A new metric for assessing the performance of 2D Lidar SLAMs

Bingxin Zi, Haiying Wang, Jose Santos and Huiru Zheng

School of Computing, Ulster University, Belfast, BT37 0QB, UK
hy.wang@ulster.ac.uk

Abstract. Simultaneous Localisation and Mapping (SLAM) is a widely studied topic in recent years and has a wide potential in the field of unmanned driving and robotics. Over the past decade, a number of SLAM algorithms have been developed, each exhibiting unique performance in their applications. This paper presents a comparative study of the performance of three well-known Light Detection and Ranging (LiDAR)-based SLAM algorithms, i.e. Gmapping, Cartographer and Hector, with an emphasis on the 2D maps constructed by each algorithm. In order to deal with incomplete maps constructed, a new evaluation metric was proposed. To reduce the human error during scene construction and equipment calibration, all experiments were carried out in the 2D simulation available within the Robot Operating system (ROS). Three well-designed maps with different sizes and complexities were introduced to investigate the features of three SLAM algorithms. Besides, to reduce the impact of randomness, each dataset was assessed 10 times to obtain the mean value and the standard deviation. The results show that, in comparison to traditional metrics such as a metric of average distance to nearest neighbour (ADNN), the proposed measurement can clearly reflect both the quality and completeness of maps built by SLAM algorithms.

Keywords: SLAM, assessment metric, 2D LiDAR.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is one of the most widely studied topics across multiple subjects including robotics, computer vision and machine learning. In the field of robotics, it gives robots the ability to explore an unknown environment by constructing a map [1] and understand the environment by processing the information from visual sensors [2]. SLAM has been applied in unmanned vehicles, autonomous driving, augmented reality (AR) among other [2]. According to different usage environment, SLAM may apply different algorithms and sensors. With the development of technology, machine learning techniques can also be used to help SLAM process the sensor information [3][4].

SLAM techniques can be divided into two main types, filter-based and graph-based. The filter-based SLAM has become the mainstream over the past decades to

This work is supported by the VCRS Research Student Award from Ulster University.

achieve localisation and mapping. The algorithm examples used in SLAM applications include MonoSLAM [5] and Gmapping [6]. MonoSLAM uses an Extended Kalman Filter (EKF) framework to realise the robot's status estimation and mapping. The EKF is a nonlinear extension of the Kalman filter which contains a status equation and an observation equation. The status equation is used for predicting the robot's current status based on the status at the last timestamp, while the observation equation is used for correcting the prediction by taking the sensor observation into account. The Gmapping algorithm applies a particle filter, in which each particle represents a potential trajectory of the robot, to handle the localisation and mapping. Each particle is associated with a weight and stands for a possible status of the robot. The final estimation is determined by the weighted mean of all particles. In recent years, graph-based SLAM has received much attention [7-10]. In this system, there are two main processes: frontend and backend. The frontend processes sensor data and calculates the dynamics of the robot while the backend receives the dynamics and generates a fusion result. At the same time, the backend is also responsible for the optimisation of the whole system, which is a procedure that the filter-based SLAM does not have. The Cartographer algorithm [10] is the classic implementation of graph-based SLAM approaches.

Each of these algorithms has its advantages and limitations. The main problem arising from their use in robotics is how to evaluate their performance. Attentions have been traditionally focused on the assessment of the accuracy of the generated maps and trajectories only. Examples include the use of a metric like average distance to the nearest neighbour (ADNN)[11-14], or evaluating the trajectory in a probabilistic approach[15]. At the same time, a ground truth independent evaluation was introduced in [16], which counts the features of estimated map to independently assess the quality of map.

This study proposes a new metric which would evaluate the SLAM's map results in terms of both the accuracy and the structural completion of the maps derived. Three well-known SLAM systems (Gmapping, Cartographer and Hector) were studied with an emphasis on the analysis of the 2D maps constructed by each algorithm.

The remainder of this paper is structured as follows: the related work is presented in Section 2, followed by a description of SLAM algorithms in Section 3. In Section 4, the experiment environment is introduced. Evaluation metrics used in the study are described in Section 5 and experiment results are analysed in Section 6. The paper concludes with a summary of conclusion and future work discussion.

2 Related works

It could be challenging to analyse and evaluate the performance of various SLAM systems because they have different sensor sources, algorithm basements and code frames. For example, Gmapping utilises light detecting and ranging (LiDAR) and an

odometer to calculate the journey of the robot and to construct the map. To apply SLAM to aircraft navigation, instead of using an odometer, Hector [17] combines 2D LiDAR with an inertial measurement unit (IMU). The Cartographer algorithm released in 2016 by Google [10] combines data from various sensors e.g. LiDAR, IMU, and odometer. In order to provide a uniform platform for implementing the multiple types of robot algorithms, the Robot Operating System (ROS) framework [18] was introduced in 2007. It defines a uniform standard of interactions among different robot subsystems. Under the uniformed framework, the different SLAM systems employ the common data inputs, data flows and outputs standard. This makes it possible to compare the performance of different SLAM algorithms under the same pattern. This paper focuses on evaluating the performance of the LiDAR-based 2D SLAMs including Hector, Gmapping and Cartographer implemented in ROS.

Normally, the evaluation of SLAM results performance can be carried on two aspects: the trajectory and the map quality. The trajectory is often evaluated in the vision SLAM (VSLAM) because the map exists in multiple forms (sparse map, dense map, semi-dense map). In a VSLAM system, it is hard to directly compare the quality of a map derived from different systems, for example, Buyval et al. [19] compared the ability of 4 different VSLAM systems to obtain features and the coverage of point clouds but did not perform any quantitative analysis. On the aspect of map comparison, Xu et al.[20] projected the feature points detected by the camera onto the ground to fake the LiDAR laser scans. Then they generated the occupied grid map by using the pseudo laser scans. Therefore, they were able to compare different type of VSLAM maps by projecting the points onto ground. However, the noises points in the space may also be projected onto the ground, thereby forming errors. Some extra procedures are required to provide ground truth data. Yagfarov et al.[11] introduced the high-precision laser tracker FARO to manually construct a ground truth map. They used FARO to get the 3D scans of the experiment room. Then they extracted the intersection lines between the floor plane and the vertical plane. Those lines were viewed as the 2D projections of the experiment room. Due to the lack of wall width information, they used some OpenCV functions to thin the SLAM estimated map's wall width to one grid. However, lacking wall width information may introduce errors. Sturm et al.[12] introduced an extra high-frequency motion capture system to provide ground truth. Filipenko and Afanasyev[13] used a much simpler way to get the ground-truth. They laid some threads on the floor. The robot was manually driven to follow the treads to obtain the estimations. However, there is no guarantee that the actual robot trajectory will perfectly fit these threads, especially in the turning areas. So they only used the straight sections of the threads for comparison. Since they found the Hector's result was closest to the ground truth, they used Hector's result as reference for the other systems. It should be mentioned that they did not get a valid Gmapping result. Santos et al. [14]implemented the evaluation in both the simulation environment STAGE and the real physical world. They evaluated Hector, Gmapping, KartoSLAM, CoreSLAM and the LagoSLAM. Their simulation experiments show that "Gmapping algorithm presents exceptional results" while the "KartoSLAM was the best performing technique in real word". The Cartographer algorithm was not

included in their study. Bayer et al.[21] pointed out that the SLAM ground truth is hard to construct. Anton et al. [16] pointed out ground truth data are not always fetchable even for a lot of open datasets. Instead of comparing with ground truth, they proposed some novel metrics: the proportion of occupied and free cell, the number of corners and the number of enclosed areas for independently evaluating the map quality without acquiring the ground truth. Besides, Le et al.[22] introduced a structural metric of Structure Similarity Index (SSIM) [23] to evaluate the map quality of different SLAM systems in the indoor environment. But the SSIM result cannot provide an intuitive feeling of the structure completion.

Since the accuracy of a trajectory relies on the accuracy of maps, this study will focus on the comparison of map quality. Moreover, because it is hard to obtain ground truth information without introducing some extra interferences, such as the calibration of a reference equipment and setting artificial markers, all the experiments in this study have been conducted in 2D simulation available in ROS.

3 2D LiDAR SLAM Algorithms

In this research, three SLAM algorithms: Gmapping, Cartographer and Hector were investigated. They utilise different algorithm frames and different sensors, as summarised below:

3.1 Gmapping

Gmapping [6] is one of the most typical filter-based SLAM algorithms. It is based on the Rao-Blackwellized Particle Filter (RBPF). In RBPF, the problem of estimating a robot pose and map is divided into two steps. In detail, RBPF takes the robot pose estimating problem as an incremental estimation problem. Based on the robot pose on the last frame and the robot dynamics on the current frame, the current robot pose can be predicted. Then the mapping can be solved by providing the current pose and observations. In naïve particle filter system [24], each particle stands for one possible status of the robot and one possible representation of the map. All particles contribute to a weighted mean estimation. To reach a closer representation of the true possibility distribution, the particle filter requires a large number of particles to expand the sampling range. But after several propagation iterations, the particles' weights concentrate on a few specific particles, which makes the other majority of particles barely contribute to the mean result. In Gmapping, a novel way was used to establish a more accurate proposal distribution by taking the laser observation into account. Furthermore, an adaptive metric was applied to guarantee that the resampling process was executed only when the weights' concentration rate was higher than a threshold. These measures allow the Gmapping algorithm to use fewer particles than the naïve system which in turn makes it works in real-time.

3.2 Hector

Hector [17] is a very straightforward algorithm that estimates the robot status by aligning the laser scan and map. Every new scan is transformed into the discrete occupied grid cell by applying the Bresenham algorithm [25]. After that, the approximate optimal transformation between the current scan and existing map is solved by applying a Gaussian-Newtown optimization. Hector accumulates each scan to form a map and to avoid falling into local minima, a multiple-resolution map principle is applied. Besides, Hector is designed to be applied in 3D spaces, so an IMU is equipped instead of the odometer.

3.3 Cartographer

Cartographer [10] is one of the most cited open-source graph-based SLAM algorithms in recent years. The main contribution of the Cartographer algorithm is the realization of the real-time loop-closure in a LiDAR SLAM system. It proposes the principle of submap that consists of recent scans. Once a new scan is acquired, the system will look up recent sub-maps to find a reasonable matching between the scan and a submap. The scan will be inserted in the submap without considering the drift during that short time. Presently, a loop closure detection will be carried out along all submaps to minimise the long-term drift. Besides, Cartographer introduces 3 types of the branch and bound algorithms to speed up the loop closure searching process.

4 Experiment environment set up

The experiments were carried out on the Ubuntu 16.04 distribution with ROS kinetic. The ROS platform provides related ROS packages to apply all the three algorithms mentioned above. In ROS, data transmission is achieved by subscribing to and publishing to certain topics. The topic defines the data structure of interactions. All the three algorithms are edited to subscribe to the same topic of "/laser_scan" to receive the laser scans. Besides, all three algorithms subscribe to the same topic of "/tf". The "/tf" is an essential ROS topic which provides the transformations among each coordinate reference of the whole system. It should also be pointed out that the Gmapping subscribes the "/tf" topic to get the odometer information. While the Hector and Cartographer subscribe to the "/imu" topic to utilize IMU information instead.

Different from Santos et al.'s method [14], the Gazebo_ROS software, instead of Stage_ROS software, was used for creating virtual environments because the Stage_ROS software does not provide the IMU simulation [26]. The algorithms were tested in three types of maps – a rectangle map, a swirl map and a corridor map. For the former two types, each type has three sizes respectively: 3m x 2m, 6m x 4m, 9m x 6m and 3m x 3m, 5m x 5m, 7m x 7m (in meter). The size of the corridor map is 2m x 10 m. Fig. 1 shows the image of the virtual environment and the initial position of the robot.

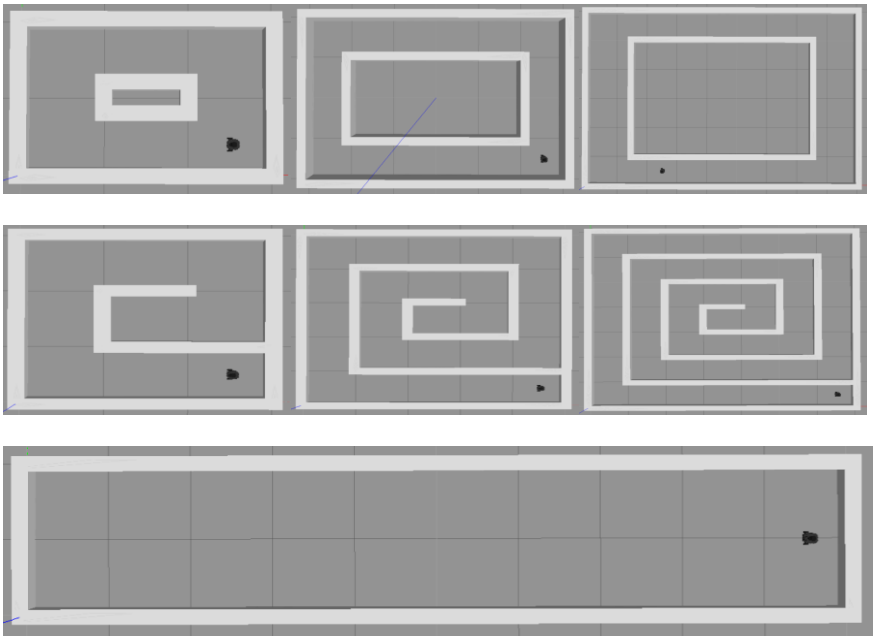


Fig. 1. The 7 simulation environments used in the study

The comparison analysis was carried out using Matlab and for mathematical convenience, each map was transformed to a dot map. The dot map keeps the same resolution – 5cm/pixel as the SLAM algorithms' setting. Therefore, in our experiment, an error of one pixel is equal to an error of 5 cm.

The robot used for simulation is the Turltlebot3 Burger [27] which has the maximum linear speed of 0.22m/s, and the maximum rotation speed of 2.84 rad/s. In the simulation, the robot's forward speed is gradually increased to the peak, but no rotating speed exceeds 0.7 rad/s. The ROS bag command is used for recording all topics' data during the walking. Each algorithm processes these records separately. The map parameters are 0.65 for occupied, 0.2 for free.

It should be noted that, due to the stochastic behaviour of the algorithms, the output from Gmapping and Cartographer could be different even with the same inputs. Therefore, unlike other studies [11][13][14], to mitigate the randomness impact, each algorithm was tested with the same dataset 10 times in this study. The standard deviation of each metric was then calculated along with their mean values.

5 Metric for evaluating

In this paper, two metrics were applied to evaluating the SLAM map quality. The ADNN metric focus on the accuracy of map while the grid portion metric we proposed assesses both the completion and quality of the estimated map.

5.1 Average distance to the nearest neighbour (ADDN)

ADNN is estimated based on the Iterative Close Point (ICP) algorithm which was used to align a SLAM map with the ground truth [11][14] and the K-Nearest Neighbours (KNN) search used for finding the correspondence between the ground truth and the estimated map. The ICP and the KNN iteratively run until the result converges. The metric is computed as the average distance from each SLAM map point to the nearest ground truth map point as defined below:

$$\mathrm{ADNN} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\left(\mathrm{P}_{\mathrm{est}}^{i} - \mathrm{P}_{\mathrm{grd}}^{i}\right) \tag{1}$$

Where n is the total point number of estimated grids, $\mathrm{P}_{\mathrm{est}}^{i}$ is the position of the i^{th} estimated grid and $\mathrm{P}_{\mathrm{grd}}^{i}$ is the position of the i^{th} ground truth grid. To avoid falling into local minima, we manually selected the 4 corner points and their average coordinates as 5 initial parameters for the ICP algorithm.

However, the ADNN metric does not perform well when dealing with incomplete estimated maps. For example, if a SLAM algorithm outputs an incomplete map but every single grid of the map is perfectly located, the ADNN is set to zero which may provide a misleading assessment.

5.2 Grid Portion Metric

Inspired by Anton et al. [16] in which three metrics of “the portion of occupied" and free cell”, “the number of corners” and “the number enclosed” are were proposed to evaluate a SLAM map from the perspective of the grid, we proposed a new metric to evaluate the map quality in terms of grid portion. Specifically, for each ground truth map, the number of occupied grids is constant. For any occupied grids, there are three possible results when aligning a SLAM map with the ground truth: a ground truth occupied grid was correctly identified as occupied in the SLAM map (True Positive (TP)); a ground truth occupied grid was wrongly regarded as free in the SLAM map (False Negative (FN)); a ground truth free grid was wrongly regarded as occupied in the SLAM map (False Positive (FP)). With this definition, the map quality can be evaluated in terms of sensitivity and precisions as defined below:

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP+FP})}, \mathrm{Sensitivity} = \frac{\mathrm{TP}}{(\mathrm{TP+FN})} \tag{2}$$

The range of these values goes from 0 to 1. The precision value describes the quality of the estimated map. If every occupied grid in the estimated map is correct, the preci-

sion value is set to 1. While 0 means every occupied grid in the estimated is wrong. Besides, this metric can also indicate the completeness of the map, where sensitivity value 1 means the estimated map's structural matches the ground truth map in 100%. And 0 means their structures are totally different. At the same time, this metric can provide an intuitive feeling of how correct the estimated map is by drawing each TP/FP/FN grids.

The metric of precision and sensitivity were widely applied to evaluating the performance in other study areas such as supervised machine learning algorithms. However, for the best of our knowledge, we are the first to utilise these metrics to evaluate the completion and the quality of the estimated map of SLAM.

6 Experiment Results and Discussion

We first used ADNN to assess the quality of maps constructed by the three algorithms as depicted in Table 1. As expected, the deviations between estimated maps and ground truth represented by ADNN tend to increase as the map's size increases for all three SLAM algorithms. As shown in Table 1, 5/7 of Cartographer's mean errors are the lowest among the three algorithms. Especially when applied to a large map, Cartographer provides fairly low errors of 0.57 and 1.02 in comparison to the errors of using the Hector and Gmapping algorithms. Hector can produce competitive results in two medium scenarios. It also has low errors which are close to Cartographer's result in small scenarios. However, the performance of the Hector algorithm deteriorates significantly when it is applied to large area scenarios. This is because Hector does not utilise the odometer which can provide precise linear dynamics. The error would accumulate inevitably over time. Unexpectedly, Gmapping always has larger mean errors then other two algorithms. It only achieves a better result than the Hector algorithm when applied to large scenarios of Rectangle_L, Swirl_L and Corridor.

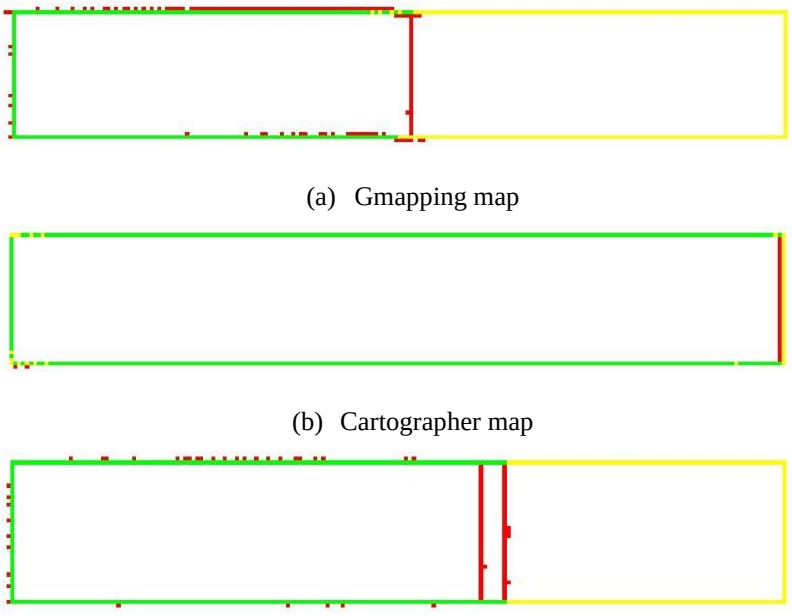
The similar observation of "Hector's performance declines in large scenario" can be found when comparing the results obtained with those presented in [14], where Hector and Gmapping's error were 0.46 and 0.42 (0.01meter/pixel) respectively, in a

Table 1. The ADNN mean and standard deviation. the unit is pixel/grid

		Rectangle_S	Rectangle_M	Rectagnle_L	Swirl_S	Swirl_M	Swirl_L	Corridor
Gmapping	mean	0.44	0.77	3.82	1.28	1.83	2.81	1.57
	std	0.11	0.25	0.94	0.57	0.45	0.54	0.17
Cartographer	mean	0.07	0.28	0.57	0.11	0.50	1.02	0.83
	std	0.03	0.20	0.15	0.01	0.07	0.05	0.12
Hector	mean	0.14	0.16	8.91	0.13	0.16	5.16	2.09
	std	0.00	0.00	0.00	0.00	0.00	0.00	0.00

about 4m*4m map. But Hector's error increased to 7.46 in a 12m*11m map while Gmapping's error only increased to 5.37. In the study [11], Cartographer is also reported having the lowest error. The error rate of the Gmapping algorithm reached the 2nd place. And Hector was the worst one. Their experiment scenario size was unclear. They tested four motion patterns (fast/slow move, sharp/smooth rotation). Hector's errors go from 20 cm to 50 cm, while all Cartographer's errors and 3/4 Gmapping's errors were less than 10 cm. But it should be noted that in [13], a totally different result was obtained. In that study, Hector produced the trajectory that closest to the ground truth. Then, they used Hector trajectory as reference but did not provide the error of it. Their Cartographer result has an error of 2.4 cm compared with Hector trajectory. Besides, they did not acquire a valid Gmapping map.

In terms of the standard deviation, Hector achieves the lowest standard deviation. When applying hector multiple times, the same map was obtained. This is due to the nature of the Hector algorithm, which relies on scan matching. It stitches together the aligned frames between consecutive scans. It uses Gaussian-Newtown optimization[17] which will always give the constant result as long as the input is the same. Gmapping, on the other hand, produces the largest variance in most of the cases, which may be attributed to the randomness feature of the particle filter employed by the algorithm.



(a) Gmapping map

(b) Cartographer map

(c) Hector map

Fig. 2. Visualisation of SLAM results on the corridor scenario based on the proposed metric. Green grids represent TP estimation, red grids denote FP estimation and yellow grids represent FN estimation.

Evaluation based on ADNN does not well describe the structure completion of a map. For example, when applied to the corridor scenario. The ADNN errors of Gmapping, Cartographer and Hector are 1.57, 0.83 and 2.09 respectively. These numbers only tell that the Gmapping and the Hector have less accuracy than Cartographer but it does not reflect the fact that the estimated map's size is only half of the ground truth's size as illustrated in Fig. 2, which was based on the visualisation of the proposed grid portion metric. Together with precision and sensitivity estimation shown in Table 2, the assessment based on grid portion has a great potential to provide a complete picture of the quality of a SLAM map. On the corridor scenario, the mean precision, mean sensitivity of the Gmapping are 0.61, 0.50. The 0.5 sensitivity tells that the Gmapping map only represents half of the ground truth map. And the 0.61 precision tells that about 61% of the estimated grids are correct. These data on the Hector map are 0.72, 0.62 respectively, as the Hector map is slightly longer than the Gmapping map and it has less FP grids. Cartographer reaches 0.92 and 0.90, when using this metric highlighted the Cartographer map is the most complete one among those three maps (Fig. 2). It is clear that the precision value intuitively reflects the correction of the map and the sensitivity value reflects the completion of the structure. Therefore, the grid portion metric along with the visualisation as illustrated in Fig. 2 is more effective in representing the quality of a map. In this view, the Hector provides second high precision in the small scenarios i.e Rectangle_S, Swirl_S and highest precision in Rectangle_M and Swirl_M. But its precision and sensitivity dramatically decrease in the large scenarios. Similar to the conclusion derived from ADNN, the Cartographer can always have relatively high precision and sensitivity in all scenarios though it has a larger variance when applied to the medium and large scenarios, which deserves further investigation.

Table 2. The mean and standard deviation of grid portion metric

			Rectangle_S	Rectangle_M	Rectagnle_L	Swirl_S	Swirl_M	Swirl_L	Corridor
Gmapping	Precision	Mean	0.67	0.44	0.09	0.39	0.20	0.13	0.61
		std	0.09	0.07	0.08	0.15	0.07	0.04	0.04
	Sensitivity	Mean	0.89	0.61	0.12	0.46	0.22	0.12	0.50
		std	0.09	0.07	0.08	0.15	0.07	0.04	0.04
Cartographer	Precision	Mean	0.96	0.78	0.52	0.92	0.63	0.44	0.92
		std	0.02	0.17	0.15	0.01	0.06	0.04	0.00
	Sensitivity	Mean	0.99	0.79	0.61	0.98	0.69	0.46	0.90
		std	0.02	0.17	0.15	0.01	0.06	0.04	0.00
Hector	Precision	Mean	0.88	0.86	0.23	0.89	0.86	0.17	0.72
		std	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Sensitivity	Mean	1.00	1.00	0.34	1.00	1.00	0.19	0.62
		std	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The last column of Table 2. shows that Cartographer (with highest precision and sensitivity) is robust to the geometry of corridor environment. Both Gmapping and Hector do not perform well in the corridor environment which has fewer features. For Gmapping, the proposed particles are distributed along the direction of the corridor. However, the observations obtained in the direction of the corridor are still highly similar, which makes it difficult to estimate the length of a trajectory. Hector uses the Gauss-Newton method to solve scan matching between two frames. However, similar structural information obtained in a featureless environment clearly has a big impact on its performance.

Overall, Hector SLAM algorithm is the 2nd best on small scenarios and the best in medium scenarios with both ADNN metric and gird portion metric. However, its performance will significantly drop in larger scenarios. While Cartographer reached the 1st place in small scenarios and the 2ed place in medium scenarios with both ADNN metric and gird portion metric. The important thing is, it has the lowest ADNN error and highest precision and sensitivity in every large scenario. Finally, Gmapping is worse than the other two algorithms in small and medium scenes and reaches second place in large scenarios also it has the greatest instability in most cases.

7 Conclusion and future work

In this work, a new grid portion metric was proposed and introduced for the assessment of the performance of SLAM algorithms. Three representative 2D LiDAR SLAMs were studied. Results show that, in comparison to traditional metrics such as ADNN, the proposed grid portion-based assessment has great potential to provide a more complete picture of the quality of SLAM maps. By visualising the SLAM results on a map, a better understanding of the map can be achieved. The numerical value of sensitivity and precision representing the proportion of FP and FN results respectively can be used to indicate the correctness and completeness of a SLAM map. In order to deal with the potential instability of the SLAM results, the mean and standard deviation of each metric were calculated which provides additional insight to the nature of each SLAM algorithm.

It is worth noting that the evaluation was based on simulation environment provided by ROS. Assessment to be carried out in a real physical world would be part of our future work. In addition, the research would be expanded to cover 3D LiDAR SLAM and vision SLAM. Besides, because each SLAM algorithm, under different scenarios, can be optimised by adjusting the certain parameters. Dynamic determination of optimal parameters represents another direction of our research.

References

1. Yuan, W., Li, Z., Su, C.-Y.: RGB-D sensor-based visual SLAM for localization and navigation of indoor mobile robot. 2016 International Conference on Advanced Robotics and

Mechatronics (ICARM). pp. 82–87 (2016).

2. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., Leonard, J.J.: Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics. 32, 1309–1332 (2016).

3. Zhang, F., Guan, C., Fang, J., Bai, S., Yang, R., Torr, P., Prisacariu, V.: Instance segmentation of lidar point clouds. ICRA, Cited by. 4, (2020).

4. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 1887–1893 (2018).

5. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: Real-time single camera SLAM. IEEE Transactions on Pattern Analysis \& Machine Intelligence. 1052–1067 (2007).

6. Grisettiyz, G., Stachniss, C., Burgard, W.: Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. Proceedings of the 2005 IEEE international conference on robotics and automation. pp. 2432–2437 (2005).

7. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE transactions on robotics. 31, 1147–1163 (2015).

8. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics. 33, 1255–1262 (2017).

9. Konolige, K., Grisetti, G., Kümmerle, R., Burgard, W., Limketkai, B., Vincent, R.: Efficient sparse pose adjustment for 2D mapping. 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 22–29 (2010).

10. Hess, W., Kohler, D., Rapp, H., Andor, D.: Real-time loop closure in 2D LIDAR SLAM. 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 1271–1278 (2016).

11. Yagfarov, R., Ivanou, M., Afanasyev, I.: Map comparison of lidar-based 2d slam algorithms using precise ground truth. 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV). pp. 1979–1983 (2018).

12. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580 (2012).

13. Filipenko, M., Afanasyev, I.: Comparison of various slam systems for mobile robot in an indoor environment. 2018 International Conference on Intelligent Systems (IS). pp. 400–407 (2018).

14. Santos, J.M., Portugal, D., Rocha, R.P.: An evaluation of 2D SLAM techniques available in robot operating system. 2013 IEEE International Symposium on Safety, Security, and

Rescue Robotics (SSRR). pp. 1–6 (2013).

15. Zhang, Z., Scaramuzza, D.: Rethinking Trajectory Evaluation for SLAM: a Probabilistic, Continuous-Time Approach. arXiv preprint arXiv:1906.03996. (2019).

16. Filatov, A., Filatov, A., Krinkin, K., Chen, B., Molodan, D.: 2d slam quality evaluation methods. 2017 21st Conference of Open Innovations Association (FRUCT). pp. 120–126 (2017).

17. Kohlbrecher, S., Von Stryk, O., Meyer, J., Klingauf, U.: A flexible and scalable slam system with full 3d motion estimation. 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics. pp. 155–160 (2011).

18. About ROS, https://www.ros.org/about-ros/.

19. Buyval, A., Afanasyev, I., Magid, E.: Comparative analysis of ROS-based monocular SLAM methods for indoor navigation. Ninth International Conference on Machine Vision (ICMV 2016). p. 103411K (2017).

20. Xu, L., Feng, C., Kamat, V.R., Menassa, C.C.: An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments. Automation in Construction. 104, 230–245 (2019).

21. Bayer, J., Cizek, P., Faigl, J.: On construction of a reliable ground truth for evaluation of visual slam algorithms. Acta Polytechnica CTU Proceedings. 6, 1–5 (2016).

22. Le, X.S., Fabresse, L., Bouraqadi, N., Lozenguez, G.: Evaluation of out-of-the-box ros 2d slams for autonomous exploration of unknown indoor environments. International Conference on Intelligent Robotics and Applications. pp. 283–296 (2018).

23. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 13, 600–612 (2004).

24. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., others: FastSLAM: A factored solution to the simultaneous localization and mapping problem. Proceedings of the National conference on Artificial Intelligence. pp. 593–598 (2002).

25. Black, P.E.: Bresenham’s algorithm, https://www.nist.gov/dads/HTML/bresenham.html.

26. stage_ros summary, http://wiki.ros.org/stage_ros.

27. TurtleBot3 - Specifications, http://emanual.robotis.com/docs/en/platform/turtlebot3/specifications/.