Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth

Conference Paper · November 2018

DOI: 10.1109/ICARCV.2018.8581131

3 authors:


Rauf Yagfarov
Innopolis University


Mikhail Ivanou
Innopolis University


Ilya Afanasyev
Innopolis University

Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth

Rauf Yagfarov, Mikhail Ivanou and Ilya Afanasyev, *Member, IEEE*

***Abstract*— This paper presents a comparative analysis of three most common ROS-based 2D Simultaneous Localization and Mapping (SLAM) libraries: Google Cartographer, Gmapping and Hector SLAM, using a metrics of average distance to the nearest neighbor (ADNN). Each library was applied to construct a map using data from 2D lidar that was placed on an autonomous mobile robot. All the approaches have been evaluated and compared in terms of inaccuracy constructed maps against the precise ground truth presented by FARO laser tracker in static indoor environment.**

I. INTRODUCTION

As is known, the process when a robot acquires an environment map while simultaneously localizing itself relatively to this map is called SLAM. At present, SLAM is a well investigated field in robotics, and there are many algorithms for localization and mapping, which are actively used both in research and in applied problems [1]–[3]. However, among the many algorithms it is difficult to choose the best and most suitable for solving certain problems. The complexity to choose the algorithm lies in the fact that it is necessary to compare the methods, after that you can choose the best one. The main problem in SLAM algorithms running is that each of them can require own environment that often is not easy to configure. To address this issue, the Robot Operating System (ROS) was presented as a flexible framework for running robot software [4]. The ROS philosophy was to build a framework that could be integrated in different robotic platforms by making little changes in a code. In addition, ROS enables researchers an option of to quick and easy simulations and real world experiments. Most of SLAM libraries support ROS api. For our investigation we choose the most widely used LiDAR-based SLAM libraries that have ROS wrappers - Gmapping [5], Google Cartographer [6], and Hector SLAM [7].

After a map construction the next important question is "How to evaluate results"? SLAM comparison is not a new topic. For example, the papers [8], [9] provide a comparison of trajectories computed by data from different visual (monocular and stereo cameras) and laser (depth sensor) sensors, using lidar-based SLAM trajectories as the ground truth. The research [10] studies onboard sensor-based odometry for a quadrotor UAV "Bebop" in outdoor environment, collating UAV trajectories for LSD-SLAM and ORB-SLAM with the built-in Bebop's visual-inertial odometry and an external observer data as the ground truth. The paper [11] presents the comparative analysis of monocular SLAM methods (ORB-SLAM, REMODE, LSD-SLAM, and DPPTAM) for indoor navigation of an autonomously guided robot in an office-style environment. The investigation [12] evaluates visual SLAM methods (ORB-SLAM, LSD-SLAM, L-SLAM, and OpenRatSLAM) using TUM RGB-D benchmark. The paper [13] proposes the original loop-closure detection algorithm for monocular vSLAM and collates it with loop-closure precision for LSD-SLAM and ORB-SLAM methods. The article [14] compares five ROS-compatible stereo visual SLAM methods (ORB-SLAM2, LIBVISO2, SPTAM, RTAB-MAP, ZED-VO), using lidar-based Hector SLAM scans as the ground truth. In the investigation [15] a framework for analyzing the results of a SLAM approach based on a metric for measuring the error of the corrected trajectory was proposed. Since this metrics used only relative displacements between robot poses and did not rely on a global reference frame, it allows to overcome serious shortcomings of approaches with error computation from a global reference frame. The research [16] compares maps generated by three ROS-based SLAM algorithms (GMapping, KartoSLAM and Hector SLAM), which processed ToF camera data, using the metrics developed by [17]. In the paper [18] a framework for quantitative evaluation of 2D laser SLAM algorithm quality is presented, which uses various metrics based on capture characteristics. This framework was tested with ROS-based SLAM methods: GMapping, tinySLAM, vinySLAM, Google Cartographer and Hector SLAM.

The work [17] conducts a study of five laser-based 2D SLAM techniques available in ROS: Hector SLAM, Gmapping, KartoSLAM, CoreSLAM and LagoSLAM, for data acquired in real world experiments with the following comparison constructed maps with the metrics based on the k-nearest neighbors. Another research with comparison of three 2D SLAM techniques available in ROS: Gmapping, Hector SLAM and CRSM SLAM, using RGB-D Kinect sensor data and CMSE (Corner Mean Square Error) and the Summation of the Differences for the map sides' length (SDS) metrices was proposed in [19]. These investigations inspired the authors of this paper to contribute and update these methodologies for Lidar-based 2D SLAM comparative analysis of Gmapping, Hector SLAM and Google Cartographer with a similar to the [17] metrics of average distance to the nearest neighbor (ADNN). In this paper, we present a method that allows to evaluate the quality of the SLAM

This research has been supported by the Russian Ministry of Education and Science within the Federal Target Program grant (research grant ID RFMEFI60617X0007.

R. Yagfarov and M. Ivanou are with the Center for Technologies in Robotics and Mechatronics Components, Innopolis University, Innopolis, Russia, {r.yagfarov, m.ivanov}@innopolis.ru

I. Afanasyev is with the Institute of Robotics, Innopolis University, Innopolis, Russia, i.afanasyev@innopolis.ru

maps compared with the ground truth constructed by the high-precision laser tracker FARO, which measures point coordinates in a map space with micrometer accuracy.

The paper is structured as follows. Section II introduces software and hardware configurations. Section III describes our robot system configuration and ground truth equipment. IV presents the methodology of experiments and map reconstruction from experimental data. Section V provides comparative analysis of reconstructed maps built by different SLAM algorithms. Finally, conclusions are drawn and future work is briefly outlined in Section VI.

II. SLAM Algorithms

A. Gmapping

Gmapping [1] is a laser-based SLAM algorithm, which uses a Rao-Blackwellized Particle Filter SLAM approach [5]. This approach uses adaptive resampling technique and overcomes the existing serious problems for usual particle filter algorithms, which have a big computational complexity and the depletion problem (when the elimination of a large number of particles from the sample set during the resampling stage can lead to correct hypothesis rejection and decrease in accuracy). The adaptive resampling technique solves the particle depletion problem, since the resampling process is limited and performed only when it is needed. Moreover, this approach computes more accurate robot localization by integrating the most recent sensor observations with odometry robot motion model, decreasing the uncertainty about the robots position in the prediction step of the particle filter. Therefore, the number of particles required has been dramatically reduced because of the quality of laser scan matching process. E.g., for our experiment, we set the parameter with the amount of particles for Gmapping in 30.

B. Google Cartographer

Cartographer [2] is positioned as a system that provides real-time SLAM in 2D and 3D across multiple platforms and sensor configurations [6]. It is an open source library, which has ROS wrapper. As far as grid mapping requires much resources, Google Cartographer does not use particle filter algorithm. Cartographer solves the problem of error accumulation during long iterations by pose estimation. Laser scans are iteratively matched with a submap (at the best estimated position) referred to as frames. Scan matching occurs at a recent submap, therefore it only depends on recent scans. After each submap is finished, all submaps and scans are automatically checked for loop closure. If they are close enough based on current pose estimates, a scan matcher tries to find the scan in the submap. The paper [6] describes the conversion process for scan points from a scan frame into a submap frame. Submaps are represented in form of probability grid points. Each grid point is a pixel that contains all the points near that grid. Hits and misses are computed during inserting a new scan. All nearest grids points are inserted into the hit set, but grid points between a scan origin and a scan point, which intersect the rays, are inserted in the miss set. Then all grid points are updated with the appropriate probabilities. Scan matching in Cartographer is based on the Ceres scan matching [20]. It maximizes the probabilities to find more accurate scan pose in the submap.

C. Hector SLAM

Hector SLAM [3] is 2D SLAM system [7], which combines robust lidar scan matching and 3D navigation approach with inertial sensing system by Extended Kalman Filter (EKF). Hector SLAM is oriented to onboard computations of actual 6DOF robot pose during motion with simultaneous high update rate lidar-based 2D mapping in real-time mode. The alignment of laser beam endpoints with a map obtained is provided by Gaussian-Newton optimization approach, where the scan matching is implicitly performed with all preceding scans [7].

III. System Setup

A. Robot system configuration

All our experiments were performed with the differential drive mobile robot platform called "Plato" shown in Fig. 1. The mobile platform is equipped with the onboard computer Jetson TX1, bumpers, sonars, Troyka IMU module, Hokuyo urg-04lx-ug01 2D lidar, Velodyne VLP-16 3D Lidar and wheel encoders that allow to calculate the wheel odometry. The robot's onboard computer uses Ubuntu 16.04 installed with ROS Kinetic.

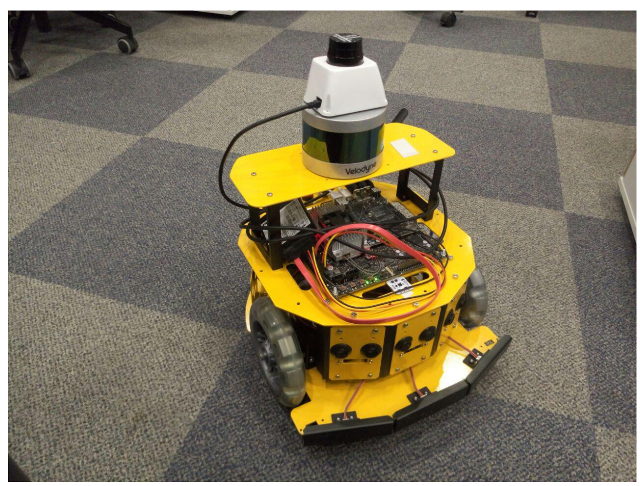


Fig. 1: Plato robot

B. FARO system

In our work we used a FARO Laser Tracker [4] to build the ground truth map. The Vantage System is a portable device

[1] Gmapping: ROS package for 2D lidar SLAM, `http://wiki.ros.org/gmapping`

[2] Cartographer: real-time 2D/3D SLAM, `http://wiki.ros.org/cartographer`

[3] hector_slam: ROS-based SLAM package, `http://wiki.ros.org/hector_slam`

[4] FARO Vantage Laser Tracker, `https://www.faro.com/products/factory-metrology/faro-laser-tracker/`

for a 3D high accuracy measurements with a range of up to 80m, which is used as the standard in workflow productivity management for metrology applications with the accuracy up to 0.015mm.

Fig. 2: FARO Laser Tracker Vantage

IV. Experiments

A. Indoor environment

For our indoor experiments we chose a classroom shown in Fig. 3, in which the tables were moved to the center and the environment was organized such way to create a loop for the mobile robot navigation.

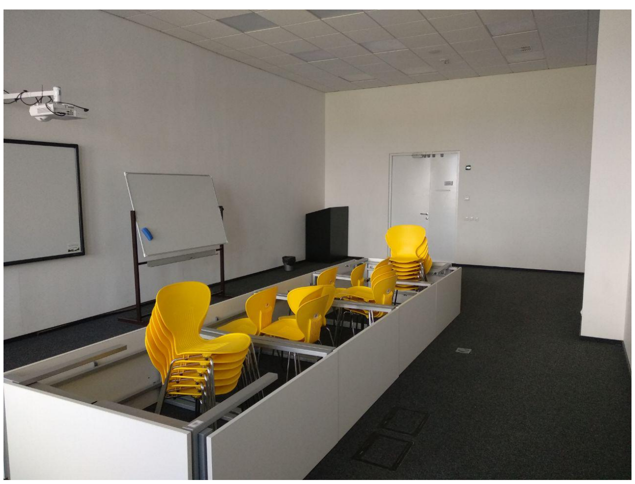


Fig. 3: Study room in Innopolis University

B. Ground truth-based map construction

The ground truth map was created by constructing the planes of walls and the floor from points obtained using FARO Laser tracker in Metrolog X4 software. After that a horizontal slice was made at the height of the lidar mounted on the mobile platform. Thus, at the beginning 3D points of the plane for walls and the floor were measured using a FARO Laser tracker. Then, these points were used to construct planes in Metrolog X4 software (see Fig. 4). After that, corners were found as the intersection of walls' planes. Finally, 2D map points were obtained as the intersection of the plane parallel to the floor at the height of the lidar mounted on mobile robot platform. The resulting points were connected by using lines and OpenCV, and then were saved in the pgm format.

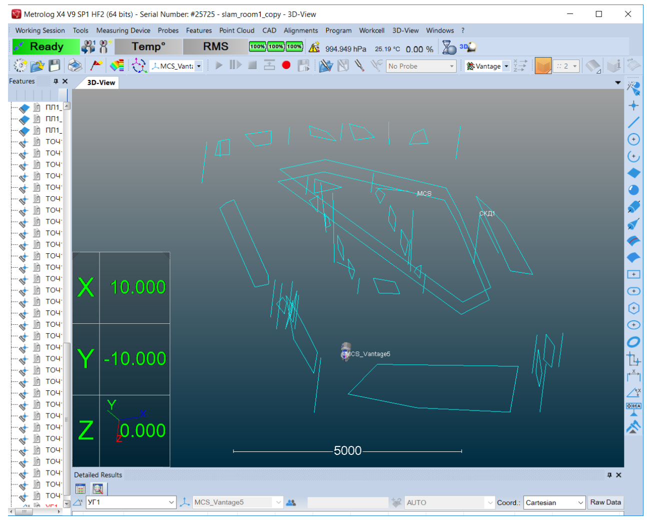


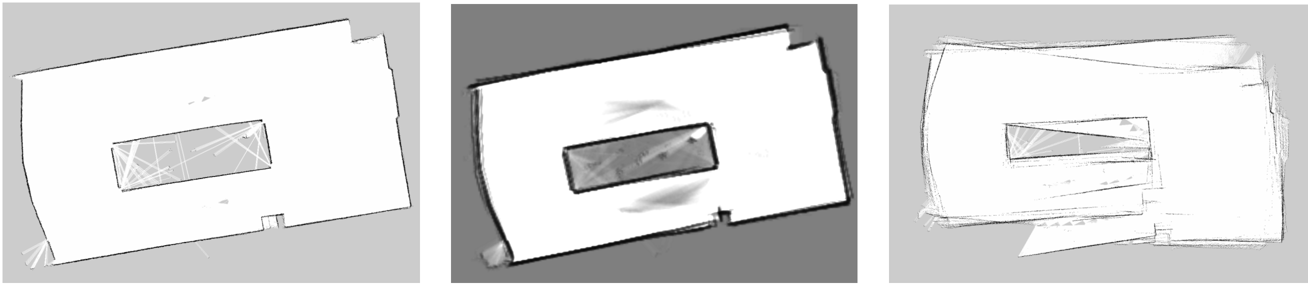
Fig. 4: Ground truth construction in Metrolog X4

C. Map construction from ROS-based SLAM algorithms

In order to make the comparison as fair as possible and in order to put SLAM algorithms on an equal conditions ROS bag files were recorded. Ros bag files are files in which data from ROS topics recorded with appropriate time sampling. For the algorithms in ROS there are no differences between data obtained from recorded bag file and real time robot data. ROS bag files recorded all ROS topics from mobile robot platform which was remotely teleoperated using joystick. Next these bag files were used offline in SLAM algorithms. Finally the maps were obtained in the Occupancy Grid form and saved in pgm format using *map_saver* node from *map_server* package. Examples of the constructed maps using different SLAM algorithms could be seen in Fig. 5. Occupancy grid is a 2D representation of a map in space, which is a grid, each cell is a state in a given space location. A cell can take one of three states: occupied, free and unknown.

V. Results and Discussion

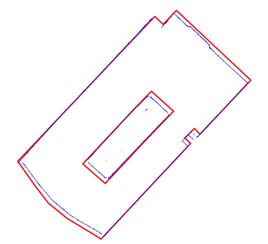
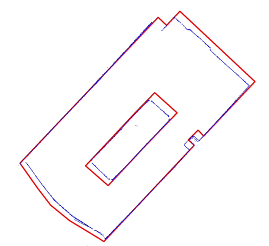
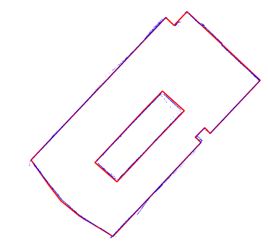
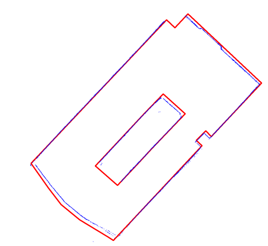
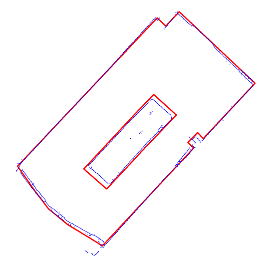
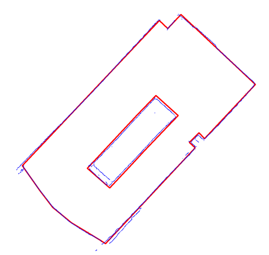
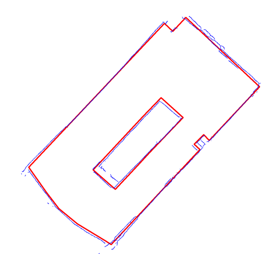
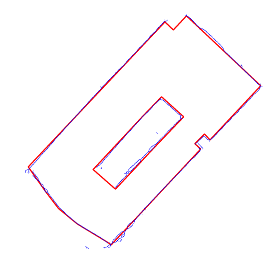
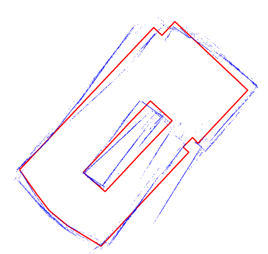
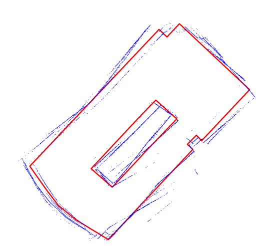
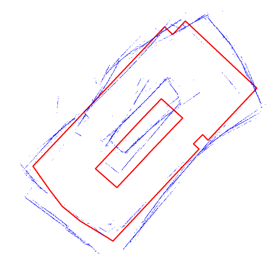
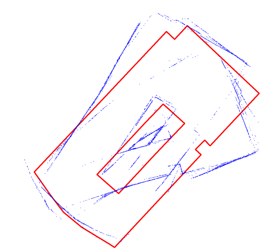
Occupancy grids of saved maps was changed so that the cells had only two meanings: occupied and free. This transformation is necessary to simplify the process of comparing maps. The ground truth lines of the map have a width of one Occupancy grid cell, so for correct comparison with maps obtained using SLAM algorithms, a thining operation using OpenCV was applied on SLAM maps, which narrows the wall lines. Next necessary steps for comparison are aligning



(a) Gmapping (b) Cartographer (c) Hector SLAM

Fig. 5: Occupancy grid maps constructed by SLAM algorithms

TABLE I: Map comparison constructed from SLAM algorithms (in blue colors) and ground truth (in red color)

SLAM method	Slow	Fast/Smooth	Fast/Sharp	No loop closure
Gmapping				
Cartographer				
Hector SLAM				

maps and calculating difference metrics. Alignment was performed using the ICP (Iterative closest point) algorithm [1], which allows to find such transformation, that gives a minimum error of the sum of distances from each point of the SLAM map to the nearest neighbor point on the ground truth map. In our work, we used the similar metric as described in the paper [17], calculating average distance to the nearest neighbor (ADNN) as a sum of all distances divided by number of occupied cells (see, equation 1).

$$\textbf{ADNN} = \frac{\sum_{i=1}^{N} \mathit{Nearest\_Neighbour}(\mathit{occupied\_grid\_cell}(i))}{N}, \quad (1)$$

where N is a number of occupied cells.

The experiments were conducted with different scenarios for the mobile robot motion, where the robot traveled: (1) slowly with smooth turns, (2) quickly with smooth turns, (3) quickly with sharp turns, and (4) slowly with smooth turns, but without loop closure. Results of alignment could be observed in Table I. The red dots show the ground truth, and the blue dots show the map obtained using SLAM algorithms.

The comparison of ADNN-based errors in map construction provided by Gmapping, Cartograoher and Hector SLAM methods relative to the ground truth is presented in Table II.

TABLE II: Map comparison: error calculated with ADNN metrics for SLAM methods relative to the ground truth, in cm

Condition	SLAM algorithm		
	Gmapping	Cartographer	Hector SLAM
Slow ride, smooth rotations, loop closure	8.05	7.41	27.95
Fast ride with smooth rotations, loop closure	11.92	5.35	19.36
Fast ride with sharp rotations, loop closure	3.21	7.37	44.03
Without loop closure	6.11	4.97	51.67

VI. Conclusions and Future work

The tests that were presented show that almost in all scenarios Google Cartographer constructs maps with the smallest error relative to the precise ground truth presented by FARO laser tracker. This algorithm is robust enough to the different type of mobile robot movements. Gmapping maps are not very far from Cartographer maps quality. It demonstrates reasonably good outcomes in 2D map construction even without loop closure. The reason is that Gmapping and Cartographer apply odometry for localization rectification and map correction. Hector SLAM uses only LIDAR data and does not provide an explicit option for loop closure. That is why Hector SLAM results are less accurate. Thus, our research shows that one of the best algorithm to generate 2D maps with LIDAR placed on mobile robot is Google Cartographer.

The comparison algorithm shown in the paper is suitable for comparing not only two-dimensional maps, but also three-dimensional. A three-dimensional map can be represented using a set of voxels as it is presented in Octomap [21]. Our next step is to extend the current algorithm to compare three-dimensional SLAM maps. Also, we plan to add more new metrics for more in-depth analysis of algorithms.

VII. ACKNOWLEDGMENTS

This research has been supported by the Russian Ministry of Education and Science within the Federal Target Program grant (research grant ID RFMEFI60617X0007).

References

[1] F. Pomerleau, F. Colas, R. Siegwart, *et al.*, "A review of point cloud registration algorithms for mobile robotics," *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[3] C. Stachniss, J. J. Leonard, and S. Thrun, "Simultaneous localization and mapping," in *Springer Handbook of Robotics*, pp. 1153–1176, Springer, 2016.

[4] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[6] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 1271–1278, IEEE, 2016.

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

[8] I. Z. Ibragimov and I. M. Afanasyev, "Comparison of ros-based visual slam methods in homogeneous indoor environment," in *Positioning, Navigation and Communications (WPNC), 2017 14th Workshop on*, pp. 1–6, IEEE, 2017.

[9] M. Filipenko and I. Afanasyev, "Comparison of various slam systems for mobile robot in an indoor environment," in *International Conference on Intelligent Systems*, IEEE, Sep. 2018.

[10] A. Gabdullin, G. Shvedov, M. Ivanou, and I. Afanasyev, "Analysis of onboard sensor-based odometry for a quadrotor uav in outdoor environment," in *International Conference on Artificial Life and Robotics (ICAROB)*, 2018.

[11] A. Buyval, I. Afanasyev, and E. Magid, "Comparative analysis of ros-based monocular slam methods for indoor navigation," in *Ninth International Conference on Machine Vision (ICMV 2016)*, vol. 10341, p. 103411K, International Society for Optics and Photonics, 2017.

[12] A. Huletski, D. Kartashov, and K. Krinkin, "Evaluation of the modern visual slam methods," in *Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (FRUCT), 2015*, pp. 19–25, IEEE, 2015.

[13] A. Bokovoy and K. Yakovlev, "Original loop-closure detection algorithm for monocular vslam," in *International Conference on Analysis of Images, Social Networks and Texts*, pp. 210–220, Springer, 2017.

[14] R. Giubilato, S. Chiodini, M. Pertile, and S. Debei, "An experimental comparison of ros-compatible stereo visual slam methods for planetary rovers," in *IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*, pp. 386–391, IEEE, 2018.

[15] R. Kümmerle, B. Steder, C. Dornhege, M. Ruhnke, G. Grisetti, C. Stachniss, and A. Kleiner, "On measuring the accuracy of slam algorithms," *Autonomous Robots*, vol. 27, no. 4, p. 387, 2009.

[16] L. Fang, A. Fisher, S. Kiss, J. Kennedy, C. Nagahawatte, R. Clothier, and J. Palmer, "Comparative evaluation of time-of-flight depth-imaging sensors for mapping and slam applications," in *Australasian Conference on Robotics and Automation 2016*, pp. 1–7, Australian Robotics and Automation Association, 2016.

[17] J. M. Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2d slam techniques available in robot operating system," in *Safety, Security, and Rescue Robotics (SSRR), 2013 IEEE International Symposium on*, pp. 1–6, IEEE, 2013.

[18] A. Filatov, A. Filatov, K. Krinkin, B. Chen, and D. Molodan, "2d slam quality evaluation methods," in *Open Innovations Association (FRUCT), 2017 21st Conference of*, pp. 120–126, IEEE, 2017.

[19] M. Rojas-Fernández, D. Mújica-Vargas, M. Matuz-Cruz, and D. López-Borreguero, "Performance comparison of 2d slam techniques available in ros using a differential drive robot," in *Electronics, Communications and Computers (CONIELECOMP), 2018 International Conference on*, pp. 50–58, IEEE, 2018.

[20] S. Agarwal, K. Mierle, *et al.*, "Ceres solver," 2012.

[21] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013. Software available at `http://octomap.github.com`.