Digital Communications and Networks (2015) 1, 253–266

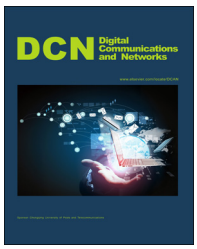
HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/dcan



Building a grid-semantic map for the navigation of service robots through human-robot interaction

Cheng Zhao[a,*], Weixing Mei[b], Wei Pan[b]

[a]*School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, United Kingdom*
[b]*Cognitive Science Department, Fujian Key Laboratory of the Brain-like Intelligent Systems, Xiamen University, Xiamen 361005, The People's Republic of China*

Received 30 July 2015; accepted 6 September 2015
Available online 30 September 2015

KEYWORDS
Visual-voice interface;
Grid-semantic map;
Global localization;
ROS

Abstract
This paper presents an interactive approach to the construction of a grid-semantic map for the navigation of service robots in an indoor environment. It is based on the Robot Operating System (ROS) framework and contains four modules, namely Interactive Module, Control Module, Navigation Module and Mapping Module. Three challenging issues have been focused during its development: (i) how human voice and robot visual information could be effectively deployed in the mapping and navigation process; (ii) how semantic names could combine with coordinate data in an online Grid-Semantic map; and (iii) how a localization-evaluate-relocalization method could be used in global localization based on modified maximum particle weight of the particle swarm. A number of experiments are carried out in both simulated and real environments such as corridors and offices to verify its feasibility and performance.

1. Introduction

The navigation of service robots traditionally relies on geometrical maps that are either priory constructed or dynamically built from sensor data. This imposes a big challenge for general public to use service robots in their daily life since they have to be trained beforehand. On the other hand, humans use the semantic map to navigate around, which is intuitive and easy to learn. It would be beneficial if a sematic map could be deployed in the robot navigation system so that users can operate service robots using semantic information such as voice and gesture. Therefore, interactive mapping and navigation for service robots is currently an active research area in the robotics community. In order to add semantic information to the map, there are two mainly branches: (i) generate the

*Corresponding author.
E-mail address: IRobotCheng@gmail.com (C. Zhao).
Peer review under responsibility of Chongqing University of Posts and Telecommunications.

http://dx.doi.org/10.1016/j.dcan.2015.09.002

semantic map automatically by extracting or classifying the data from sensors, and (ii) generate the semantic map via human–robot interaction using visual and voice signals.

Buschka and Saffiotti proposed an approach to segment and detect the room spaces for navigation using range data in an office environment [1]. Using the anchoring technique, Galindo et al. labelled the topological map with semantic information for navigation [2]. The work in [3] and [4] introduced an approach to learn topological maps from geometric maps by applying a semantic classification procedure based on associative Markov networks and AdaBoost. In addition, many works use visual features from camera sensors to extract semantic information via place categorization [5].

Since semantic information automatically extracted from sensors is limited and not robust enough, more and more researchers turn to the second branch: semantic mapping based on human–robot interaction, which extracts richer semantic information. A wearable gesture interface in an ear-mounted FreeDigiter device was proposed in [6] to label doors in the topological map. Several IMUs are used in [7] to detect movements of a person and door opening and closing events labelled as topological nodes in a graph-based SLAM framework. A contextual topological map was built by making the robot follow the user and verbal commentary [8]. A probabilistic model was proposed in [9] for recognition and classification of spaces into separate semantic regions and can use this information for generation of the topological map.

Kruijff et al. built the semantic map through natural language dialogues between human and robot [10]. The system in [11] integrates laser and vision sensors for place and object recognition as well as a linguistic framework, which creates a conceptual representation of the human-made indoor environment. Randelli et al. summarized many multi-modal interactions such as speech, gestures and vision for semantic labelling, which assists the robot in obtaining rich environment knowledge without many pre-requisites features [12]. Pronobis and Jensfelt presented a probabilistic framework combining heterogeneous, uncertain information such as object observations, shape, size, appearance of rooms and human input for semantic mapping [13]. It abstracts multi-modal sensory information and integrates it with conceptual common-sense knowledge, which makes the semantic map more descriptive, and the system is more scalable and better adapted for human interaction. The work in [14,15] proposed similar approaches for learning environmental knowledge about the grounding of expressions from task-based human-robot dialogs. In a survey [16], many semantic mappings for mobile robotics tasks are summarised in detail.

Fig. 1 The platform snapshot in this paper.

This paper presents the construction of a Grid-Semantic map for the navigation of service robots, which is based on human-robot interaction. A novel localization-evaluate-relocalization for global localization is applied to navigation. The rest of the paper is organized as follows. Section 2 introduces the platform, the software and the basic system configuration. Section 3 outlines the methodologies deployed in this research, including interaction, control, mapping and navigation. In Section 4, a number of experimental results are presented to verify the feasibility and performance of the proposed approach. Finally, a brief conclusion and future work are given in Section 5.

2. System overview

2.1. Platform and software configuration

The platform in this paper includes a Pioneer 3-AT robot (P3AT in short), a Sick LMS100 laser, a Kinect sensor, a notebook and a cell phone, as shown in Fig. 1. The whole system is based on the ROS [17] framework. The related libraries include: Fuerte ROS (Gmapping and navigation Package), PrimerSensor (5.1.2.1), NITE (1.5.2.21), OpenNI (1.5.4.0), JDK (1.6.0_20), Android SDK (2.3.3), IFLYTEC SDK (1013), Ekho (5.6), Aria (2.7.5.2).

2.2. Workflow

In our daily life, when guests come to our house for the first time, we show them around the house so that they could find a way around during their stay. Inspired by this, we propose an interactive navigation system for service robots, as shown in Fig. 2. It contains four modules and operates sequentially as follows:

1. The robot moves in an unknown indoor environment, following the user's gestures gathered by the Kinect-based skeleton tracking.
2. During the following process, a real time grid map is generated based on the RBPF algorithm using laser and odometer data.
3. At the same time, the user can use the vocal APP based on IFLYTEK on a cell phone to label places on the grid map.
4. Then the robot can combine the semantic names from voice recognition with the coordinate values from the Grid map together to build a Grid-Semantic map.
5. The robot obtains its current location through a novel localization–evaluate–relocalization method and acquires the destination from the vocal APP sent by user.
6. The robot transforms a Grid map to a Cost-Map, then makes a path planning using Dijkstra's algorithm, and finally reaches the destination.

All the experiment videos can be watched on Youtube [18].

3. Methodology

3.1. Interactive Module

3.1.1. Skeleton Tracking

In terms of visual interaction, the Kinect based Skeleton Tracking [19] is adopted because of the following four reasons.

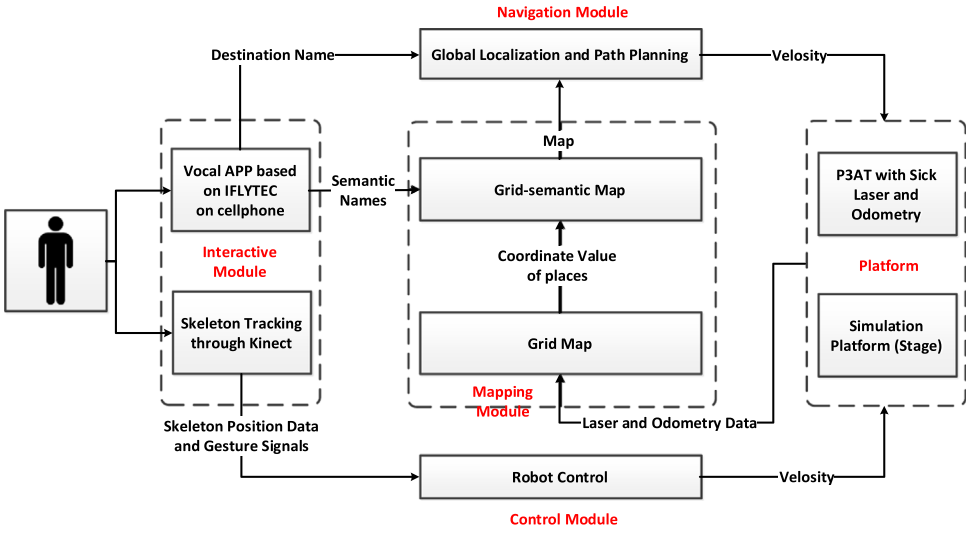


Fig. 2 The flow chart of the proposed system.

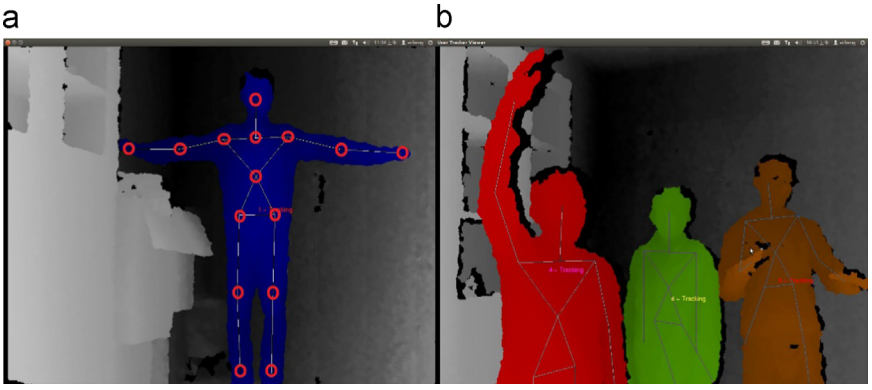


Fig. 3 (a) The 15 skeletal points. (b) Multiple persons in the scene.

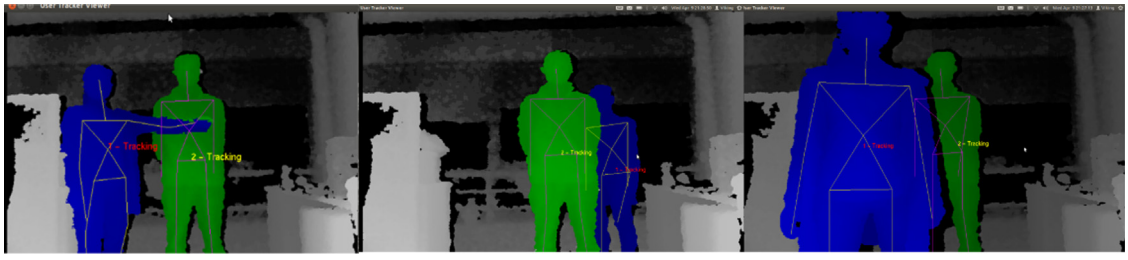


Fig. 4 Interference person (blue) shelters the tracked user (green).

1. It provides the ability to track the position (*X*, *Y*, *Z*) of 15 skeletal joints (Fig. 3(a)) and center of mass for each video frame in real time with a confidence level.
2. The Random Forest classifier in the Skeleton Tracking can identify multiple persons and assign everyone a unique and persistent ID (Fig. 3(b)), which allows the robot to follow a specific user under multiple people interference.
3. The skeleton Tracking algorithm can track the user reliably under the strong shelter interference (Fig. 4).
4. When the user disappears from the scene for a few seconds, his or her color and ID will remain because the routine saves the skeleton calibration data of identified person. It enables the robot to take some actions such as searching for the lost user at this time.

In order to more robustly control the robot, the visual interaction generates some gesture signals to the control model, such as “raise the right hand over the torso”, as shown in Tab. 1. These gesture signals are generated by comparing the coordinate value of different skeleton points. For example, if the *y* coordinate value of right hand is less (the positive plus of *y* is to the ground) than that of torso, the signal “raise the right hand over the torso” is generated.

Tab. 1 The corresponding actions of P3AT according to different signals.

User's motions or voices	Signal	P3AT's actions
Find the user	0	Follow the person
Raise the right hand over the torso	1	Stop temporarily
Raise the left hand over the neck	−1	Stop and end the program
Lose the user	−2	Rotate to search for the person
Barrier distance is lower than threshold	2	Stop and end the program
Voice command “begin mapping”	3	Begin mapping mode
Voice command “begin navigation”	4	Begin navigation mode
Voice command “end mapping”	−3	Return to selecting mode
Voice command “end navigation”	−4	Return to selecting mode

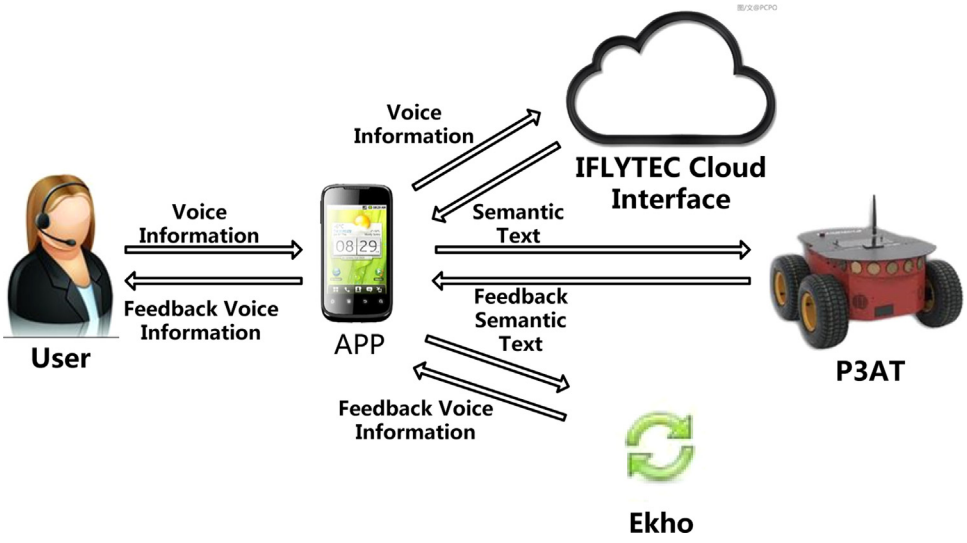


Fig. 5 The vocal interaction framework.

3.1.2. Vocal Marking

The user communicate with the robot using a vocal APP on the cell phone. The IFLYTEC cloud interface is used for the voice recognition [20] and the open source library Ekho [21] is used for the voice synthesis. Fig. 5 shows the framework for the voice interaction. IFLYTEC is a very popular commercial voice recognition interface for Chinese based on Hidden Markov Model [22] and Deep Neural Networks [23], which has a huge number of training samples and high accuracy rate. Our voice interaction in this system has three modes: selecting mode, mapping mode and navigation mode. The default mode at APP startup is selecting mode. It is changed between them through voice commands such as “begin mapping”, “end mapping”, “begin navigation” and “end navigation”. In order to decrease the workload of the natural language processing, an effective Augmented Backus-Naur Form (ABNF) file should be designed according to the technical documentation of IFLYTEC [20]. Those three modes have their own ABNF files and those ABNF files are uploaded to the IFLYTEX cloud.

During the semantic labeling in the mapping process, labeling the place could be described using similar but different sentences like “here is office 506”, “this place is office 506” or “office 506 is here”. This problem is addressed by designing a reasonable mapping ABNF file. Then the IFLYTEC cloud interface returns the same text according to similar voice input using ABNF files. During the navigation process, going to the same destination also can be described using similar but different sentences, such as “office 506”, “go to office 506” or “office 506 is the destination”. This also can be addressed by designing a reasonable navigation ABNF file. After transforming the voice to text, the simple key word (such as “office 506”) string matching technique is used for parsing text. In order to avoid misrecognizing voice commands, the vocal APP will give voice feedback like “Are you sure here is office 506?” when it receives the voice input. Only if the user responds “yes”, then this voice command will be performed. When the robot completes the task, it can give a feedback, such as “global localization was successful”, “I have reached office 506”. These feedbacks can be transformed from text to speech using open source library Ekho.

3.2. Control Module

After the Control Module receives signals (Fig. 6) generated from the interaction module, the robot takes different actions according to corresponding signals in Tab. 1. As mentioned in vocal interaction, three system modes, namely selection mode, mapping mode and navigation mode, can be changed by voice commands (signal 3, −3, 4, −4). In the mapping mode: Signal

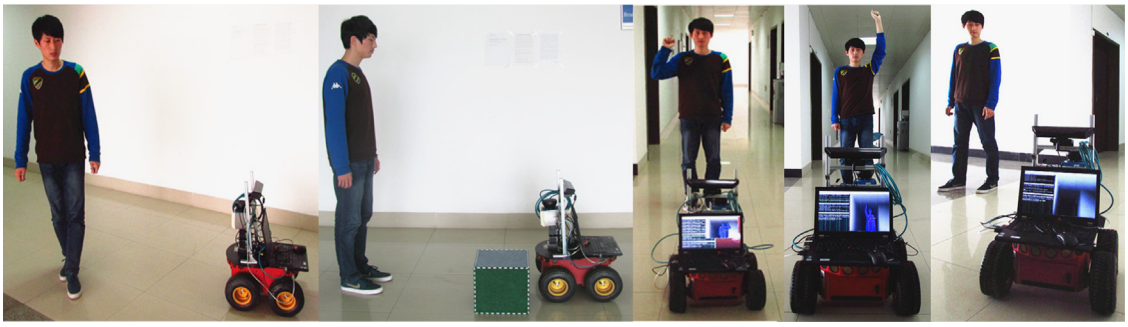


Fig. 6 Different signals generated from interaction module.

Tab. 2 The mathematic relation between the centre of mass' position and the velocity of P3AT.

The centre of mass' position	Angular velocity	Linear velocity
$270 \leqq X \leqq 370$, $1000 \leqq Z \leqq 1500$	0	0
$X < 270$	$[-(270-X)/(270-0)-0.5]*(10/180*\pi)$	Null
$X > 370$	$[(X-370)/(640-370)+0.5]*(10/180*\pi)$	Null
$Z < 1000$	Null	$[-(1500-Z)/(1500-0)-0.5]*0.4$
$Z > 1500$	Null	$[(Z-1500)/(3000-1500)+0.5]*0.4$

0 means that the robot finds the user and will follow him. Signal −2 means that the robot loses the user and will rotate to search the user.

In order to make the control robust during the process of semantic labeling,the user raising their right hand (signal 1) makes the robot stop temporarily to get the current position in the grid map. When the user puts down his right hand, the robot will keep following the user again. The user raise left hand (signal −1) makes the robot stop and ends the system. In the navigation mode, the robot can avoid obstacles automatically using the cost map. However in mapping mode, if the barrier distance is below a specify threshold (signal 2), the robot will stop and end the system for safety reasons.

In terms of following the user, the major principle is keeping the position of the user's mass center in central view of Kinect and the distance between the user and robot is within a specified range [24]. According to the position of the user's mass center, it will be divided into 5 different areas. In different areas, the robot will take different linear and angular velocities (Tab. 2), which makes the robot come back to the center area. In Tab. 2, the units of the *X* and *Z* axes are pixel and mm respectively. The unit of angular velocity is rad/s (*Plus* represents turning to left and *minus* represents turning to right). The unit of linear velocity is m/s (*Plus* represents going forward and *minus* represents going back).

3.3. Mapping Module

3.3.1. Grid Map

The GMapping package in ROS is deployed to build a real-time indoor grid map during the following process, which is based on Rao-Blackwellized Particle Filter (RBPF) [25,26]. We decompose the GMapping framework into 2 phases: (i) updating the robot's state through measurement from an odometer and laser scanner, and (ii) updating the map based on RBPF. In RBPF, each particle carries an individual map of the environment. In order to update its estimate, the sampling-calculate weight-resampling process is performed as follows:

1. Each position of the particles is updated using the odometer data and the map in each particle is drawn using the laser data.
2. Assign a weight to each particle using a modified Vasco Scan-Matching which compares the new scan with the existing particle map. The weight is proportional to the number of points that match the existing map.
3. During resampling, the target distribution is obtained from the weight proposal. Particles with the weight that is lower than a certain threshold will be deleted and replaced by new particles.

In addition, the noise points of the user in the grid map are filtered by RBPF automatically.

3.3.2. Grid-Semantic Map

During the grid mapping, the user can label the spots using a cell phone. When the robot receives a voice instruction, it will combine the semantic name from voice recognition with its coordinate value on the grid map to create a semantic marker, as shown in Fig. 7. They are stored in the map container.

$$MAP_{Nodes} \equiv \left\{ \langle sem_i, coor_i \rangle | sem_i \in S, coor_i \in R^3 \right\} \quad (1)$$

where sem_i is the semantic name such as “Office 506” and “toilet”. $coor_i$ is $\langle x, y, \theta \rangle$ containing the coordinate values on the grid map and the position data of the robot. *S* is the semantic name set.

Finally, a Grid-Semantic map is built for navigation. Voice markers establish a relationship between the semantic name and the grid coordinate value, which gives the robot and human a common description for a same place. The

interactive mapping overcomes the problem of extracting and describing markers because it extracts markers through human interaction and describes them with human language which has rich semantic information.

3.4. Navigation Module

Three central problems face robot navigation: “where am I”, “where shall I go”, and “how to get there”. For the first problem, we propose a novel localization-evaluate-relocalization global localization method based on a modified maximum particle weight. For the second problem, the user can tell the robot the destination name using the vocal APP. For the third problem, the off-the-shelf Cost-Map with the Dijkstra search algorithm is used to make a path plan.

3.4.1. Global localization

The localization-evaluate-relocalization global localization method is based on an Adaptive Particle Filter [27] which performs a sampling-calculate weight-KLD resampling [28] framework and uses Kullback-Leibler (KL) to measure the quality of the particles swarm.

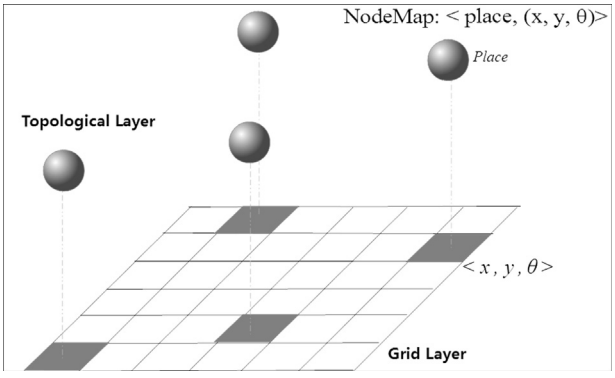


Fig. 7 Vocal marking.

Two features: the modified maximum particle weight and the radius of the particle swarm, are the criterions for a successful localization. The strategy is: firstly the whole particle swarm is distributed evenly in the whole map. Then the radius of the particle swarm is assessed continuously during the process of particle swarm convergence. When the radius is less than 1 m, the modified maximum weight of the particle swarm is examined. If the modified maximum weight is more than the given threshold δ_{global}, the localization is successful. Otherwise, the distribution of the particle swarm is renewed and the next round begins. The maximum time of relocalization is 12. If the global localization is not successful within 12 iterations, the global localization fails. Because it is easy to explain by combining with experiment results, this approach will be described in more detail later in the experiment part in Section 4.2.

3.4.2. Path planning

In terms of path planning, the Cost-Map with the Dijkstra search algorithm is adopted. Transforming the grid map to the Cost-Map needs an inflation process [29,30]. Firstly, the influence area of every obstacle is extended. Then the distance between the grid center and the obstacle is compared with the robot radius, and a decay function [31] is used to calculate and assign a cost value for each grid. After transforming the grid map to the Cost-Map, the Dijkstra algorithm is used to search the Cost-Map to find a route that has a minimal sum of the cost value. Finally, the robot reaches the destination using the Path Tracking algorithm [32].

4. Experiment and evaluation

4.1. Mapping experiment

The Grid map can be generated through the Gmapping package in ROS [33] by using laser and odometer data. Then the semantic markers are labelled on the grid map through vocal marking. Finally the grid-semantic map is built for

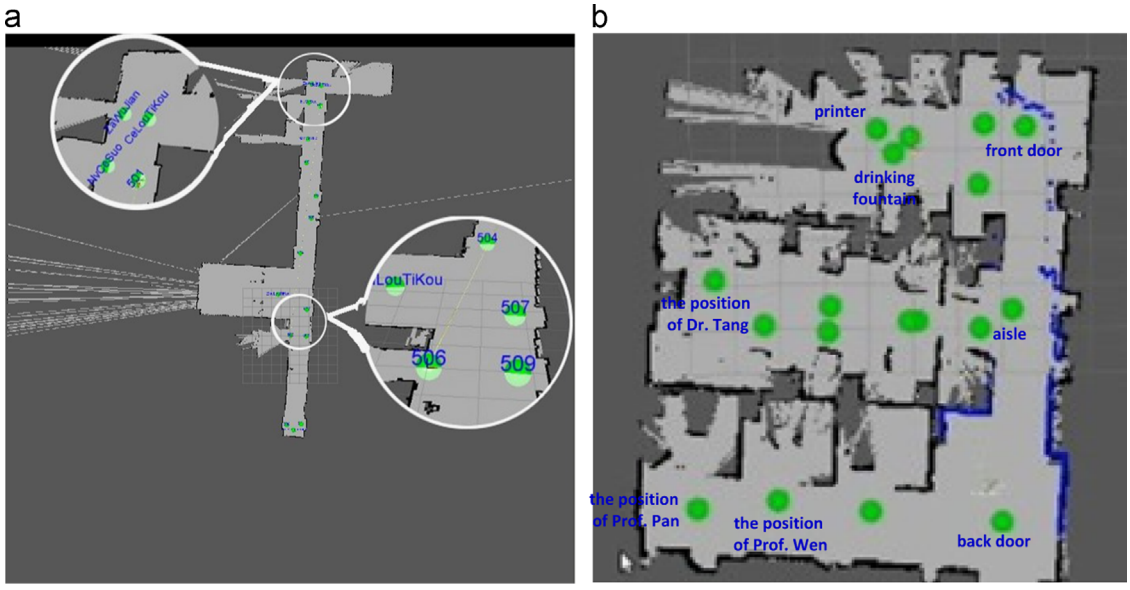


Fig. 8 (a) Grid-Semantic map of corridor. (b) Grid-Semantic map of office 506.

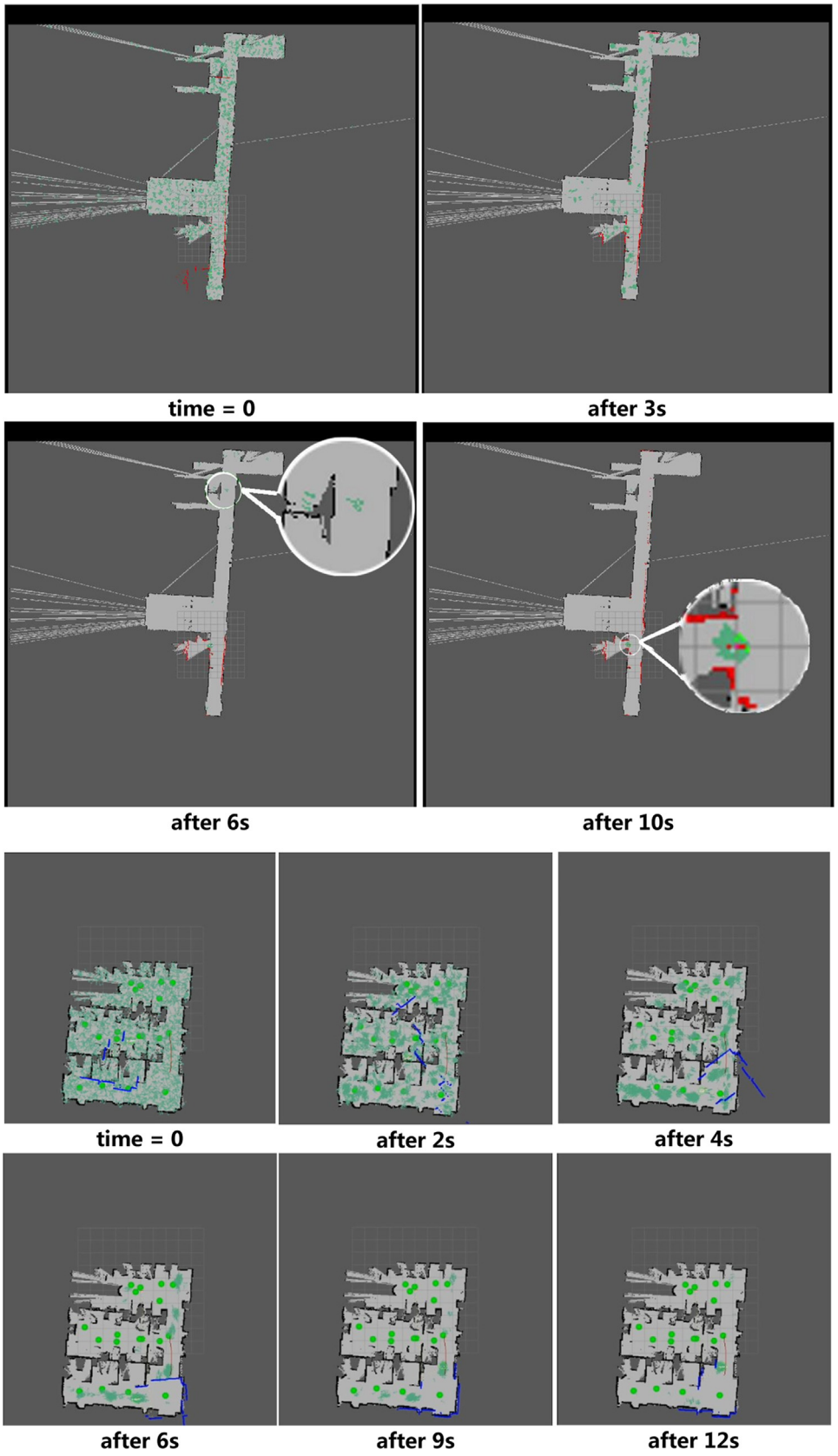


Fig. 9 The convergence of the particle swarm in global localization.

navigation. Fig. 8 shows the grid-semantic maps of a corridor and an office environment. The grey color represents the grid map. The green dots are the markers with semantic name and coordinate values. The blue line is the laser scan data.

4.2. Global localization experiment

4.2.1. Feature selection for global localization

Fig. 9 shows the distribution and convergence situations of the particle swarm in the corridor and office 506. The red lines are the laser scans, the small green dots are the particles and the big green dots are the semantic markers. At the beginning, particles are distributed evenly in the entire map and the laser scan reading does not match the map. After 3–4 s, the particle swarm begins to gradually converge. After 6–9 s, most particles converge to one position, but a minority particles are still located at another position. Only after 12 s, all particles converge to the same position and the laser scan readings match the map.

As shown in Fig. 10, 80 global localization tests were performed in the passageway and office 506 using the simulation platform Stage [34] and Rviz [35] in ROS. A green dot represents a position where a global localization test was performed. The real position and test position of every dot is obtained through Rviz and laser scanning (see motion strategy), and then they are compared. If the offset distance between them is less than 1 m, global localization is considered to be successful and this position is marked as green in Fig. 11. Conversely, if the offset distance between them is greater than 1 m, global localization is considered to have failed and this position is marked as red. The statistical results are shown in Tab. 3 and the detailed illustration of Fig. 11 is presented in Tab. 4.

As can be seen in Fig. 11, intuitively the swarm radius should be the criteria for successful localization because the swarm radius is usually very small when the particle swarm converges to the initial position. However in Fig. 11A, when samples were projected to the *Y* axis which stands for the swarm radius, the red and green samples cannot be totally separated. So the swarm radius is the only criteria for the convergence of the particle swarm rather than the successful localization. In Fig. 11B, when the samples were projected to the *X* axis which stands for the swarm radius, the red and green samples still cannot be separated. Yet, when the samples were projected to the *Y* axis which stands for the maximum particle weight of the swarm, the red and green samples can be partially separated. Therefore, we need to modify the particle weight calculation method to increase the distance between red and green samples.

In the particle filter localization, the particle weight is calculated using the following likelihood model [36].

$$p(z_t|x_t,m)=\prod_{i=1}^{K}(z_{hit}\cdot p_{hit}(z_t|x_t,m)+z_{rand}\cdot p_{rand}(z_t|x_t,m)) \tag{2}$$

where $p_{hit}(z_t|x_t,m)\sim N_{dist}(0,\delta_{hit}^2)$, $p_{max}(z_t|x_t,m)=\delta(dist-z_{max})$, $p_{rand}(z_t|x_t,m)=1/z_{max}$. The dist is the distance between the position of predicted obstacle and the position of real obstacle. z_{hit} and z_{rand} are the weighting factors.

In order to increase the distance between the red and green samples, we cube the maximum particle weight w_{max} by considering the computation complexity. Since the result of the accumulated multiplicative is very small after cubing, accumulated addition is adopted rather than accumulated multiplicative. The modified weight calculation method is shown below.

$$p(z_t|x_t,m)=\sum_{i=1}^{K}(z_{hit}\cdot p_{hit}(z_t|x_t,m)+z_{rand}\cdot p_{rand}(z_t|x_t,m))^3 \tag{3}$$

As shown in Fig. 11C, the maximum particle weight is calculated through the modified weight calculation method.

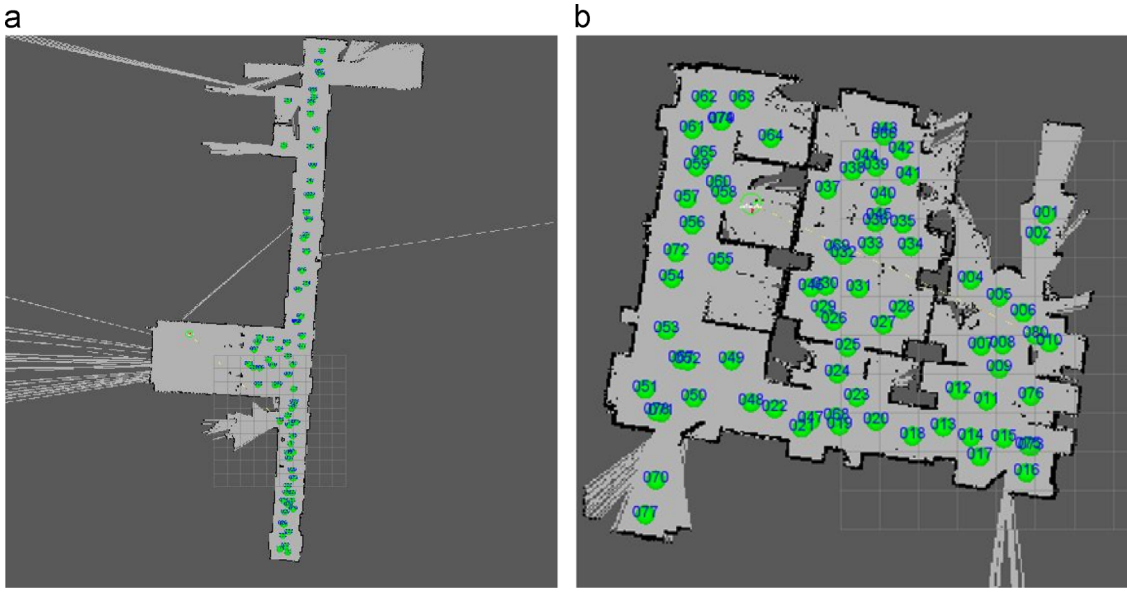


Fig. 10 (a) 80 Global localization tests in different positions of passageway. (b) 80 Global localization tests in different positions of office 506.

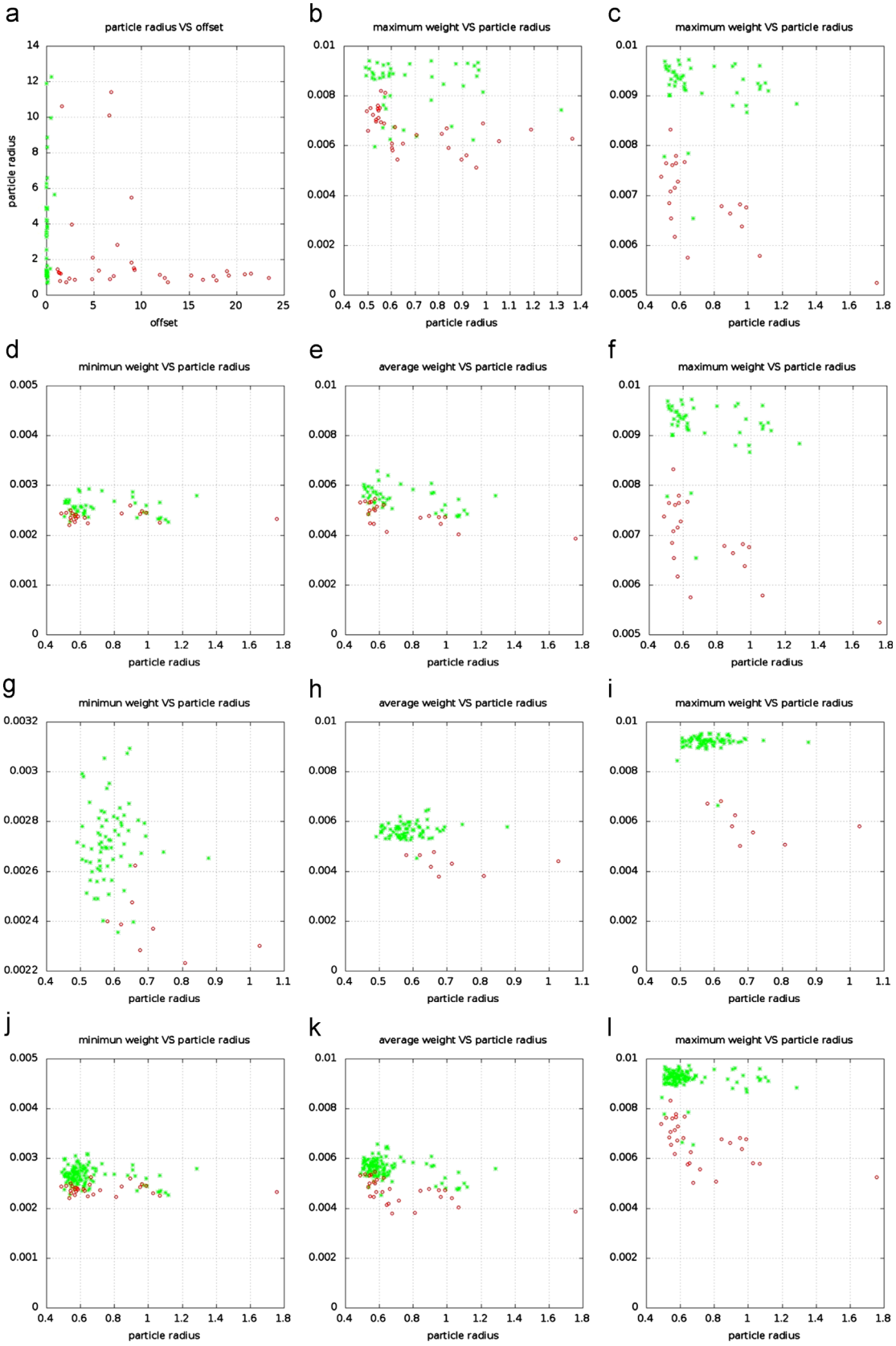


Fig. 11 Features selection for globalization localization.

The red and green samples are separated by projecting samples to Y axis. As the maximum, minimum, mean particle weights describe the localization of particle swarm in the particle filter, we also use these three features based on the modified weight calculation method to evaluate the global localization in a corridor (Fig. 11D-F) and an office 506 (Fig. 11G-I) respectively. Finally, we put the two different experiment data into one coordinate system (Fig. 11J-L) according to the same features.

Using the minimum weight of a particle swarm, as shown in Fig. 11D, G and J, we cannot separate the red and green samples in the corridor, office 506, and their mixture. Using the mean weight of the particle swarm, as shown in Fig. 11E, H, K, we can separate the red and green samples in office but not in the corridor and their mixture. Using the maximum weight of the particle swarm, as shown in Fig. 11F, I, L, we separate the red and green samples basically in the corridor, office 506 and their mixture. In other words, the global localization based on modified the maximum particle weight can be implemented for different environments.

In conclusion, the swarm radius is the only criteria for the convergence of a particle swarm. When the radius is less than a given threshold, e.g. 0.8 m or 1 m, we conclude that the particle swarm has already converged.The modified maximum particle weight of the particle swarm is the criteria for successful localization. To prove it theoretically, we describe every sample dots using two features, i.e. the radius r of the particle swarm and the maximum particle weight w_{max} of the particle swarm. Then we analyze the experimental samples using Fisher linear classification and calculate the best projection vector. The calculation method of project vector is shown below.

Tab. 3 The statistical result of the localization test.

	Corridor	Office
Effective recording	67/80	79/80
Successful localization dots	46/67	71/79
Failed localization dots	21/67	8/79

$$w = \left(\sum_{x \in pos} (x - m_{pos}) * (x - m_{pos})' + \sum_{x \in neg} (x - m_{neg}) * (x - m_{neg})' \right)^{-1} \cdot (m_{pos} - m_{neg}) \quad (4)$$

where the 1st bracket calculates the scatter matrix of the positive and negative samples. m_{pos} and m_{neg} are the mean value centers of the positive and negative samples respectively. w is the best project vector.

We reduce the dimensions of samples using w and guarantee the classification of the sample's projection in this direction is best. The projection process is shown below.

$$s * w = [r, w_{max}] \cdot \begin{bmatrix} w_1, \\ w_2 \end{bmatrix} = r * w_1 + w_{max} * w_2 \quad (5)$$

where w is [0.088 478, 98.408 442] in the corridor environment and w is [−0.19135, 270.72736] in the office 506 environment. Those two vectors are approximately parallel to Y axis. In other words, the projection value approximates w_{max} which can replace the vector $\langle r, w_{max} \rangle$.

4.2.2. Motion strategy in global localization

The motion strategy of the robot during the global localization process is that keeping rotation in situ and the angular velocity is 60°/s.This motion strategy does not change the initial position of robot and also can avoid obstacles. The filter iteration only uses the localization information of the robot in one direction of the current position.In the experiment, each iterationis about 0.5 s and each completed rotation is 3 s. So, we obtain 7 complete data rotations in 20 s. Then we average this data to obtain the test localization. After comparing the test position and real position on the map, if the offset between them is less than 1 m, the global localization is successful. Otherwise, it has failed.

4.2.3. Relocalisation results analysis

The global localization is likely to fail sometimes especially in a simple environment like the corridor because the particle filter uses a randomized distribution method. If it

Tab. 4 The illustration of Fig. 11.

Fig. 11	X axis	Y axis	Modified weight calculation	Environment
A	Offset distance	Particle swarm radius	No	Corridor
B	Particle swarm radius	Maximum particle weight	No	Corridor
C	Particle swarm radius	Maximum particle weight	Yes	Corridor
D	Particle swarm radius	Minimum particle weight	Yes	Corridor
E	Particle swarm radius	Mean particle weight	Yes	Corridor
F	Particle swarm radius	Maximum particle weight	Yes	Corridor
G	Particle swarm radius	Minimum particle weight	Yes	Office 506
H	Particle swarm radius	Mean particle weight	Yes	Office 506
I	Particle swarm radius	Maximum particle weight	Yes	Office 506
J	Particle swarm radius	Minimum particle weight	Yes	Corridor & Office 506
K	Particle swarm radius	Mean particle weight	Yes	Corridor & Office 506
L	Particle swarm radius	Maximum particle weight	Yes	Corridor & Office 506

fails within one time global localization, we will renew the distribution of the particle swarm evenly in the whole environment and begin the next round. We set the maximum times of relocalization is 12. The relocalization tests are carried out in the same positions in the corridor and office 506 environments as shown in Fig. 10. Tab. 5 shows the statistic results and Fig. 12 shows the histogram of the relocalization times.

Except for the 3 failed localizations in the office 506, the outcome of global localization in the office 506 is obviously better than that in the corridor. This is because that the corridor environment is relative simple and the laser scan data in different positions is usually very similar. So, the particle swarm easily converges to the wrong position. This is why we cannot select the radius of particle swarm as the criteria for successful localization. In terms of the office 506, the environment is relative complex and the laser scan data in different positions varies a lot. It is easy to converge to the right position for particle swarm. As shown in Fig. 11, the distance between red and green dots in I is much bigger than that in F (Note: the unit of the *Y* axis in I and F are different).

Tab. 5 The statistical result of relocalization test.

	Corridor	Office
Effective recording	78/80	79/80
Successful localization dots	78/78	76/79
Success rate of localization	100%	96.20%
Success rate of localization with one time	62.82%	89.87%
Maximum times of localization	6	12

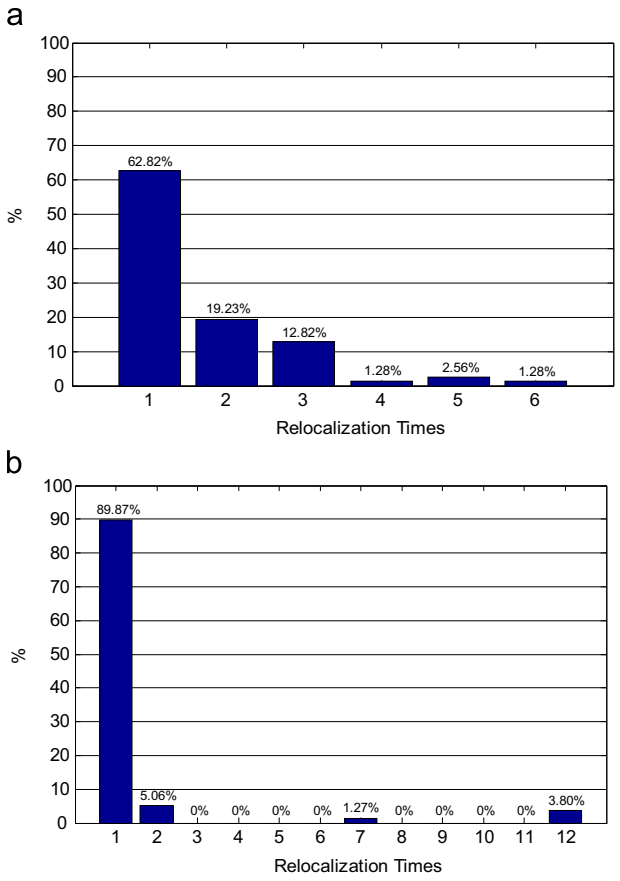


Fig. 12 (a) The histogram of relocalization times in the corridor. (b) The histogram of relocalization times in the office 506.

Finally, we analyze why there are three failed positions during the global localization in the office 506. These three positions are marked on the map in Fig. 13. They all have a large-scale area without map information around them. During the mapping process, those large-scale areas were not scanned by laser because of other objects. Since these areas may be free space or obstacles, but they all are marked as unknown space on the map. For the position fail-1 in the top-right corner, the red line is the laser scan data. The robot sees some obstacles (in the green circles) that do not exist around this position on the map.

Now because of the lack of some surrounding map data, the particle weight in the right position decreases a lot so that it cannot be selected through resampling. Then the particle swarm cannot converge in time or may converge to the wrong position. However, some positions with no surrounding map data localize successfully, which depends on whether the unknown surrounding area is big enough to influence selecting the right particle from the particle swarm.

To summarize, except for the positions which lack of surrounding map information, the success rate of localization-evaluate-relocalization can reach 100%.

4.2.4. Threshold δ_{global} setting in a real environment

In order to get the real position of the robot in the real environment, we just need set a threshold δ_{global} for localization because only one feature is used. First, we calculate the mean value of the positive and negative samples, and then use these two centers as the weight. It should be noted that the risk of classifying wrong samples to right samples is much bigger than that of classifying right samples to wrong samples. During the navigation process, if the initial localization is wrong, a series of later localizations are all wrong because localization depends on the last

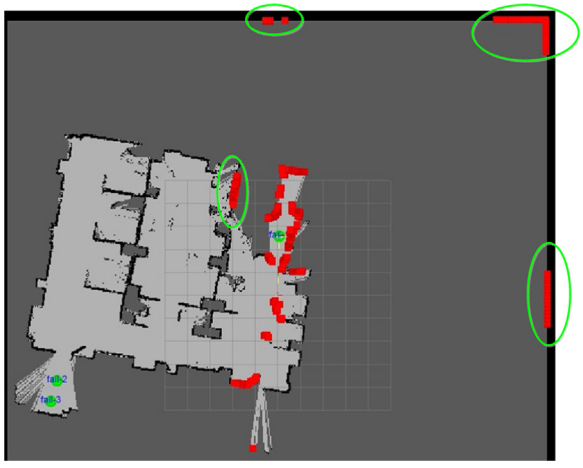


Fig. 13 The three failed positions of relocalization in office 506.

Tab. 6 The experiment result of relocalization.

Evaluation item	Corridor	Office 506	Mixture
Size	12.5 m × 4 0m	8.5 m × 1 2m	-
Experiment times	78	79	157
The mean times of relocalization	1.654	1.544	1.599
δ_{global}	0.008769	0.008299	0.008375
Success rate of classification	92.68%	100.0%	96.63%
Misclassification rate of negative samples	0.0%	0.0%	0.0%

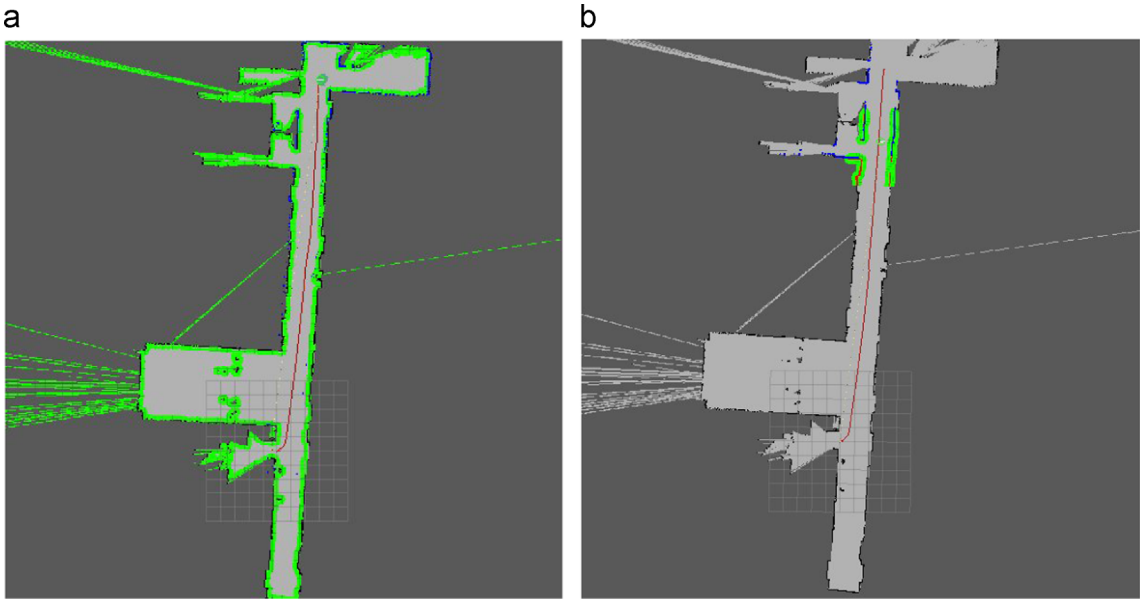


Fig. 14 (a) Global Cost-Map of corridor. (b) Local Cost-Map of corridor.

moment position. Nevertheless if the right samples are classified as wrong samples, they can be modified through relocaliszation. So the weight of positive samples is much bigger than that of negative samples. In our experiment α=0.75.

$$\delta_{global} = \alpha \bullet \frac{1}{N_{pos}} \bullet \sum_{x_i \in pos} x_i + (1-\alpha) \bullet \frac{1}{N_{neg}} \bullet \sum_{x_j \in neg} x_j \alpha \in (0.5, 1) \quad (6)$$

As shown in Tab. 6, no wrong localization samples are classified as the correct localization samples. For the global localization in a real environment, if the modified maximum weight of the particle swarm is more than δ_{global}, the localization is successful, otherwise the distribution of the particle swarm is renewed and the next round begins.

4.3. Path planning experiment

The navigation package [34] in ROS based on Cost-Map is used to complete the path planning. Fig. 14 shows the global Cost-Map and local Cost-Map. Because of the dynamic variations in the circumstances, a local Cost-Map [35] is built for local real-time motion planning of the robot during the moving process. It uses the laser data from the current position of the robot rather than the whole circumstance. The red line is the outcome of path planning and the blue line is the laser scan. The obstacles and unknown areas are marked as red, indicating that a collision will occur if the robot is on this grid. The free spaces are marked as off-white, indicating that a collision will not occur if the robot is on this grid. The inflated areas around the obstacles are marked as green, indicating that a collision may occur if the robot is on this grid (depending on the robot shape). In Fig. 14 (b), only the grid map around the current position of the robot is transformed to a Cost-Map.

Fig. 15 shows the outcome of path planning using the Dijkstra search algorithm in the Cost-Map. The red line is the planned path and the blue line is the laser scan. The green dots are the particles swarm which are around the robot all the time for real-time localization.

5. Conclusion

In this paper, we build a novel interactive mapping and navigation system based on the ROS framework. It contains four modules: Interaction, Control, Mapping and Navigation, which are easy to configure and extend. Three contributions have been made in this research: (i) applying a visual-voice interface in the mapping and navigation process so that semantic information is added to the map. (ii) Building a Grid-Semantic map which combines the semantic names with the coordinate values to help the robot have a common description for the same places with human language. (iii) A

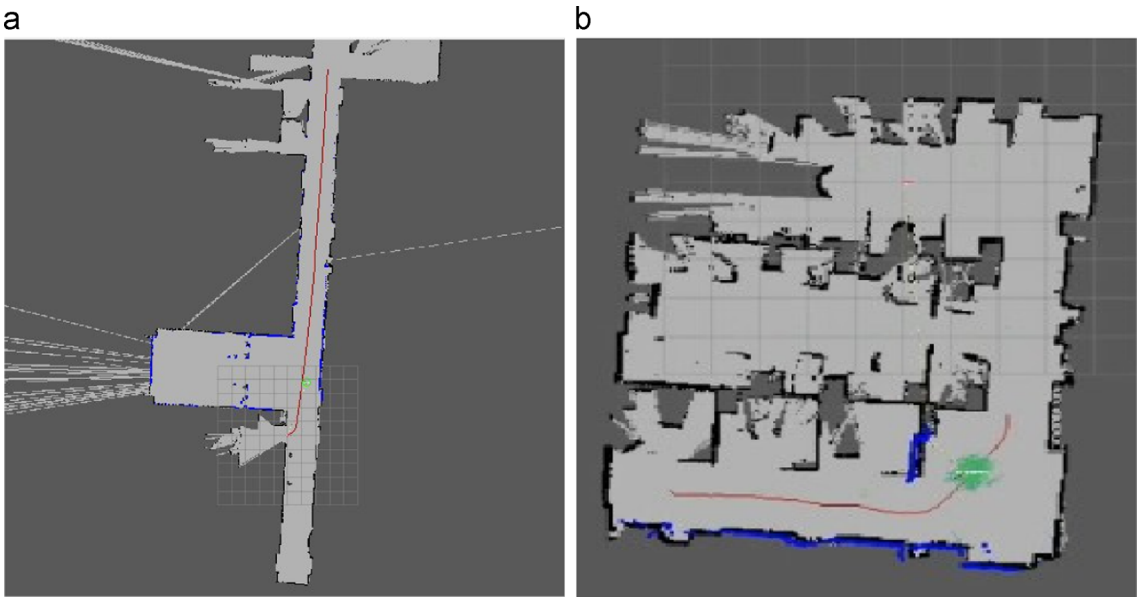


Fig. 15 The outcome of path planning through the Dijkstra search algorithm in Cost-Map in (a) passageway and (b) office 506.

novel localization–evaluate–relocalization global localization method based on modified maximum particle weights is proposed to increase the success rate of global localization. This system is implemented on a service robot in an indoor environment to improve the quality of life for the disabled and elderly people, as well as anyone in needs. The future work will be focused on extending from manual semantic labeling to automatic semantic labeling and an efficient way to add more rich semantic information to the map.

Author contributions

All authors contributed to this work.

Conflicts of interest

The authors declare no conflict of interest.

Acknowledgments

The 1st author is financially supported by scholarships from China Scholarship Council and University of Essex, U.K.

References

[1] P. Buschka, A. Saffiotti, A virtual sensor for room detection, In: IEEE/RSJ International Conference on Intelligent and Robototic Systems, vol. 1, no. October, 2002, pp. 637–642.
[2] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.A. Fernández-Madrigal, J. González, Multi-hierarchical semantic maps for mobile robotics, In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2005, pp. 3492–3497.
[3] O.M. Mozos W. Burgard Supervised learning of topological maps using semantic information extracted from range data, 2006 IEEE/RSJ Int. Conf. Intell. Robot. Syst. 2006 2772 2777.
[4] E. Brunskill T. Kollar N. Roy Topological mapping using spectral clustering and classification, In: Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 3491–3496.
[5] J. Wu, H.I. Christensen, J.M. Rehg, Visual place categorization: Problem, dataset, and algorithm, In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009, 2009, pp. 4763–4770.
[6] G. Schindler, C. Metzger, T. Starner, A wearable interface for topological mapping and localization in indoor environments, Location-and Context-Awareness, Springer, Berlin Heidelberg67–73.
[7] S. Grzonka, F. Dijoux, A. Karwath, W. Burgard, Mapping indoor environments based on human activity, In: Proceedings of IEEE International Conference on Robotics and Automation, 2010, pp. 476–481.
[8] A. Diosi, G. Taylor, L. Kleeman, Interactive SLAM using laser and advanced sonar, In: Proceedings of IEEE International Conference on Robotics and Automation, vol. 2005, no. April, 2005, pp. 1103–1108.
[9] C. Nieto-Granda, J.G. Rogers, A.J.B. Trevor, H.I. Christensen, Semantic map partitioning in indoor environments using regional analysis, In: Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 1451–1456.
[10] G.-J.M. Kruijff, H. Zender, P. Jensfelt, H.I. Christensen, Clarification dialogues in human-augmented mapping, In: Proceeding of the 1st ACM SIGCHI/SIGART Conference on Human–robot Interaction, HRI'06, 2006, pp. 282.
[11] H. Zender, O. Martínez Mozos, P. Jensfelt, G.J.M. Kruijff, W. Burgard, Conceptual spatial representations for indoor mobile robots, Rob. Auton. Syst. 56 (6) (2008) 493–502.
[12] G. Randelli, T.M. Bonanni, L. Iocchi, D. Nardi, Knowledge acquisition through human–robot multimodal interaction, Intell. Serv. Robot. 6 (1) (2013) 19–31.
[13] A. Pronobis, P. Jensfelt, Large-scale semantic mapping and reasoning with heterogeneous modalities, In: Proceedings of IEEE International Conference on Robotics and Automation, 2012, pp. 3515–3522.
[14] T. Kollar, V. Perera, D. Nardi, M. Veloso, Learning environmental knowledge from task-based human–robot dialog, In: Proceedings of IEEE International Conference on Robotics and Automation, 2013, pp. 4304–4309.

[15] E. Bastianelli, D.D. Bloisi, R. Capobianco, F. Cossu, G. Gemignani, L. Iocchi, D. Nardi, On-line semantic mapping in 2013, In: 16th International Conference on Advanced Robotics, ICAR 2013, 2013.
[16] I. Kostavelis, A. Gasteratos, Semantic mapping for mobile robotics tasks: a survey, Rob. Auton. Syst 66 (2015) 86–103.
[17] M. Quigley, K. Conley, ROS: an open-source robot operating system, ICRA Workshop on Open Source Software, vol. 3, no. 3.2, 2009, pp. 5.
[18] Cheng Zhao's Youtube. [Online]. Available: ⟨https://www.youtube.com/channel/UC-Tl5R4MBKlXsGI0sFlJuXQ⟩ (accessed 01.05.15).
[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, A. Blake, Real-time human pose recognition in parts from single depth images, Stud. Comput. Intell. 411 (1) (2013) 119–135.
[20] IFLYTEC. [Online]. Available: ⟨http://www.iflytek.com/en/audioengine/list_3.html⟩ (accessed: 01.03.13).
[21] Ekho [Online]. Available: ⟨http://www.eguidedog.net/ekho.php⟩ (accessed 01.03.13).
[22] L.R. Rabiner, Tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[23] G. Hinton, L. Deng, D. Yu, G. Dahl, A.R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[24] C. Zhao, W. Pan, O. Hu, Interactive indoor environment mapping through visual tracking of human skeleton, Int. J. Model. Identif. Control 20 (4) (2013) 319.
[25] G. Grisetti C. Stachniss W. Burgard Improving grid-based SLAM with Rao-blackwellized particle filters by adaptive proposals and selective resampling, In: Proceedings of IEEE International Conference on Robotics and Automation, no. April, 2005, pp. 2432–2437.
[26] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with Rao-blackwellized particle filters, IEEE Trans. Robot. 23 (1) (2007) 34–46.
[27] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots, Artif. Intell. 128 (1–2) (2001) 99–141.
[28] S. Sun, T. Li, T.P. Sattar, Adapting sample size in particle filters through KLD-resampling, Electron. Lett. 49 (12) (2013) 740–742.
[29] J. King, M. Likhachev, Efficient cost computation in cost map planning for non-circular robots, Intell. Robots Syst. (2009) 3924–3930.
[30] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, K. Konolige, The office marathon: robust navigation in an indoor office environment, In: Proceedings of IEEE International Conference on Robotics and Automation, 2010, pp. 300–307.
[31] 2D Costmap [Online]. Available: ⟨http://wiki.ros.org/costmap_2d?distro=fuerte⟩ (accessed: 01.03.14).
[32] B.P. Gerkey, K. Konolige, planning and control in unstructured Terrain, In: ICRA Workshop on Path Planning on Costmaps, 2008.
[33] slam_Gmapping [Online]. Available: ⟨http://www.ros.org/wiki/gmapping⟩ (accessed 01.03.13).
[34] Stage [Online]. Available: ⟨http://wiki.ros.org/stage⟩ (accessed 01.03.13).
[35] Rviz [Online]. Available: ⟨http://wiki.ros.org/rviz⟩ (accessed: 01.03.13).
[36] D. Fox, S. Thrun, W. Burgard, Probabilistic Robotics, MIT press, Cambridge, 2005.