Eötvös Loránd University

Faculty of Informatics

Erasmus Mundus Joint Master in

Intelligent Field Robotic Systems

Failure-resilient Graph-based SLAM for Autonomous Robotic Exploration in GNSS-Denied Environments

Author:

Vsevolod Hulchuk

Bachelor of Science

Internal supervisor:

Zoltan Istenes

Associate Professor

External supervisors:

Jan Bayer — Ing.

Jan Faigl — prof. Ing., Ph.D.

Prague / Budapest, 2023

"In the canvas of exploration, SLAM paints a vibrant tapestry, blending sensor data strokes and probabilistic hues to compose a masterpiece of understanding."

"Na plátně průzkumu maluje SLAM živou tapisérii, na níž se mísí tahy dat ze senzorů a pravděpodobnostní odstíny, aby vzniklo mistrovské dílo porozumění."

"En el llenç d'exploració, SLAM pinta un tapís vibrant, combinant traços de dades del sensor i matisos probabilistes per compondre una obra mestra de comprensió."

"На полотні дослідження SLAM малює яскравий гобелен, поєднуючи мазки сенсорних даних та імовірнісні відтінки, щоб створити шедевр розуміння."

- ChatGPT [1]

STATEMENT

OF THESIS SUBMISSION AND ORIGINALITY

I hereby confirm the submission of the Master Thesis Work on the IFRoS MSc course with author and title:

Name of Student: Vsevolod Hulchuk

Code of Student: HYVBV0

Title of Thesis: Failure-resilient Graph-based SLAM for Autonomous Robotic Exploration in GNSS-Denied Environments

Supervisor: Dr. Zoltán Istenes

....

at Eötvös Loránd University, Faculty of Informatics.

In consciousness of my full legal and disciplinary responsibility I hereby claim that the submitted thesis work is my own original intellectual product, the use of referenced literature is done according to the general rulesof copyright.

I understand that in the case of thesis works the following acts are considered plagiarism:

- literal quotation without quotation marks and reference;
- citation of content without reference;
- presenting others' published thoughts as own thoughts.

Budapest, 2023.

....

student

EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

MASTER THESIS TOPIC DECLARATION FORM

Name of Student: Vsevolod Hulchuk

Neptun code: HYVBV0

Training: Full-time

Technical major: Robotics

Email Address: vsevolod.hulchuk@gmail.com

Phone Number: + 34666523119

Name of ELTE supervisor: Dr. Zoltán Istenes

Department: Faculty of Informatics

Contact (email, phone): istenes@inf.elte.hu, +36306552545

Name of Industrial supervisor: prof. Ing. Jan Faigl, Ph.D.

Department: Department of Computer Science, Faculty of Electrical Engineering
CTU in Prague

Contact (email, phone): faiglj@fel.cvut.cz, +420-22435-7400

Information about the internship

Name of the company: Czech Technical University in Prague

Starting date of internship: 21.1.2023

Closing date of internship: 28.5.2023

Weekly schedule: 8 hours per day, Monday-Friday

The purpose of the admission declaration is to certify that the student of the MSc in Intelligent Field Robotics System at ELTE Faculty of Informatics may complete the mandatory internship in the selected institution within the framework detailed hereby and in accordance with the learning outcomes required by the program.

Title of the thesis

Autonomous Robotic Exploration in Communication-Denied Environments

Topic of the thesis (1 – 1,5 pages)

During autonomous robotic exploration, human supervision is desired to monitor the mission performance or assist in specific cases where autonomy cannot safely resolve the mission progress. In communication-denied environments, suitable communication infrastructure is unavailable, and the robot needs to build and maintain communication connectivity with a supervision base station. The thesis goal is to investigate localization methods and active perception strategies in combined spatial exploration and communication infrastructure building.

The following achievements are expected.

- Familiarize with the existing autonomous exploration frameworks [1,2] and communication accessibility models based on interpolation [3] and extrapolation [4].
- Propose an integration of existing state-of-the-art Simultaneous Location and Mapping (SLAM) methods [5], such as GTSMA [6], into the exploration framework [7] used in a single robot exploration scenario [8].
- Propose exploration strategy in active perception scenario combining spatial exploration with communication infrastructure building [9] with deployable communication modules [4].
- Develop and integrate proposed solutions using a multi-legged walking robot or wheeled ground robotic platform of the Computational Robotics Laboratory at the Czech Technical University in Prague (CTU).
- Propose ideas for an eventual generalization for the multi-robot systems.
- Develop and integrate proposed solutions using a multi-legged walking robot or wheeled ground robotic platform of the Computational Robotics Laboratory at the Czech Technical University in Prague (CTU).
- Propose ideas for an eventual generalization for a multi-robot exploration framework [9].

The following research questions are supposed to be answered.

- How does the proposed SLAM system improve navigation for a restricted set of sensors?
- What is the performance effect of combining spatial and communication exploration models?
- How does the proposed exploration system perform with multiple robots?

The following tools and methods are expected to be used.

- The development would be performed using ROS1 and C++ programming language. For simulation, the DARPA Subt simulator[1] will be used. It could be launched using docker.
- For Localization, the robot would use 3D point clouds outputted by the 3D Lidar. Other sensors may be integrated. An Iterative Closest Point (ICP) method combined with Inertial Measurement Unit

[1] https://github.com/osrf/subt

(IMU) may be used for basic odometry. On top of that, SLAM frameworks, such as GTSAM[2], may be added to fuse with other measurements.

- For mapping the environment, a 2.5D map representation would be used. The ANYbots library[3] will be used for the first tests, with a possible switch to a custom mapping implementation inspired by it. The Eigen library will likely be used. In any case, the map could be represented using the GridMap[4] library or by a custom implementation inspired by it.
- Custom code or Robotic Systems Lab's package[5] may be used to generate traversability maps.
- The move_base[6] package could be used for the navigation stack during the early stage. Afterward, custom nodes for path planning and path following could be implemented.
- Custom frontier detection and different frontier fusion could be used for exploration management.

Keywords: Mobile robotic exploration, autonomous navigation, localization, SLAM, communication infrastructure building.

[1] Lluvia, I., Lazkano, E., and Ansuategi, A.: Active mapping and robot exploration: A survey, Sensors, vol. 21, no. 7, 2021. https://doi.org/10.3390/s21072445.

[2] Amigoni, F., Banfi, J., and Basilico, N.: Multirobot Exploration of Communication-Restricted Environments: A Survey, IEEE Intelligent Systems, vol. 32, no. 6, pp. 48-57, 2017. https://doi.org/10.1109/MIS.2017.4531226.

[3] Quattrini Li, A., Penumarthi, P.K., Banfi, J. et al.: Multi-robot online sensing strategies for the construction of communication maps. Autonomous Robots, vol. 44, no. 3, 299–319, 2020. https://doi.org/10.1007/s10514-019-09862-3

[4] Zoula, M., Prágr, M., and Faigl, J.: On Building Communication Maps in Subterranean Environments, 2020 Modelling and Simulation for Autonomous Systems (MESAS), 2021, pp. 15-28. https://doi.org/10.1007/978-3-030-70740-8_2.

[5] Grisetti, G., Kümmerle, R., Stachniss, C. and W. Burgard: A Tutorial on Graph-Based SLAM, IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31–43, 2010, https://doi.org/10.1109/MITS.2010.939925.

[6] Dellaert, F. et al.: GTSAM4, https://gtsam.org/ [cited on 2022-10-14].

[7] Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Azayev, T., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Krátký, V., Petráček, P., Baril, D., Vaidis M., Kubelka, V., Pomerleau, F., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., and Krajník, T. (2022). System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA Subterranean Challenge. Field Robotics, 2, 1779–1818. https://doi.org/10.55417/fr.2022055

[8] Bayer, J. and Faigl, J.: Speeded Up Elevation Map for Exploration of Large-Scale Subterranean Environments, 2019 Modelling and Simulation for Autonomous Systems (MESAS), 2020, pp. 190–202. https://doi.org/10.1007/978-3-030-43890-6_15.

[9] M. Zoula: Communication Infrastructure Building in Mobile Robot Exploration, Master thesis, Czech Technical University in Prague, 2023.

[2] https://gtsam.org/
[3] https://github.com/ANYbotics/elevation_mapping
[4] https://github.com/ANYbotics/grid_map
[5] https://github.com/leggedrobotics/traversability_estimation
[6] http://wiki.ros.org/move_base

[10] Bayer, J. and Faigl, J.: Decentralized Topological Mapping for Multi-robot Autonomous Exploration under Low-Bandwidth Communication, European Conference on Mobile Robots (ECMR), 2021. https://doi.org/10.1109/ECMR50962.2021.9568824.

Encryption of the topic is necessary: **NO**

I ask for the acceptance of my thesis topic.

Budapest, 31/01/2023

Vsevolod Hulchuk

Student

I approve of the suggested topic of the Master's Thesis:

Budapest, …./…./2023

Dr. Zoltán Istenes

ELTE supervisor

I approve of the suggested topic of the internship, and in the name of the institution (organization, company) above I agree, that the named student will carry out his/her internship along the conditions detailed above:

Budapest, …./…./2023

prof. Ing. Jan Faigl, Ph.D.

Industrial supervisor

The topic of the thesis and internship is approved by the Dean of Faculty of ELTE Informatics

Budapest, …./…./2023

Kozsik Tamás

Digitally signed by
Dr. Kozsik Tamás
Date: 2023.02.09
19:19:11 +01'00'

…………………………………………………

Dr. Tamás Kozsik

Dean of Faculty of Informatics

Contents

Acknowledgements **3**

Abstract **4**

1 Introduction **5**

1.1 Background and context 5

1.2 Problem Statement 7

1.3 Methodology and Approach 8

1.4 Thesis's Structure Overview 10

2 Literature Review **11**

2.1 Overview of Relevant Literature 11

2.2 Theoretical framework 14

2.2.1 Exponential Smoothing Filter 14

2.2.2 Outlier Detection 15

3 Methodology **17**

3.1 System Overview 17

3.2 Assumptions and Limitations 18

3.3 Localization 19

3.3.1 Localization Failure Detection 19

3.3.2 Visual Odometry Integration and Scale Self-adjustment 21

3.4 Exploration 23

3.4.1 Mapping and Traversability Estimation 23

3.4.2 Exploration Strategy 25

3.4.3 Path Planning and Following 26

3.5 Tools and Technologies Used 27

4 Results **29**

4.1 Localization Failure Detection Indicator 32
4.2 Parameterization of the Failure Detection 33
4.3 Localization Performance in the Campus Scenario 33
4.4 Localization Performance in the Parking Lot Scenario 36
4.5 Exploration 37

5 Discussion **39**
5.1 Localization 39
5.2 Exploration 40
5.3 Future Research Directions 41

6 Conclusion **43**

Bibliography **44**

Acknowledgements

I would like to express my deepest gratitude to Jan Bayer, my direct supervisor from the COMROB laboratory at the Czech Technical University in Prague (CTU) for providing unwavering support throughout the thesis work, providing valuable resources, ideas, advice, feedback, and assistance with experiments. I extend my thanks to prof. Ing. Jan Faigl, my external supervisor at CTU, for his wise guidance, timely insights, and valuable contributions to the research. I am also grateful to all the other members of the COMROB laboratory at CTU who provided assistance and advice on various aspects of the thesis. We use the term "we" throughout the thesis because the work was done in a close collaboration with my supervisors from CTU.

I would like to express my sincere gratitude to Dr. Narcís Palomeras from the University of Girona (UdG) for imparting to me a deep appreciation and understanding of autonomous systems. I am also grateful to Dr. Pere Ridao of UdG for introducing me to the world of robot localization. I am grateful to Miguel Castillón of UdG for sparking my interest in graph-based SLAM. I would like to thank Dr. Nuno Gracias and Dr. Rafael Garcia of UdG for teaching me the fundamentals of various aspects of robotics, including computer vision.

The presented work has been supported by:

- Erasmus Mundus Joint Master in Intelligent Field Robotic Systems (IFRoS);
- The support under the OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics," and R&D project from program NCK, No. TN02000028 "Center for advanced machines and manufacturing technology" is also gratefully acknowledged.

Abstract

In this thesis, motivated by the autonomous exploration problem, we address the problem of autonomous robot localization using Simultaneous Localization and Mapping (SLAM) with Light Detection and Ranging (LiDAR) perception enhanced by black-box visual odometry in scenarios where laser scan matching can be ambiguous because of a lack of sufficient features in the scan. We propose to develop a novel localization method based on the Graph SLAM approach that benefits from fusing data from multiple types of sensors to overcome the drawbacks of using only LiDAR data. The proposed localization method uses a failure detection model based on the quality of the LiDAR scan matching and inertial measurement unit (IMU) data. The failure model improves LiDAR-based localization by an additional localization source, including low-cost black-box visual odometers like the Intel RealSense T265. The proposed method is compared to the state-of-the-art localization system LIO-SAM in cluttered and open urban areas. Based on the performed experimental deployments, the proposed failure detection model with black-box visual odometry sensor yields improved localization performance measured by the absolute trajectory and relative pose error indicators. Furthermore, we developed elevation mapping and traversability estimation to employ the proposed localization method in autonomous robotic exploration that is based on the frontier-based exploration strategy. The proposed localization method has been experimentally validated within the developed exploration framework in the outdoor field experimental deployments in the campus backyard, where it allows building successfully aligned map of the environment.

Chapter 1

Introduction

This section starts with describing the context of the SLAM problem for exploration in Section 1.1. The specific addressed problem is explained in Section 1.2. The section continues by summarizing the proposed approach to face the problem in Section 1.2. Finally, in Section 1.4, the structure of the thesis remainder is described.

As the research progressed, it became apparent that there was a need to address the issue of failure-resilient simultaneous localization and mapping (SLAM) in autonomous robotic exploration, especially in environments without Global Navigation Satellite System (GNSS) signals. Consequently, the title of the thesis was changed from "Autonomous Robotic Exploration in Communication-Denied Environments" to "Failure-Resilient Graph-based SLAM for Autonomous Robotic Exploration in GNSS-Denied Environments" to emphasize the importance of effectively addressing this problem.

1.1 Background and context

The autonomous robotic exploration problem appears in scenarios where a map of the operational environment is not a prior known, such as search-and-rescue missions, and the robot is requested to build such a map. Hence, spatial robotic exploration aims to explore an unknown environment and build a desired environment representation, a map. In autonomous exploration, it is necessary to address the autonomous robot navigation and decision-making to determine where to navigate the robot next, represented as the exploration strategy. The navigation itself consists of localization, mapping, path planning, and path-following tasks. Mapping is needed to interpret the environment; elevation maps [2], which store the height of the environment, are suitable environment representations for outdoor scenarios with rough terrains. However, to build a map from the sensory data, it is necessary to align them properly and consistently, which requires a sufficiently precise estimation of the robot (sensor) pose (position and orientation of the robot and its sensory equipment). Having a map, the exploration strategy decides where the robot should move to maximize exploration of the reachable parts of the environment. One of the commonly used exploration strategies is frontier-based exploration [3] that steers the robot navigation towards the boundary (frontier) of the already explored and not yet explored parts of the environment and thus increases the explored parts of the environment.

The performance of the exploration highly depends on the localization system that enables environment mapping for which the robot needs to know its pose. In scenarios where external localization systems, such as satellite navigation, are unavailable or do not work reliably because of signal reflections from tall structures, the robot's sensors-based localization is required. In the presented thesis, autonomous robotic exploration is the motivation to develop a reliable localization system.

The widely adopted method for localizing a robot in an unknown environment using its sensors is *Simultaneous Localization and Mapping* (SLAM) [4]. The approach builds a map of the environment and, at the same time, localizes the robot with respect to (w.r.t.) such a map. SLAM can be based on data from various sensors, including *Light Detection and Ranging* (LiDAR) laser scanners [5], visual cameras [6], *inertial measurement units* (IMU), or wheeled odometry, to name just a few.

Using exteroceptive sensors to build a map of the operational environment within which the robot is localized allows for decreasing the localization drift compared to purely proprioceptive incremental methods such as odometry and dead reckoning. Even matching only consecutive frames in *Visual Odometry* (VO) [7] might help to overcome drifts of IMU measurements or slippage of the wheeled odometry. Nevertheless, the map's quality is important and related to the data quality, specifically the depth estimates of the range measurements. LiDAR sensors provide relatively precise range measurements and can have a resolution in hundreds or thousands of scanning lines [8]. These properties make them suitable for localization, especially in cluttered environments, where LiDAR scans can be precisely matched w.r.t. each other [9]. However, the scan matching may be ambiguous in long corridors or flat fields, leading to localization failure or a high drift.

Incremental localization methods, such as IMU and odometry-based methods, including VO, might help to overcome areas where LiDAR scan matching is ambiguous locally, albeit it can lead to higher drift than the LiDAR-based SLAM in the long run. Thus, combining multiple data sources can be advantageous in SLAM, and two main sensor fusion approaches can be found in the literature. The first is *tightly-coupled* methods that account for all the sensor raw data to get the localization estimation, such as in *LiDAR Inertial Odometry via Smoothing and Mapping* [10] (LIO-SAM), where an IMU displacement measurement serves as an initial guess for the scan-matching of LiDAR scans. The second class of sensor fusion methods is *loosely-coupled* approaches that fuse multiple localization sources, meaning that two displacement outputs from localization systems are fused at the top. Consequently, the resulting estimation tends to be more robust because a failure of one source does not provoke the failure of another. Besides, loose coupling allows the integration of several independent localization systems, making the whole system modular and easily replaceable compared to tightly-coupled systems. However, in scenarios where both localization sources function normally, tightly-coupled methods might improve the performance better than loosely-coupled methods. For example, a loosely-coupled method for the fusion of LiDAR and IMU would not give an advantage of a good initial guess for the scan matching.

In the presented thesis, we report on the developed failure-resilient localization system to address the localization problem leveraging both tight and loose coupling. Motivated by the deployment of exploration, we developed an exploration framework that combines the proposed localization, mapping, exploration strategy with path planning and path following to validate the proposed localization method and reveal potential improvements of the method.

1.2 Problem Statement

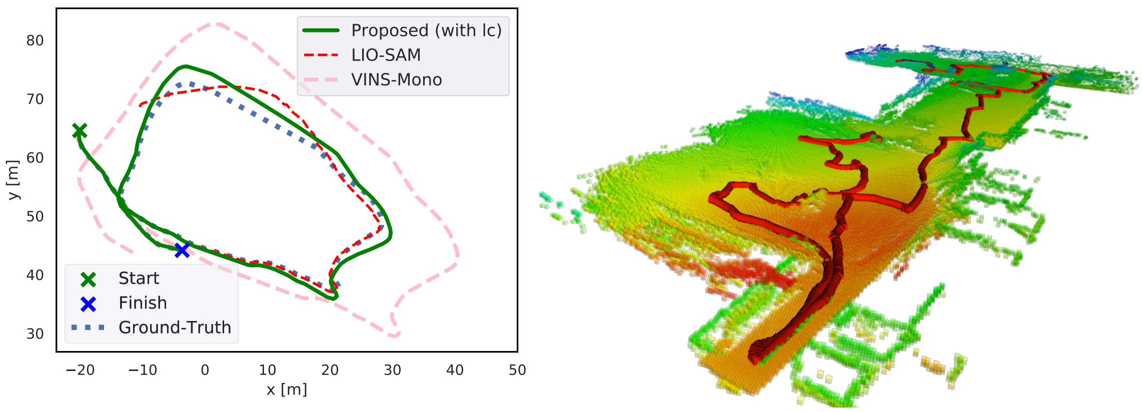
Localization is an active research field, specifically in challenging environments with limited availability of the global satellite signal. Within the thesis, we face a failure-resilient localization problem using an additional odometry source. We consider localization a critical component of autonomous exploration; therefore, we aim the evaluate the localization performance within the exploration. We present the developed localization system in the exploration context and address the following research questions in the thesis.

1. Is the proposed localization method resilient against LiDAR-based localization failure?
2. What is the performance of the proposed localization method compared to state-of-the-art methods?
3. Is the proposed exploration method sufficient for exploring the outdoor environment?
4. Can the proposed localization system support the developed exploration framework in the case the employed LiDAR odometry fails?

1.3 Methodology and Approach

We propose building the developed localization system on the existing work and an extension of the Pose-Graph SLAM [11], combining tightly and loosely coupled ideas. We propose to use tightly coupled sensory fusion between LiDAR and IMU, similar to LIO-SAM [10]. Besides, the developed solution allows utilizing additional sources of pose estimates in a loosely-coupled manner, improving the SLAM performance when LiDAR-based localization fails. Various methods of incremental localization can be used in loose-coupling, and we opt for the black-box embedded stereo visual localization system, the *Intel RealSense T265* (T265) [12], and visual-inertial localization VINS-Mono [13]. We propose a relatively straightforward failure detection model that triggers incorporating an additional low-quality pose estimate into the developed Pose-Graph SLAM. The model assesses LiDAR scan matching quality to indicate possible matching failure and IMU-based pose change prediction to confirm the failure for switching the pose estimate source. Incorporating the additional localization source is enhanced by an auto-scaling mechanism and improved graph structure.

The triggering threshold for failure detection has been experimentally established using a real robotic system, and the proposed Graph-based SLAM has been deployed in several deployments and compared with the selected state-of-the-art LiDAR-based



(a) Resulted localization in the parking field (b) Resulted map of the explored campus backyard

Figure 1.1. Results of the developed methods. Notice that even though the matching of the LiDAR scans was unsuccessful in some areas, the proposed method can use scale and pose drifting visual localization VINS-Mono to overcome such areas and close the loop.

SLAM. Based on the experimental results, the proposed method demonstrates improvement of the localization performance by the additional source of the incremental localization while not sacrificing LiDAR-based performance in scenarios where LiDAR scan matching performs well, see Fig. 1.1a.

For the environment representation of the exploration part, we developed an instance of the elevation mapping method, from which the traversability map is computed to detect untraversable areas (obstacles) that are avoided by path planning. We employed frontier-based exploration strategy [3] to set the exploration goals, an external path planner to plan towards these goals, and a follow-the-carrot path follower to follow the planned path. The developed exploration framework can navigate the robot in the unknown environment and construct a representation of the environment supported by the developed localization system as shown in Fig. 1.1b. LiDAR-based localization itself, with no failure detection and additional odometry incorporation, was not able to localize the robot in such an environment. Thus, the presented thesis successfully faced the challenged research questions.

We consider the main contributions of the proposed approach as follows.

- Modular enhancement of existing Pose-Graph SLAM by a loosely coupled unreliable additional localization system.
- Two-step failure detection model, allowing detection of LiDAR-based localization failure.
- Outdoor exploration and elevation mapping framework.

1.4 Thesis's Structure Overview

The rest of the thesis is organized as follows. Chapter 2 overviews the related literature, including a brief description of the selected reference LIO-SAM framework. The proposed method, including exploration and localization, is described in Chapter 3. Experimental results are reported in Chapter 4 and discussed in Chapter 5. Finally, the thesis is concluded in Chapter 6.

Chapter 2

Literature Review

2.1 Overview of Relevant Literature

Autonomous exploration is one of the fundamental problems in robotics. We consider the exploration task as a problem to autonomously build a map of (possibly a priory unknown) environment. Existing map representation approaches include occupancy gridmap [14] (2D) and octomap [15] (3D). While a pure 2D map (grid map) is designed for flat terrains, an octomap is a more complex 3D representation requiring complex and computationally demanding algorithms. Therefore, a 2.5D elevation map [2] is a suitable trade-off for a wide range of outdoor autonomous exploration tasks. It combines 3D information (elevation) sufficient for terrain mapping and computational efficiency of projection to a 2D-matrix representation. Implementation [16] proposes an efficient robot-centric elevation map. Although it is designed to map only the local area of the robot's surroundings, it inspired our solution.

The exploration strategy to explore unknown parts of the environment can be based on the frontier-based exploration [3] that utilizes a border between unknown and unknown parts of the environment (called frontier) to selects the next navigation waypoint at the frontier and thus steer the robot toward exploring not yet explored parts of the environments. Planning a path to reach the desired waypoint is desirable to assess multiple possible waypoints. Several methods, such as grid-based planning or RRT-based approaches [17], are reported to be fast and suboptimal. Next, the path following can be based on multiple approaches described in [18], where the carrot-chasing algorithm is mentioned as one of the simplest yet effective approaches.

Calculating traversability from the elevation map [19] shifts exploration and path planning to 2D space, simplifying further data processing. However, all the above methods are highly dependent on the localization quality and would fail in the case of localization failure. Therefore, SLAM [4] technique can be used to address high-quality and reliable localization.

Diverse SLAM systems have been proposed [5, 6] and evaluated using the available Kitti benchmark dataset [20]. Based on the reported results, most top-performing methods use LiDAR measurements for robot pose estimation. One of the top-performing LiDAR-based methods is LOAM [21], albeit it lacks an explicit loop closure and is limited to only one type of sensor. On the other hand, multiple possible sensors are used in the RTAB-Map [22], which is a general tightly-coupled LiDAR-Visual SLAM framework using multiple graph frameworks. However, failure handling is not resolved in the framework yet, and the authors indicated it as a future research direction.

Another approach is shown in [23], where the authors loosely coupled several localization sources to explicitly handle failure of one of the sources. The first step of the coupling is the sanity check, where localization failures are identified for each localization source using the dynamic model of the vehicle. Then, Chamfer distance-based [24] score is used to select the best pose estimate. The advantage of [23] is its high robustness, but since the localization sources are completely independent, the visual odometry cannot help the LiDAR-based SLAM to close the loop in the case of temporal LiDAR-based SLAM failure. Furthermore, the Chamfer distance-based score measures the alignment of the LiDAR scans. Thus, it does not directly detect when the perfect alignment of LiDAR scans may correspond to a wrong displacement in monotonous corridors or fields.

In [25], the authors review available sensory fusion approaches for LiDAR-Visual SLAM. They mention that the graph-based SLAM [4] is often used for sensor fusion because it abstracts from raw measurements. The approach represents measurements, poses, and observations in a graph structure. Pose-graph SLAM [26] is a specific kind of graph-based SLAM that is widely used. It restricts the graph's nodes to be poses and positions of the robots and landmarks and edges to be measurements-based constraints between them. The authors of [27] demonstrate the computational advantages of the pose-Graph SLAM for large-scale maps, comparing the solution with conventional filter-based approaches. The approach is further explored in [28],

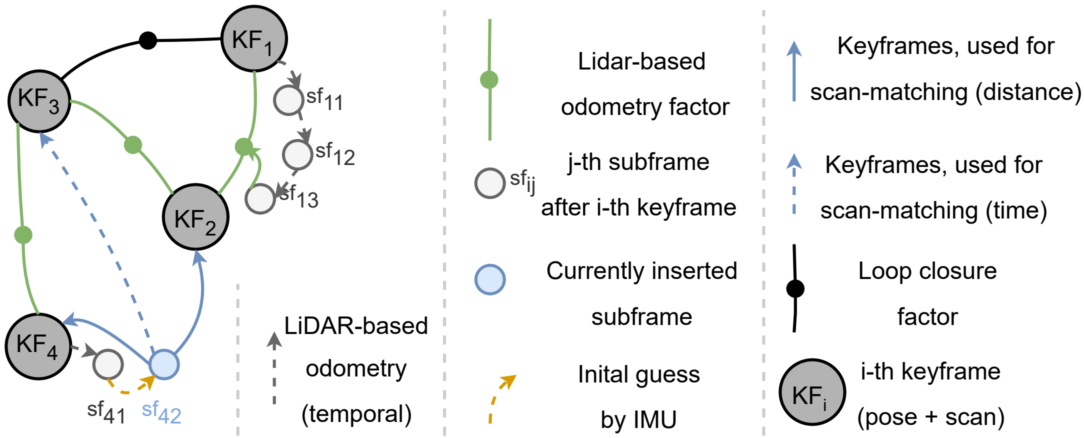


Figure 2.1. Map optimization graph in LIO-SAM [10].

where the authors review iSAM2 [29], which iteratively re-optimizes only nodes influenced by new observations. Multiple graph optimization frameworks have been proposed. ORB-SLAM3 [30] uses the g2o library [31] in Loop Closure for Bundle Adjustment [32] to improve the Visual-Inertial Odometry. In VINS-Mono [13], the authors present a Visual-Inertial SLAM solution that fuses a monocular camera and IMU in a tightly-coupled manner for obtaining odometry and optimizing the global trajectory with pose-graph SLAM using Ceres [33] optimization framework.

LiDAR-Inertial odometry is the core of LIO-SAM [10] that uses scan matching based on LOAM [21], where the initial guess of the LiDAR pose is based on integrated IMU measurements. The scans aligned by LiDAR odometry are marked as keyframes if the distance from a pose corresponding to the previous keyframe is above a certain threshold. Otherwise, the pose is treated as a temporal sub-frame. The relations between the keyframes are represented by constraints that are used to construct a sparse graph within the GTSAM [34] optimization framework. Loop closure is performed as a parallel process using the *Iterative Closest Point* (ICP) [35], and the loop constraints are added if the ICP converges. For the loop closure detection, the latest keyframe is attempted to be matched against the nearby keyframes, including recent keyframes and keyframes that are close to the current robot pose. If the matching of the keyframes is successful, the transformation between them is inserted into the graph as a constraining factor. The graph structure is illustrated in Fig. 2.1.

LIO-SAM is further extended by tightly-coupled *Visual Odometry* (VO) in *LiDAR-Visual-Inertial Odometry via Smoothing and Mapping* (LVI-SAM) [36]. LVI-

SAM tightly couples LIO-SAM with Visual SLAM VINS-mono [13] to improve performance in challenging scenarios using sensor-specific failure detectors for LiDAR and VO. However, such an approach does not support flexibility in changing the source of additional localization systems and restricts end-users to specific additional sensors (camera) and algorithms (VINS-mono).

Based on the literature review, we opt for LIO-SAM as a suitable base system for integrating the additional sensor for localization. It provides the advantage of a great performance of LiDAR-based methods [20] while avoiding the disadvantage of the tightly-coupled visual odometry of LVI-SAM, which supports only the specific method of visual odometry. LIO-SAM framework accounts for ambiguities of the scan-matching by checking scan-matching convergence. The convergency is then reflected in uncertainties while optimizing IMU measurements. On the other hand, the system is developed for structure-rich environments. Besides, it does not explicitly handle situations where the scan-matching results are completely unusable. Both the drawbacks are addressed by the proposed loosely-coupled combination of LiDAR-Inertial SLAM and VO.

Here, it is worth mentioning that in the available literature, localization systems are usually evaluated on pre-recorded datasets only, while we also aim for more challenging show-case usage of the developed method in the online exploration task and custom datasets.

2.2 Theoretical framework

In this section, we briefly introduce concepts of *exponential smoothing filter* and *outlier detection methodology* to make the thesis self-contained, as these techniques are used in the proposed solution. The exponential smoothing filter is used to estimate the heights of the environment in the mapping module. The outlier detection methodology is used for the failure detection part of the proposed localization system.

2.2.1 Exponential Smoothing Filter

Exponential smoothing is a way of processing sequential numerical data, described in [37]. Although it is a commonly known theory and a widely used technique, we recap it here for completeness. In the presented work, we use it to process the sequential observation of the heights of the map to maintain the most recent observation without being hardly influenced by dynamic obstacles such as walking people or noise. The technique works as follows.

Having a sequence of observations $\{S_i\} = \{S_0, S_1, \ldots, S_t\}$, we aim to estimate the most recent height of the map point $\bar{S}_t$. A general approach is to estimate it with the weighted average between the *most recent estimation* $\bar{S}_{t-1}$ and the *new observation* S_t as in (2.1).

$$\bar{S}_t = (1 - A) \cdot \bar{S}_{t-1} + A \cdot S_t \tag{2.1}$$

where the parameter $A \in (0, 1)$ regulates how much the recent observations weights w.r.t. the old ones. With increasing value of A, the more recent observations are reflected in the value of $\bar{S}_t$.

This simple yet effective approach allows estimating possibly dynamic value represented by the sequence of observations, accounting for all past observations but reflecting the most recent changes.

2.2.2 Outlier Detection

Outlier detection [38] faces the problem of detecting atypical observation values $\hat{\xi}$ of a given distribution ξ. Essentially, it is the way to understand if the currently observed value $\hat{\xi}$ is unlikely to happen based on the knowledge about the modeling distribution ξ. We employ the outlier detection methodology to detect the failure of the LiDAR-Inertial odometry as follows.

Let ξ be a random distribution for which the distribution law is unknown. The *quantile* concept of the distribution is employed to detect the outliers. For a random variable, α-quantile q_α is defined as a value, for which the probability that an observation is above q_α equals to α as in (2.2).

$$P\{\xi > q_\alpha\} = \alpha \tag{2.2}$$

The value of the quantile has to be estimated as it can be used for further outlier detection. As the nature of the distribution is unknown, we directly estimate the quantile instead of fitting a distribution and using the fitted distribution's quantile. By doing that, the estimation is resilient to outliers in the modeling set of observations. Given a set of N distribution observations $\{\hat{\xi}_i, i = 1, \ldots, N\}$, the estimation of the quantile $\bar{q_\alpha}$ can be estimated by ensuring the condition (2.3).

$$|\{\hat{\xi}_i : \hat{\xi}_i > \bar{q_\alpha}\}| = [\alpha \cdot N] \tag{2.3}$$

where $|\cdot|$ operation defines the number of elements of the set, and $[\cdot]$ is the integer part of the number. In practice, $\bar{q_\alpha}$ may be calculated by taking the $[(1-\alpha)\cdot N]$-th item of the ordered list of observations $\{\hat{\xi}_{i_k} : \hat{\xi}_{i_{k-1}} < \hat{\xi}_{i_k}\}$.

The described method can be performed if almost all the observed values are coming from the distribution ξ, and the distribution fully represents the model. Having the value $\bar{q_\alpha}$, the observation $\hat{\xi}$ is detected as an outlier if the condition $\hat{\xi} > \bar{q_\alpha}$ is true. The smaller α is, the less amount of false-positive detections are conducted. With increasing α, more false-negative detections are conducted.

In our case, the distribution ξ corresponds to the non-failure LiDAR-Inertial odometry behavior; thus $\bar{q_\alpha}$ value is estimated using the data, where the LiDAR-Inertial odometry works with no failure. The process of inferring $\bar{q_\alpha}$ is referred to as *system modeling*. Afterward, the outlier detection condition is applied to a challenging dataset, where the observations may not belong to the distribution ξ, detecting the failures as outliers.

Chapter 3

Methodology

3.1 System Overview

The proposed method consists of two parts: localization and exploration. The localization and exploration parts are two separate modules. Since the localization module itself does not depend on the exploration module, the developed localization system can be tested separately. Besides, the proposed modular structure allows for the potential substitution of either exploration or localization modules using the defined interface. An overview of the proposed method's structure is presented in Fig. 3.1, and the localization and exploration parts briefly work as follows.

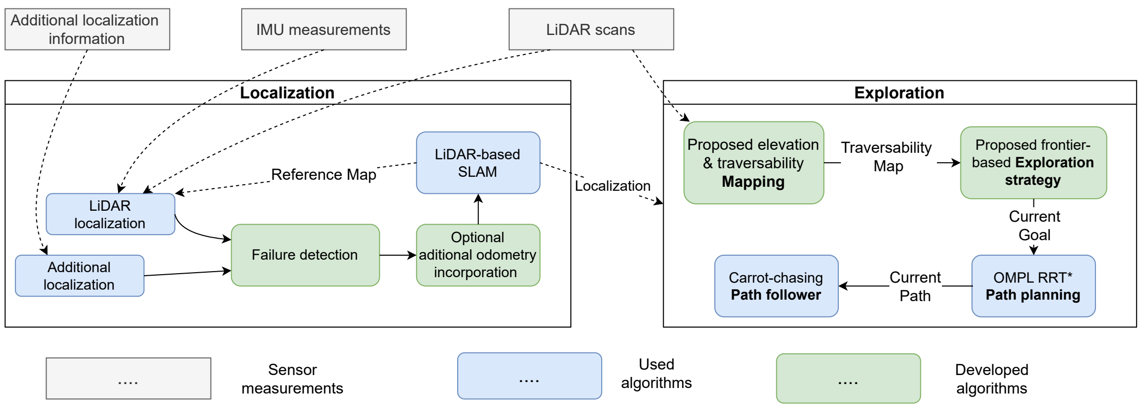


Figure 3.1. Overview of the proposed method.

The proposed localization system starts by estimating *LiDAR-based localization* (odometry) from received IMU and LiDAR measurements, using previously created *reference map* if available. If the developed *failure detection* system detects a failure, corresponding *additional odometry* is integrated into the underlying graph-SLAM structure. Hence, the resulting *LiDAR-based SLAM* system outputs the most recent *localization information* based on globally optimized trajectory.

For the exploration part, the *elevation mapping and traversability estimation* module creates a map of the environment from the LiDAR scans having a robot pose. Then the employed *exploration strategy* determines the next exploration waypoint, and afterward the path is planned to the waypoint using the RRT* [17] path-planner from the *The Open Motion Planning Library* (OMPL) [39]. Finally, the *carrot-chasing path follower* controls the robot to follow the planned path. Once the waypoint is reached by following the path or any other trigger (such as the traveled distance or period), the process is repeated until all reachable parts of the environment are explored or a supervising operator terminates the autonomous exploration. Besides, the waypoint is reselected if the robot follows the waypoint for more than t_{limit} seconds, making the process more resilient against unexpected failure situations.

In the rest of the chapter, we explain assumptions for the developed methods. After that, we describe the modules of the proposed localization system and exploration framework in detail, following the order of how data are processed from sensors: localization and failure detection, additional localization source integration, mapping, exploration strategy, and path planning and following.

3.2 Assumptions and Limitations

Localization part – Although the proposed localization approach is general, to present the proposed concept, we consider Visual-Inertial odometry (VIO) as the additional localization system that produces a 6 DoF (degrees of freedom) robot pose estimate. The following assumptions are made in the design of the proposed method.

- For simplicity of the description, only a single additional localization system VIO is used, albeit multiple additional localization sources can be straightforwardly utilized.
- The additional localization system provides pose estimates w.r.t. to the same coordinate frame as the LiDAR-based odometry.
- All sensors' data is synchronized in time.

Exploration part – Since the main focus of the thesis is on improving localization, evaluated within the exploration, we focused our solution on the following scope.

- The ceiling height is above 3 meters, albeit the limit is not sufficient nor necessary; it would work in most exploration situations, specifically in the planned validating experimental deployments.
- The explored area is more or less static. The presence of highly-dynamic objects is handled, but the system does not explicitly handle semi-static objects.
- The explored area does not contain fog, glass, or other factors that may corrupt LiDAR's sensor measurements.

3.3 Localization

3.3.1 Localization Failure Detection

The localization system exploits *LiDAR-Inertial odometry* part of LIO-SAM [10], partly described in Section 2.1, and an *additional localization system*, VIO in our case. The first step to make the developed localization a failure-resilient system is to detect when the LiDAR-Inertial odometry fails.

The failure detection of LiDAR-Inertial odometry starts with the failure indication defined by the *Failure indicator* I_{fail}. If the indication is positive, *Failure resolution* determines if VIO provides a more suitable pose estimate than the LiDAR-based odometry. The overview of the failure detection process is depicted in Fig. 3.2, and it works as follows.

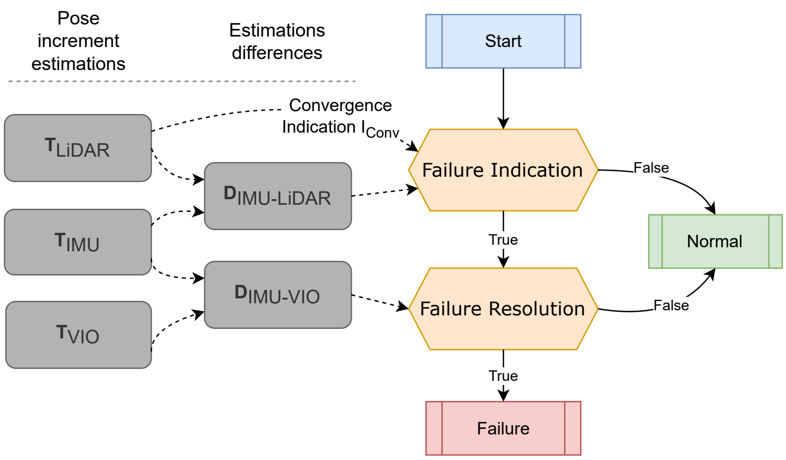


Figure 3.2. Failure detection algorithm.

The failure indicator I_{Fail} is combined from two components: *convergence indicator* I_{Conv} and *IMU-based indicator* I_{IMU} as

$$I_{\text{Fail}} = I_{\text{IMU}} \texttt{ or } I_{\text{Conv}}. \tag{3.1}$$

I_{Conv} is triggered when the LiDAR scan matching does not converge, but it might not cover all the cases when it is suitable to switch to VIO. Therefore, we also use I_{IMU} to increase the failure detection rate, which is supported by the experimental results reported in Section 4.2.

The advantage of I_{IMU} is that it is not directly influenced by a lack of spatial and visual features in the environment. The indicator uses a rough estimation of the robot motion by IMU-based odometry increment $T_{\text{IMU}} \in SE(3)$ to estimate the adequacy of the LiDAR-based odometry increment $T_{\text{LiDAR}} \in SE(3)$. As the rate of the LiDAR measurements is mostly bigger than 1 Hz, the translational drift accumulated by the IMU is small enough to not affect failure indication. The difference of the increments $D_{\text{IMU-LiDAR}}$ is computed as

$$D_{\text{IMU-LiDAR}} = T_{\text{IMU}} \cdot T_{\text{LiDAR}}^{-1}. \tag{3.2}$$

We analyze the norm of the rotational component and a translational component of the difference defined by

$$\begin{aligned} r_{\text{IMU-LiDAR}} &= ||\,\text{rot}(D_{\text{IMU-LiDAR}})||^{\text{ANG}} \\ t_{\text{IMU-LiDAR}} &= ||\,\text{trans}(D_{\text{IMU-LiDAR}})|| \end{aligned} \tag{3.3}$$

where $\text{rot}(D_{\text{IMU-LiDAR}}) \in SO(3)$ is the rotational component and $\text{trans}(D_{\text{IMU-LiDAR}}) \in R^3$ is the translational component of $D_{\text{IMU-LiDAR}}$. The term $||\cdot||^{\text{ANG}}$ denotes the angular metric of the rotation that is determined as a rotation angle of the angle-axis representation of the rotation.

The IMU-based indicator I_{IMU} works as an outlier detector [38] (described in Section 2.2.2), and it is defined as the logical `or` of detecting translational or rotational differences as outliers with the defined quantile values (thresholds), c_r and c_t correspondingly.

$$I_{\text{IMU}} = (r_{\text{IMU-LiDAR}} > c_r) \texttt{ or } (t_{\text{IMU-LiDAR}} > c_t) \tag{3.4}$$

The thresholds are estimated experimentally using outlier detection methodology.

The failure resolution begins if the failure indicator I_{Fail} (3.1) is triggered. The VIO pose estimate is used if it is significantly closer to the IMU-based odometry than the LiDAR-based odometry. Thus, the resolution is defined by the condition

$$\begin{gathered}(r_{\text{IMU-VIO}} < \beta \cdot r_{\text{IMU-LiDAR}}) \\ \texttt{and} \\ (t_{\text{IMU-VIO}} < \beta \cdot t_{\text{IMU-LiDAR}})\end{gathered} \tag{3.5}$$

where $r_{\text{IMU-VIO}}$ and $t_{\text{IMU-VIO}}$ are defined similarly to the IMU-LiDAR difference $D_{\text{IMU-LiDAR}}$ defined in (3.3).

The LiDAR-based odometry failure might be indicated based on I_{Fail}. However, the failure resolution (3.5) would not activate the usage of the VIO pose estimate if the latter does not improve the LiDAR-based one. We incorporate a constant factor of $\beta = 0.8$ in the IMU-LiDAR difference when considering additional odometry over LiDAR to ensure the significance of any potential improvement by additional odometry and to account for the IMU noise.

3.3.2 Visual Odometry Integration and Scale Self-adjustment

Let us suppose the LiDAR-based odometry failure is indicated, and VIO provides more precise localization according to the rule (3.5). In that case, VIO is incorporated into the factor graph in place of the LiDAR-based odometry, introducing a constraint between the keyframes if the keyframe is inserted. Since the additional odometry (such as visual or wheeled) might suffer from a wrong scale or slow scale drift, the proposed method performs dynamic scale self-adjustment, estimating the scale of the odometry when the LiDAR-based localization is considered sufficiently precise.

We propose to utilize the median value of the moving window to compute the scale. In particular, 500 keyframes-long window includes the past ratios of the absolute values of the translations $c_{adjustment} = t_{\text{VIO}}/t_{\text{LiDAR}}$, where $t_{\langle\text{source}\rangle}$ is the norm of the translational part of the odometry increment from the corresponding source. An additional odometry is scaled by the estimated median value of $c_{adjustment}$ before the incorporation. Besides, to synchronize additional pose increment estimation with the LiDAR-based one, linear interpolation of the additional pose estimations is performed. Thus, the rate of the additional localization should be at least bigger than the rate of the LiDAR-based localization. Then, the factor graph structure is created according to the scheme depicted in Fig. 3.3 as follows.

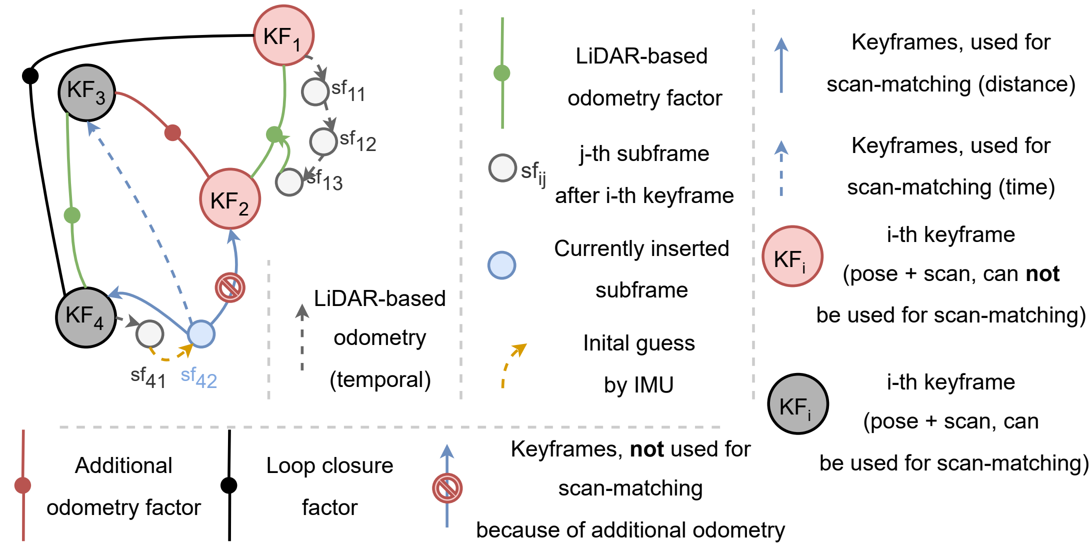


Figure 3.3. The proposed method for combining the LiDAR-based odometry with VIO-based pose estimate increments.

- The LiDAR-based odometry creates constraints between the previous and the new keyframes based on scan-matching when the LiDAR-based odometry works successfully. When the new LiDAR scan (frame) is available, it is scan-matched against a reference map combined from the nearby keyframes to create such a constraint. Similarly to LIO-SAM, only if the estimated pose increment exceeds a configurable threshold the frame is inserted into the map as a keyframe. Otherwise, it is treated as a temporal sub-frame to improve the initial guess of the next frame pose and output localization information.

- On the other hand, the VIO constraint is inserted instead of the LiDAR-based one if the failure is detected. However, in contrast to LiDAR-based constraints, the VIO-based ones are not guaranteed to be optimized for the keyframes alignment as they optimize visual features alignment and may suffer from the incorrect and drifting scale. Thus, combining keyframes connected with the VIO-based constraints can result in a poorly aligned reference map and a new LiDAR scan would not be successfully matched against such a reference map. Therefore, only keyframes inserted after the last VIO usage are combined in the reference map when the new LiDAR scan is processed.

Finally, it is necessary to properly handle loop closure constraints of the graph-based SLAM that aim to match keyframes that are far from each other. These constraints might fix the drift introduced by the VIO constraints. However, false loop closures may appear for the structure-less keyframes, consequently breaking the graph. Therefore, keyframes corresponding to the LiDAR-based odometry failure are deemed unsuitable for loop closures. Further, the inserted VIO constraints are set to have 10 times larger uncertainty than the LiDAR-based ones to ensure that loop closure constraints will fix mostly VIO constraints without affecting LiDAR-based constraints significantly. The effect of the proposed loop closing system has been experimentally examined, and results are reported in Chapter 4; in particular, the effect is demonstrated in Fig. 4.5.

3.4 Exploration

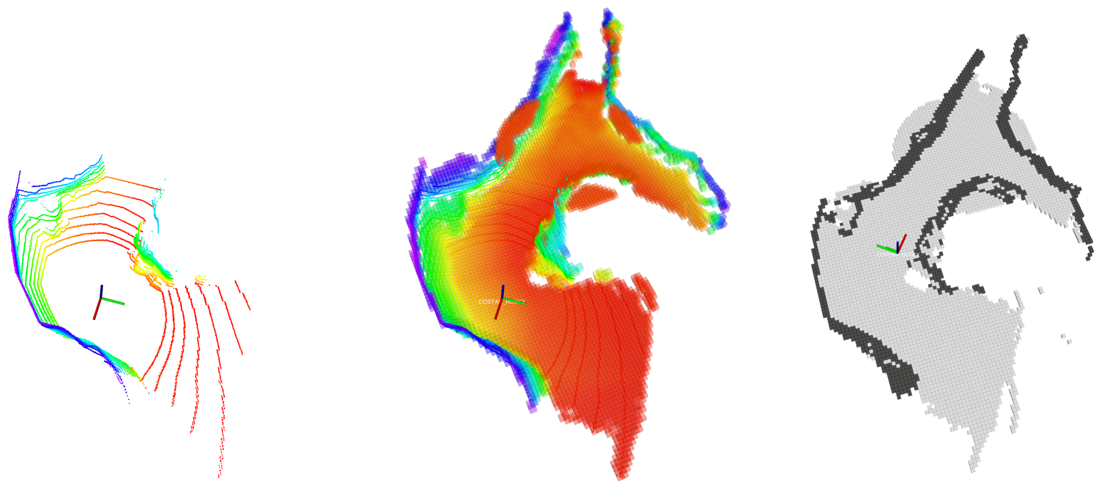
An obtained failure-resilient localization information is further used by the exploration module. The robot maps the environment based on its estimated pose and sensor measurements; based on the estimated map, exploration strategy defines an exploration goal, and path planning and following ensure that the robot reaches the goal.

3.4.1 Mapping and Traversability Estimation

Mapping is an integration of the sensor measurements (LiDAR scans) into a map using the pose estimate. We developed an implementation of *the elevation map* that represents the environment as a 2D grid, where each cell contains information about the elevation of the corresponding area. At each timeframe when the LiDAR scan is available, it is transformed to the map reference frame knowing the robot's pose from the localization system and down-sampled as in Fig. 3.4a. Afterwards, each point $p = \{p_x, p_y, p_z\} \in \mathbb{R}^3$ is assigned to a grid cell $c = \{c_i, c_j\} \in \mathbb{N}^2$ that corresponds to it using (3.6).

$$c = [(p_{xy} - o_{xy})/r] + [N/2], \tag{3.6}$$

where o denotes coordinates of the map's origin, r is the resolution of the map, $\cdot_{xy}$ extracts xy-coordinates of the point, N is the size of the map in cells, and $[\cdot]$ extracts integer part of the number.



(a) LiDAR Scan (b) Elevation map (c) Binary traversability map

Figure 3.4. Example of mapping in a simulated cave environment using [40].

Further, each cell c with correspondences chooses the observed point p with the highest z-coordinate out of the ones corresponding to it. If the elevation of the cell el_c does not have a previous estimation, it is assigned to the current measurement. Otherwise, the elevation of the cell c is calculated using the exponential filter described in Section 2.2.1 with the parameter $A = 0.1$ as in (3.7).

$$el_c[t] = \begin{cases} 0.1 \cdot p_z + 0.9 \cdot el_c[t-1] & \text{if } el_c[t-1] \text{ is defined;} \\ p_z & \text{otherwise,} \end{cases} \tag{3.7}$$

where $el_c[t]$ is the estimated elevation of the cell c at the time t, and p_z is the z-coordinate of the observed point. The map is initialized with 0 elevation on the radius R_{init} around the robot. It is a blind zone of the LiDAR, and initialization happens only at the time $t = 0$. An example of the resulting elevation map can be seen in Fig. 3.4b.

To extract 2D obstacle information from the elevation map, firstly the slope $slope_c$ of every cell c is calculated as in (3.8).

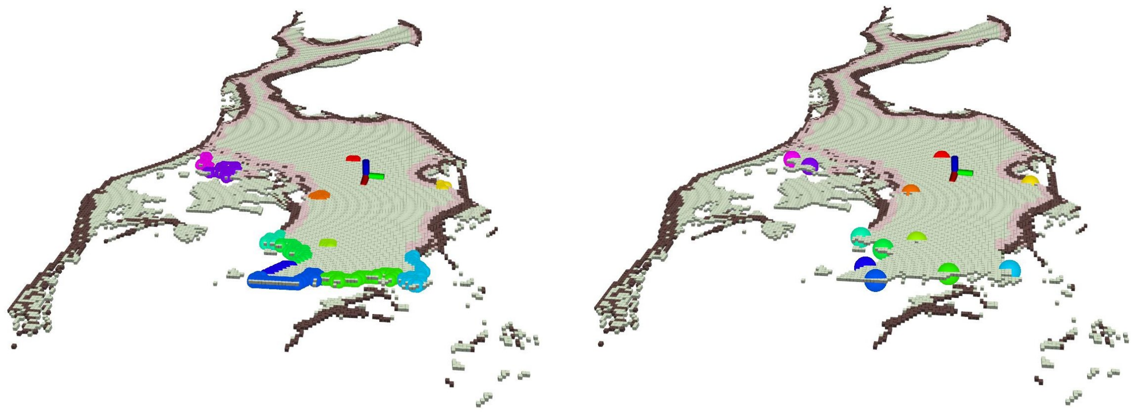
$$slope_c = \max_{c' \in N_4(c)} (|el_c - el_{c'}|)/r, \tag{3.8}$$

where $N_4(c)$ is a 4-connectivity neighborhood of the cell c. A user-defined traversability threshold value th_{sl} defines traversability $trav_c$ of each cell c as in (3.9), forming a binary traversability map shown in Fig. 3.4c, that represents information about obstacles in the environment.

$$trav_c = \begin{cases} 1 & \text{if } slope_c > th_{sl}; \\ 0 & \text{otherwise,} \end{cases} \tag{3.9}$$

3.4.2 Exploration Strategy

With a traversability map based on integrating and processing the localized LiDAR's scans, the *frontier detection* starts to determine the next exploration waypoint. We exploit the *Wave Front Detection* (WFD) frontier detection algorithm [41]. The WFD determines frontiers starting from the robot's position using the *Breadth-first search* (BFS) algorithm. It propagates through traversable cells, searching for the frontier points. However, before running the WFD frontier detection, the obstacles of the traversability map are expanded to ensure that the detected frontiers are reachable by the robot. The side effect of the obstacles' expansion is that it would not be needed to propagate the wave through the narrow corridors, thus saving computational time. Moreover, the robot would not try to reach waypoints close to obstacles, increasing collision avoidance. In Fig. 3.5, it can be noticed that the frontiers are detected only in the region that is reachable by the robot.



(a) Frontier clusters (b) Corresponding waypoint candidates

Figure 3.5. Frontier detection in a simulated cave environment using [40]. Pink cells are expanded obstacles.

Each detected frontier (an edge of the boundary between the explored free space and unknown part of the environment) is limited to a user-defined length l_{max}, in our case set to $l_{max} = 2\,\text{m}$. Frontiers with a length smaller than l_{min} are not being detected. As a result, the frontier is clustered in sub-frontiers as shown in Fig. 3.5a according to the maximum length. Next, on each sub-frontier, the closest frontier point to its geometrical center is chosen as the frontier representative forming a set of candidates for the next waypoint; see Fig. 3.5b. The closest candidate to the robot's position is then chosen as the next waypoint to follow.

Note that after the waypoint is chosen once, a disk-shaped region with the radius 2 m around that waypoint is marked as unexplorable, meaning that candidate waypoints in that region can not be selected in the future. It ensures that the robot is navigated toward each frontier only once, as some frontiers might not change even after the robot approaches them. However, if the waypoint is reselected due to the path following timeout t_{limit}, the region around such a waypoint is not marked as unexplorable.

3.4.3 Path Planning and Following

After the exploration strategy selects the next waypoint, the employed RRT* path planner plans the path toward it. We use out-of-the-box implementation from the OMPL [17] because it optimizes a combined objective from two objectives: *path length* and *path clarity*. The path length is a standard path optimization objective, while the path clarity optimizes the distance to the nearest obstacle along the path, thus enhancing collision avoidance. These distances are calculated for every cell and stored in the costmap, that is used by the planner, as shown in Fig. 3.6. However, before the planning and costmap calculation, the obstacles are expanded similarly to the previous step, but unexplored space is considered an obstacle for path planning. If the robot is at the untraversable area, the nearest traversable cell is determined to be the path start. This situation might happen due to path following imperfections, dynamic obstacles, localization imperfections, or other unexpected factors. The same procedure occurs if the waypoint is at the untraversable area. During the exploration task, the selected waypoint is always considered being at the untraversable area because, as described above, when the obstacles are expanded, unexplored space is considered an obstacle for path planning.

Once the path is determined, the robot follows the path with a *carrot-chasing* algorithm [18]. The idea of the carrot-chasing is that the robot employs a simple kinematic control to reach the l_{carrot} meters away point on the path. The point that is followed is reselected if the robot approaches at a closer distance than $l_{reselection}$ meters. An example of the path following can be seen in Fig. 3.6.

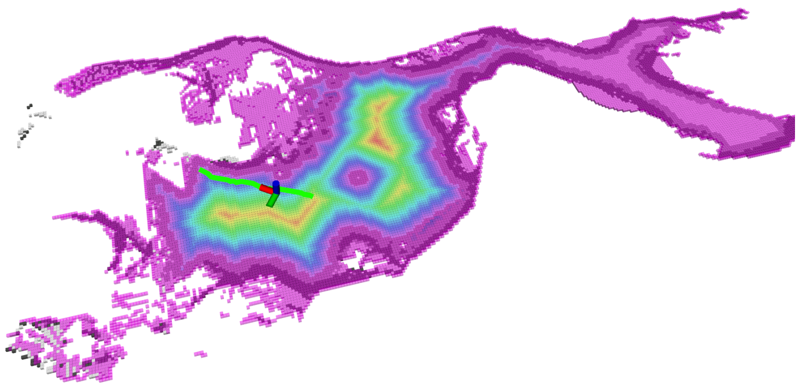


Figure 3.6. An example of the costmap, planned path, and path following in progress. Note that the costmap is calculated only in the region of the map that is reachable by the robot, ignoring the right part of the map.

When the planned path is executed, the process of frontiers detection, subsequent waypoint detection, and path planning with the path following repeats until the termination condition of the exploration is not triggered.

3.5 Tools and Technologies Used

- The proposed loosely coupled VO with the graph-based LiDAR-Inertial SLAM leverages LIO-SAM [10]. It uses the same way of calculating LiDAR-Inertial odometry (referred to as LiDAR-based odometry). However, we modify the factor graph construction to incorporate measurements from an additional localization system, such as VO. For VO, depending on the experiment, the black-box localization system T265 [12] and VINS-Mono [13] Visual SLAM system were used.

- LIO-SAM uses the GTSAM framework [34] to support the underlying graph structure and employs iSAM2 for iterative graph optimization.

- For implementation debugging and statistical modeling of outlier detection the `SciPy` library [42] for Python (version three) has been used. Besides, the ROS framework [43] and C++ programming language have been used to test and deploy the proposed method on the real robotic system.

- The developed exploration framework has been extensively tested in the virtual environment before its deployment on a real robot. The available DARPA SubT simulator [40] has been used.

- The path planning has been implemented using the available planners from the OMPL [39] available at [44].
- Finally, the Eigen library [45] has been widely used throughout the developed solution.

Chapter 4

Results

The proposed method's localization and exploration parts have been experimentally validated using a four-wheeled skid-steered robot Husky, depicted in Fig. 4.1a. The robot is equipped with the Ouster OS0 LiDAR with 128 channels and a maximum range of approximately 50 m, 9-axis IMU Xsens MTi-30, and fisheye stereo tracking camera, the *Intel RealSense T265* (T265). T265 provides out-of-the-box VIO odometry, but its internal loop closures have been disabled to make it compliant with the assumptions made on the additional localization systems. The 3 DoF ground truth localization of the crystal mounted on the robot has been recorded using the Leica TS16 total station, shown in Fig. 4.1b.

(a) Used wheeled robot

(b) Total station setup

Figure 4.1. The robot Husky by Clearpath Robotics used in experimental validation of the proposed solution and the Leica TS16 total station for 3 DoF ground truth localization.

Two experimental deployment sites have been considered for the system evaluation. The first environment is the backyard area of the Czech Technical University campus at Charles Square; see Fig. 4.2a. The environment has been used to record

(a) Urban scenario

(b) Parking lot scenario

(c) Campus exploration setup

Figure 4.2. Experimental testing sites: (a) urban scenario at the Czech Technical University campus at Charles Square; (b) parking lot scenario in Prague's outskirts; and (c) exploration scenario at Charles Square campus.

a dataset for localization evaluation and to test qualitatively the exploration with integrated localization online. The second environment is a parking lot at Prague's outskirt, depicted in Fig. 4.2b, that has been only used to record a dataset for localization evaluation.

While the first environment can be considered structure-rich, the parking lot in the rural area contains wide-open locations where LiDAR scans do not provide sufficient features for successful scan matching. The testing environments are denoted as *campus* and *parking lot* scenarios. The length of the traveled path for the localization system evaluation is 285 m and 300 m for the campus and parking lot scenarios, respectively. The proposed localization method is examined with different failure indicators to justify the combined indicator denoted IMU + Convergence. Besides, its performance is compared with the LIO-SAM [10] as the former localization method to show the benefits of the proposed loosely-coupled VIO.

The evaluation of the localization method is based on the methodology [46] using medians of the *relative pose error* RPE_t and *absolute trajectory error* ATE_t indicators considering the translational parts of the localization error. In particular, the median RPE_t estimates the local consistency of the localization (drift), while ATE_t evaluates the global accuracy of the trajectory.

Calculating RPE_t and ATE_t is based on the estimated poses P_i that are first matched with the corresponding ground truth poses Q_i using timestamps, resulting in a set of corresponding pairs $\{\hat{P}_j, \hat{Q}_j\}$.

For calculating RPE_t, the pairs are downsampled to the set $\{\hat{P}_{j_k}, \hat{Q}_{j_k}\}$ so that the distance between consecutive estimated poses $\hat{P}_{j_{k-1}}$ and $\hat{P}_{j_k}$ is bigger than the defined parameter Δ. For simplicity, instead of j_k, we use indexes i in further explanation of RPE_t. Afterward, a relative pose error at the time step i is defined by (4.1) and is calculated for every consecutive pair of downsampled poses.

$$E_i = (\hat{Q}_i^{-1}\hat{Q}_{i+1})^{-1}(\hat{P}_i^{-1}\hat{P}_{i+1}) \tag{4.1}$$

In essence, E_i is a difference between the estimated pose increment $\hat{P}_i^{-1}\hat{P}_{i+1}$ and the actual pose increment $\hat{Q}_i^{-1}\hat{Q}_{i+1}$. The median of RPE_t is calculated as the median value of the set of absolute values of relative pose errors: $med(\{||E_i||\})$.

For calculating ATE_t, first, the rigid-body transformation S that aligns the estimated trajectory with the ground truth is calculated as described in [46]. Afterward, for every pair $\{\hat{P}_j, \hat{Q}_j\}$, the absolute trajectory error F_i at the time i is calculated as in (4.2), and the median of such a set of values is calculated as $med(\{||F_i||\})$.

$$F_i = Q_i^{-1}SP_i \tag{4.2}$$

For RPE_t, the step Δ is set to 1 m, which corresponds to the minimum distance between consecutive poses. Besides, the standard deviation STD_t of RPE_t is reported to account for outliers. The indication `Fail` is used in cases when the system received corrupted odometry, which led to a wrong IMU bias estimation. Such situations prevent the localization system from recovering.

The exploration framework has been tested experimentally in the campus environment, showing the sufficiency of the proposed localization method for the exploration task and the successful exploration of the outdoor environment using the developed framework. The reported results are structured as follows. The following section presents an evaluation of the localization failure detection indicator followed by failure detection parameterization. The localization performance in the campus scenario is reported in Section 4.3. The results from the localization method deployment in the parking lot scenario are in Section 4.4. The exploration deployment is presented in Section 4.5.

4.1 Localization Failure Detection Indicator

The failure detection indicator has been examined in the campus scenario, where a human operator has operated the robot, and the total station has provided the ground truth data for evaluation. The localization performance is examined based on the scan-matching failure indicator I_{Conv} only and with both indicators I_{Conv} and I_{IMU}. The scan-matching ambiguity is induced by limiting the LiDAR range to 10 m.

Table 4.1. Localization performance in with and w/o failure detection

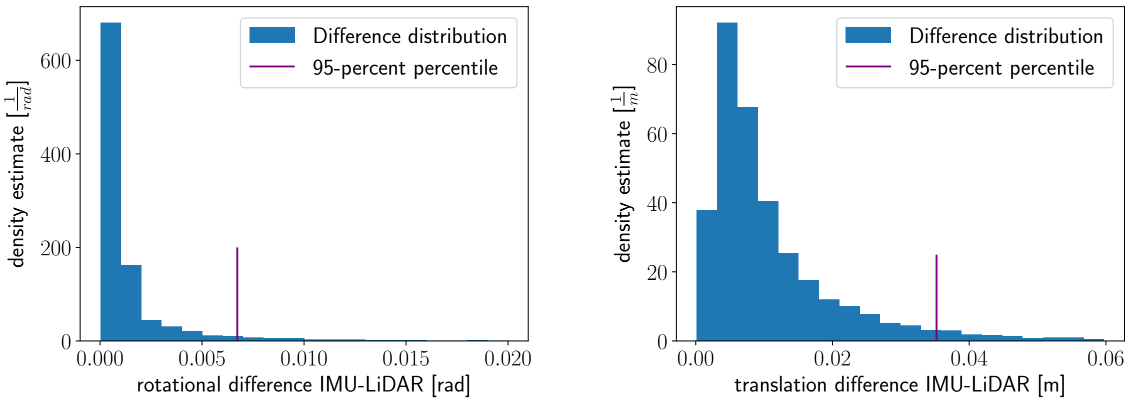
Method / Failure Indicator	ATE_t [m]	RPE_t [m]	STD_t [m]
LIO-SAM [10] (No indicator)	Fail	Fail	Fail
Proposed IMU	Fail	Fail	Fail
Proposed Convergence	5.35	0.08	0.22
Proposed IMU + Convergence	**4.70**	**0.06**	**0.26**

Fail indicates the method has not been able to produce reasonable results.

The results summarized in Table 4.1 indicate that a solo IMU-based indicator cannot detect failure by itself but significantly improves the performance when combined with the convergence-based indicator, reflected in more precise localization results.

4.2 Parameterization of the Failure Detection

The failure detection model's parameters have to be estimated using the experimental data collected in the campus scenario. The proposed IMU-based failure detection model is based on outlier detection [38] (see Section 2.2.2) for differences between the IMU-based and LiDAR-based pose increments $D_{\text{IMU-LiDAR}}$ as of (3.4) with two established threshold values c_r and c_t (α-quantiles).



(a) Rotational part (b) Translational part

Figure 4.3. Histograms of $D_{\text{IMU-LiDAR}}$ differences in the campus scenario, translational and rotational parts. The threshold values c_r and c_t are established as 95 percent quantiles depicted by the vertical line segment.

The thresholds are set based on the estimate of the quantiles for the baseline distributions of differences $r_{\text{IMU-LIDAR}}$ and $t_{\text{IMU-LIDAR}}$ in the non-failure scenario. We set the outliers thresholds to the values of such quantiles as shown in Fig. 4.3. Regarding the presented results, the LiDAR-based odometry provides satisfactory results that can be treated as "Non-Failure" in the campus scenario with full range LiDAR scans. Thus, this data is used to estimate the quantiles to fit the outlier detection model. Note that the data used to fit the model does not intersect with data from the campus dataset used for evaluation, ensuring that the localization system tuning and evaluation are performed using different data.

4.3 Localization Performance in the Campus Scenario

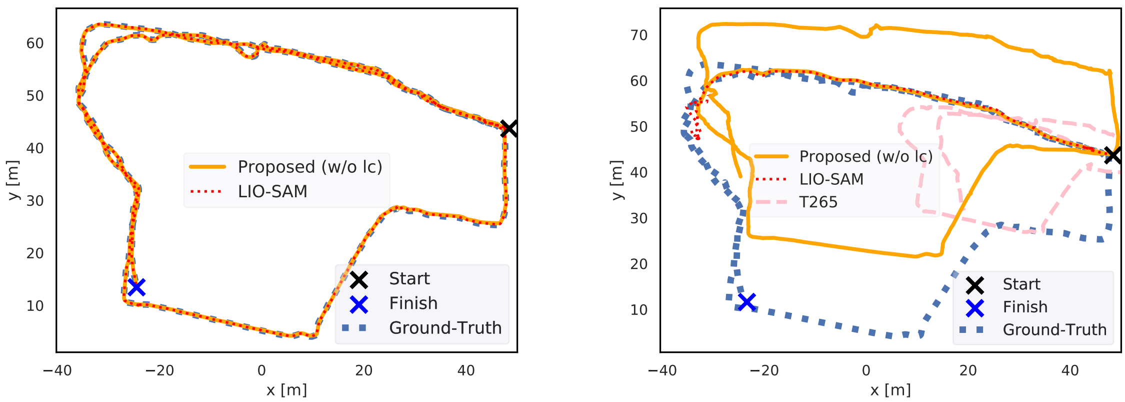
The proposed localization and the reference LIO-SAM performance evaluation in the campus scenario are made for two setups: *full range* and *limited range*. Besides, we consider the localization method without and with the *loop closure* activated (further denoted as lc). The methods are fed with data directly captured by LiDAR without any range restrictions for the full range. However, for the limited range, LiDAR's range is cropped to 10 m to examine the localization system performance under conditions where LiDAR scan matching might be ambiguous. Black-box localization from the T265 sensor was used as an additional source of odometry in this experiment.

Table 4.2. Localization performance in the campus scenario

Method	Full range			Limited range		
	ATE_t [m]	RPE_t [m]	STD_t [m]	ATE_t [m]	RPE_t [m]	STD_t [m]
LIO-SAM	0.08	0.04	0.03	Fail	Fail	Fail
T265	16.40	0.83	0.50	16.40	0.83	0.50
T265 scaled	7.06	0.20	0.30	7.06	0.20	0.30
Proposed method	0.13	0.04	0.03	4.70	**0.06**	**0.26**
Proposed method (lc)	0.13	0.04	0.03	**2.8**	0.08	0.3

Fail indicates the method has not been able to produce a reasonable results.

In addition to LIO-SAM and the proposed localization method, we evaluate the localization provided by the T265 with its (denoted T265) and with the optimal scale (denoted T265 scaled). The optimal scale is the scale that minimizes ATE_t metric for the T265-provided trajectory, it is estimated after the experiment and applied to the entire T265 trajectory. The optimal scale is considered to estimate the best possible reachable result only using T265 with the constant scale. Nevertheless, the proposed method is inputted with the raw T265 localization data, and the scale is estimated dynamically using the method introduced in Section 3.3.2. The performance indicators are depicted in Table 4.2, and the trajectories are shown in Fig. 4.4.

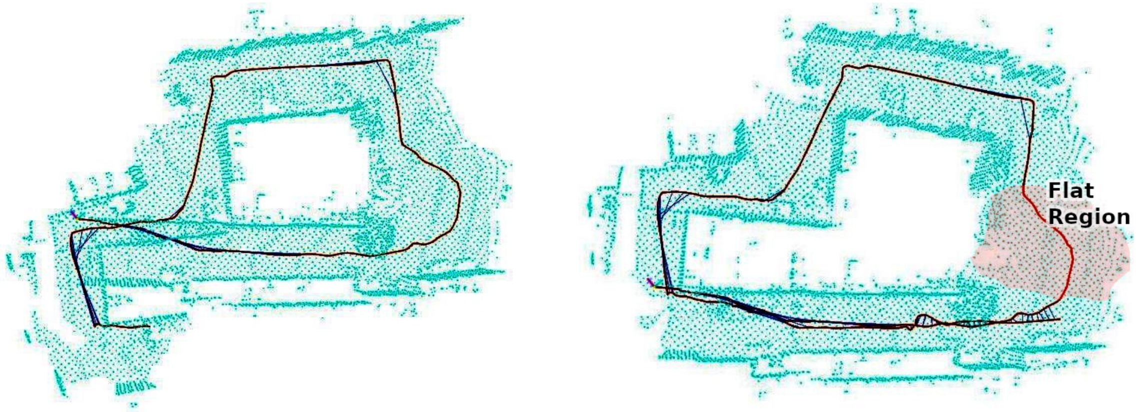


(a) Full LiDAR range

(b) Limited LiDAR range to 10 m

Figure 4.4. Aligned trajectories in the campus dataset.

Besides, we examined the loop closure of the proposed method qualitatively; the obtained maps and trajectories before and after the loop closure are depicted in Fig. 4.5. [1]



(a) Before the loop closure

(b) After the loop closure

Figure 4.5. Impact of the proposed loop closure for data in the campus scenario. Note that the LiDAR scans (the map) remain aligned, as optimization mostly affects VIO-based constraints that were used in the flat region, which is ambiguous for LiDAR-based odometry. Length of the depicted trajectory is around 200 m.

[1] The video of the localization together with the loop closure can be accessed at https://youtube.com/playlist?list=PLpjtKkt1Mu3sbAcw8JcYJ0aOTCRqh-Cen.

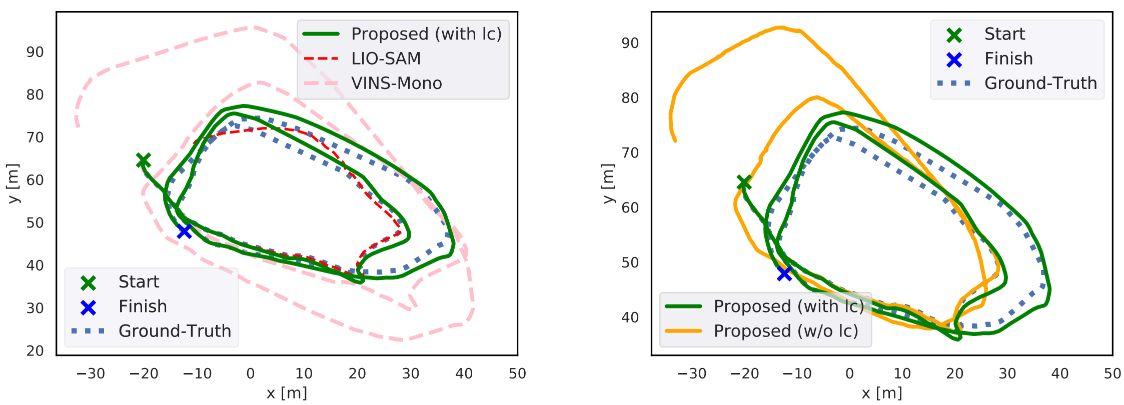
4.4 Localization Performance in the Parking Lot Scenario

In the parking lot scenario with wide open areas, we use fisheye images from the T265 processed by the VINS-mono [13] odometry to show the flexibility of the proposed method to incorporate measurements from various types of additional localization systems. Thus, we examine the performance of LIO-SAM, VINS-Mono, and two variants of the proposed method, without and with loop closure (lc). Similarly to the previous experiment, we also evaluate the scaled variant of the VINS-Mono trajectory. The results are summarized in Table 4.3, and robot trajectories provided by the evaluated methods can be seen in Fig. 4.6a.

Table 4.3. Localization performance in the parking lot scenario

Method	ATE_t [m]	RPE_t [m]	STD_t [m]
LIO-SAM	Fail	Fail	Fail
VINS-Mono	10.9	0.39	0.25
VINS-Mono (scaled)	4.97	0.42	0.18
Proposed method	7.7	0.19	**0.13**
Proposed method (lc)	**2.4**	**0.15**	1.0

Fail indicates the method has not been able to produce a reasonable results.



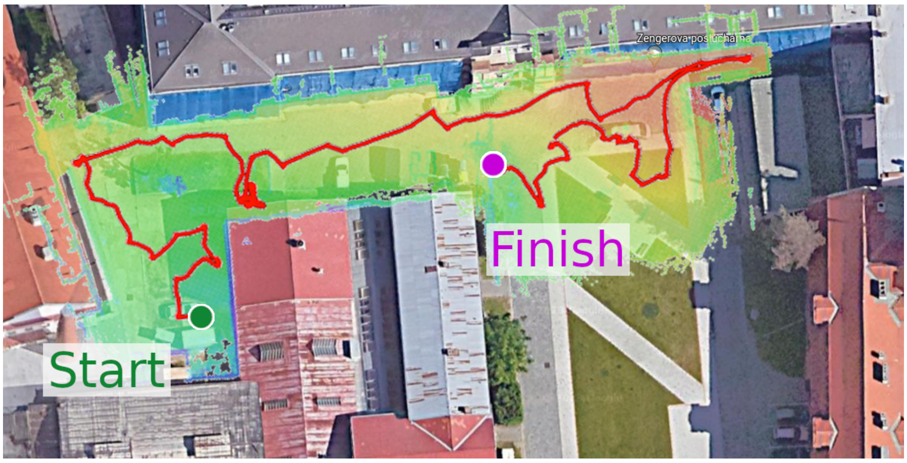
(a) Full trajectories

(b) Loop closure effect

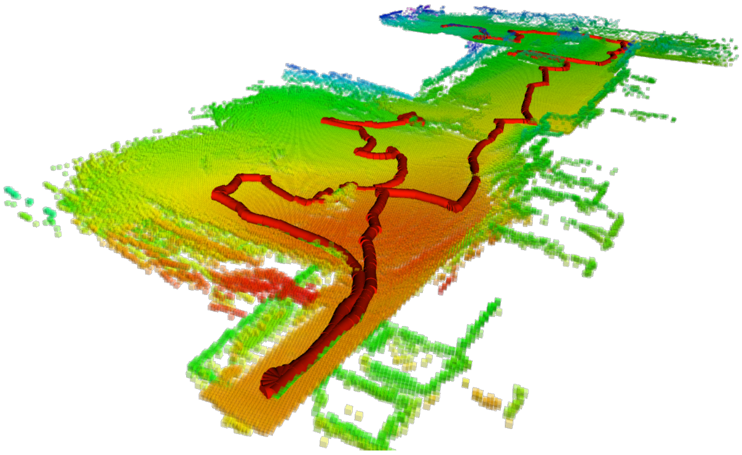
Figure 4.6. Trajectories provided by evaluated methods in parking lot scenario.

4.5 Exploration

The proposed localization system employed by the developed exploration framework was experimentally validated in a part of the campus site. The explored area with the overlaid robot trajectory is shown in Fig. 4.7a, and the obtained elevation map together with the trajectory of the robot is shown in Fig. 4.7b. The robot used the proposed localization method during the experiment, with the LiDAR range restricted to 10 m. Besides, we also compare the resulting map with the one obtained by using only the LIO-SAM method for localization as shown in Fig. 4.8.[2]



(a) Exploration scenario with the overlaid traveled path



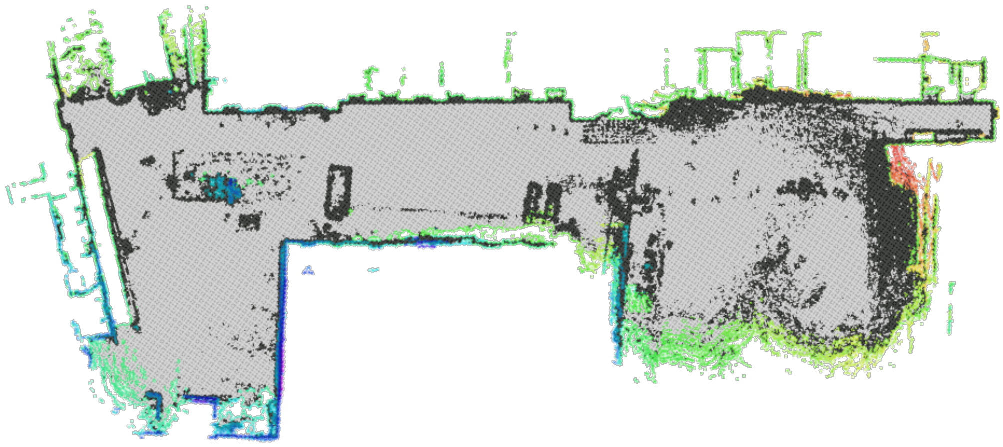
(b) Resulting elevation map with the exploration path

Figure 4.7. Exploration results in the campus site using the proposed localization method with LiDAR range restricted to 10 m.

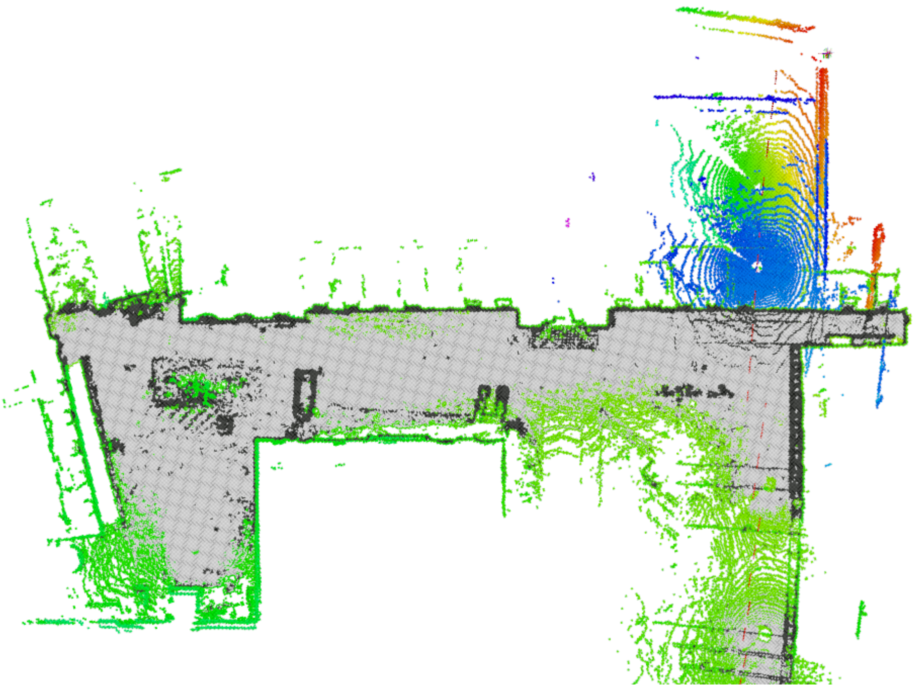
The time of the exploration has been limited to 30 min during which the robot explored about 2000 m^2, traveled 360 m long path with the speed limited to 0.3 $m\,s^{-1}$.

[2]The video with the exploration can be accessed at
https://youtube.com/playlist?list=PLpjtKkt1Mu3sbAcw8JcYJ0aOTCRqh-Cen

For the exploration experiment, only qualitative analysis of the resulting map and the exploration path is made using the final obtained maps depicted in Fig. 4.8. During the experiment, manual introduction in the robot control was done in case the robot was heading too close to the obstacle.



(a) Exploration using the proposed localization method



(b) Exploration using the former LIO-SAM

Figure 4.8. Exploration results in the campus environment using: the (a) proposed localization method; and the (b) former LIO-SAM.

Chapter 5

Discussion

5.1 Localization

For the campus dataset, the presented results support the assumption that the environment is structure-rich for the full LiDAR range and that LIO-SAM provides competitive results in such a case. On the other hand, the T265 suffers from localization drift and provides worse results, but as rarely used, it only slightly worsens the performance of the proposed method compared to LIO-SAM. However, when the LiDAR range is cropped to 10 m, LIO-SAM fails to output any feasible result once the robot enters the area where it is too far from the buildings. The limited LiDAR scans are ambiguous for the scans-matching algorithm, and the whole localization fails. The proposed method handles these ambiguous LiDAR scans by substituting constraints with the VIO-based ones, as shown in Fig. 4.4b. Although it introduces a drift caused by the additional odometry, it outperforms the other localization methods.

The loop closure compensates for the drift introduced by the relatively low-quality VIO, as observed in Fig. 4.5. The resulting map is aligned because the loop closure constraint mostly optimized the trajectory where the LiDAR-based odometry is ambiguous, which is the flat region at the right part of the map. At the same time, LiDAR-based constraints that align keyframes with no ambiguity are altered less than the VO-based constraints because the former ones have much lower uncertainty in the graph structure.

In the rural dataset, we can see in Table 4.2 that the proposed method performed better than LIO-SAM since it did not fail. VINS-mono provided the robot with smooth but scale and pose drifted odometry. It can be seen in Fig. 4.6b that due to the loop closure, the proposed method is able to re-estimate the whole trajectory, mainly altering the part where the additional odometry is used.

Overall, the presented method successfully handles LiDAR-deprived situations, handling LiDAR odometry failure and reoptimizing the map after the loop closure, dealing with the resulting drift.

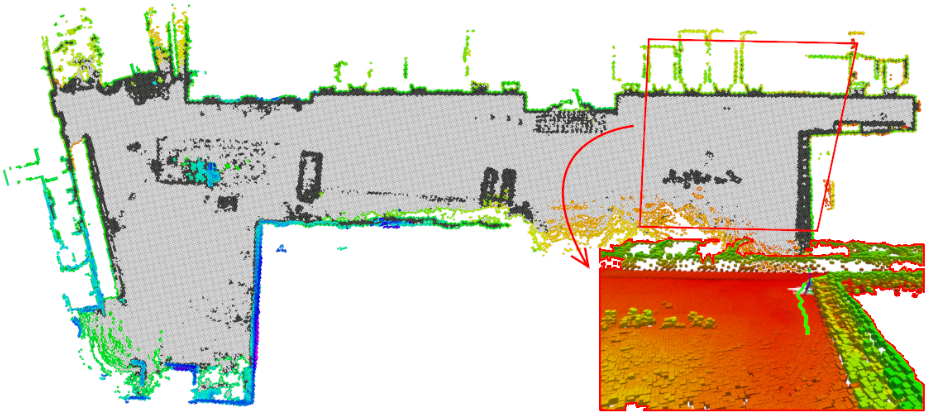
5.2 Exploration

Employing the proposed localization system in the developed exploration framework yields a successful exploration of the unknown environment and creates an elevation map of the environment. The map is visually aligned, meaning that the localization system has provided reliable data for exploration. In contrast, using LIO-SAM for localization during exploration yields inaccurate localization data, not allowing the experiment to continue as can be seen in Fig. 4.8b.

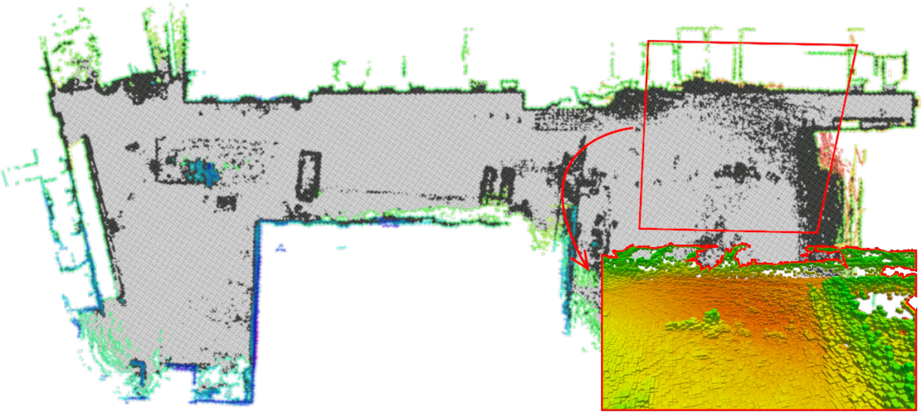
The elevation map that resulted from the proposed method looks adequate to the environment, but the resulting traversability map includes unexpected noisy obstacles in the right part of the map in Fig. 5.1b. These obstacles appeared as a result of observations incorporated after excessive usage of the additional odometry. As seen in Fig. 5.1a, the elevation map of the highlighted area is smooth, as the LiDAR features are close to the robot, and the robot does not use additional odometry extensively yet.

In contrast, when the robot entered the area where the LiDAR features were far from the robot, it started relying on the additional odometry only, the visual odometry in our case. As visual constraints are not optimized for the point cloud alignment, the usage of additional odometry creates a small tilt of the ground plane that creates a traversability clutter for the previously explored area, as can be seen in Fig. 5.1b. Despite that, these obstacles appear after exploring the corresponding area, and the area around the robot is detected as traversable as local scans are more consistent with each other than with the rest of the map.

Despite the resulting noisy obstacles, the usage of unreliable additional odometry made the proposed localization system resilient enough to keep the robot pose estimation adequate during autonomous exploration of the environment. This shows



(a) Before extensive usage of unreliable additional odometry



(b) After extensive usage of unreliable additional odometry

Figure 5.1. Binary traversability maps of the campus environment with corresponding parts of the elevation maps for the proposed localization system before and after exploiting the unreliable additional odometry.

the potential of the proposed method and reveals possible future research directions that would ensure smoother exploration results.

Another problem that might appear is related to dynamic obstacles. Exponential filtering handles the problem if it observes the area after the dynamic obstacle disappears. However, if the absence of an obstacle has not been observed for a significant period of time, remnants of the dynamic obstacles may remain on the map.

5.3 Future Research Directions

There can be multiple ways in which the developed system can be further improved. Here, we presented selected ideas that might be considered for future work.

- A combination of the localization employed in the exploration might be further investigated within the Active SLAM (ASLAM) context. The exploration strategy aims to keep the LiDAR-based localization reliable by choosing an appropriate path.

- Another possible future work might combine both types of features for joint optimization in a common space to handle failures effectively. For instance, in the field, the LiDAR scan-matching is ambiguous, but such an ambiguity still fixes some degrees of freedom of the motion, such as changes of z, roll, and pitch. Such a constraint might be useful in combination with visual odometry, which would define the rest motion degrees of freedom.

- The exploration can be improved in multiple ways. Most promising, after the loop closure, the elevation mapping module can recalculate the whole map as it might be affected by the newly gained information. Moreover, segmenting the ceiling and not adding it to the elevation map could account for more challenging scenarios.

- Out of less fundamental implementation details, the safety behavior might be implemented in the exploration framework. A more inclusive formula of the traversability might be fitted to account for the difference between the tilted plane and the stairs. Such a formula can also partly resolve the cluttered elevation map from Fig. 5.1b. Modularity for even better scalability might be implemented in the memory representation of the map, such that only the closest parts of the map are processed in computer memory. Moreover, a dynamic adjustment of the map size might be implemented instead of initializing the whole map from the beginning.

Chapter 6

Conclusion

In this thesis, we propose an augmentation of the graph-based SLAM based on LiDAR-Inertial odometry in a modular way for incorporating additional localization sources in a loosely-coupled manner. We demonstrate the performance of the proposed localization system in two outdoor experimental test sites. Besides, the system performance is further demonstrated using the developed exploration module, revealing insights for practical improvements of the localization system.

The resulting factor graph substitutes LiDAR-based odometry with the additional one in case the former fails, and identifies loop closures based on the LiDAR data accounting for which scans are ambiguous for matching and fixing the drift of the additional odometry. The proposed improvement is based on failure detection by an IMU model, setting the graph constraints uncertainties according to the nature of localization sources and setting the selection rules for keyframes usage.

The proposed localization method has been tested in urban and rural scenarios demonstrating competitive results compared to LIO-SAM when LiDAR scan matching is not ambiguous. In scenarios where scan matching is ambiguous, the proposed method utilized additional localization source improving the results of both LiDAR-based odometry and additional odometry.

Such a system's usage is demonstrated in the exploration task, supporting it with a proposed failure-resilient localization. As a result, the given area has been successfully explored, and a map of the environment has been obtained. This demonstrated practical usage of the developed system, at the same time pointing to unobvious points of improvement of the localization system that are not reflected in localization metrics.

Bibliography

[1] OpenAI's ChatGPT. *Quote on SLAM problem in robotics and exploration.* Retrieved from OpenAI's ChatGPT model (Version GPT-3.5, trained on a knowledge cutoff of 2021). 2023.

[2] Dominik Belter, Przemysław Łabcki, and Piotr Skrzypczyński. "Estimating terrain elevation maps from sparse and uncertain multi-sensor data". In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2012, pp. 715–722.

[3] Brian Yamauchi. "A frontier-based approach for autonomous exploration". In: *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*. 1997, pp. 146–151.

[4] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005.

[5] Leyao Huang. "Review on LiDAR-based SLAM Techniques". In: *International Conference on Signal Processing and Machine Learning (CONF-SPML)*. 2021, pp. 163–168.

[6] Iman Abaspur Kazerouni et al. "A survey of state-of-the-art on visual SLAM". In: *Expert Systems with Applications* 205.117734 (2022).

[7] David Nistér, Oleg Naroditsky, and James Bergen. "Visual odometry". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2004.

[8] Ricardo Roriz, Jorge Cabral, and Tiago Gomes. "Automotive LiDAR technology: A survey". In: *IEEE Transactions on Intelligent Transportation Systems* 23.7 (2021), pp. 6282–6297.

[9] François Pomerleau et al. "Comparing ICP Variants on Real-World Data Sets". In: *Autonomous Robots* 34.3 (2013), pp. 133–148.

[10] Tixiao Shan et al. "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 5135–5142.

[11] Edwin Olson, John Leonard, and Seth Teller. "Fast Iterative Optimization of Pose Graphs with Poor Initial Estimates". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2006, pp. 2262–2269.

[12] *Intel RealSense Tracking Camera T265*. `https://www.intelrealsense.com/tracking-camera-t265/`. accessed Apr 9, 2023.

[13] Tong Qin, Peiliang Li, and Shaojie Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Transactions on Robotics* 34.4 (2018), pp. 1004–1020.

[14] Péter Fankhauser and Marco Hutter. "A universal grid map library: Implementation and use case for rough terrain navigation". In: *Robot Operating System (ROS) The Complete Reference (Volume 1)* (2016), pp. 99–120.

[15] Armin Hornung et al. "OctoMap: An efficient probabilistic 3D mapping framework based on octrees". In: *Autonomous robots* 34 (2013), pp. 189–206.

[16] Péter Fankhauser, Michael Bloesch, and Marco Hutter. "Probabilistic Terrain Mapping for Mobile Robots with Uncertain Localization". In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 (2018), pp. 3019–3026.

[17] Iram Noreen, Amna Khan, and Zulfiqar Habib. "Optimal path planning using RRT* based approaches: a survey and future directions". In: *International Journal of Advanced Computer Science and Applications* 7.11 (2016).

[18] Masoud Emam and Ahmad Fakharian. "Path following of an omnidirectional four-wheeled mobile robot". In: *Artificial Intelligence and Robotics (IRANOPEN)*. IEEE. 2016, pp. 36–41.

[19] Yiyuan Pan et al. "GPU accelerated real-time traversability mapping". In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2019, pp. 734–740.

[20] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012, pp. 3354–3361.

[21] Ji Zhang and Sanjiv Singh. "LOAM: Lidar odometry and mapping in real-time." In: *Robotics: Science and Systems*. Vol. 2. 9. 2014, pp. 1–9.

[22] Mathieu Labbé and François Michaud. "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation". In: *Journal of Field Robotics* 36.2 (2019), pp. 416–446.

[23] Andrzej Reinke, Xieyuanli Chen, and Cyrill Stachniss. "Simple But Effective Redundant Odometry for Autonomous Vehicles". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 9631–9637.

[24] Harry G Barrow et al. "Parametric correspondence and chamfer matching: Two new techniques for image matching". In: *Proceedings: Image Understanding Workshop*. Science Applications, Inc. 1977, pp. 21–27.

[25] César Debeunne and Damien Vivet. "A review of visual-LiDAR fusion based simultaneous localization and mapping". In: *Sensors* 20.7 (2020), p. 2068.

[26] Giorgio Grisetti et al. "A tutorial on Graph-based SLAM". In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43.

[27] Kurt Konolige et al. "Efficient Sparse Pose Adjustment for 2D mapping". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2010, pp. 22–29.

[28] Xiaobin Xu et al. "A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR". In: *Remote Sensing* 14.12 (2022).

[29] Michael Kaess et al. "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 3281–3288.

[30] Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM". en. In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890.

[31] Rainer Kümmerle et al. "g2o: A general framework for graph optimization". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2011, pp. 3607–3613.

[32] Bill Triggs et al. "Bundle adjustment—a modern synthesis". In: *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*. 2000, pp. 298–372.

[33] Sameer Agarwal, Keir Mierle, et al. "Ceres solver". In: (2012).

[34] Frank Dellaert. *Factor graphs and GTSAM: A hands-on introduction*. Tech. rep. Georgia Institute of Technology, 2012.

[35] K. S. Arun, T. S. Huang, and S. D. Blostein. "Least-Squares Fitting of Two 3-D Point Sets". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5 (1987), pp. 698–700.

[36] Tixiao Shan et al. "LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 5692–5698.

[37] Charles C Holt. "Forecasting seasonals and trends by exponentially weighted moving averages". In: *International Journal of Forecasting* 20.1 (2004), pp. 5–10.

[38] Peter Filzmoser, Robert G Garrett, and Clemens Reimann. "Multivariate outlier detection in exploration geochemistry". In: *Computers & Geosciences* 31.5 (2005), pp. 579–587.

[39] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. "The open motion planning library". In: *IEEE Robotics & Automation Magazine* 19.4 (2012), pp. 72–82.

[40] Open Robotics. *DARPA SubT Virtual Competition Software*. `https://github.com/osrf/subt`. accessed Apr 9, 2023. 2019.

[41] Matan Keidar and Gal A Kaminka. "Efficient frontier detection for robot exploration". In: *The International Journal of Robotics Research* 33.2 (2014), pp. 215–236.

[42] *SciPy*. `https://scipy.org/`. accessed Apr 9, 2023.

[43] *ROS - Robot Operating System*. `https://www.ros.org/`. accessed Apr 9, 2023.

[44] *The Open Motion Planning Library*. `https://ompl.kavrakilab.org/`. accessed Apr 9, 2023.

[45] Gaël Guennebaud, Benoît Jacob, et al. *Eigen v3*. http://eigen.tuxfamily.org. accessed Apr 9, 2023. 2010.

[46] Jürgen Sturm et al. “A benchmark for the evaluation of RGB-D SLAM systems”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 573–580.