2018 International Conference on Intelligent Systems (IS)

Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment

Maksim Filipenko
Institute of Robotics
Innopolis University
Innopolis, Russia
m.filipenko@innopolis.ru

Ilya Afanasyev
Institute of Robotics
Innopolis University
Innopolis, Russia
i.afanasyev@innopolis.ru

***Abstract*—This article presents a comparative analysis of a mobile robot trajectories computed by various ROS-based SLAM systems. For this reason we developed a prototype of a mobile robot with common sensors: 2D lidar, a monocular and ZED stereo cameras. Then we conducted experiments in a typical office environment and collected data from all sensors, running all tested SLAM systems based on the acquired dataset. We studied the following SLAM systems: (a) 2D lidar-based: GMapping, Hector SLAM, Cartographer; (b) monocular camera-based: Large Scale Direct monocular SLAM (LSD SLAM), ORB SLAM, Direct Sparse Odometry (DSO); and (c) stereo camera-based: ZEDfu, Real-Time Appearance-Based Mapping (RTAB map), ORB SLAM, Stereo Parallel Tracking and Mapping (S-PTAM). Since all SLAM methods were tested on the same dataset we compared results for different SLAM systems with appropriate metrics, demonstrating encouraging results for lidar-based Cartographer SLAM, Monocular ORB SLAM and Stereo RTAB Map methods.**

***Keywords*—simultaneous localization and mapping (SLAM), visual odometry, indoor navigation, ROS**

I. Introduction

The map building problem for an unknown environment with use of onboard sensors while solving the localization problem at the same time is known as Simultaneous Localization and Mapping (SLAM) [1], [2]. One of the classical approaches to solve this problem is filter-based SLAM. This method is based on the Bayesian filtering theory. It has two main steps: (1) the prediction step, when robot localization and map state are updated using the previous information about the system state and input control commands; (2) the measurement update, where current sensor data are matched to predicted system state in order to make new system state prediction. This approach has various implementations. The earlier SLAM system were based on Extended Kalman filter (EKF-based SLAM, [3]), or on Particle Filter (e.g., FastSLAM, [4]). For introduction to the basics of SLAM system we direct to surveys [5], for more detailed information about filter-based systems recommend [6].

The SLAM problem can be solved with the use of different sensors and the choice of suitable sensors plays a special role in the effective operation of Unmanned Ground Vehicle (UGV) due to the limited autonomous resource. Most mobile robots have Inertial Measurement Unit (IMU), which includes accelerometers, gyroscope, and magnetometer, which measure orientation, angular velocity, and acceleration correspondingly. This approach to getting mobile robot localization at using only Inertial Navigation System (INS) can give significant navigation errors [7]. Therefore the main sensor for indoor robot navigation and SLAM is usually lidar. 2D lidar SLAM systems are currently presented in different packages like GMapping [8] (which uses Rao-Blackwellized particle filer [9] to learn grid maps from 2D lidar data), Hector SLAM [10] (that is another very popular ROS-based SLAM), and Cartographer [11] (which is one of the most recent systems).

On the other hand, monocular and stereo cameras are good low-cost passive sensors, which can effectively solve SLAM problem working as a single source of information about an environment that referring to Visual SLAM (V-SLAM [12], [13]). All methods of V-SLAM can be divided into two groups: (1) feature-based, which use selected features for map building, and (2) direct that work with entire images as a whole. The earlier investigations on visual navigation were carried out with a binocular stereo camera [14], [15] and a monocular camera (that was called MonoSLAM [16]). Over the past decade, numerous methods have been proposed, including Parallel Tracking and Mapping (PTAM, [17], [18]), REgularized MOnocular Depth Estimation (REMODE, [19]), Oriented FAST [20] and Rotated BRIEF [21] (ORB-SLAM, [22], [23]), Dense Tracking and Mapping in Real-Time (DTAM, [24]), Large-Scale Direct monocular SLAM (LSD-SLAM, [25]), Large-Scale Direct SLAM with Stereo Cameras (Stereo LSD-SLAM, [26]), Fast semi-direct monocular visual odometry (SVO,[27]), Real-Time Appearance-Based Mapping (RTAB map, [28]), Dense Piecewise Parallel Tracking and Mapping (DPPTAM, [29]), Direct Sparse Odometry (DSO, [30]), ElasticFusion (Dense SLAM without a Pose Graph [31]), Convolutional Neural Networks SLAM (CNN-SLAM, [32]), Stereo Parallel Tracking and Mapping (S-PTAM, [33]). Each of these methods has its advantages and disadvantages. Common features include weak resistance to unfavorable conditions, experimentation, poor performance with a weak geometric diversity of the environment, sensitivity to pure rotations [34], [35]. These methods do not provide metric information that is required in some applications (this problem is also known as Scale ambiguity). The solution of this problem is discussed in computer vision and robotics society. Thus, e.g., the paper [36] suggested to use additional sensors like IMU, whereas the authors [37] proposed the approach for restoring metric information using information about the geometric sensor arrangement. The other important things, which can contribute to SLAM quality, are map initialization [38],[39], a loop closure [40] (that compensates errors accumulated during laps running by UGV) and a type of camera shutter, which is used [41].

978-1-5386-7097-2/18/$31.00

More detailed description of features for Visual SLAM system [42], which we investigated in this research, is presented in the Table I.

TABLE I: DESCRIPTION OF FEATURES FOR VISUAL SLAM SYSTEMS INVESTIGATED IN THIS RESEARCH

System	Method	Feature type	Feature descriptor	Map density	Loop Closure
PTAM	feature	FAST	local patch of pixels	sparse	none
SVO	semi-direct	FAST	local patch of pixels	sparse	none
DPPTAM	direct	intensity gradient	local patch of pixels	dense	none
LSD SLAM	direct	intensity gradient	local patch of pixels	dense	Bag of Words place recognition
ORB SLAM	feature	FAST	ORB	semi-dense	FabMap
DSO	direct	intensity gradient	local patch of pixels	dense	none
RTAB map	feature	GFTT, FAST, ...	SIFT, ORB, ...	sparse	Bag of Words place recognition
S-PTAM	feature	GFTT, FAST, ...	SIFT, ORB, ...	sparse	Bag of Words place recognition

In this paper, we mainly focus on a real robotics application for existing SLAM systems, which are implemented in Robot Operating System (ROS, [43]). For this reason, we investigate the following SLAM systems: (a) 2D lidar-based: GMapping, Hector SLAM, Cartographer; (b) monocular camera-based: LSD SLAM, ORB SLAM, DSO; and (c) stereo camera-based: ZEDfu, RTAB map, ORB SLAM, S-PTAM. The brief information about ROS-based SLAM methods which were studied in this research are presented in the Table II.

TABLE II: ROS-BASED SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM) SYSTEMS STUDIED IN THIS RESEARCH

Year	System	Sensor	Ref.
2007	GMapping	2D lidar	[8]
2007	Parallel Tracking and Mapping (PTAM)	mono	[17]
2011	Hector SLAM	2D lidar	[10]
2014	Semi-direct Visual Odometry (SVO)	mono	[27]
2014	Large Scale Direct monocular SLAM (LSD SLAM)	mono	[25]
2014	Real-Time Appearance-Based Mapping (RTAB map)	stereo	[28]
2015	ORB SLAM	mono, stereo	[22]
2015	Dense Piecewise Parallel Tracking and Mapping (DPPTAM)	mono	[29]
2016	Direct Sparse Odometry (DSO)	mono	[30]
2016	Cartographer	2D lidar	[11]
2017	Stereo Parallel Tracking and Mapping (S-PTAM)	stereo	[33]

The paper is organized as follows: Section III describes the hardware and software used, Section IV presents the methodology of the experiment and metrics for evaluation, Section V considers visual SLAM algorithm implementation, Section VI demonstrates the achieved results. Finally, we conclude in Section VII.

II. Related work

In this section we review the research papers dealt with comparative analysis of various ROS-based SLAM methods. There are plenty of recent investigations, which compare SLAM methods for indoor navigation in mobile robotics applications [44], [45], [46], [47], [48]. Some studies compare only efficiency of lidar-based SLAM approaches [49], [50], [51], [52]. The paper [50] compares five ROS-based Laser-related SLAM methods: Gmapping, KartoSLAM, Hector SLAM, CoreSLAM and LagoSLAM, demonstrating more favourable outcomes for the last three.

The paper [45] describes the qualitative comparison of monocular SLAM methods (ORB-SLAM, REMODE, LSD-SLAM, and DPPTAM) performed for indoor navigation. The research [46] extends the comparative analysis, considering a mobile platform trajectories calculated from different sensors data (mono and stereo cameras, lidar, and Kinect 2 depth sensor) with various ROS-based SLAM methods in homogeneous indoor environment. The author used monocular ORB-SLAM, DPPTAM, stereo ZedFu and Kinect-related RTAB map packages, verifying the trajectories with a ground truth dealt with lidar-based Hector SLAM.

III. System setup

We developed a prototype of an unmanned ground vehicle (UGV, see, Fig. 1) for conducting our indoor experiment. This platform has a computational module as well as a set of sensors, including 2D lidar, a monocular camera, and ZED stereo camera. The detailed description of the software and hardware used is presented in this section.

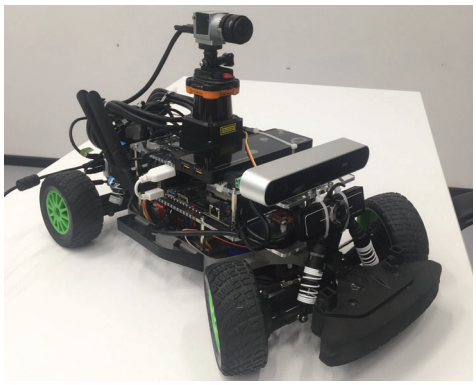
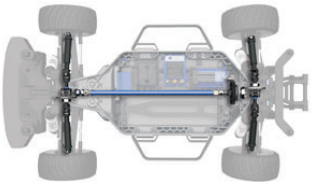


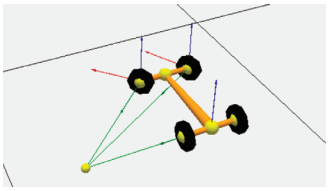
Fig. 1: Innopolis UGV prototype: Labcar platform with Lidar, Stereo and Mono camera

A. Hardware and software

The experiment consists of two parts: (1) the dataset acquisition, and (2) the dataset execution from the recorded sensors data. We used Labcar platform based on Traxxas 7407 Radio-Controlled Car Model [53], [54] for the dataset acquisition and a ground station for the dataset execution.



(a) 4WD Traxxas #74076



(b) Ackerman steering model

Fig. 2: Chassis of Labcar UGV based on Traxxas Radio-Controlled Car Model.

Labcar platform has chassis with Ackermann steering model, which is presented in Fig. 2. As far as this model does not assume rotations at relatively small speeds, it has a positive impact on robustness of V-SLAM methods (since they could have a bad performance at rotational movements). Our Labcar platform uses NVidia Jetson

TX1 embedded controller as the computation module with Ubuntu 16.04 operation system and custom ROS Kinetic Kame. The platform includes the following on-board sensors: 2D lidar, a monocular camera, and ZED stereo camera. Detailed information about robot platform is presented in Table III. Since we processed our results

TABLE III: HARDWARE AND SOFTWARE SPECIFICATION OF THE LABCAR PLATFORM

PARAMETERS	CONFIGURATION
Chassi	4WD Traxxas #74076
Hardware	Jetson TX1
Processor	Quad ARM A57
GPU	NVIDIA Maxwell
RAM	4 GB
Sensors	
Lidar	Hokuyo UTM-30LX
Camera	Basler acA1300-200uc
Stereo camera	ZED camera
Software	
JetPack	3.1
OS	Ubuntu 16.04
ROS	Kinetic Kame

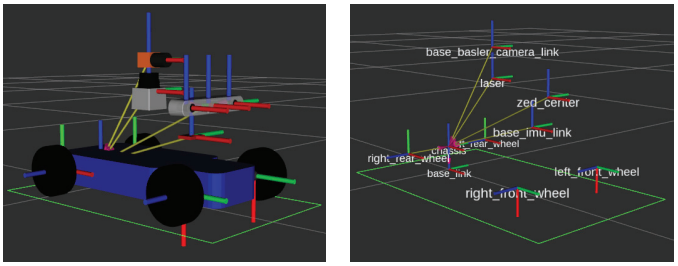
offline, we used the ground station to run different SLAM system on the dataset data which was acquired from Labcar platform sensors. The detailed information about the ground station is presented at the Table IV.

TABLE IV: HARDWARE AND SOFTWARE SPECIFICATION OF THE GROUND STATION

Parameters	Configuration
Processor	Intel Core i7 6500U
GPU	NVIDIA GeForce GTX 950M
RAM	12 GB
Software	**Configuration**
OS	Ubuntu 16.04
ROS	Kinetic Kame

B. Robot model

All tested SLAM systems have integration with ROS (Robot Operating System), that allows to use ROS advantages and improve the experiment quality. Some of them provide full integration, but some of them do this only in form of wrappers. For robot modeling, we applies URDF and *tf* package, transferring all data about UGV positions at the same coordinate frame, and what is more having data from different SLAM systems at the same coordinates with the *base_link*. It provides also more informative dataset representation and allows reducing human errors at conducting the experiments. The URDF robot model for Labcar platform is presented in Fig. 3.



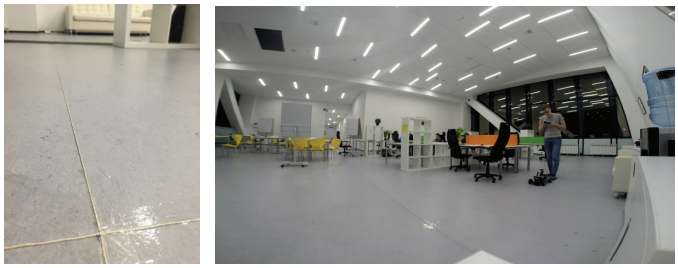
(a) Robot visualization in RViz (b) The model coordinate system

Fig. 3: The Labcar platform modelling in ROS/Gazebo simulator environment.

IV. METHODOLOGY OF THE EXPERIMENT

A. Experimental environment

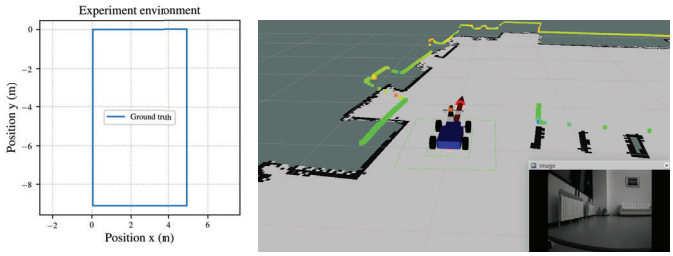
The experiment was conducted with a mobile robot launched within teleoperated closed-loop trajectory along a known perimeter of a rectangular work area for indoor environment of a typical office with monochrome walls. The experimental environment is presented in Fig. 5.



(a) Marked line (b) Test environment

Fig. 4: Indoor environment for the experiment with a mobile robot navigation.

We used a thread to mark the perimeter of UGV trajectory. The test area is presented in Figure 5a. We refer to this marked line as a Ground truth, although the robot trajectory coincided only during direct movement. The transition between two straight lines occurred with a turn radius of approximately 1 m. Data from sensors was collected in *ROS_bag* for Basler Camera, ZED stereo camera, and 2D lidar. Ground station was used to run SLAM algorithms based on the collected dataset.



(a) Test area (b) Lidar & camera data with robot model

Fig. 5: The test polygon model: lidar & camera data visualization with the robot model in RViz.

The Fig. 5b shows the visualization of the dataset in RViz, demonstrating the robot model, a window of the forward camera, 2D lidar point cloud and the map built.

B. Metrics for evaluation

We used Absolute Trajectory Error (ATE) for experiment evaluation, comparing the collect data from the tested system with more accurate data about UGV trajectory. We used data from 2D lidar-based Hector SLAM to verify visual SLAM systems, assuming that Hector SLAM provides higher accuracy that will be confirmed further.

Let (x_i, y_i, t_i) represent x and y coordinates at time t_i at the Hector SLAM system-based coordinate frame. Then $(\dot{x_i}, \dot{y_i}, \dot{t_i})$ correspond to data for the exploring method. We interpolate Hector SLAM trajectory with the first degree polynomial. Let denote $x(t)$ and $y(t)$ as interpolated Hector SLAM trajectory data. Then we compute the absolute trajectory error by the following equation 1:

$$ATE(\dot{t_i}) = \|(x(\dot{t_i}), y(\dot{t_i})) - (\dot{x_i}, \dot{y_i})\| \quad (1)$$

To represent this evaluation function we use statistical metrics: Root Mean Square Error (RMSE), Mean, Median, Standard deviation (Std.), Minimum, Maximum. Dataset was used to run different ROS-based SLAM system on the ground station, where all metrics were collected and evaluated.

V. Implementation

A. Lidar SLAM methods

In this section, we discuss SLAM systems based on 2D lidar data. In ROS architecture we use *urg_node*, which produces *sensor_msgs/LaserScan message* that contains the array of ranging information. Each range information corresponds to the angle. We focused on popular systems and compare their performance in terms of map building and localization after two laps.

1) GMapping: GMapping was developed in 2007, and it is still the one of the most common systems for robot application. The system uses Particle filter and creates the grid-based map [8]. The system subscribes to the *sensor_msgs/LaserScan message* and publishes *nav_msgs/OccupancyGrid*, *tf* transformation. The final map after two laps is presented in the Fig. 6*(a)*. Different parameters of the system were tested. The final map does not represent the real map of the environment. The trajectory of the robot cannot be computed since there is no robust map of the environment.

2) Hector SLAM: Hector SLAM system [10] was developed in 2011. The system subscribes to the *sensor_msgs/LaserScan message* and publishes *nav_msgs/OccupancyGrid*, *tf* transformation, and the pose with covariance message *geometry_msgs/PoseWithCovarianceStamped*. This information can be used for sensor fusion of systems based of Hector SLAM data. The system produces 2D map of environment precisely that can be used for navigation of mobile robot. The final map after two laps is presented in the Fig. 6*(b)*. The trajectory obtained from Hector SLAM system was compared with the marked line on the floor (which we refer as the ground truth), with conclusion of good fitting one to another. The trajectories are shown in the Fig. 6*(c)*. As far as the Hector SLAM path fits the ground truth, we make reasonable assumption that it produces the accurate measurements and can be used for trajectory validation of other SLAM systems.

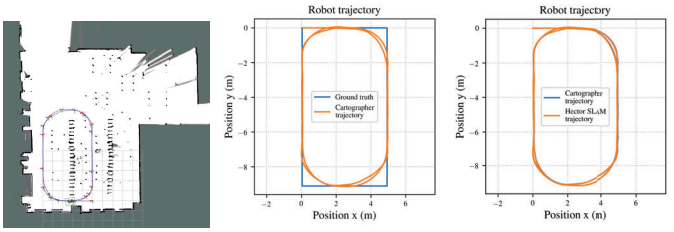
3) Cartographer: Cartographer [11] system was developed in 2016. The system subscribes to the *sensor_msgs/LaserScan message* and publishes *nav_msgs/OccupancyGrid*, *tf* transformation. The final map generated by this system after two UGV laps is presented in the Fig. 6*(d)*, and shows the best fitting from 2D Lidar maps to the real environment. The trajectory obtained from Cartographer system was compared with the ground truth, and the Fig. 6*(e)* demonstrates the good result, which is relevant for indoor robotics applications.

4) Summary: We tested three 2D Lidar SLAM systems: GMapping, Hector SLAM and Cartographer. The first system did not provide reliable results. The most important result in this section is comparison between Hector SLAM and Cartographer trajectories, which is presented in the Fig. 6*(f)*. The trajectories fit perfectly. We use absolute trajectory error which is defined at the methodology section IV-B. The results are presented in the Table V. The difference between two trajectories is less than 3 cm in RMSE, that allows to conclude that two systems produce actually the same results.

B. Monocular SLAM methods

In this section, the results of ROS-based monocular SLAM systems are presented. All systems were tested on data from monocular Basler camera with the frame rate of about 20 fps. The processing of SLAM systems was provided by an external PC. The main disadvantage of these systems is scale ambiguity problem, which means that there is no information about the environment in an absolute scale. We use manual handling to solve this problem.

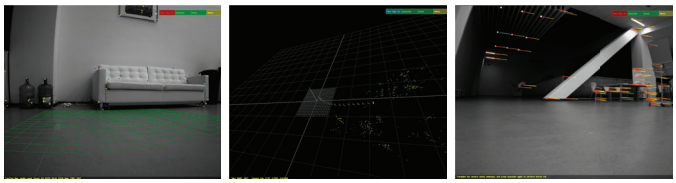
(a) GMapping map (b) Hector SLAM map (c) Hector SLAM path



(d) Cartographer map (e) Cartographer path (f) Hector vs Cartographer

Fig. 6: Maps and UGV trajectories recovered from 2D Lidar SLAM methods.

1) Parallel Tracking and Mapping (PTAM): PTAM is one of the keyframe-based visual SLAM system that allows to estimate 3D robot pose and build 3D map by tracking FAST features [17]. The system subscribes to the *sensor_msgs/Image* and publishes *geometry_msgs/PoseWithCovarianceStamped*, *tf* transformation. The system does not output a map to the ROS environment, therefore the map can be observed using the build-in tool. The feature detection and map visualization with camera pose estimation are presented in the Fig. 7*(a-b)*. We did not succeed to get reliable data from this system after two UGV laps. The system lost the track when robot had a turn, demonstrating a lack of robustness to lateral maneuvers. The process of map reinitialization after losing a track is presented in the Fig. 7*(c)*.



(a) Feature detection (b) Point cloud and path (c) Reinitialization

Fig. 7: The UGV camera-based visualization with Monocular PTAM method.

2) Semi-direct Visual Odometry (SVO): SVO [27] has a good integration with ROS, subscribes to the *sensor_msgs/Image* and publishes *geometry_msgs/PoseWithCovarianceStamped* for a robot pose, *sensor_msgs/PointCloud* for a map, and *tf* for transformation. The process of feature detection and map with pose estimation are presented in the Fig. 8. The system is not robust for this application since it loses track on robot turns, but can be used for additional information for straight UGV motion segments.

3) Dense Piecewise Parallel Tracking and Mapping (DPPTAM): DPPTAM is a direct system for pose estimation, which uses an assumption that regions with the same color belong to approximately the same plane [29]. We can see the key point detection and plane fitting in the Fig. 9. As far as the system lost tracking on robot turns, the system is not enough robust for this indoor application.

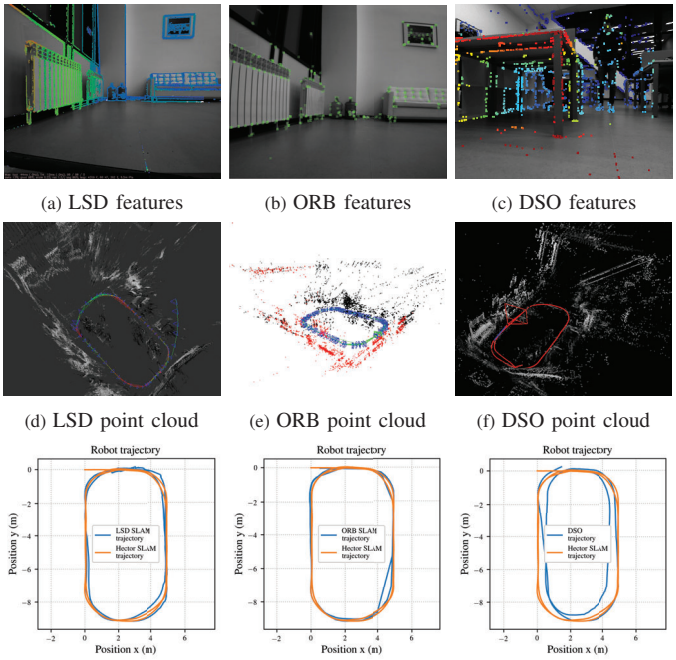
4) Large Scale Direct monocular SLAM (LSD SLAM): LSD SLAM is the direct method, which uses a local image to

(a) Feature detection (b) Point cloud and trajectory

Fig. 8: The UGV camera-based visualization with Monocular SVO method.

(a) Feature detection (b) Feature detection

Fig. 9: The UGV camera-based feature visualization with DPPTAM method.



(a) LSD features (b) ORB features (c) DSO features

(d) LSD point cloud (e) ORB point cloud (f) DSO point cloud

(g) LSD-based path (h) ORB-based path (i) DSO-based path

Fig. 10: The features *(a-c)*, maps *(d-f)* and UGV trajectories *(g-f)* recovered from the following Monocular SLAM methods: LSD, ORB and DSO.

image fitting, providing a pose prediction and building a dense map [25]. LSD SLAM system has a loop closure option that gives an advantage for drift error compensation. The system has a poor ROS integration, subscribing to the *sensor_msgs/Image* and publishing *tf* transformation. The process of key points detection and map with pose estimation are presented in the Fig. 10*(a,d)*. After scale correction, the UGV trajectory was recovered and compared with Hector SLAM trajectory (the result is presented in the Fig. 10*(g)*). The computed metrics for trajectory evaluation is present in the Table V. The system has a problem with a robot pose shift at the beginning of the map building since the map is not enough dense for precise localization. We can observe such a shift in the Fig. 10*(g)*. This shift can be filtered with a robot motion model. Thus, the system is robust and after absolute scale recovery can be used for robot pose estimation and map building.

5) ORB SLAM (mono): The ORB SLAM is a feature-based system, which tracks ORB features for robot pose estimation [22]. It creates sparse point cloud as the map. The system has loop closure detection and widely used for indoor robotics application. The process of features detection and map with pose estimation are presented in the Fig. 10*(b,e)*. The system requires an additional development to be integrated in ROS. The camera trajectory can be evaluated with *txt*-file only after dataset processing. The map is the good approximation of test environment. The trajectory of the robot from ORB SLAM system in comparison with Hector SLAM is shown in the Fig. 10*(h)*. The pose estimation is provided only during keyframes, and the information about the camera location is more sparse. The metrics for the trajectory evaluation is presented in the Table V. The ORB SLAM system is very robust and provides the good approximation of robot trajectory if the task of absolute scale recovery is solved.

6) Direct Sparse Odometry (DSO): DSO is the fully direct method [30]. The system uses the wrapper for ROS integration and requires an additional development for real robotics implementation. The main advantage of this system is the dense map. The process of key points detection and map with robot trajectory are presented in the Fig. 10*(c,f)*. DSO publishes robot pose as *tf* transformation. The DSO-based UGV trajectory in comparison with Hector SLAM is presented in the Fig. 10*(i)*. The system used without a loop closure, therefore the robot pose drift was accumulated during the first lap and created doubled environment objects in the map when the robot did the second lap. It is also possible that calibration difficulties [55] could lead to additional errors in our experiments. The metrics for the trajectory evaluation is presented in the Table V. DSO creates the dense map and the system is robust in terms of pose tracking, but the lack of loop close makes this map noisy. The system improved with a loop closure detection will give more precise results at the condition of careful calibration.

7) Summary: Monocular SLAM systems provide the information about UGV localization and mapping. But since there is no absolute scaling, it is required additional modules for scale recovery. Thus, monocular systems can be used as the source of additional information for pose estimation and map building.

C. Stereo SLAM methods

The main advantages of stereo SLAM systems that they provide with absolute scale estimation. The systems were processed offline by an external PC, using the dataset collected from ZED Stereolabs camera with the data stream of about 10 fps.

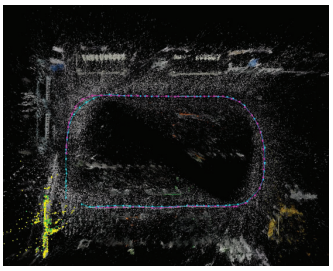
1) ZEDfu: ZEDfu is the software solution with closed source code, which was provided by ZED camera developer to solve SLAM problem. The UGV trajectory is computed on-board, using GPU-accelerated code. The robot trajectory is presented in the Figure 11*(g)*. The metrics for trajectory evaluation are presented in the Table V. For our indoor experiment the method provided sufficiently large RMSE (almost two robot sizes). Thus, the system is robust in terms of robot pose tracking, but the trajectory is not accurate for such type of robot navigation.

2) Real-Time Appearance-Based Mapping (RTAB map): RTAB map is the feature-based SLAM system [28], which has a loop close detection and good integration with ROS. The feature extraction, map and robot trajectory after two laps are presented in the Fig. 11*(a,c,e)*. The system sometimes has the problem with pose estimation when the robot moves closer to monotonous walls. But the

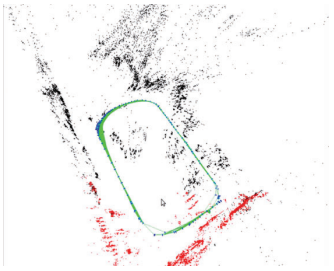
Failure Recovery System is able to detect the failure and determine a robot pose properly. The robot trajectory from RTAB map system is presented in the Fig. 11*(e)*. It has a shift at the end of the first lap, but after a loop close detection it relocates the robot position. The metrics for trajectory evaluation are presented in the Table V. The system is robust and accurate. It solves localization problem with accuracy comparable with Lidar methods without additional manipulations that allows to use it for indoor robotics applications.

(a) RTAB map features

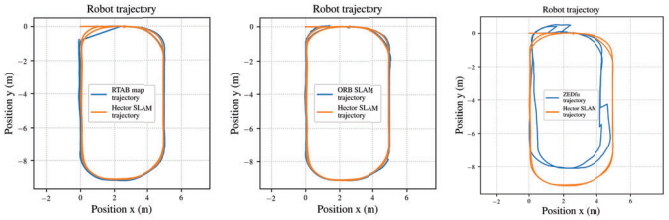
(b) ORB features



(c) RTAB map point cloud



(d) ORB point cloud

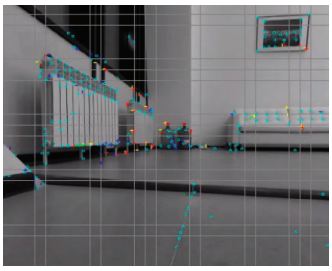


(e) RTAB map path (f) Stereo ORB path (g) ZEDfu path

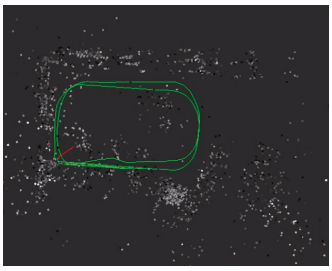
Fig. 11: The features *(a,b)*, maps *(c,d)* and UGV trajectories *(e-g)* recovered from Stereo SLAM methods: RTAB map, Stereo ORB and ZEDfu.

3) ORB SLAM (stereo): ORB SLAM system has also the stereo mode [22], which provides absolute scaling in metres. The feature detection, mapping and UGV pose estimation are presented in the Fig. 11*(b,d,f)*. The metrics for UGV trajectory evaluation re presented in the Table V. The ORB SLAM with stereo camera provides accurate pose estimation and create sparse 3D map, allowing robust robot localization and pose tracking for indoor applications.

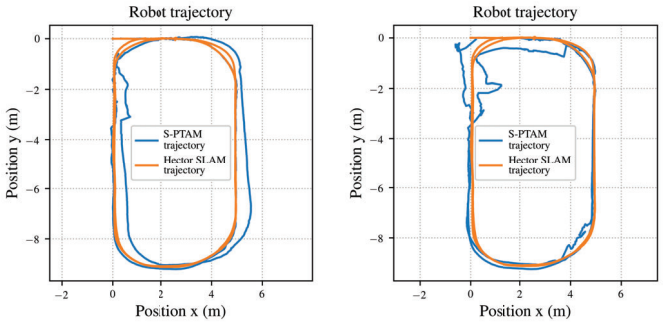
4) Stereo Parallel Tracking and Mapping (S-PTAM): One of the most modern systems is feature-based S-PTAM [33], which has a framework for SLAM problem with a loop closure detection. It has also a possibility to choose any key point detector and feature descriptor available in OpenCV. The feature detection, mapping and UGV pose estimation are presented in the Fig. 12. The system was tested in two modes with loop closure and without, and compared with Hector SLAM-based trajectory in the Fig. 12*(c,d)*. The metrics for trajectory evaluation are presented in the Table V. The system is very flexible in terms of keypoint detection and selection of feature descriptors. It has loop closure option but it can demonstrate problems with the default "Bag of Words"-based place recognition dataset. It is required to create the Bag of Words for indoor office application and test loop closure. The system is robust in terms of pose tracking, and has the sparse map. But the localization accuracy can be not enough for robot navigation.



(a) Feature detection



(b) Point cloud and trajectory



(c) Loop closure - off (d) Loop closure - on

Fig. 12: The features *(a)*, maps *(b)* and UGV trajectories *(c,d)* recovered from the following S-PTAM method.

5) Summary: Stereo visual SLAM systems provide metric information about UGV localization. The results for stereo systems: Real-Time Appearance-Based Mapping (RTAB map), ORB SLAM, Stereo Parallel Tracking and Mapping (S-PTAM) can be accurate enough at solving localization problems for indoor robotics applications.

VI. Evaluation and Discussion

A. Odometry analysis

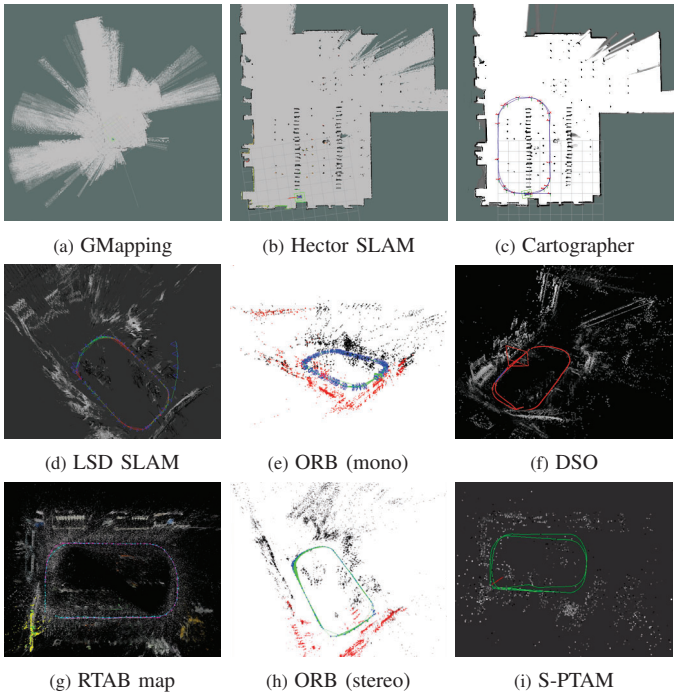
The metrics, which evaluate mobile robot localization with different SLAM methods, are presented in Table V.

TABLE V: ABSOLUTE TRAJECTORY ERROR FOR DIFFERENT SYSTEMS BASED ON HECTOR SLAM TRAJECTORY

System	RMSE (m)	Mean (m)	Median (m)	Std. (m)	Min (m)	Max (m)
Cartographer	0.024	0.017	0.013	0.021	0.001	0.07
LSD SLAM	0.301	0.277	0.262	0.117	0.08	0.553
ORB SLAM (mono)	0.166	0.159	0.164	0.047	0.047	0.257
DSO	0.459	0.403	0.419	0.219	0.007	0.764
ZEDfu	0.726	0.631	0.692	0.358	0.002	1.323
RTAB map	0.163	0.138	0.110	0.085	0.004	0.349
ORB SLAM (stereo)	0.190	0.151	0.102	0.115	0.004	0.414
S-PTAM (no loop cl.)	0.338	0.268	0.244	0.206	0.001	0.768
S-PTAM (loop cl.)	0.295	0.257	0.242	0.145	0.006	1.119

From the tested SLAM methods, RTAB map can be considered as one of the best method in terms of RMSE for mobile robot localization problem in homogeneous indoor office environment. The monocular ORB SLAM may be estimated as the second most accurate method, whereas ZEDfu demonstrated the worse results for our experiments.

B. Maps analysis



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

(d) LSD SLAM (e) ORB (mono) (f) DSO

(g) RTAB map (h) ORB (stereo) (i) S-PTAM

Fig. 13: Maps generated by various SLAM methods

The maps generated by different SLAM systems are presented in Figure 13.

GMapping, Hector SLAM, Cartographer lidar based methods are provides *nav_msgs/OccupancyGrid* map. In our experiments, GMapping built an inaccurate map, whereas Hector SLAM and Cartographer provided quite similar maps. However, it should be noted that since Cartographer uses global map optimization cycle and local probabilistic map updates, it makes this system more robust to environmental changes. Therefore, at present we can recommend Cartographer as the best choice for 2D lidar SLAM. This conclusion is supported by this research [52].

LSD SLAM, DSO are direct methods, which provide dense point cloud maps and allow to make 3D scene recovery and object detection.

ORB SLAM, RTAB map, S-PTAM are feature-based methods that build sparse point cloud maps. Therefore, there are difficulties to make 3D scene recovery, but they are good enough for solving navigation problems.

VII. Conclusion

As the result of our work the mobile robot prototype to provide experiments in a typical office environment was developed. The robot was launched indoor within teleoperated closed-loop trajectory along a known perimeter of a square work area, recording sensors telemetry data for offline processing. Thus, the dataset from on-board sensors (2D lidar, monocular and stereo cameras) was collected and processed using the following SLAM methods: (a) 2D lidar-based: GMapping, Hector SLAM, Cartographer; (b) monocular camera-based: Large Scale Direct monocular SLAM (LSD SLAM), ORB SLAM, Direct Sparse Odometry (DSO); and (c) stereo camera-based: ZEDfu, Real-Time Appearance-Based Mapping (RTAB map), ORB SLAM, Stereo Parallel Tracking and Mapping (S-PTAM). The results of execution for SLAM systems were investigated and compared with the use of metrics.

The main results of our investigations:

- 2D lidar SLAM systems: Hector SLAM and Cartographer provide accurate solutions for UGV localization and map building. The methods provide almost the same results with RMSE of Absolute Trajectory Error (ATE) at 0.024 m. Both trajectories coincide with the marked line on the floor. However, since Cartographer uses global map optimization cycle and local probabilistic map updates, it makes this system more robust to environmental changes.
- Monocular visual SLAM systems: Parallel Tracking and Mapping (PTAM), Semi-direct Visual Odometry (SVO), Dense Piecewise Parallel Tracking and Mapping (DPPTAM) failed the experiments since they lost track due to lack of features.
- Monocular visual SLAM systems: Large Scale Direct monocular SLAM (LSD SLAM), ORB SLAM, Direct Sparse Odometry (DSO) can be used for solving localization problem with an additional module for scale recovery.
- No monocular SLAM system could handle scale ambiguity problem without additional information about environment for scale recovery.
- Stereo visual SLAM systems: ZEDfu, Real-Time Appearance-Based Mapping (RTAB map), ORB SLAM, Stereo Parallel Tracking and Mapping (S-PTAM) provide metric information about localization without additional scaling modules, also building 3D metric point cloud.
- Visual SLAM system: RTAB map demonstrated the best results for localization problem in our experiments with RMSE ATE of 0.163 m, but it has the problem with the track lost close to monochrome walls. The most robust and stable between tested system is ORB SLAM with RMSE ATE of 0.190 m.

Since the performance of visual SLAM systems strongly depends on computational resources of a mobile robot, our future studies can concern hardware limitations and their influences on efficiency of SLAM applications.

References

[1] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, no. 91, pp. 1442–1447, 1991.

[2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and 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] T. Bailey, J. Nieto, J. Guivant, M. Stevens, and E. Nebot, "Consistency of the ekf-slam algorithm," in *Int. Conference on Intelligent Robots and Systems (IROS)*, pp. 3562–3568, IEEE, 2006.

[4] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *Aaai/iaai*, vol. 593598, 2002.

[5] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[6] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: A survey," *Frontiers in Artificial Intelligence and Applications*, vol. 184, no. 1, pp. 363–371, 2008.

[7] B. Barshan and H. F. Durrant-Whyte, "Inertial Navigation Systems for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 328–342, 1995.

[8] 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.

[9] K. P. Murphy, "Bayesian map learning in dynamic environments," *Advances in Neural Information Processing Systems 12*, pp. 1015–1021, 2000.

[10] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," *9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011*, pp. 155–160, 2011.

[11] 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.

[12] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2012.

[13] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[14] S. Se, D. Lowe, and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[15] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Robotics and Autonomous Systems*, vol. 43, no. 4, pp. 215–229, 2003.

[16] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," *ICCV*, vol. 2, pp. 1403–1410, 2003.

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, IEEE, 2007.

[18] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (slam): Part ii," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[19] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *Robotics and Automation (ICRA), International Conference on*, pp. 2609–2616, IEEE, 2014.

[20] E. Rosten and T. Drummond, "Machine Learning for High Speed Corner Detection," *Computer Vision – ECCV 2006*, vol. 1, pp. 430–443, 2006.

[21] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, pp. 778–792, Springer, 2010.

[22] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Trans. on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*, pp. 2564–2571, IEEE, 2011.

[24] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 2320–2327, IEEE, 2011.

[25] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.

[26] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct slam with stereo cameras," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 1935–1942, IEEE, 2015.

[27] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 15–22, 2014.

[28] M. Labbé and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," *IEEE Int. Conference on Intelligent Robots and Systems*, pp. 2661–2666, 2014.

[29] A. Concha and J. Civera, "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5686–5693, 2015.

[30] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[31] T. Whelan, S. Leutenegger, R. Salas Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM Without A Pose Graph," *Robotics: Science and Systems XI*, 2015.

[32] K. Tateno, F. Tombari, I. Laina, and N. Navab, "Cnn-slam: Real-time dense monocular slam with learned depth prediction," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[33] T. Pire, T. Fischer, G. Castro, P. DeCristóforis, J. Civera, and J. JacoboBerlles, "S-PTAM: Stereo Parallel Tracking and Mapping," *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.

[34] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Hollerer, "Live tracking and mapping from both general and rotation-only camera motion," in *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pp. 13–22, IEEE, 2012.

[35] C. Pirchheim, D. Schmalstieg, and G. Reitmayr, "Handling pure camera rotation in keyframe-based slam," in *Mixed and Augmented Reality (ISMAR), International Symposium on*, pp. 229–238, IEEE, 2013.

[36] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and vision for absolute scale estimation in monocular SLAM," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 61, no. 1-4, pp. 287–299, 2011.

[37] D. Zhou, Y. Dai, and H. Li, "Reliable scale estimation and correction for monocular visual odometry," in *Intelligent Vehicles Symposium (IV)*, pp. 490–495, IEEE, 2016.

[38] A. Mulloni, M. Ramachandran, G. Reitmayr, D. Wagner, R. Grasset, and S. Diaz, "User friendly SLAM initialization," *2013 IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013*, no. October, pp. 153–162, 2013.

[39] C. Arth, C. Pirchheim, J. Ventura, D. Schmalstieg, and V. Lepetit, "Instant Outdoor Localization and SLAM Initialization from 2.5D Maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 21, no. 11, pp. 1309–1318, 2015.

[40] P. Newman and K. Ho, "Slam-loop closing with visually salient features," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 635–642, IEEE, 2005.

[41] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras," *Procedings of the British Machine Vision Conference 2013*, pp. 93.1–93.11, 2013.

[42] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular slam: design, survey, and future directions," *Robotics and Autonomous Systems*, vol. 98, pp. 67–88, 2017.

[43] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Mg, "ROS: an open-source Robot Operating System," *Icra*, vol. 3, no. Figure 1, p. 5, 2009.

[44] 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.

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

[46] 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.

[47] M. Sokolov, O. Bulichev, and I. Afanasyev, "Analysis of ros-based visual and lidar odometry for a teleoperated crawler-type robot in indoor environment," in *Proc. Int. Conf. on Informatics in Control, Automation and Robotics (ICINCO)*, 2017.

[48] 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.

[49] 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.

[50] 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), Int. Symposium on*, pp. 1–6, IEEE, 2013.

[51] 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.

[52] R. Yagfarov, M. Ivanou, and I. Afanasyev, "Map comparison of lidar-based 2d slam algorithms using precise ground truth," in *Int. Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2018.

[53] I. Shimchik, A. Sagitov, I. Afanasyev, F. Matsuno, and E. Magid, "Golf cart prototype development and navigation simulation using ros and gazebo," in *MATEC Web of Conferences*, vol. 75, p. 09005, EDP Sciences, 2016.

[54] T. S. Sheikh and I. Afanasyev, "Stereo vision-based optimal path planning with stochastic maps for mobile robot navigation," in *Int. Conference on Intelligent Autonomous Systems (IAS)*, 2018.

[55] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2878–2885, 2018.