A Brief Review of Simultaneous Localization and Mapping

Zhiwei Kong and Qiang Lu
School of Automation, Hangzhou Dianzi University, Hangzhou, China
Email:162060161@hdu.edu.cn and lvqiang@hdu.edu.cn

***Abstract*—This paper reviews the development history of simultaneous localization and mapping (SLAM) and concentrates on two mainstream methods: the filter-based method and the vision-based graph optimization method. FastSLAM and Real-Time Appearance-Based Mapping (RTAB-MAP) as two examples are adopted in the real experiments. The experiments are implemented on TurtleBot with Kinect in a small laboratory and a large circular corridor. The experimental results show that the error is small in the small laboratory, but in the highly unknown large scale environment, the loop closure detection is less effective and the error accumulation is obvious. The results show the accuracy and robustness of two algorithms need to be further improved when robots are in the large-scale unknown environments.**

***Index Terms*—SLAM, FastSLAM, RTAB-MAP, Loop closure detection**

I. Introduction

SLAM has gotten increasingly attention in the past 30 years. SLAM plays a key part in the fields of driverless vehicle, autonomous mobile robot and unmanned aerial vehicle, and attracts a large number of researchers. Meanwhile, SLAM has become an important subject in the field of robotics in the 21st century. Essentially, SLAM aims at estimating a mobile agent's pose in an unknown environment and mapping the environment that includes many features called landmarks. SLAM has become an important function of robot perception of the environment.

SLAM has formed two mainstream approaches: the filter-based method and the vision-based graph optimization method which are also recognized as filtering SLAM and Visual-SLAM (VSLAM), respectively. The filtering SLAM approach was first proposed by Smith [1] in 1986. Their approach modeled the environment by Extended Kalman Filters (EKF), estimated the relative position between the robot and landmarks, filtered the Gaussian noise in the observation data. EKF was the most popular method in earlier studies. Actually, the earliest real-time VSLAM also used EKF to compute the change of sensors' spatial pose [2]. Then, FastSLAM [3] and Particle Filter [4] began to attract attention gradually. The two approaches take the robot's pose, the sensors' input data and the observed data of last moment into account to estimate robot's current pose by recursive Bayesian estimation. The difference of various filtering methods comes from the different assumptions about the prior probability and the different solutions to the posterior probability. Filtering methods include EKF, particle filtering, maximum likelihood estimation and so on. Moreover, with the development of the research, many new methods have appeared such as Rao-Blackwellised Particle Filter [4], extended information filter, etc.

The graph optimization method springed up around 2004. This method mostly adopted Bundle Adjustment [5] to solve the problem of pose change. Bundle Adjustment was applied in the fields of structure from motion (SfM) [6] which aims at minimizing the error arising from pose projection transformation between the observed image points and the predicted image points. Graph optimization method [7] consists of graph construction and graph optimization. Graph construction is the front-end of the optimization method and is used to estimate the movement of the robot by the change of the images. While the graph optimization belongs to the back-end, which is used to optimize the robot's pose. Since all the observation and control information in the robot movement are used to optimize the whole trajectory and the result of the map building, this approach is also called Full SLAM.

The commonly used graph optimization methods include nonlinear least square method [8], relaxation optimization method [9], stochastic gradient-descent [10]. Depending on the camera types, VSLAM is devided into three classes: MonoSLAM, stereo-SLAM and RGB-D SLAM. In recent years, many RGB-D 3D-SLAM methods based on graph optimization have emerged. Compared with ordinary cameras, RGB-D camera can capture the depth information of the image. It not only saves time for extracting deep data from 2D data which improves the real-time performance of the algorithm, but also builds an environment map accurately. Compared with the filter method, the graph optimization method has obvious advantages in robustness, consistency and accuracy. And the graph optimization method is better for loop closure detection. However, when the robot in a highly uncertain environment, the filter method may be better than graph optimization method. Thus, how to make full use of the advantages of the two kind of methods in their respective environments and scheme out the SLAM system consisting of both Bundle Adjustment and probabilistic filtering will be the development trend in this field.

II. SLAM Algorithms

Essentially, SLAM is a posterior probability estimation problem. SLAM algorithms can be divided into filter based

978-1-5386-1127-2/17/$31.00 ©2017 IEEE

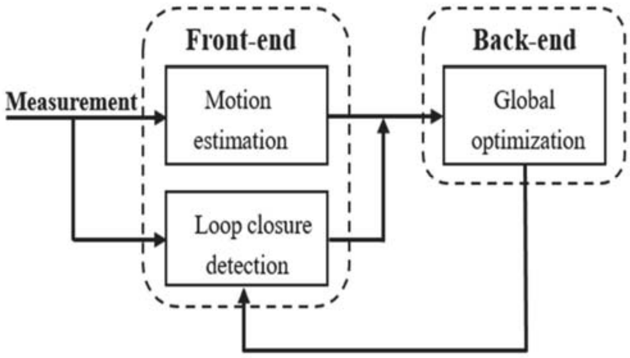

Fig. 1. Graph optimization algorithm structure

algorithms and graph optimization based algorithms. At the beginning, SLAM was solved by filter methods. But, in the 21st century, SLAM researchers began to draw lessons from SfM and used graph optimization methods to solve SLAM.

A. Filter based algorithm

Most of the SLAM solutions are based on probabilistic filtering algorithms that including EKF, particle filter (PF), maximum likelihood (ML) estimation and expectation maximization (EM) algorithm, etc. EKF and PF are the most commonly used algorithms. However, these methods are relatively successful in small space. The filter algorithms are limited when navigating in large-scale environment because of linearization and update efficiency.

Kalman filter (KF) is the optimal linear filter that can estimate the internal state of a dynamic system from noise-filled measurements. However, since KF can only be used in linear systems, EKF is proposed to solve the probabilistic estimation problem in nonlinear systems. But EKF-SLAM is sensitive to erroneous data associations, since an incorrect measurement makes the filter to diverge. In addition, it is difficult for EKF used in a large-scale map because EKF-SLAM has a quadratic calculation process for landmarks.

Gristetti [4] proposed a Rao-Blackwellized Particle Filter (RBPF) which has well scalability for maps with many landmarks. Therefore, data associations errors can be solved efficiently by RBPF. FastSLAM is one of the most typical examples of RBPF. FastSLAM treats the robot position distribution as a set of Rao-Blackwellized particles, and each particle represents a robot trajectory and uses EKF to maintain local map such that the computational complexity of SLAM is greatly reduced.

B. Graph optimization based algorithm

In recent years, due to the development of the direct linear solver, graph optimization algorithm have been widely used. Commonly used optimization methods include relaxation method, gradient descent method, flow method, etc.

The graph optimization algorithm is composed of three parts: motion estimation, loop closure detection and graph optimization. Motion estimation and loop closure detection constitute the front-end part, and graph optimization belongs to the back-end part. The algorithm structure is showed in Fig.1.

The graph optimization algorithm uses motion estimation and loop closure detection for data association, and then graph construction can be completed. The purpose of the front-end part is to obtain the constraint relationship of robot's pose graph nodes. And the back-end part is used to optimize pose graphs. The current graph optimization algorithms include ORB-SLAM, LSD-SLAM, RTAB-MAP, etc.

III. Classical Algorithms

A. FastSLAM algorithm

FastSLAM is a very classical algorithm in the existing 2D-SLAM projects. The core of the algorithm is the RBPF that reduces the dimension of the estimation by decomposing the state space of the Bayesian filter. The core formula of the FastSLAM can be expressed as follows.

$$\begin{aligned} Bel(x_{1:t}, m) &= p(x_{1:t}, m | z_{1:t}, u_{1:t-1}) \\ &= p(m | x_{1:t}, z_{1:t}) \cdot p(x_{1:t} | z_{1:t}, u_{1:t-1}) \end{aligned} \quad (1)$$

where $x_{1:t}$ is the robot trajectory; $u_{1:t}$ is the odometer information; m is the map; $z_{1:t}$ is the observation information. The joint posterior probability is decomposed into the product of two posterior probabilities by this equation. $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ represents the posterior probability of the path which can be estimated by the particle filter. $p(m|x_{1:t}, z_{1:t})$ represents the posterior probability of the map under a given path which can be computed by the Kalman filter after $p(x_{1:t}|z_{1:t}, u_{1:t-1})$ is given. Actually, RBPF, shown in Table I, can be divided into two parts: particle filter and Kalman filter. The particle filter approximates the probability density function by finding a set of random samples and uses the sample mean instead of the integral operation to obtain the minimum variance estimate of the system state which is a sequential importance sampling method. By the importance sampling principle, the probability of the ith sample can be expressed by an importance weight ω_t^i.

$$\tilde{\omega}_{x_{1:t}} = \frac{p(x_{1:t}|z_{1:t}, u_{1:t-1})}{\pi(x_{1:t}|z_{1:t}, u_{1:t-1})} \quad (2)$$

where $\pi(x_{1:t}|z_{1:t}, u_{1:t-1})$ is the proposal distribution. The normalized weights are:

$$\omega_t^{(i)} = \frac{\tilde{\omega}\,(x_{1:t}^{(i)})}{\sum_{k=1}^{N} \tilde{\omega}\,(x_{1:t}^{(k)})} \quad (3)$$

The posterior probability distribution $p(x_{1:t}, m|z_{1:t}, u_{1:t-1})$ can be calculated by $\omega_t^{(i)}$ and $\{x_{1:t}^{(i)}, m^{(i)}\}$. Suppose that N weighted samples $\{x_{1:t}^{(i)}, \omega_t^{(i)}\}_{i=1}^{N}$ are used to represent the posterior probability distribution of the path $p(x_{1:t}|z_{1:t}, u_{1:t-1})$.

$$P_N(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta_{x_{1:t}^{(i)}, m}(x_{1:t}, m) \quad (4)$$

TABLE I
THE STANDARD STEPS OF RBPF

Steps	Approaches
Step1	Sampling from a prior distribution $P(x_t\|x_{t-1}^{(i)})$ by Monte Carlo method.
Step2	Sequential importance sampling. Compute $p(x_{1:t}\|z_{1:t}, u_{1:t-1})$ and weight $\widetilde{\omega}_t^{(i)}$.
Step3	Resampling. Discard low weight particles and retain high weight particles.

$$P_N(x_{1:t}|z_{1:t}, u_{1:t-1}) = \sum_{i=1}^{N} \omega_t^{(i)} \delta_{x_{1:t}^{(i)}}(x_{1:t}) \quad (5)$$

where $\delta(\cdot)$ is the impulse function. Therefore, the posterior probability of the map can be expressed as

$$P_N(m|z_{1:t}, u_{1:t-1}) = \sum_{i=1}^{N} \omega_t^{(i)} p(m|z_{1:t}, u_{1:t-1}, x_{1:t}^{(i)}) \quad (6)$$

In the RBPF mapping algorithm, each particle contains robot's trajectories and environment map information which are expressed by mean and variance, respectively. Both mean and variance can be obtained by Kalman filter.

B. RTAB-MAP algorithm

According to the number of map points, 3D SLAM is divided into three classes: the sparse map, the semi-dense map and the dense map. In the 3D reconstruction experiment, the RTAB-MAP, which is a classic dense RGB-D SLAM solution, is adopted. This approach is available in wiki ROS.

The purpose of RTAB-MAP is to provide a solution that is based on localization and graph composition and independent of time and scale. The idea of the solution is to satisfy some real-time constraints, and the loop closure detection uses only a limited number of location points, which are based on the assumptions that more frequent access points are more possible to form loop closure than other location points. RTAB creates the signature for the image by BoW. The signature is expressed by a set of words in the dictionary. The advantage of using an online incremental dictionary is that there is no need for pre-training for a particular environment. RTAB uses OpenCV to extract features from pictures to obtain visual words and each visual word is a description of the SURF feature which is expressed as a 64 dimensions vector. In order to construct a good signature and find the match between the words in the dictionary, RTAB-MAP uses the nearest neighbor distance method in the comparison of the SURF features. The process of RTAB is shown in Table II.

The RTAB-MAP uses memory management to maintain a graph. This kind of memory management is achieved by detecting picture feature points and can be online managed. The memory consists of Short-Term Memory (STM), Working Memory (WM) and Long-Term Memory (LTM). The STM is the entry point that adds the new node to the graph whenever new data is received. And STM has a fixed size S. When the size of STM reaches S nodes, the node with longest storage time will be moved to WM for loop closure detection.

The loop closure hypothesis relies heavily on Bayesian filters. A loop closure probability is estimated by the current location point and the location point stored in the WM. Let S_t be a random variable at time t. $S_t = i$ denotes that the L_t is a loop closure with an already visited point L_i, that is, L_t and L_i are the same location point. $S_t = -1$ denotes a new location point. The filter estimates all posterior probability distributions $p(S_t|L^t)$ ($i = -1, \ldots, t_n$ where t_n is the time index of the latest location point stored in WM).

$$p(S_t|L^t) = \rho p(L_t|S_t) \sum_{i=1}^{t_n} p(S_t|S_{t-1} = i) p(S_{t-1} = i|L^{t-1}) \quad (7)$$

where ρ is a standardized coefficient, $L^t = L_{-1}, \ldots, L_t$ is a sequence of location points and only contains location points from WM and STM. L^t changes such that it is different from the classic Bayesian filter that L^t is a fixed length sequence.

$p(S_t|L_t)$ is the observation model which is used to measure the similarity between S_t and L_t. A likelihood function $\gamma(S_t|L_t)$ is used to distinguish the similarities between the different location points. And $p(S_t|L_t = j)$ can be calculated as follows.

$$\begin{aligned} p(L_t|S_t = j) &= \gamma(S_t = j|L_t) \\ &= \begin{cases} \frac{S_j - \sigma}{\mu} & \text{if } S_j \geq \mu + \sigma \\ 1 & \text{otherwise} \end{cases} \end{aligned} \quad (8)$$

where μ is the similarity mean; σ is the standard deviation. If L_t is a new location, the possibility is computed as follows.

$$p(L_t|S_t = -1) = \gamma(S_t = -1|L_t) = \frac{\mu}{\sigma} + 1 \quad (9)$$

The calculation is related to the ratio between the mean and the standard deviation of the similarity. L_t is a new location point if $\gamma(S_t = -1|L_t)$ is quite large.

$p(S_t|S_{t-1} = i)$ is used to predict the distribution of S_t under the condition of knowing the distribution of S_{t-1}. This is similar to predict the motion of robots from $t-1$ to t moment. Together with the $p(S_t|S_{t-1} = i)$, the confidence of the next loop closure detection is constructed. $p(S_t|L^t)$ is calculated and normalized. If the $p(S_t = -1|L^t)$ is smaller than the given threshold, then the loop closure assumption with the maximum value in $p(S_t|L^t)$ is considered valid.

When a loop closure assumption is established, the new location point L_t and former location point L_i form a closure loop. The weight of L_t is updated based on the former weight plus the weight of L_i. The effect of closure loop is to find the adjacent nodes of the former location point and calculate the transfer model used in the Bayesian filter. With the completion of the loop closure detection, if the location points with the highest loop closure probability have adjacent points and these adjacent points are not in WM, their adjacent points will

TABLE II
THE MAIN PROCESS OF RTAB ALGORITHM

Steps	Process
Step1	Create location points L_t with time index t and image. signature which is generated by BoW.
Step2	Update weights. Compare L_t with the last location point in STM.
Step3	Update Bayesian filter that records closure-loop assumptions by estimating probability of forming closure-loop.
Step4	Loop closure hypothesis selection.
Step5	The adjacent point of the location point with the highest probability of closure is retrieved from LTM to WM.
Step6	The longest stored point in the minimum weight is transferred from LTM to WM.

be taken back from LTM to WM. However, if the image processing time exceeds threshold T_h, the longest stored point in the minimum weight will be transfered from WM to LTM. These steps are the memory management process of RTAB-MAP in the loop closure detection.

IV. EXPERIMENTAL RESULTS

The experiment was implemented in our laboratory and larger circular corridors. In our experiments, the hardware platform we used include Kinect, TurtleBot and computer with i5 CPU@2.6 GHz and 4 GB RAM. Turtlebot is a relatively inexpensive robot development platform made by Willow Garage. TurtleBot is equipped with Kobuki mobile chassis, Kinect vision sensors, 2200mAh batteries and other loading and unloading modules. As shown in Fig.2, the top of the TurtleBot is Kinect and the bottom is Kobuki mobile chassis. Kinect has three lenses where the middle lense is the RGB camera, which is used to collect color images. The left and right lenses are infrared transmitter and infrared CMOS camera, respectively.

Moreover, TurtleBot is also equipped with the most popular Robot Operating System (ROS). ROS is an open source operating system for robots. It provides some tools for programs and libraries to acquire, build, write and run multi-machine integration programs. The ROS version adopted in this experiment is Indigo Igloo which installed on the Ubuntu Trusty (14.04 LTS).

We first implemented experiments in our laboratories. The experimental results of 2D and 3D are shown in following figures. FastSLAM experimental results generate only 2D map while RTAB-MAP generate 3D points cloud map and 2D map using the visual odometry data input from Kinect. In the 2D mapping experiment, we still use Kinect to simulate the laser sensor.

The laboratory's real environment and the 3D points cloud map have been shown in Fig.3 and Fig.5, respectively. The experiment was implemented with handheld Kinect. The loop closure detection is realized by matching the feature points, the feature points are marked with green and red points on the right side of the Fig.4. The blue line is the trajectory of the Kinect. As you can see from the 3D points cloud in the Fig.5, the laboratory environment is clearly reconstructed.

Fig. 2. TurtleBot with Kinect

Fig. 3. Laboratory environment

But, from the experimental results, we can see that RTAB is not perfect. Some objects in the map show ghosting and the trajectory is not precise enough. The 2D laboratory map generated by FastSLAM is shown in Fig.6. As the laboratory piled up many obstacles, the map presents an irregular pattern. Compared to the 3D points cloud map, the 2D map doesn't have rich environmental information. Hence, it is obvious that 2D map has unique advantages. FastSLAM is faster and requires less computing resources than complex 3D reconstruction. In addition, 2D map is sufficient to achieve localization for indoor mobile robots.

The the corridor environment is shown in Fig.7. The results of corridor experiments are shown in Fig.8. The top view of corridors is shown in Fig.9. The white part in the figures is seriously affected by sunlight, such that Kinect can not get complete environment information. The 2D map in Fig.10 is

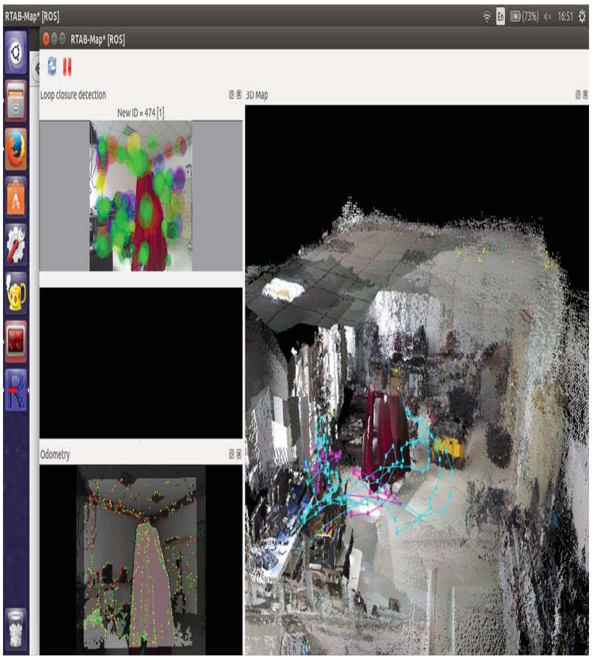

Fig. 4. Feature points detection

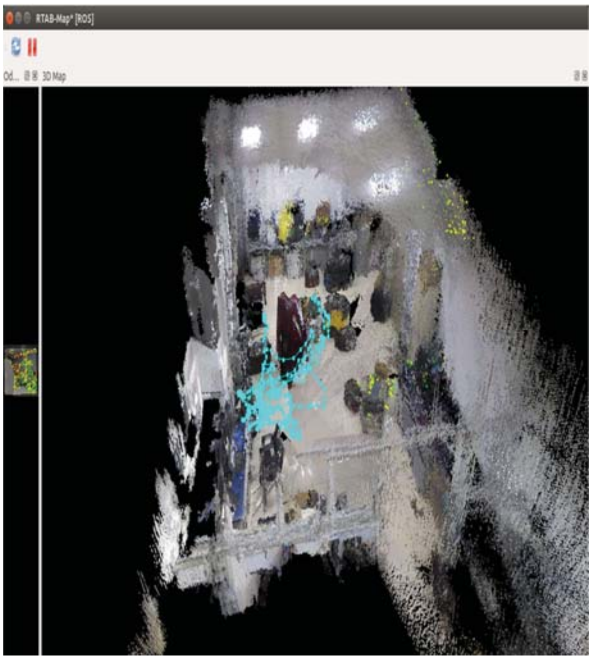

Fig. 5. Laboratory 3D points cloud map

the result of removing the 3D points cloud from Fig.8. As you can see from the figures, the corridors can not be well closed and the trajectory of the corridor is partial curving. The experimental results demonstrate that the robot's pose and trajectory estimation are not accurate enough and loop closure detection is not satisfactory in this kind of highly unknown large scale environment.

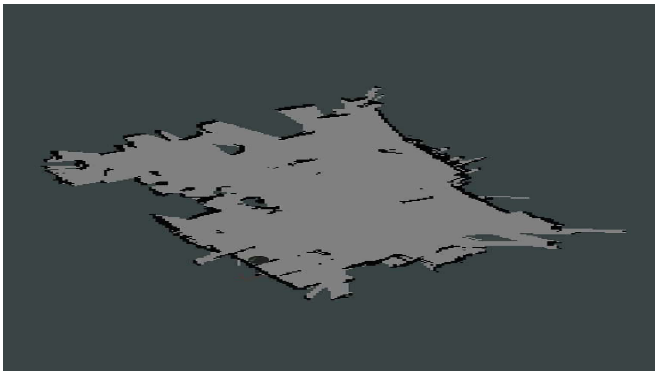

Fig. 6. FastSLAM laboratory 2D map

Fig. 7. Corridors environment

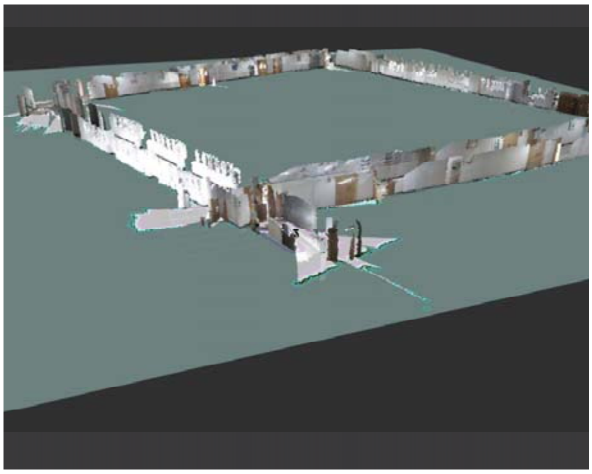

Fig. 8. Corridors 3D points cloud

V. Conclusion

The two mainstream SLAM methods we have implemented demonstrate that the indoor environment with rich feature points is easier for loop closure detection than corridor with less feature points. In the indoor environment 3D reconstruc-

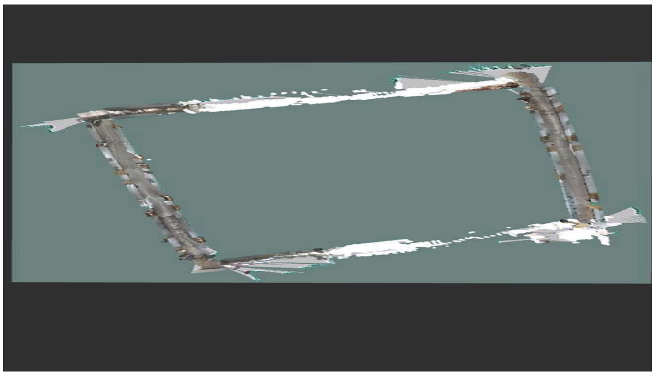

Fig. 9. Top view of corridors 3D points cloud

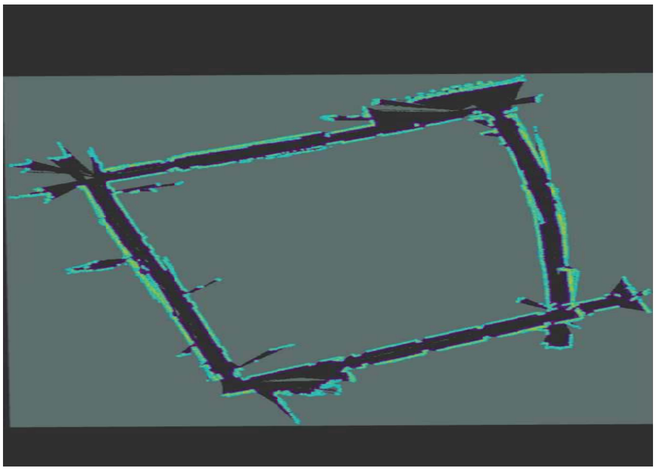

Fig. 10. RTAB-MAP corridors 2D map

tion experiment, some objects show ghosting in the map. And the robot's motion trajectory error accumulates greatly and loop closure detection is not accurate enough in the large-scale unknown environment. The next work will concentrate on improving the accuracy of loop closure detection and reduce the robot's motion trajectory errors in large-scale environment.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant 61375104.

REFERENCES

[1] R. Smith, M. Self, P. Cheeseman, Estimating uncertain spatial relationships in robotics[M]. *Springer-Verlag New York*,1990, 4(5):167-193.
[2] S. Se, D. Lowe, J. Little, Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks[J]. *The international Journal of robotics Research*, 2002, 21(8):735-758.
[3] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem[C]. *Eighteenth National Conference on Artificial Intelligence*, 2002, pp.593-598.
[4] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with rao-blackwellized particle filters[J]. *IEEE Transactions on Robotics*, 2007, 23(1):34-46.
[5] B. Triggs, P.F. McLauchlan, R.I. Hartley, A.W. Fitzgibbon, Bundle adjustment - a modern synthesis[C]. *International workshop on vision algorithms*, 1999, 1883(1883):298-372.
[6] J.J. Koenderink, A.J. Van Doorn, Affine structure from motion[J]. *Journal of the Optical Society of America A Optics and Image Science*, 1991, 8(2):377-385.
[7] N. Snderhauf, P. Protzel, Towards a robust back-end for pose graph slam[C]. *IEEE International Conference on Robotics and Automation(ICRA)*, 2012, 24(7):1254-1261.
[8] F. Dellaert, M. Kaess, Square Root SAM: Simultaneous localization and mapping via square root information smoothing[J]. *The International Journal of Robotics Research*, 2006, 25(12): 1181-1203.
[9] U. Frese, P. Larsson, T. Duckett, A multilevel relaxation algortithm for simultaneous localization and mapping[J]. *IEEE Transactions on Robotics*, 2005, 21(2):196-207.
[10] G. Grisetti, C. Stachniss, S, Grzonka, W. Burgard, A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent[C]. *Robotics: Science and Systems*, 2007, pp.27-30.
[11] M. Bryson, S. Sukkarieh, Inertial Sensor-Based Simultaneous Localization and Mapping for UAVs[J]. *Springer Netherlands*, 2014, pp.401-431.
[12] H. Sankrit, B.J. Panwala, P. Mudgal, Indoor SLAM using Kinect Sensor[J]. *International Journal of Science Technology and Engineering(IJSTE)*, vol.2, Issue 10, 2016, pp.1226-1231.
[13] S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots[J]. *Artificial Intelligence*, 2001, 128(1-2):99-141.
[14] G. Tian, Q. An, C. Ji, B. Gu, H. Wang, J. Zhao, Simultaneous localization and mapping based on Gray EKF for intelligent agricultural vehicle[J]. *Transactions of the Chinese Society of Agricultural Engineering*, 2012, 28(19):17-25.
[15] T. Dumont, S.L. Corff, Online EM for indoor simultaneous localization and mapping[C]. *IEEE International Conference on Acoustics*, 2017:6431-6435.
[16] Q. Lu, Q. Han, S. Liu, A cooperative control framework for collective decision on movement behaviors of particles[J]. *IEEE Transactions on Evolutionary Computation*, 2016, 20(6):859-873.
[17] C. Yuuto, T. Kanji, A. Masatoshi, Part-based SLAM for partially changing environments[C]. *IEEE International Conference on Robotics and Biomimetics*, 2013, pp.1629-1634.
[18] F.R. Fabresse, F. Caballero, L. Merino, A. Ollero, Active perception for 3D range-only simultaneous localization and mapping with UAVs[C]. *International Conference on Unmanned Aircraft Systems*, 2016, pp.394-398.
[19] X. Zhao, S. Zhang, Distributed strong tracking unscented particle filter for simultaneous localization and mapping[C]. *Control Conference*, 2014, pp.978-983.
[20] B. Brown, E. Laurier, The Trouble with Autopilots: Assisted and Autonomous Driving on the Social Road[C]. *Chi Conference on Human Factors in Computing Systems*, 2017, pp.416-429.
[21] Y. Hu, J. Gong, Y. Jiang, G. Xiong, A sub-map-based simultaneous localization and mapping technique for intelligent vehicles[J]. *Qiche Gongcheng/automotive Engineering*, 2015, 37(2):224-229.
[22] T. Sato, K. Sakaki, H. Arisawa, Construction of Motion Evaluation/Simulation System Based on Musculoskeletal Human Body Description[C]. *International Joint Conference on Artificial Intelligence*, 2003, 133(1):1151-1156.
[23] J. Yang, J. Zhang, G. Wang, M. Li, Semantic Map Building Based on Object Detection for Indoor Navigation[J]. *International Journal of Advanced Robotic Systems*, 2015, 12:1.
[24] N.K. Dhiman, D. Deodhare, D. Khemani, Where am I? Creating spatial awareness in unmanned ground robots using SLAM: A survey[J]. *Sadhana*, 2015, 40(5):1385-1433.