalman filter (EKF, integrated in the ROS robot\_pose\_ekf package), and a 3D point cloud map was constructed. A 2D grid map was generated from the 3D point cloud for autonomous exploration and motion control.

(2) **Combined strategy of partial map simulation and global frontier search.** Based on the forward simulation algorithm, the goal pose  $[x_g, y_g, \theta_g]^T$  of the exploration process was generated using the 2D grid map and the current pose  $[x_c, y_c, \theta_c]^T$ , with the simulation strategy using partial information from the grid map, and the search strategy using global frontier points from the grid map.

(3) **Dynamic window motion control based on motion constraints.** According to the robot's current pose  $[x_c, y_c, \theta_c]^T$ , a 2D grid map  $M_c$ , and a goal pose  $[x_g, y_g, \theta_g]^T$  from the exploration strategy, the robot's motion control command was generated. To ensure the stability of the mapping process, cost function was added with the motion constraint, based on a dynamic window control algorithm [35].

In the exploration algorithm, the distance between an available goal and the current position was set to be between 0.5 and 10 m, the maximum speed of the robot was  $0.5 \text{ m}\cdot\text{s}^{-1}$ , and the cycle time was below 1.0 s. During the exploration process toward the next goal, the point cloud map, grid map, and pose were updated to provide environment information. The point cloud and grid map update rate was 1 Hz, and the pose update rate could be increased up to 20 Hz. Thus, these parameters were able to meet the demands of the exploration process.

## 5. Experiments and results

### 5.1. Experimental settings

The autonomous exploration and mapping environment settings are shown in Fig. 9(a). The mobile robot set off in the starting

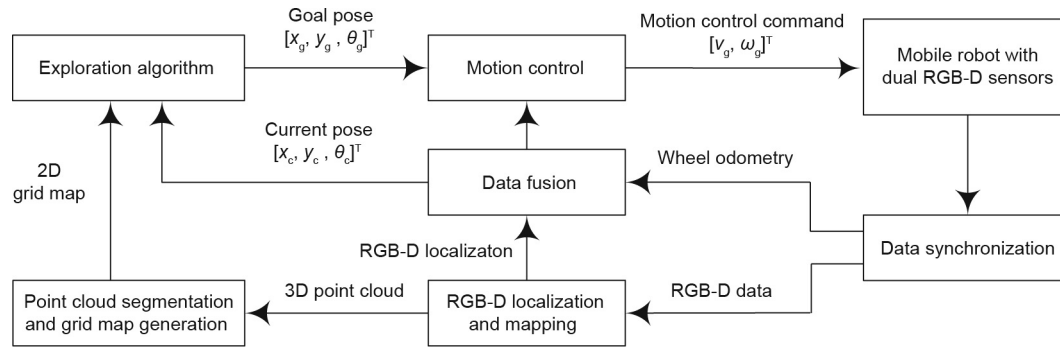


Fig. 8. The workflow of the mobile robot for autonomous exploration.

region and executed the autonomous exploration and mapping task. The task was finished when the robot arrived at the target region. Eight kinds of indoor scenes were tested, as shown in the figure. Scene (i) was a rectangular area with obstacles on three sides and only one exit. Scenes (ii–iv) were aisles with widths of 1.43 and 0.85 m, respectively ((iii), (iv) for the entrance and exit). Scenes (v) and (vii) were regions with various obstacles and aisle widths of 0.90 and 0.78 m, respectively. Scenes (vi) and (viii) were different perspectives of the open area.

According to the mobile robot exploration process in the laboratory, the scenes were divided into two types: single connected regions and multi-branch regions. Scenes (i–iv) were single connected regions, with only one aisle connecting scene (i) to scene (iv). The mobile robot made a  $90^\circ$  right turn from scene (i) to scene (ii), and then a  $90^\circ$  right turn to scenes (iii) and (iv), in which the view of the sensors was restricted by the baffle plates and the width of the aisle. The height of the baffle plates was 1.5 m, which was greater than the vertical range of the RGB-D sensors. Scenes (v–viii) were multi-branch regions; the order in which the mobile robot explored these scenes was not unique due to the stochastic

nature of the exploration algorithm. Thus, the experimental task was activated from the start region of the indoor environment and was stopped after entering the target region. The starting region was a rectangular area in the aisle, with a length of 1.18 m from the edge of the cabinet on the right side to the rear obstacle, and a width of 1.12 m. The target region was a rectangular area in scenes (vi) and (viii) bounded by the door and the wall, with a length of 2.68 m and a width of 2.44 m.

In the single connected regions, the FoV was limited, regardless of whether single or dual RGB-D sensors were deployed. In the multi-branch regions, the scene was wider, and the dual RGB-D sensors allowed more information to be detected and fed into the system. Thus, the experiment was able to demonstrate the efficacy of utilizing dual RGB-D sensors by comparing the experimental performances in the single connected and multi-branch regions.

## 5.2. Results

Combining the methods of partial map simulation with a global frontier search ensures that the simulated goal pose is the

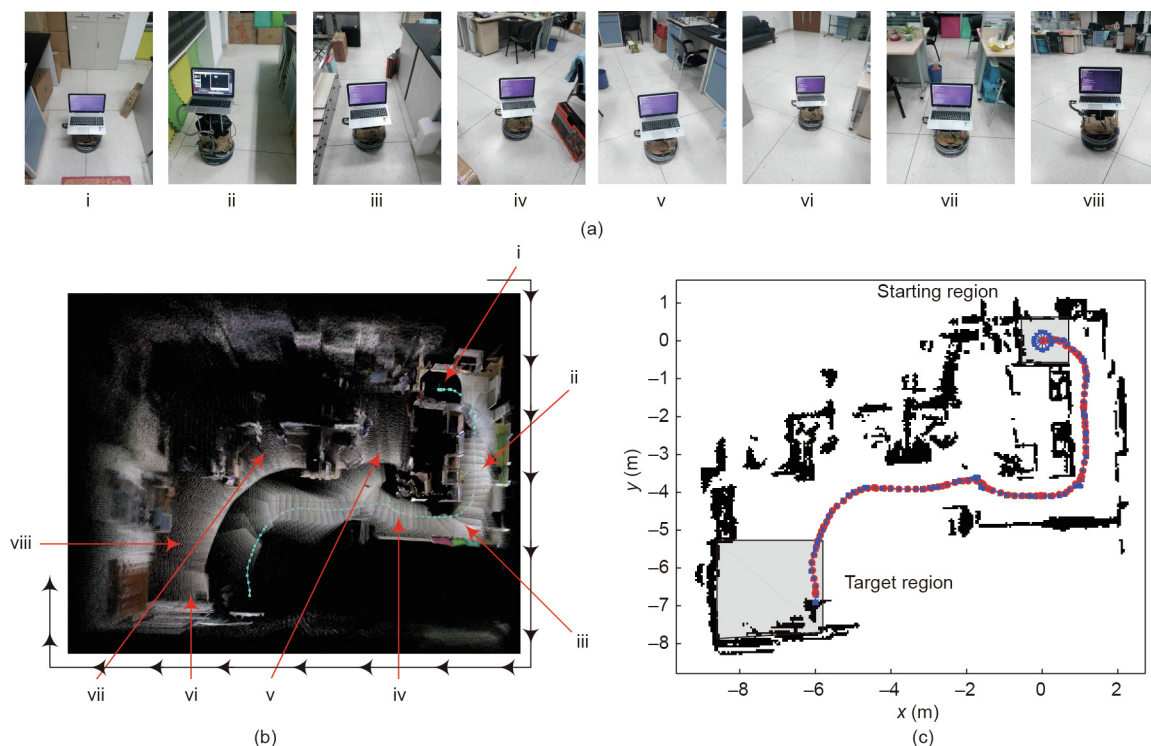


Fig. 9. The autonomous exploration and mapping experiment. (a) The experiment environment from scene (i) to (viii). (b) The constructed 3D point cloud map and (c) generated 2D grid map and mobile robot trajectory from the autonomous mapping and exploration process.

optimum in the observation scope, and that a local optimum can be avoided by searching the map frontier. Thus, the exploration task was successfully accomplished.

Fig. 9(b) depicts the 3D point cloud map constructed from the mobile robot autonomous mapping and exploration experiment. This laboratory 3D point cloud map was constructed using the mobile robot odometry and the information from the dual RGB-D sensors; it includes the narrow aisles, obstacles, and open area. The constructed 3D point cloud map corresponded well with the actual indoor environment; examples of good correspondence included the rear box in scene (i), the color blocks near the wall in scene (ii), the obstacles in scenes (v) and (vii), the door in scene (vi), and the open area in scenes vi and viii. The 3D point cloud map was incrementally constructed during the autonomous exploration process, thus preserving spatial consistency with effective stitching.

Fig. 9(c) depicts the 2D grid map that was generated from the 3D point cloud map and the mobile robot trajectory. In this 2D grid map, the length of the x axis is 11.40 m, and the length of the y axis is 9.40 m. The actual dimensions of the laboratory were 11.04 m by 7.93 m. The dimensions of the starting region were 1.25 m by 1.20 m, and the dimensions of the target region were 2.80 m by 2.43 m. The size, location of obstacles, and dimensions of the 2D map matched well with the real situation.

With the combined method of partial map simulation and global frontier search, the mobile robot continuously obtained environment information during the autonomous mapping and exploration process. Furthermore, in circumstances with insufficient information, the trajectory to the destination could be recovered with sufficient information by triggering the frontier search method. If a goal pose with sufficient information could not be simulated in the current robot pose, the global frontier search strategy was triggered in order to change the view of the robot sensors for further exploratory orientation to the target region; for example, the scene might change from (iv) to (v), corresponding to  $(-1.7, -3.6)$  in Fig. 9.

Fig. 10 shows the time-region curve of the autonomous mapping and exploration process, with a comparison between the use of a single RGB-D sensor and dual RGB-D sensors. The area was measured by the generated 2D grid map. A total area of  $31.0 \text{ m}^2$  was explored over 196 s with the dual RGB-D sensors, whereas a total area of  $27.6 \text{ m}^2$  was explored over 297 s with a single RGB-D sensor. In the single connected regions, the view obtained by the autonomous mobile robot system was limited by the aisle; thus, the exploration efficiency with a single RGB-D sensor was similar to that of the system with dual RGB-D sensors. In the multi-branch regions, the exploration cost time with the dual RGB-D sensors was significantly less than with a single RGB-D sensor; furthermore, the exploration area covered with the dual RGB-D sensors was larger than the area covered with a single RGB-D sensor. The mobile robot equipped with dual RGB-D sensors obtained more environment information than when equipped with a single RGB-D sensor. In the autonomous mapping and exploration process, the map region stably increased over time, thus demonstrating the efficacy of the proposed method.

### 5.3. Discussion

The exploration process will be aborted if neither the partial map simulation algorithm nor the global frontier search algorithm can generate an available exploration goal. The partial map simulation algorithm may fail if no goals with sufficient information can be generated. Furthermore, the global frontier search algorithm may fail if no feasible connected frontier point can be discovered. As shown in Fig. 6, if both algorithms fail, the current

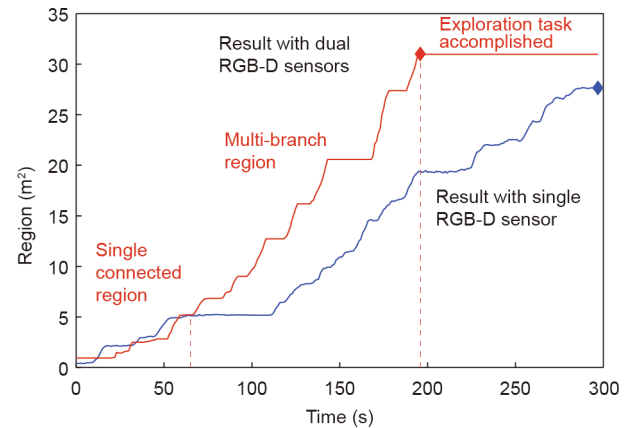


Fig. 10. The time-region curve in the autonomous mapping and exploration process comparing a single RGB-D sensor with dual RGB-D sensors.

exploration process is stopped. The robot can remain in place, return to the starting point, or execute orders from the human operator.

In the autonomous mapping and exploration process, the use of the dual RGB-D sensors improved the exploration efficiency. If more RGB-D sensors are deployed, the localization accuracy may be further improved, as has been observed in other applications [25]. More sensors can be integrated into the system for application-specific measurements [6]. The number and type of sensors can be determined with respect to application, computational cost, real-time capability, and accuracy requirements.

Multiple robots can collaborate in the exploration process in order to obtain more information and improve the exploration efficiency [36]. The fusion of autonomous exploration with teleoperation from human operators holds promise for overcoming possible pitfalls in robot autonomy [37].

## 6. Conclusion

In this work, we have presented a systematic approach for the autonomous exploration and mapping of an unknown environment using dual RGB-D sensors. The synchronized and processed RGB-D data provided environment color and depth information. Both a 3D cloud map and a 2D grid map were incrementally generated. Partial map simulation and global frontier search methods were combined for autonomous exploration, and dynamic action constraints were utilized in motion control in order to avoid a local optimum. The experimental results demonstrated the high robustness and efficiency of the developed system and methods.

Future work will focus on a semantic analysis of the 3D point cloud map in order to achieve abundant environment information in the exploration process.

## Acknowledgements

The authors would like to thank Mathieu Labbé from Université de Sherbrooke, Canada, and Mikko Lauri from Tampere University of Technology, Finland, for their helpful suggestions.

The authors would like to acknowledge the National Natural Science Foundation of China (61720106012 and 61403215), the Foundation of State Key Laboratory of Robotics (2006-003) and the Fundamental Research Funds for the Central Universities for the financial support of this work.



## Compliance with ethics guidelines

Ningbo Yu and Shirong Wang declare that they have no conflict of interest or financial conflicts to disclose.

## References

- [1] Ström DP, Bogoslavskyi I, Stachniss C. Robust exploration and homing for autonomous robots. *Robot Auton Syst* 2017;90:125–35.
- [2] Zhang Z, Nejat G, Guo H, Huang P. A novel 3D sensory system for robot-assisted mapping of cluttered urban search and rescue environments. *Intell Serv Robot* 2011;4(2):119–34.
- [3] Huang J, Ri M, Wu D, Ri S. Interval type-2 fuzzy logic modeling and control of a mobile two-wheeled inverted pendulum. *IEEE Trans Fuzzy Syst* 2017;26(4):2030–8.
- [4] Gai S, Jung EJ, Yi BJ. Multi-group localization problem of service robots based on hybrid external localization algorithm with application to shopping mall environment. *Intell Serv Robot* 2016;9(3):257–75.
- [5] Pirbhual S, Zhang H, Alahi M, Ghayvat H, Mukhopadhyay SC, Zhang Y, et al. A novel secure IoT-based smart home automation system using a wireless sensor network. *Sensors* 2016;17(1):69–87.
- [6] Huang J, Xu W, Mohammed S, Shu Z. Posture estimation and human support using wearable sensors and walking-aid robot. *Robot Auton Syst* 2015;73:24–43.
- [7] Krajník T, Santos JM, Duckett T. Life-long spatiotemporal exploration of dynamic environments. In: *Proceedings of the European Conference on Mobile Robots*; 2015 Sep 2–4. Lincoln, UK. New York: IEEE; 2015. p. 1–8.
- [8] Yamauchi B. A frontier-based approach for autonomous exploration. In: *Proceedings of 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*; 1997 Jul 10–11. Monterey, CA, USA. New York: IEEE; 1997. p. 146–51.
- [9] Lee K, Doh NL, Chung WK. An exploration strategy using sonar sensors in corridor environments. *Intell Serv Robot* 2010;3(2):89–98.
- [10] Ryu H, Chung WK. Local map-based exploration for mobile robots. *Intell Serv Robot* 2013;6(4):199–209.
- [11] Keidar M, Kaminka GA. Efficient frontier detection for robot exploration. *Int J of Rob Res* 2014;32(2):215–36.
- [12] Cieslewski T, Kaufmann E, Scaramuzza D. Rapid exploration with multi-rotors: a frontier selection method for high speed flight. In: *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent robots and Systems*; 2017 Sep 24–28. Vancouver, Canada. New York: IEEE; 2017. p. 2135–42.
- [13] Carrillo H, Dames P, Kumar V, Castellanos JA. Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Renyi Entropy. In: *Proceedings of the 2015 IEEE International Conference on Robotics and Automation*; 2015 May 26–30; Seattle, WA, USA. New York: IEEE; 2015. p. 487–94.
- [14] Lauri M, Ritala R. Planning for robotic exploration based forward simulation. *Robot Auton Syst* 2016;83:15–31.
- [15] Bai S, Wang J, Chen F, Englot B. Information-theoretic exploration Bayesian optimization. In: *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2016 Oct 9–14; Daejeon, Korea. New York: IEEE; 2016. p. 1816–22.
- [16] Shade R, Newman P. Choosing where to go: complete 3D exploration with stereo. In: *Proceedings of the 2011 IEEE International Conference on Robotics and Automation*; 2011 Sep 9–13; Shanghai, China. New York: IEEE; 2011. p. 2806–11.
- [17] Yu N, Wang S, Xu C. RGB-D based autonomous exploration mapping of a mobile robot in unknown indoor environment. *ROBOT* 2017;39(6):860–71.
- [18] Umari H, Mukhopadhyay S. Autonomous robotic exploration and on multiple rapidly-exploring randomized trees. In: *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots Systems*; 2017 Sep 24–28; Vancouver, Canada. New York: IEEE; 2017. p. 1396–402.
- [19] Vallvé J, Andrade-Cetto J. Potential information fields for mobile t exploration. *Robot Auton Syst* 2015;69(205):68–79.
- [20] Henry P, Krainin M, Herbst E, Ren X, Fox D. RGB-D mapping: using depth cameras for dense 3D modeling of indoor environments. In: *Proceedings of the 12th International Symposium on Experimental Robotics*; 2010 Dec 18–21; New Delhi, India. Berlin: Springer; 2010. p. 22–5.
- [21] Engelhard N, Endres F, Hess J, Sturm J, Burgard W. Realtime 3D visual SLAM with a hand-held RGB-D camera. In: *Proceedings of the RGB-D Workshop on 3D Perception in Robotics at the Euro Robotics Forum*. Sweden: Vasteras; 2011. p. 1–5.
- [22] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. In: *Proceedings of the 6th IEEE and ACM International Symposium Mixed and Augmented Reality*; 2007 Nov 13–16; Nara, Japan. Berlin: Springer; 2007. p. 225–34.
- [23] Labbé M, Michaud F. Online global loop closure detection large-scale multi-session graph-based SLAM. In: *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2014 Sep 14–18; Chicago, IL, USA. Berlin: Springer; 2014. p. 2661–6.
- [24] Yu N, Li Y, Xu C, Liu J. Localization and motion planning and control of mobile robots with RGB-D SLAM. *J Syst Sci Math Sci* 2015;35(7):838–47.
- [25] Yang S, Yi X, Wang Z, Wang Y, Yang X. Visual SLAM using Multiple RGB-D cameras. In: *Proceedings of the 2015 IEEE International Conference on Robotics and Biomimetics*; 2015 Dec 6–9; Zhuhai, China. Berlin: Springer; 2015. p. 1389–95.
- [26] Chow JC, Lichti DD, Hol JD, Belluscio G, Luinge H. IMU and multiple RGB-D camera fusion for assisting indoor stop-and-go 3D terrestrial laser scanning. *Robotics* 2014;3(3):247–80.
- [27] Munaro M, Basso F, Menegatti E. Openprtrack: Open source multi-camera calibration and people tracking for RGB-D camera networks. *Robot Auton Syst* 2016;75:525–38.
- [28] Huang C, Yang T. Line detecting and tracking of a mobile robot with multiple RGB-D cameras. In: *Proceedings of the 2016 IEEE 13th International Conference on Networking, Sensing, and Control*; 2016 Apr 28–30; Mexico City, Mexico. Berlin: Springer; 2015. p. 1–4.
- [29] Grisetti G, Grzonka S, Stachniss C, Pfa P, Burgard W. Efficient estimation of accurate maximum likelihood maps in 3D. In: *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2007 Oct 29–Nov 2; San Diego, CA, USA. Berlin: Springer; 2007. p. 3472–8.
- [30] Rusu RB. Semantic 3D object maps for everyday manipulation in human living environments. *Künstliche Intelligenz* 2010;24(4):345–8.
- [31] Sarbolandi H, Leoch D, Kolb A. Kinect range sensing: Structured-light versus time-of-flight Kinect. *Comput Vis Image Underst* 2015;139:1–20.
- [32] Thrun S, Burgard W, Fox D. Probabilistic robotics. *Commun ACM* 2002;45(3):52–7.
- [33] Khoshelham K, Elberink SO. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors* 2012;12(2):1437–54.
- [34] Faion F, Friedberger S, Zea A, Hanebeck UD. Intelligent sensor-scheduling for multi-Kinect-tracking. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2012 Oct 7–12; Vilamoura, Portugal. New York: IEEE; 2012. p. 3993–9.
- [35] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Trans Robot Autom* 1997;4(1):23–33.
- [36] Zheng S, Zhang Q, Zheng R, Huang B, Song Y, Chen X. Combining a multi-agent system and communication middleware for smart home control: a universal control platform architecture. *Sensors* 2017;17(9):2135–59.
- [37] Yu N, Wang K, Li Y, Xu C, Liu J. A haptic shared control algorithm for flexible human assistance to semi-autonomous robots. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2015 Sep 28–Oct 2; Hamburg, Germany. Berlin: Springer; 2015. p. 5241–6.