

UNIVERSITY OF APPLIED SCIENCES OF THE GRISONS

BACHELOR THESIS

**SLAM implementation using ROS2 for a
differential drive robot**

Author:
Florian BÜTTIKER
21-172-010

Supervisor:
Prof. Dr. Philipp ROEBROCK

Client:
Sandro RABBIOSI
ANT Robotics

*A thesis submitted in fulfillment of the requirements
for the degree of Bachelor in Photonics with a specialisation in Mobile Robotics*

in the

Institute for Photonics and Robotics
Applied Future Technologies

August 9, 2024

UNIVERSITY OF APPLIED SCIENCES OF THE GRISONS

Abstract

Applied Future Technologies
Institute for Photonics and Robotics

Bachelor in Photonics with a specialisation in Mobile Robotics

SLAM implementation using ROS2 for a differential drive robot

by Florian BÜTTIKER

This thesis examines the utilisation of Simultaneous Localization and Mapping (SLAM) through the ROS2 in a differential drive robot, with a particular emphasis on its deployment in polytunnels. The research assesses the efficacy of visual SLAM techniques in agricultural contexts, where robots are deployed for tasks such as crop monitoring and pesticide spraying.

Simulated polytunnel conditions, which revealed that visual-inertial SLAM provides superior accuracy to visual-only methods, particularly in challenging terrain. The system's sensitivity to factors such as mist or dirt on the camera lens represents a limitation in its performance. While the SLAM system shows promise for agricultural use, further refinement is required. Future research should seek to improve robustness against environmental disturbances and include field testing. This study contributes to the development of autonomous agricultural robots, offering insights for more reliable farming solutions.

Acknowledgements

Completing this bachelor thesis has been a rewarding journey, and I am sincerely thankful to everyone who contributed to this achievement.

I would especially like to thank Prof. Dr. Philipp ROEBROCK for his insightful lectures which helped lay the foundation for my project. Furthermore, I would like to express my gratitude for the feedback and support provided during the course of this thesis. Their enthusiasm and passion for the subject inspired me and fostered a deep appreciation for the field.

Furthermore, I would like to thank the Team at ANT Robotics, namely Sandro RABBIOSI and Andrew THOMBRA for making this project possible in the first place, but also for their support and insights throughout its development.

Finally, I am grateful to my fellow students and friends for their assistance and inputs during the course of this thesis.

Contents

Abstract	i
Acknowledgements	ii
1 Introduction	1
1.1 Company	1
1.2 Problem	1
1.3 State of the Art	2
1.3.1 SLAM	2
1.3.2 Navigation in Polytunnels	3
1.4 Objective and Scope	3
1.5 Significance	3
2 Theoretical Foundation	4
2.1 Introduction into SLAM	4
2.2 Core Concepts	4
2.3 Visual SLAM	4
2.4 Inertial Measurements	6
2.4.1 Random Walk in IMUs	6
3 Methodology	7
3.1 Testing Environment	7
3.2 Experimental Setup	7
3.3 Camera	8
3.4 IMU	9
3.5 LIDAR	9
3.6 Data Analysis	9
4 Results and Analysis	10
4.1 Results Classroom	10
4.1.1 Localisation	10
4.1.2 Mapping	12
4.2 Results Classroom with Mist	12
4.2.1 Localisation	13
4.2.2 Mapping	14
5 Discussion	15
5.1 Localisation	15
5.2 Mapping	15
5.3 Suitability for Polytunnels	16

5.4	Visual or Visual-Inertial	16
5.5	Spraying	16
5.6	Fusion with Wheel Odometry	17
5.7	Difficulties	17
5.8	Performance Issues	17
5.9	Further Steps	17
6	Conclusion	19
	List of Aids	24
A	Appendix	25
B	Attachments	34
	Declaration of Authorship	37

List of Figures

1.1	Cultivation of Strawberries in a Polyntunnel [1]	1
1.2	Overview of SLAM History with milestones [4]	2
2.1	Example of a visual-inertial SLAM (viSLAM) process explained visually [22].	5
3.1	Classifying Objects from Static to Dynamic [31].	7
3.2	Testing Setup used with Horizontal Camera and Horizontal LIDAR	8
3.3	Testing Setup used with Angled Camera at 30° and Horizontal LIDAR	8
3.4	Testing environment used to simulate the polyntunnel with the poles	8
4.1	Absolute Positional Error from classroom run. The left trajectory shows the result of visual SLAM, the right shows the visual-inertial SLAM run	10
4.2	Mapped corridor as displayed in RVIZ	12
4.3	Image taken in corridor from approximately the same position	12
4.4	Camera after being sprayed with water	12
4.5	Stereo Image taken in RVIZ with sprayed lens	12
4.6	Absolute Positional Error from sprayed run. The left trajectory shows the result of visual SLAM, the right shows the visual-inertial SLAM run. The red arrow shows the position where the camera was sprayed with water	13
4.7	Testing environment XYZ from the sprayed classroom run. The red line marks the point in time where the lens was sprayed with water	14
4.8	Testing environment RPY from the sprayed classroom run. The red line marks the point in time where the lens was sprayed with water	14
A.1	Allan Deviation: Accelerometer Data ZED 2i	25
A.2	Allan Deviation: Gyroscope Data ZED 2i	25
A.3	Allan Deviation: Raw Accelerometer Data ZED 2i	26
A.4	Allan Deviation: Raw Gyroscope Data ZED 2i	26
A.5	Histogram visual-inertial from classroom run with outliers from the tracking failure	28
A.6	Histogram visual-inertial from classroom run before failure (without the outliers)	28
A.7	Testing environment XYZ from the classroom run	29
A.8	Testing environment RPY from the classroom run	29
A.9	Testing environment speed from the classroom run	29
A.10	Visual SLAM mapping corridor side view	30
A.11	Visual SLAM mapping side view	30
A.12	Mapping from the whole classroom run as displayed in RVIZ	31
A.13	LIDAR map top view taken from the classroom run	32

List of Tables

4.1	APE error calculations for SLAM in classroom environment	11
4.2	APE error Calculations for classroom environment with sprayed Lens	13
A.1	Accelerometer Specifications	26
A.2	Gyroscope Specifications	27
A.3	Raw Accelerometer Specifications	27
A.4	Raw Gyroscope Specifications	27
A.5	NVIDIA vSLAM on KITTI dataset APE sequences 11-15 (calculated) . . .	32
A.6	ORB-SLAM2 on KITTI dataset APE sequences 11-15 (calculated)	33
A.7	SOFT2 on KITTI dataset APE sequences 11-15 (calculated)	33

Chapter 1

Introduction

1.1 Company

This thesis was written in collaboration with ANT Robotic, a Hamburg-based company that produces collaborative robots designed to assist farmers. Given the predicted shortage of qualified workers, particularly in cultivation and harvesting, these autonomous robots have the potential to address this challenge. The robots are designed for specific purposes, including harvesting, pesticide spraying and crop monitoring.

1.2 Problem

In order for these autonomous machines to function correctly, it is essential to obtain precise positioning data. This information is then employed to facilitate navigation of the environment. In order to optimise the movement of the robot, it is best to have a map of the surrounding environment. This map can also be saved and used for the next mission in that environment, thus saving time on the farmer's side.

With a working system, the initial configuration can also be reduced, and the robot can learn about the environment as it navigates through it.

The robot that was considered is a differential drive robot, designed for the use in poly-tunnel environments.



FIGURE 1.1: Cultivation of Strawberries in a Polytunnel [1]

As illustrated in Figure 1.1, the floor of the polytunnel may be constructed from soil or a more solid material. This variation in composition affects the grip of the tyres, which may not always be consistent. Additionally, the ground surface is not always perfectly flat. The crop is typically cultivated at an elevated height using poles.

1.3 State of the Art

1.3.1 SLAM

The SLAM problem has been a long-standing challenge in the field of computer vision and robotics. The pioneers in this field were at the forefront of research in the 1980s and 1990s, working on this problem [2, 3]. After a period of intense focus on expensive LIDAR sensors, there was a shift towards the use of more affordable sensors and real-time applications. This led to the development of new strategies to solve the problem, as illustrated in Figure 1.2.

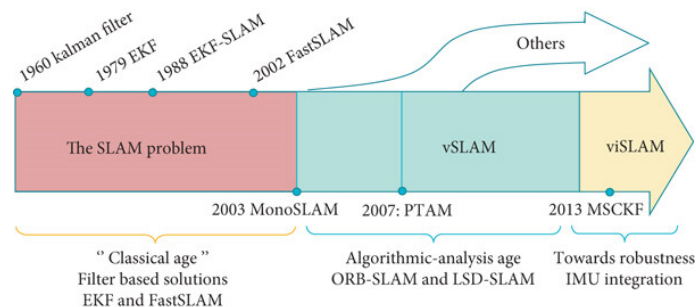


FIGURE 1.2: Overview of SLAM History with milestones [4]

In recent years, a considerable number of packages have been published. These can be classified according to their underlying methodology, which may be feature-based, direct, or RGB-D based. RGB-D is the only method that requires some form of stereo vision; the others do not necessarily require it, but often support it. Notable representatives of the direct method include the LSD-SLAM package [5] and the 2018 DSO package [6]. Additionally, the ORB-SLAM series [7, 8, 9] exemplifies a prominent feature-based method. In the RGB-D category, the ElasticFusion package [10] is a prominent example.

The use of images and, consequently, the generation of more data has led to a significant increase in computational costs. This is why many of these algorithms run on GPUs.

Another promising package is the NVIDIA-vSLAM package [11], as it has been optimised to run on the NVIDIA GPU, rather than solely on CPUs. This is particularly beneficial for embedded mobile applications utilising visual SLAM (vSLAM) in real-time, which are battery-powered and require minimal power consumption. It can be deployed on a System-on-Module (SOM) such as the NVIDIA Jetson Nano, with a power consumption of 10W to 15W.

Tao Peng et al. [12] has shown that using such platforms can improve performance and reduce the power consumption in the same time. This while using ORB-SLAM 2 [8] which isn't necessarily optimized for hardware-acceleration and according to [13] performs worse than NVIDIA's vSLAM when benchmarked with the KITTI dataset [14].

1.3.2 Navigation in Polytunnels

By considering navigation and localisation in polytunnels or greenhouses, there are a number of different approaches that can be taken. The straightforward approach to localisation employed in open fields utilising GNSS or GPS-RTK comes to mind. However, this method is susceptible to disruptions in polytunnels [15]. Nevertheless, even if a real-time kinematics setup is employed, it is a costly solution that only provides a location, thus it is necessary to have a map of the tunnel in place. Alternative approaches in strawberry polytunnels utilise 2D LIDAR and an IMU [16].

1.4 Objective and Scope

The objective of this study is to assess the viability of a visual SLAM approach in a polytunnel environment.

Given the potential benefits of using cameras on agricultural robots, for instance for analysing plant health or fruit ripeness, this study focuses on visual solutions, contrary to the conventional approach to SLAM systems which employs LIDAR.

1. How reliable is a camera-based SLAM solution in a polytunnel?
2. Visual vs. Visual-Inertial SLAM?
3. How does an obstructed FOV (mist/dirt) affect the performance?
4. Is using the wheel odometry beneficial to the performance?

These questions guide the investigation into optimizing autonomous robot navigation within polytunnel environments.

1.5 Significance

The findings presented here can be of vital importance for agricultural robots operating in these tunnels. The insights gained could facilitate more effective and reliable navigation, thereby enhancing operational efficiency and reducing the necessity for human intervention. This can result in a reduction in operational costs and an overall decrease in investment.

Chapter 2

Theoretical Foundation

2.1 Introduction into SLAM

Simultaneous Localisation and Mapping represents a foundational technology in the domain of mobile robotics. SLAM enables robots to construct and update a map of unknown environments while simultaneously tracking their location in said environments. The utilisation of SLAM is vital for autonomous navigation in complex settings, including urban environments, indoor spaces and extraterrestrial terrains [17].

2.2 Core Concepts

The term **Localisation** is used to describe the process of determining the position and orientation of a robot in relation to a reference frame. In the context of SLAM, the localisation process is achieved by integrating data from a variety of sensors, including those used for odometry, inertial measurement units, and exteroceptive sensors such as cameras and LIDAR [18].

The process of creating a representation of the environment that a robot navigates is known as **mapping**. The map may be represented in a variety of ways, including as occupancy grids, feature-based maps, or 3D point clouds. The accuracy of a map is a crucial element in enabling a robot to navigate with reliability and to avoid obstacles [18].

The concept of **Sensor Fusion** involves integrating data from multiple sensor types, which can be employed to stabilise the SLAM process. This approach can assist in circumventing errors in the map and guaranteeing the accurate tracking of the robot's location [19].

Loop closure represents a fundamental component of SLAM, whereby a robot is enabled to recognise locations that it has previously visited by detecting similarities in features or landmarks observed from disparate viewpoints. This process reduces the drift of sensors over time, thereby facilitating the generation of a more accurate map [20, 21].

2.3 Visual SLAM

As mentioned, visual SLAM is a method of solving the SLAM problem using camera systems. This solution is used to get comparable results using cheaper hardware. Another advantage of this is the versatility of a camera, since mapping usually is not the main task of a rover, it can be used for other tasks as well [21]. See Figure 2.1 for a flowchart of what the inner workings might look like.

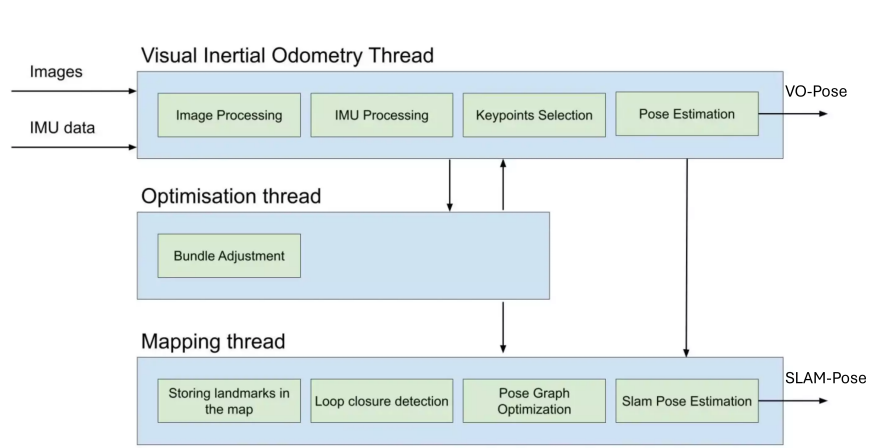


FIGURE 2.1: Example of a visual-inertial SLAM (viSLAM) process explained visually [22].

Some insight on how the inner workings Figure 2.1 might look like:

Visual Inertial Odometry (VIO) Thread [23, 24]

- **Image Processing:** Extracts and tracks keypoints across image frames to determine relative motion. By using features (such as corners, edges or blobs) that are distinct and can be reliably tracked across frames. Techniques such as SIFT (Scale-Invariant Feature Transform), ORB (Oriented FAST and Rotated BRIEF) or others can be used.
- **IMU Processing:** Processes and filter inertial data to predict short-term motion and correct biases.
- **Keypoints Selection:** Chooses robust keypoints for accurate pose estimation.
- **Pose Estimation:** Combines visual and IMU data to estimate the camera's position and orientation (*VO-Pose*).

Optimization Thread [25]

- **Bundle Adjustment:** Refines the camera poses and 3D keypoint positions by minimizing reprojection errors across multiple frames, enhancing accuracy and reducing drift.

Mapping Thread [24]

- **Storing Landmarks:** Updates the map with optimized keypoints (landmarks).
- **Loop Closure Detection:** Recognizes when the camera revisits a location, correcting accumulated errors (drift).
- **Pose Graph Optimization:** Refines the entire camera trajectory for global consistency.
- **SLAM Pose Estimation:** Provides the final, globally consistent camera pose (*SLAM-Pose*).

Flow of Information

- **VO-Pose:** Initial pose estimate from the VIO thread.
- **SLAM-Pose:** Final, refined pose estimate from the Mapping thread, ensuring accurate localization and mapping.

2.4 Inertial Measurements

It is of critical importance to ensure that the viSLAM system is provided with accurate readings from the IMU in order to achieve optimal results. The IMU included with the ZED 2i camera was satisfactory initially, but subsequent use revealed a number of deficiencies. Even when utilising the presumed factory-calibrated data, the issues remained evident. These sensors are inherently noisy, meaning that slight variations in their output are produced even in the absence of actual movement. Over time, the accumulated noise causes a gradual shift in the measured values since this data usually is integrated. [26].

2.4.1 Random Walk in IMUs

IMUs are affected by random walk, where sensor noise accumulates over time, causing output drift. This affects applications such as navigation and motion tracking. Random walk in IMUs is typically the result of inherent noise characteristics in the sensors, particularly in gyroscopes and accelerometers. This noise manifests as random fluctuations or errors that accumulate over time, resulting in an increasing divergence from the true values. In gyroscopes, for instance, this noise can cause the angular velocity measurements to deviate from their true value, while in accelerometers, it affects the measurement of linear acceleration [27].

Since these parameters were not supplied by the manufacturer in this case, they had to be obtained by measuring and calculating them.

In order to address these issues, there are a number of potential solutions. In this case, it was decided that the most appropriate approach was to use the Allan standard deviation, given its widespread use as a standardized procedure [28].

Chapter 3

Methodology

3.1 Testing Environment

The data collected from testing can be analysed in different ways, as there are multiple objectives to be met in the result. One area for examination is the mapping capabilities, which pertains to the accuracy of the map produced. Additionally, the localisation aspect is a focus. This encompasses the precision of the path and the poses visited during the test run.

To standardize the benchmarks of SLAM algorithms, public datasets are used and the results compared, for example the ETH3D dataset [29, 30] or as mentioned before the KITTI dataset [13].

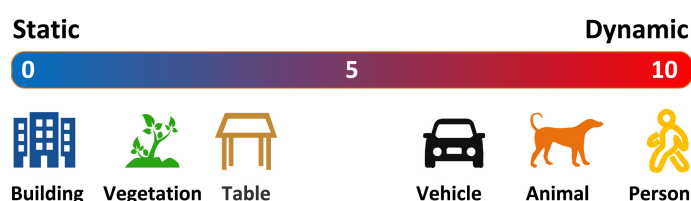


FIGURE 3.1: Classifying Objects from Static to Dynamic [31].

To effectively assess the suitability of the package in question, a test setup was employed, mirroring the actual use case of ANT-Robotics. An environment can be classified from static to dynamic as shown in Figure 3.1. In the case of a polytunnel, the value is low. It can be assumed that the environment is static. This information helps in picking the right SLAM method and test data.

3.2 Experimental Setup

For the testing setup, two different rigs were selected. One employed both LIDAR and stereo camera positioned horizontally, as illustrated in Figure 3.2. The rig depicted in Figure 3.3 utilised an angled camera with the objective of enhancing the Field of View and increasing the visibility of features, particularly when the camera was mounted in close proximity to the ground.

The tests were conducted on a trolley in order to facilitate mobility while attempting to replicate the movement of the ADIR-robot. As a consequence of the difficulties encountered with the OS and drivers on the NVIDIA Jetson Nano, a desktop computer was employed

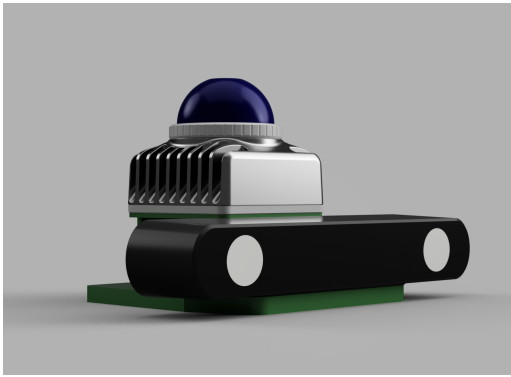


FIGURE 3.2: Testing Setup used with Horizontal Camera and Horizontal LIDAR

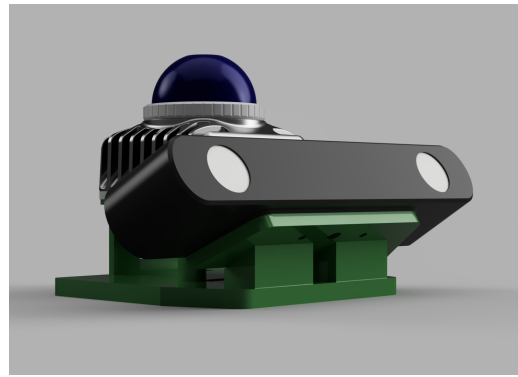


FIGURE 3.3: Testing Setup used with Angled Camera at 30° and Horizontal LIDAR

as an alternative. The PC was mounted on the trolley for processing the data in real time. To conduct the test, the trolley was pushed through the environment. This was conducted in an indoor setting displayed in Figure 3.4. This with the objective of replicating the configuration employed in Figure 1.1, the future workplace of the robot. This is in the FHGR building B in room B 1.03.

The open-source ROS2 framework [32] was utilised to oversee the execution of all packages, as this is standard practice in the industry and is used by ANT-Robotics.



FIGURE 3.4: Testing environment used to simulate the polytunnel with the poles

3.3 Camera

In order to gather visual data, a Stereolabs ZED 2i was utilised. A ROS2 package has already been developed for this camera, which allows the user to select different resolutions and frame rates. In order to utilise this package, the machine in question must have

a NVIDIA GPU built-in and at least one USB 3.0 port, as outlined in the Stereolabs specifications [33]. The camera is equipped with a built-in IMU.

3.4 IMU

Using the tools supplied in the Kalibr package [34] and Allan-Variance package [35] the analysis was conducted. The IMU was fixed still in place and the data was recorded for approximately 32 hours to a ROSBAG. After that the Allan-Variance software was used to plot the data and obtain the parameters. The calculated values are shown in Table A.1 and Table A.2. The plots also can be found in the appendix. These calculated values were used for the drift reduction of the viSLAM.

3.5 LIDAR

For referencing the path, a solid-state Livox Mid-360 LIDAR was used. Using the driver Livox ROS2 driver [36] as a interface. For SLAM the popular FAST-LIO package [37] was used.

3.6 Data Analysis

To check the trajectory data the absolute pose error (APE) was used as a metric. This is usually used to compare SLAM packages. To plot this data and calculate the errors, the evo package [38] was used. The recorded paths from the ROSBAG can be read and analysed with this software.

With these values, the goodness and precision of a model can be measured and compared to each other.

Chapter 4

Results and Analysis

All runs were recorded during daylight with additional lighting. The Python package `evo` was used for the evaluation of the trajectories.

4.1 Results Classroom

The data for this particular run was gathered at the FHGR building B, specifically on the first floor. This can be considered a typical office environment.

This run, along with several others, was conducted on multiple occasions. However, due to the inherent differences between each run, it was not possible to overlay them. A representative example was chosen to illustrate the results gathered from the runs.

4.1.1 Localisation

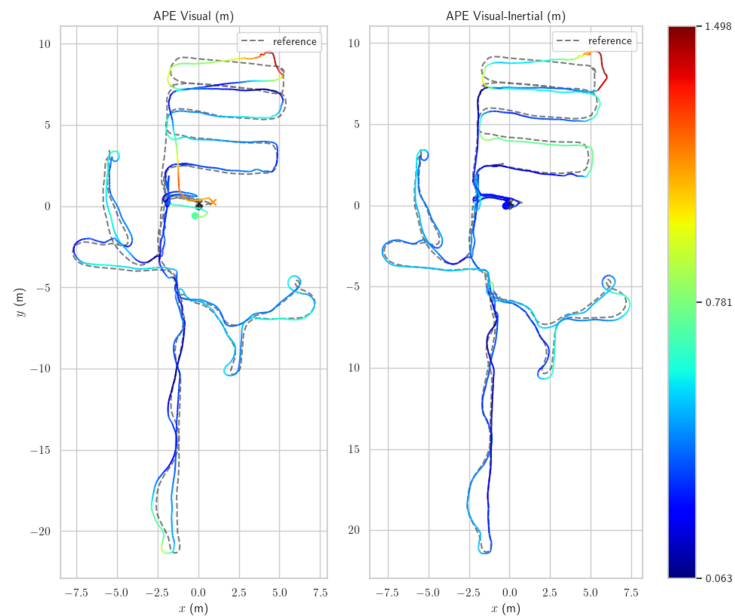


FIGURE 4.1: Absolute Positional Error from classroom run. The left trajectory shows the result of visual SLAM, the right shows the visual-inertial SLAM run

Looking at Figure 4.1, there are three different trajectories visible, the reference is the same for both runs. These were all recorded simultaneously, so they all travelled the same path. The trajectory of the LIDAR using FAST-LIO was chosen as a reference to align the others. The plot shows the path in a plane. The rectangle from $(-2.5\text{ m}, -1\text{ m})$ to $(6\text{ m}, 10\text{ m})$ in Figure 4.1 shows the environment displayed in Figure 3.4.

The path was projected to the XY-plane since it was in a planar environment. The deviation in the Z-Axis is shown in Figure A.7. This barley has an influence on the accuracy of the system.

TABLE 4.1: APE error calculations for SLAM in classroom environment

	RMSE (m)	mean (m)	median (m)	σ (m)	min (m)	max (m)
Visual-Inertial	0.472	0.418	0.364	0.220	0.066	1.393
Visual	0.635	0.552	0.461	0.313	0.063	1.498
Visual-Inertial before fail	0.414	0.392	0.365	0.132	0.070	0.833
Visual before fail	0.436	0.405	0.407	0.160	0.027	0.990

In Table 4.1, the absolute pose error is displayed in comparison with the reference. It can be observed that these values are relatively high. When comparing the metrics root mean squared error (RMSE), mean, median and standard deviation (σ), it can be seen that the visual-inertial model demonstrates a higher level of accuracy. This suggests that the measurement is more stable.

The two minimums are comparable, while the maximums are significantly elevated. This phenomenon can be observed in Figure A.7 around the 740 s mark or in Figure Figure 4.1 at $(x = 5\text{ m}, y = 9.5\text{ m})$. This was caused by the loss of tracking due to insufficient visible features.

Upon closer examination of the calculated values and the APE in Figure A.5, the distribution exhibits characteristics of a normal distribution. Following the removal of outliers resulting from tracking failure, the graph in Figure A.6 more closely resembles the expected distribution. The distributions exhibits a shift to the right. After this adjustment, the standard deviation improved by 40% to 13.2 cm.

Figure A.8 illustrates the Euler angles. Upon examination of the roll and pitch angles, it indicates the paths of viSLAM and vSLAM are analogous yet offset from one another. The reference exhibits minimal rotation around the roll and pitch axes, resulting from a relatively flat trajectory.

Upon analysis of the yaw rotation, it is observed that both the visual and visual-inertial paths closely adhere to the trajectory of the reference. It is likely due to the majority of rotations occurred around the Z-axis.

A comparison of the speed across all axes in Figure A.9, reveals that the differences are minimal. However, the visual signal demonstrates a slower reaction to rapid changes.

4.1.2 Mapping

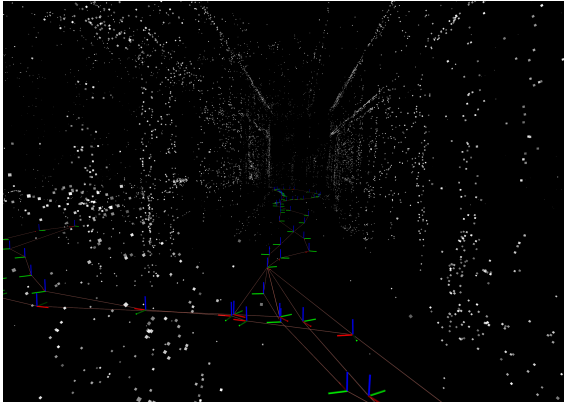


FIGURE 4.2: Mapped corridor as displayed in RVIZ



FIGURE 4.3: Image taken in corridor from approximately the same position

In Figure 4.2 the mapped corridor is shown, which was created using a 720p resolution. The Figure 4.3 illustrates the same scene as an image captured by a other camera. Upon comparison, it is evident that certain features more prominent in the point cloud than others. Given the lack of distinctive features in the environment, the number of mapped points is relatively limited.

A comparison of the top view in Figure A.12 with Figure 4.1 reveals that the path taken can be retraced, thereby providing some insight into the mapped environment. For a more comprehensive understanding, it is recommended to examine Figure A.10 and Figure A.11 . The map constructed using the reference LIDAR is shown in Figure A.13.

4.2 Results Classroom with Mist

This run was similar to the previous one. The difference was that the camera lenses sprayed with a water as illustrated in Figure 4.4. The run was started with a clean camera. Subsequently, the lenses were misted with a water spray bottle.



FIGURE 4.4: Camera after being sprayed with water



FIGURE 4.5: Stereo Image taken in RVIZ with sprayed lens

4.2.1 Localisation

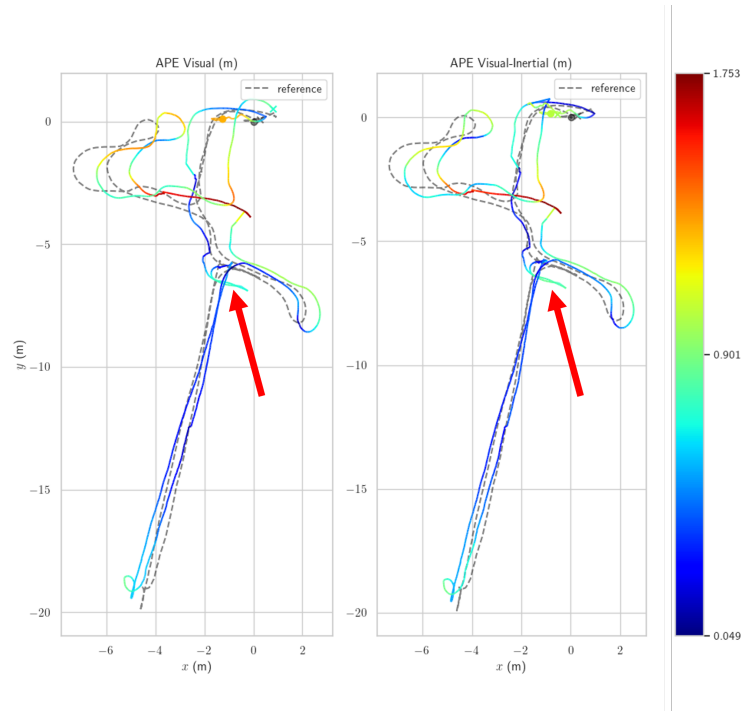


FIGURE 4.6: Absolute Positional Error from sprayed run. The left trajectory shows the result of visual SLAM, the right shows the visual-inertial SLAM run. The red arrow shows the position where the camera was sprayed with water

The position where the camera was sprayed is indicated by an arrow in Figure 4.6. After this, the run was continued, allowing for the observation of potential changes. For a more comprehensive understanding, the taken path Figure 4.7 can be consulted, where the sprayed position is marked with a red line.

TABLE 4.2: APE error Calculations for classroom environment with sprayed Lens

	RMSE (m)	mean (m)	median (m)	σ (m)	min (m)	max (m)
Visual-Inertial SLAM	0.617	0.524	0.478	0.326	0.012	1.477
Visual SLAM	0.822	0.710	0.613	0.413	0.049	1.753

An examination of the data in Table 4.2 reveals elevated values. However, a comparison with the data in reference Table 4.1 before correction indicates that the former are not significantly inferior.

Upon examination of the provided graphs, including reference Figure 4.8, it becomes evident that the alteration subsequent to the application of the water is marginal.

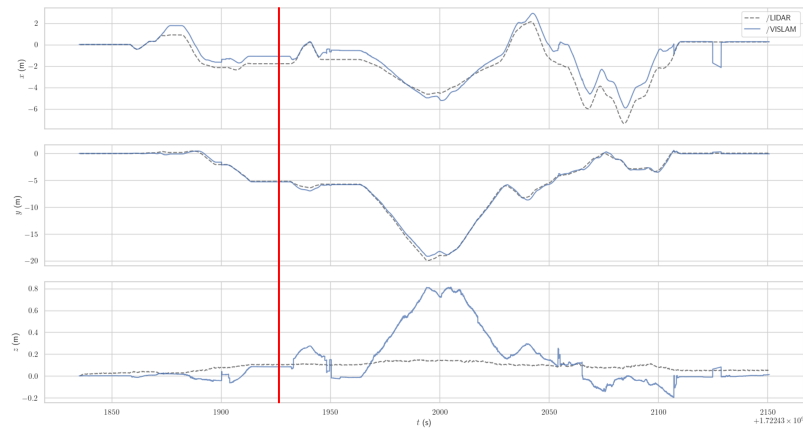


FIGURE 4.7: Testing environment XYZ from the sprayed classroom run. The red line marks the point in time where the lens was sprayed with water

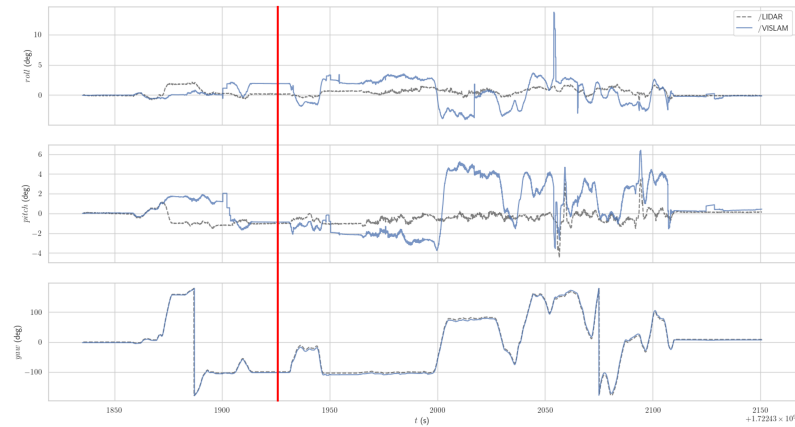


FIGURE 4.8: Testing environment RPY from the sprayed classroom run. The red line marks the point in time where the lens was sprayed with water

4.2.2 Mapping

Following the application of water to the camera, the mapping performance was found to be impaired. This resulted in a reduction in the number of matching points incorporated into the point cloud. The quantity of water present on the lens was identified as a contributing factor.

The influence of water mist sprayed in front of the camera, without causing it to become wet, was also examined within this context. However, the resulting data could not be effectively visualised due to the minimal impact on the mapping capabilities.

Chapter 5

Discussion

5.1 Localisation

The pathways illustrated in Figure 4.1 provide a valuable insight into the findings. A comparison of the LIDAR path with the visual ones for the entire run in the classroom reveals a improvable result. As previously stated, this is a consequence of the numerous outliers generated when tracking failed in the confined and poorly textured area. Upon examination of the tracking data prior to the failure, as illustrated in Table 4.1, the outcomes appear considerably more favourable. When looking at the change in distribution when removing the outliers in Figure A.5 and going to Figure A.6, along with changes in RMSE and mean values in Table 4.1, suggests the presence of a systematic error in the measurement. This is supported by the distribution of the measurement errors.

It is possible that this offset was caused by a faulty transformation between the two sensors, resulting in unequal origins in the coordinate systems. This could explain the shift observed in Figure A.6 Otherwise, the standard deviation is relatively low and aligns with findings in the literature for this and comparable systems, as documented in [4]. Additionally, a comparison with a previous version of the NVIDIA VSLAM was conducted using the KITTI dataset, of which the result was uploaded online for reference. The data was used to calculate the values in Table A.5. For comparison purposes, the popular packages ORB-SLAM2 Table A.6 and SOFT2 [39] in Table A.7, which represents the best in its class, were also considered.

However, it should be noted that the datasets were of a disparate nature, with varying sample sizes. Consequently, the results must be interpreted with caution.

Furthermore, there is a possibility that the reference provided by the LIDAR is carrying an error, this is supported by the literature [40, 41]. The APE standard deviation for the KITTI dataset in recorded sequences 05 to 09 is observed to range from 0.25 m to 2.09 m. The NVIDIA vSLAM package currently lacks the functionality required to perform global localisation, thus preventing an evaluation of the kidnapped robot problem.

5.2 Mapping

In regard to mapping, the outcomes of the visual-inertial SLAM demonstrate a satisfactory level of performance. However, the sparse maps are unable to accurately detect slim objects such as poles. The preference of the algorithm for objects situated in the central portion of the image may result in the neglect of finer details located at the borders of the image. This could prove problematic if the navigation is based solely on the poles from the created map. Should this be required, an additional program could be developed to detect the said

poles and map them independently to the map. Additionally, the Nvblox package [42] from NVIDIA could be employed to get an enhanced 3D reconstruction of the environment. Following the application of a lens spray containing water, the mapping capabilities were found to be compromised. This could be attributed to the reduced number of points that could be matched, although there remained a sufficient number for the purposes of localisation and odometry.

5.3 Suitability for Poly tunnels

It is challenging to provide a definitive answer to this question, given the limited field testing conducted as part of the thesis. The validation conducted within the tunnel is challenging to assess due to the absence of a ground truth. It is unfortunate that the visual SLAM datasets in comparable settings could not be obtained due to their unavailability or incompatibility with ROS2 [43, 44].

However, other factors present in poly tunnels, namely changes in lighting or the spraying, were successfully tested and demonstrated favourable outcomes, ranging from acceptable to excellent. In the instance of brightness changes, the visual-inertial system once again exhibited an advantage in its capacity to sustain operation despite the disruption of the tracking process.

5.4 Visual or Visual-Inertial

The results and subsequent analysis have demonstrated that, in the context of a robust system, the visual-inertial system exhibits a distinct advantage over the only visual system. Even though the IMU movements are only taken into account in the calculations in the event of a visual tracking failure, the performance of this package is enhanced when the IMU tracking takeover is enabled. However, it should be noted that there are some limitations to the IMU tracking takeover, as it is only stable for a couple of seconds. After this period, the drift can influence it to a significant extent, as can be observed in the Classroom scenario, even when the correct drift correction values are applied.

5.5 Spraying

In regard to the issue of a dirty or sprayed lens, the findings in this study were inconclusive. This finding is also consistent with the results of the conducted study [45]. In some instances, the impact was more pronounced than in others. Nevertheless, the outcome was more favourable than anticipated. This could be enhanced by the utilisation of a secondary camera positioned differently or alternative sensor fusion techniques. The application of the spraying in front of the lens had a negligible impact on the results. Nevertheless, this is a challenging aspect to control when the robot is tasked with spraying. In future stages, a mist/droplet removal algorithm, such as that described in [46], could be employed to filter the images prior to matching. In [45], the use of DNNs for this purpose were also proposed.

The impact of water on the LIDAR was not within the scope of this study.

5.6 Fusion with Wheel Odometry

It is difficult to provide a definitive answer to this question, as the odometry of the wheels could not be tested in the test environment due to the robots size. The results produced suggest that this could enhance the robustness of the positioning, particularly when the robot is stationary or tracking is lost. However, when considering the different surfaces on which the robot is moving, the implementation, especially when moving or turning, could present a challenge.

5.7 Difficulties

It proved very hard to find a working vSLAM solution for this application. Since many of these packages are not maintained anymore, only a small percentage of them support ROS2. Some of that also could be due to the industry being hesitant to migrate their entire codebase from ROS1 to ROS2. Even for those packages found for ROS2, very little of them actually worked and even less with the hardware available. This led to the use of the closed-source NVIDIA vSLAM package [11]. There were no publications found using this algorithm. This is not an optimal solution since less of the inner workings and limitations of the system are not known. However, as mentioned this package it is still kept up-to-date and new features are developed [22, 47].

The ZED 2i camera has a topic called `/imu/data` and `/imu/data/raw`. However when comparing them to the Figure A.3 and Figure A.4 as well as to the Table A.3 and Table A.4 the difference is minimal, even when looking at the change over an hour. It is not really clear what the factory calibration from the camera did.

5.8 Performance Issues

Some issues were encountered when using the Stereolabs ZED 2i in conjunction with the hardware employed in this experiment. As previously stated in the methodology section, an older desktop computer was utilised. It was observed that when ROSBAG recordings were taken, especially when recording the images provided by the camera, the performance was significantly reduced. This was noticeable by unexpected jumping from the position calculated by the SLAM software. This could have been due to the large amount of data being transported over USB and subsequently through the USB controller or the RAM speed, that is quite low. The CPU load was not significantly impacted when recording fewer topics to the external SSD; however, the performance of SLAM improved, as was evidenced by fewer images being dropped and a reduction in the jumping of position. Similar issues were noticed when replaying ROSBAGS from the external drive. This was only a problem when recording the ROS topics, this problem did not occur during normal real-time operation.

5.9 Further Steps

Further testing is required with embedded hardware in the field to resolve the issues noticed during this project. Additionally, testing different camera angles could prove beneficial,

particularly in mapping the poles close to the ground.

To enhance overall accuracy, a superior IMU and the use of multiple stereo cameras, which are supported by the package, are recommended.

Chapter 6

Conclusion

This study examined the deployment of SLAM (Simultaneous Localization and Mapping) using ROS2 in a differential drive robot operating within polytunnel environments. The objective was to integrate visual and inertial methods to enhance the robot's localisation and mapping capabilities. The research demonstrated that while visual-inertial SLAM provides significant improvements over visual-only approaches, several challenges must be addressed to ensure robust performance in agricultural settings.

The findings indicate that the SLAM system developed in this thesis is effective in standard indoor environments. However, its performance can be compromised by real-world conditions such as mist or dirt on the camera lens, which can obstruct the field of view and degrade mapping accuracy. Nevertheless, the system demonstrated potential for localisation and mapping within simulated polytunnel conditions, suggesting a valuable contribution to agricultural automation. However, the study also identified limitations related to software compatibility, hardware performance and environmental adaptability. These issues indicate that further research and development are necessary to refine the SLAM system for practical deployment in agriculture. In particular, enhancing the robustness of the system against environmental disturbances, improving sensor integration and conducting extensive field tests are essential steps for future work.

In conclusion, the implementation of SLAM using ROS2 has been demonstrated to be a viable approach for differential drive robots. However, further refinement is required for its application in polytunnel environments. Addressing the identified challenges will be essential for advancing the use of autonomous robots in agriculture, ultimately contributing to increased efficiency and reduced human intervention in farming operations.

Bibliography

- [1] Lights Science and Technology. *How to Grow Indoor Strawberries*. Accessed: July 10, 2024. 2024. URL: <https://lightsciencetech.com/how-to-grow-indoor-strawberries/>.
- [2] Randall C. Smith and Peter Cheeseman. "On the Representation and Estimation of Spatial Uncertainty". In: *The International Journal of Robotics Research* 5.4 (1986), pp. 56–68. DOI: [10.1177/027836498600500404](https://doi.org/10.1177/027836498600500404). eprint: <https://doi.org/10.1177/027836498600500404>. URL: <https://doi.org/10.1177/027836498600500404>.
- [3] J.J. Leonard and H.F. Durrant-Whyte. "Simultaneous map building and localization for an autonomous mobile robot". In: *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*. 1991, 1442–1447 vol.3. DOI: [10.1109/IROS.1991.174711](https://doi.org/10.1109/IROS.1991.174711).
- [4] Myriam Servières et al. "Visual and Visual-Inertial SLAM: State of the Art, Classification, and Experimental Benchmarking". In: *Journal of Sensors* 2021.1 (2021), p. 2054828. DOI: <https://doi.org/10.1155/2021/2054828>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/2054828>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2021/2054828>.
- [5] J. Engel, T. Schöps, and D. Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. Sept. 2014.
- [6] J. Engel, V. Koltun, and D. Cremers. "Direct Sparse Odometry". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Mar. 2018).
- [7] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), 1147–1163. ISSN: 1941-0468. DOI: [10.1109/tro.2015.2463671](https://doi.org/10.1109/tro.2015.2463671). URL: <http://dx.doi.org/10.1109/TRO.2015.2463671>.
- [8] Raúl Mur-Artal and Juan D. Tardós. "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras". In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262. DOI: [10.1109/TRO.2017.2705103](https://doi.org/10.1109/TRO.2017.2705103).
- [9] Carlos Campos et al. "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM". In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890.
- [10] Thomas Whelan et al. "ElasticFusion: Real-time dense SLAM and light source estimation". In: *The International Journal of Robotics Research* 35.14 (2016), pp. 1697–1716. DOI: [10.1177/0278364916669237](https://doi.org/10.1177/0278364916669237). eprint: <https://doi.org/10.1177/0278364916669237>. URL: <https://doi.org/10.1177/0278364916669237>.
- [11] NVIDIA. *Isaac ROS Visual SLAM*. https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_visual_slam. Accessed: June 10, 2024.

- [12] Tao Peng et al. "Evaluating the Power Efficiency of Visual SLAM on Embedded GPU Systems". In: *2019 IEEE National Aerospace and Electronics Conference (NAECON)*. 2019, pp. 117–121. DOI: [10.1109/NAECON46414.2019.9058059](https://doi.org/10.1109/NAECON46414.2019.9058059).
- [13] Alexander Korovko et al. "Realtime Stereo Visual Odometry". Accessed: March 11, 2024. URL: https://www.cvlibs.net/datasets/kitti/eval_odometry_detail.php?&result=252948fc1b95f8b635cf63df526e4864cb6c0f41.
- [14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2012.
- [15] Tuan Le et al. "A Cost-Effective Global Localization System for Precision Agriculture Tasks in Polytunnels". In: *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. 2020, pp. 1572–1577. DOI: [10.1109/CASE48305.2020.9216823](https://doi.org/10.1109/CASE48305.2020.9216823).
- [16] Ya Xiong et al. "An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation". In: *Journal of Field Robotics* 37.2 (2020), pp. 202–224. DOI: <https://doi.org/10.1002/rob.21889>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21889>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21889>.
- [17] Shuran Zheng et al. "Simultaneous Localization and Mapping (SLAM) for Autonomous Driving: Concept and Analysis". In: *Remote Sensing* 15.4 (2023). ISSN: 2072-4292. DOI: [10.3390/rs15041156](https://doi.org/10.3390/rs15041156). URL: <https://www.mdpi.com/2072-4292/15/4/1156>.
- [18] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha. "Visual Simultaneous Localization and Mapping: A Survey". In: *Artificial Intelligence Review* 43.1 (2015), pp. 55–81. DOI: [10.1007/s10462-012-9365-8](https://doi.org/10.1007/s10462-012-9365-8). URL: <https://doi.org/10.1007/s10462-012-9365-8>.
- [19] Simon Lynen et al. "A robust and modular multi-sensor fusion approach applied to MAV navigation". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 3923–3929. DOI: [10.1109/IROS.2013.6696917](https://doi.org/10.1109/IROS.2013.6696917).
- [20] Brian Williams et al. "A comparison of loop closing techniques in monocular SLAM". In: *Robotics and Autonomous Systems* 57.12 (2009). Inside Data Association, pp. 1188–1197. ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2009.06.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889009000876>.
- [21] Xiang Gao and Tao Zhang. *Introduction to visual SLAM: from theory to practice*. Springer Nature, 2021.
- [22] Sagar Dhatrak and Naitik Nakrani. *AMR Navigation Using Isaac ROS, VSLAM, and NVBlox with Intel RealSense Camera*. Accessed: May 5, 2024. 2023. URL: <https://www.einfochips.com/blog/amr-navigation-using-isaac-ros-vslam-and-nvblox-with-intel-realsense-camera/>.
- [23] Thomas Schultz. *Feature Detection*. Lecture notes, Department of Computer Science, University of Zurich. 2022. URL: https://rpg.ifi.uzh.ch/docs/teaching/2022/06_feature_detection_2.pdf.

- [24] Jin-Chun Piao and Shin-Dug Kim. "Adaptive Monocular Visual-Inertial SLAM for Real-Time Augmented Reality Applications in Mobile Devices". In: *Sensors* 17.11 (2017). ISSN: 1424-8220. DOI: [10.3390/s17112567](https://doi.org/10.3390/s17112567). URL: <https://www.mdpi.com/1424-8220/17/11/2567>.
- [25] Mingyang Li. *Visual-inertial odometry on resource-constrained systems*. University of California, Riverside, 2014.
- [26] H. Zhao et al. "A time-controllable Allan variance method for MEMS IMU". In: *Industrial Robot* 40.2 (2013), pp. 111–120. DOI: [10.1108/01439911311297702](https://doi.org/10.1108/01439911311297702).
- [27] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. "Analysis and Modeling of Inertial Sensors Using Allan Variance". In: *IEEE Transactions on Instrumentation and Measurement* 57.1 (2008), pp. 140–149. DOI: [10.1109/TIM.2007.908635](https://doi.org/10.1109/TIM.2007.908635).
- [28] "IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros". In: *IEEE Std 952-1997* (1998), pp. 1–84. DOI: [10.1109/IEEESTD.1998.86153](https://doi.org/10.1109/IEEESTD.1998.86153).
- [29] Thomas Schops et al. "A Multi-view Stereo Benchmark with High-Resolution Images and Multi-camera Videos". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. DOI: [10.1109/cvpr.2017.272](https://doi.org/10.1109/cvpr.2017.272).
- [30] ETH Zurich. *ETH3D SLAM Benchmark*. https://www.eth3d.net/slam_benchmark. Accessed: May 3, 2024. 2024.
- [31] Linhui Xiao et al. "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment". In: *Robotics and Autonomous Systems* 117 (2019), pp. 1–16. ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2019.03.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889018308029>.
- [32] Steven Macenski et al. "Robot Operating System 2: Design, architecture, and uses in the wild". In: *Science Robotics* 7.66 (2022), eabm6074. DOI: [10.1126/scirobotics.abm6074](https://doi.org/10.1126/scirobotics.abm6074). URL: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>.
- [33] Stereolabs. *Hardware Specifications - StereoLabs*. Accessed: May 2, 2024. 2024. URL: <https://www.stereolabs.com/docs/installation/specifications>.
- [34] ETH Zurich Autonomous Systems Lab. *Kalibr: Toolbox for Camera-IMU Calibration*. <https://github.com/ethz-asl/kalibr>. Accessed: May 29, 2024. 2024.
- [35] Autoliv Research. *allan_variance_ros2: ROS2 Package for Allan Variance Analysis*. https://github.com/Autoliv-Research/allan_variance_ros2. Accessed: May 30, 2024. 2024.
- [36] Livox-SDK. *livox_ros_driver2: Livox ROS Driver 2*. https://github.com/Livox-SDK/livox_ros_driver2. 2024.
- [37] Xieyuanli Chen et al. *FAST-LIO: Fast LiDAR-Inertial Odometry*. https://github.com/hku-mars/FAST_LIO. Accessed: May 30, 2024. 2020.
- [38] Michael Grupp. *evo: Python package for the evaluation of odometry and SLAM*. <https://github.com/MichaelGrupp/evo>. 2017.
- [39] Igor Cvišić, Ivan Marković, and Ivan Petrović. "SOFT2: Stereo Visual Odometry for Road Vehicles Based on a Point-to-Epipolar-Line Metric". In: *IEEE Transactions on Robotics* (2022), pp. 1–16. DOI: [10.1109/TR0.2022.3188121](https://doi.org/10.1109/TR0.2022.3188121).

- [40] Wei Xu et al. "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry". In: *IEEE Transactions on Robotics* 38 (Aug. 2022), pp. 1–21. DOI: [10.1109/TR0.2022.3141876](https://doi.org/10.1109/TR0.2022.3141876).
- [41] Huimin Pan et al. "LiDAR-IMU Tightly-Coupled SLAM Method Based on IEKF and Loop Closure Detection". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* PP (Jan. 2024), pp. 1–16. DOI: [10.1109/JSTARS.2024.3357536](https://doi.org/10.1109/JSTARS.2024.3357536).
- [42] NVIDIA Corporation. *Isaac ROS Nvblox*. Accessed: July 21, 2024. 2024. URL: https://nvidia-isaac-ros.github.io/repositories_and_packages/isaac_ros_nvblox/index.html.
- [43] Shuo Wang et al. *LFS-Dataset*. 2023. DOI: [10.21227/s85q-cf10](https://doi.org/10.21227/s85q-cf10). URL: <https://dx.doi.org/10.21227/s85q-cf10>.
- [44] Fernando Cañadas-Aránega et al. "Multimodal Mobile Robotic Dataset for a Typical Mediterranean Greenhouse: The GREENBOT Dataset". In: *Sensors* 24.6 (2024). ISSN: 1424-8220. DOI: [10.3390/s24061874](https://doi.org/10.3390/s24061874). URL: <https://www.mdpi.com/1424-8220/24/6/1874>.
- [45] Lorenzo Sarti et al. "Towards Robust Visual Odometry Systems Against Camera Lens Failures". In: *2023 IEEE 34th International Symposium on Software Reliability Engineering Workshops (ISSREW)*. 2023, pp. 164–165. DOI: [10.1109/ISSREW60843.2023.00068](https://doi.org/10.1109/ISSREW60843.2023.00068).
- [46] Da He, Xiaoyu Shang, and Jiajia Luo. "Adherent mist and raindrop removal from a single image using attentive convolutional network". In: *Neurocomputing* 505 (2022), pp. 178–187. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2022.07.032>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231222008918>.
- [47] intermodalics. *NVIDIA Isaac ROS In-Depth: CUVSLAM and the DP3.1 Release*. Accessed: June 15, 2024. 2023. URL: <https://www.intermodalics.ai/blog/nvidia-isaac-ros-in-depth-cuvslam-and-the-dp3-1-release>.
- [48] Raúl Mur-Artal and Juan D. Tardós. *KITTI Vision Benchmark Suite - Odometry Evaluation*. https://www.cvlibs.net/datasets/kitti/eval_odometry_detail.php?&result=162e869388fd5fd235b88dac7718149241fdc255. Accessed: June 21, 2024. 2024.
- [49] Igor Cvišić, Ivan Marković, and Ivan Petrović. *KITTI Vision Benchmark Suite - Odometry Evaluation*. https://www.cvlibs.net/datasets/kitti/eval_odometry_detail.php?&result=988dcafb412eaaeae1b54b12cb83ca2618dae2c4. Accessed: June 21, 2024. 2016.

List of Aids

This thesis was written with the support of DeepL Write, an AI-based writing assistant, to help reformulate phrases.

Appendix A

Appendix

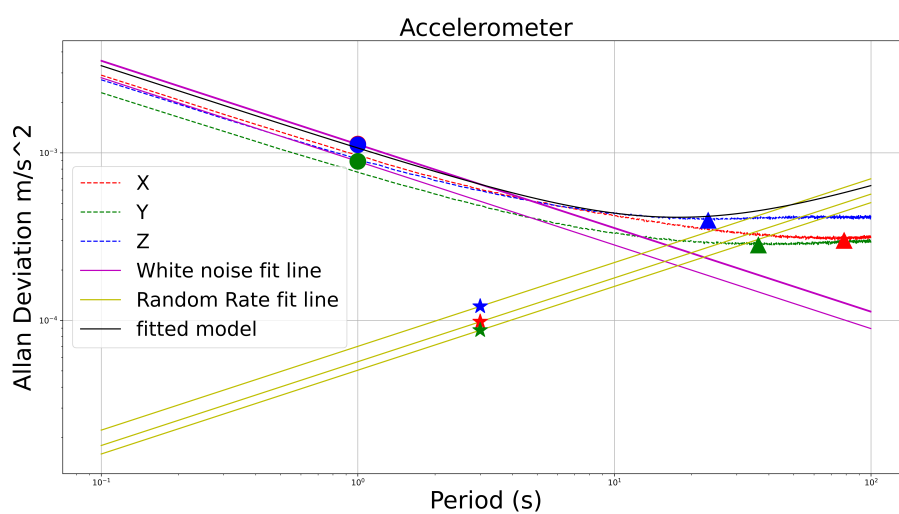


FIGURE A.1: Allan Deviation: Accelerometer Data ZED 2i

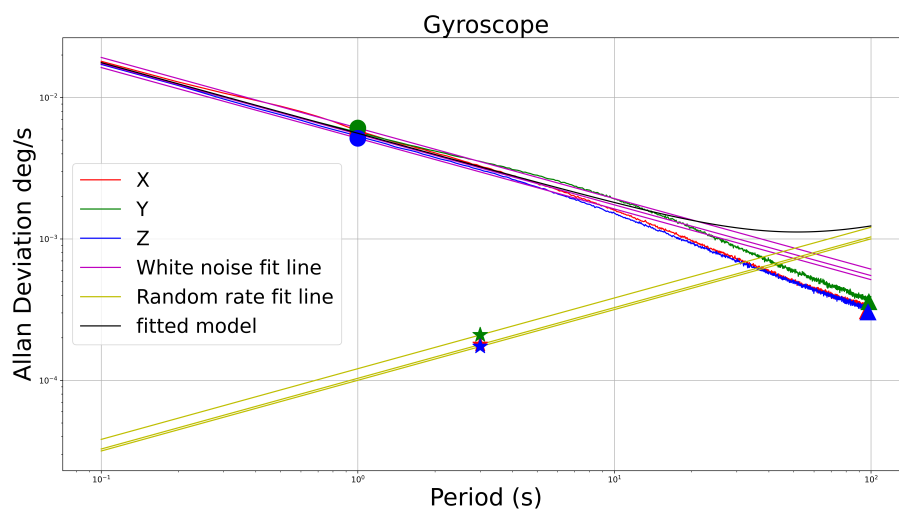


FIGURE A.2: Allan Deviation: Gyroscope Data ZED 2i

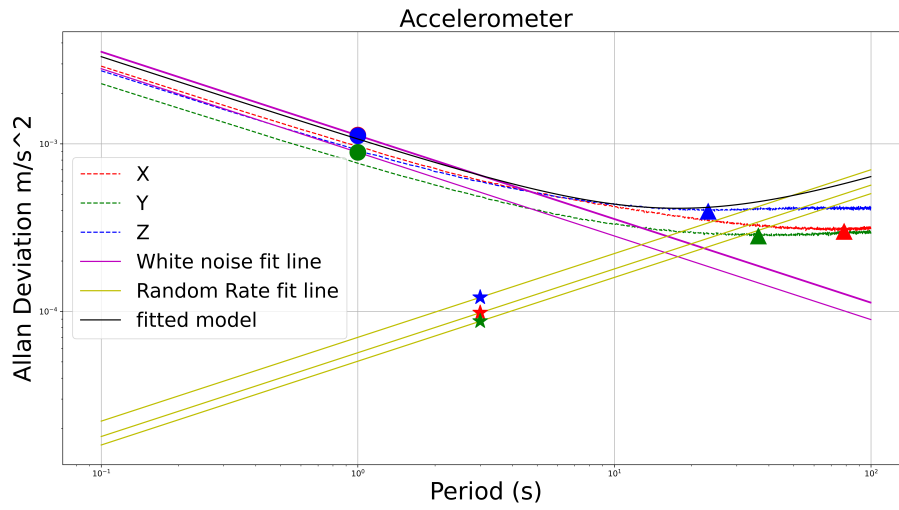


FIGURE A.3: Allan Deviation: Raw Accelerometer Data ZED 2i

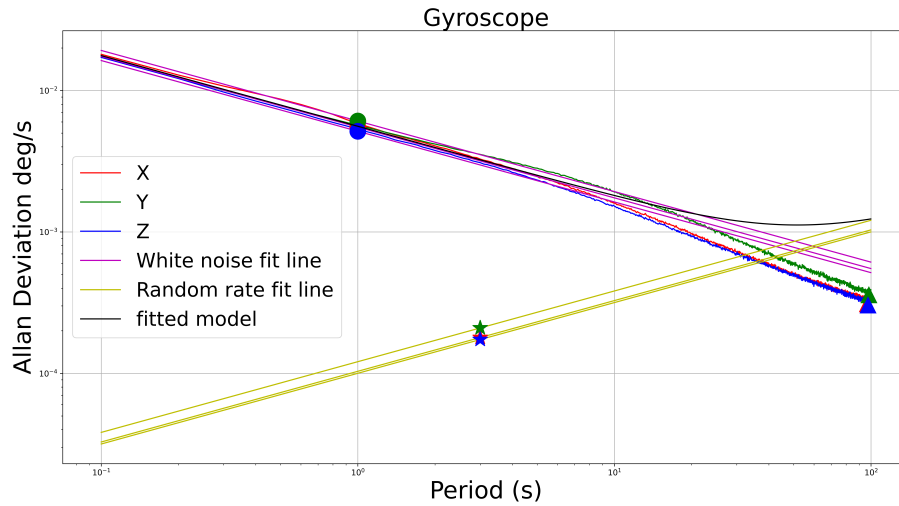


FIGURE A.4: Allan Deviation: Raw Gyroscope Data ZED 2i

TABLE A.1: Accelerometer Specifications

Specification	Value
X Velocity Random Walk	0.06767 m/s/√hr
Y Velocity Random Walk	0.05336 m/s/√hr
Z Velocity Random Walk	0.06703 m/s/√hr
X Bias Instability	3886.10525 m/hr ²
Y Bias Instability	3648.37478 m/hr ²
Z Bias Instability	5123.08800 m/hr ²
X Accel Random Walk	0.00010 m/s ² /√s
Y Accel Random Walk	0.00009 m/s ² /√s
Z Accel Random Walk	0.00012 m/s ² /√s

TABLE A.2: Gyroscope Specifications

Specification	Value
X Angle Random Walk	0.33064 deg/ $\sqrt{\text{hr}}$
Y Angle Random Walk	0.36550 deg/ $\sqrt{\text{hr}}$
Z Angle Random Walk	0.30938 deg/ $\sqrt{\text{hr}}$
X Bias Instability	1.11593 deg/hr
Y Bias Instability	1.29324 deg/hr
Z Bias Instability	1.09518 deg/hr
X Rate Random Walk	0.00018 deg/s/ $\sqrt{\text{s}}$
Y Rate Random Walk	0.00021 deg/s/ $\sqrt{\text{s}}$
Z Rate Random Walk	0.00017 deg/s/ $\sqrt{\text{s}}$

TABLE A.3: Raw Accelerometer Specifications

Specification	Value
X Velocity Random Walk	0.06767 m/s/ $\sqrt{\text{hr}}$
Y Velocity Random Walk	0.05336 m/s/ $\sqrt{\text{hr}}$
Z Velocity Random Walk	0.06704 m/s/ $\sqrt{\text{hr}}$
X Bias Instability	3886.41110 m/hr ²
Y Bias Instability	3648.30998 m/hr ²
Z Bias Instability	5122.19246 m/hr ²
X Accel Random Walk	0.00010 m/s ² / $\sqrt{\text{s}}$
Y Accel Random Walk	0.00009 m/s ² / $\sqrt{\text{s}}$
Z Accel Random Walk	0.00012 m/s ² / $\sqrt{\text{s}}$

TABLE A.4: Raw Gyroscope Specifications

Specification	Value
X Angle Random Walk	0.33067 deg/ $\sqrt{\text{hr}}$
Y Angle Random Walk	0.36552 deg/ $\sqrt{\text{hr}}$
Z Angle Random Walk	0.30937 deg/ $\sqrt{\text{hr}}$
X Bias Instability	1.11606 deg/hr
Y Bias Instability	1.29233 deg/hr
Z Bias Instability	1.09486 deg/hr
X Rate Random Walk	0.00018 deg/s/ $\sqrt{\text{s}}$
Y Rate Random Walk	0.00021 deg/s/ $\sqrt{\text{s}}$
Z Rate Random Walk	0.00017 deg/s/ $\sqrt{\text{s}}$

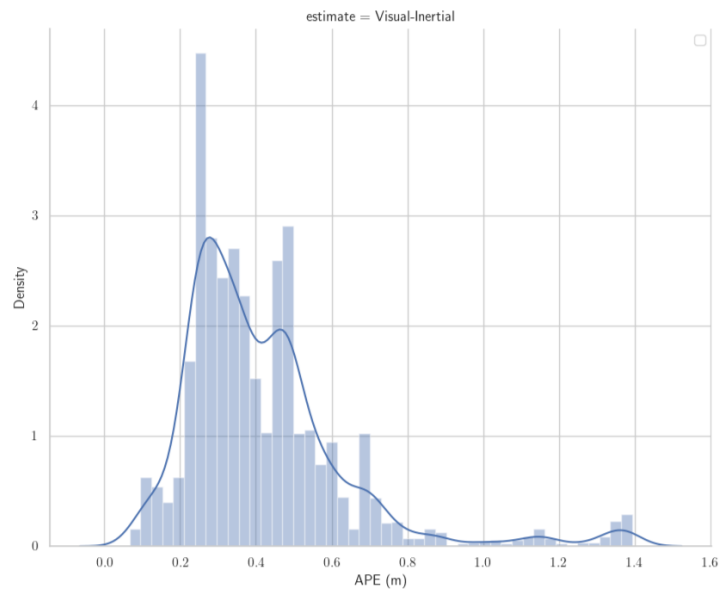


FIGURE A.5: Histogram visual-inertial from classroom run with outliers from the tracking failure

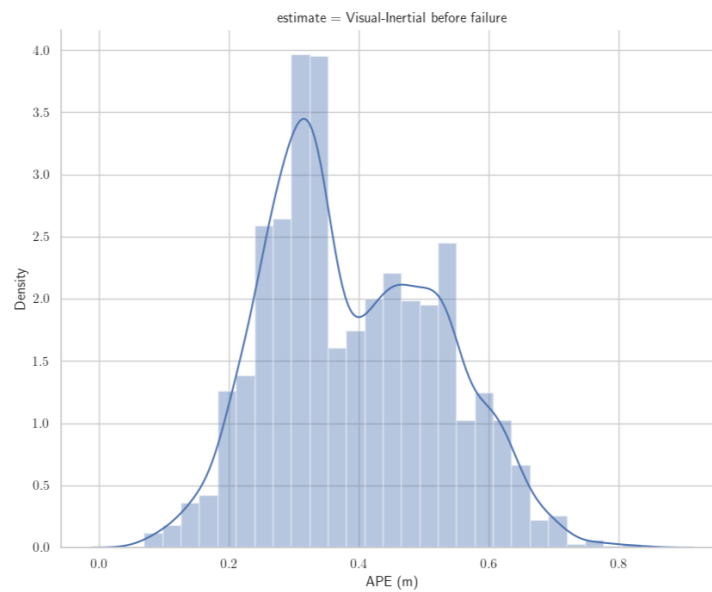


FIGURE A.6: Histogram visual-inertial from classroom run before failure (without the outliers)

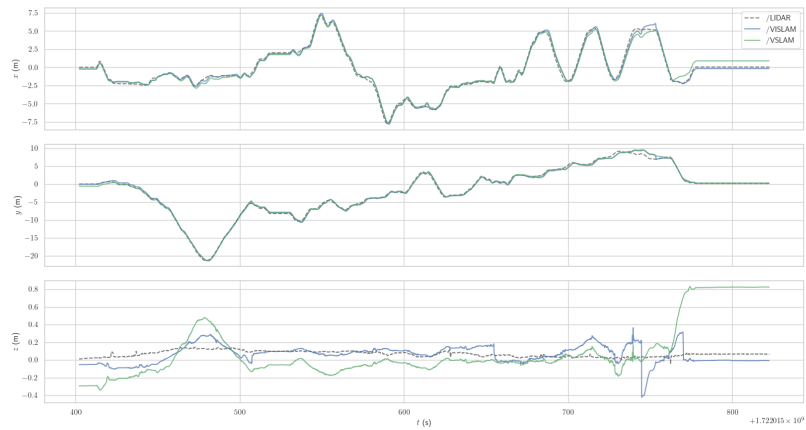


FIGURE A.7: Testing environment XYZ from the classroom run

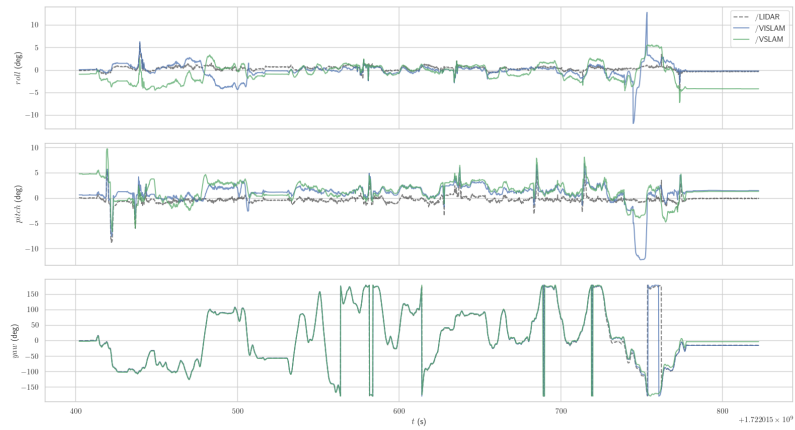


FIGURE A.8: Testing environment RPY from the classroom run

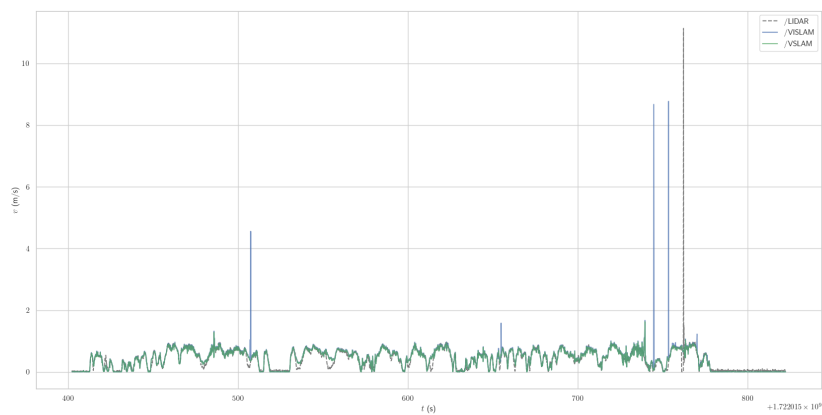


FIGURE A.9: Testing environment speed from the classroom run

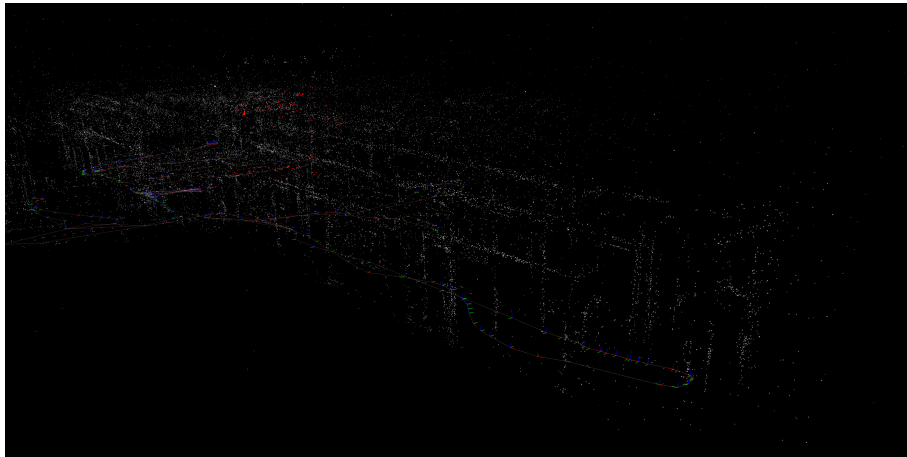


FIGURE A.10: Visual SLAM mapping corridor side view

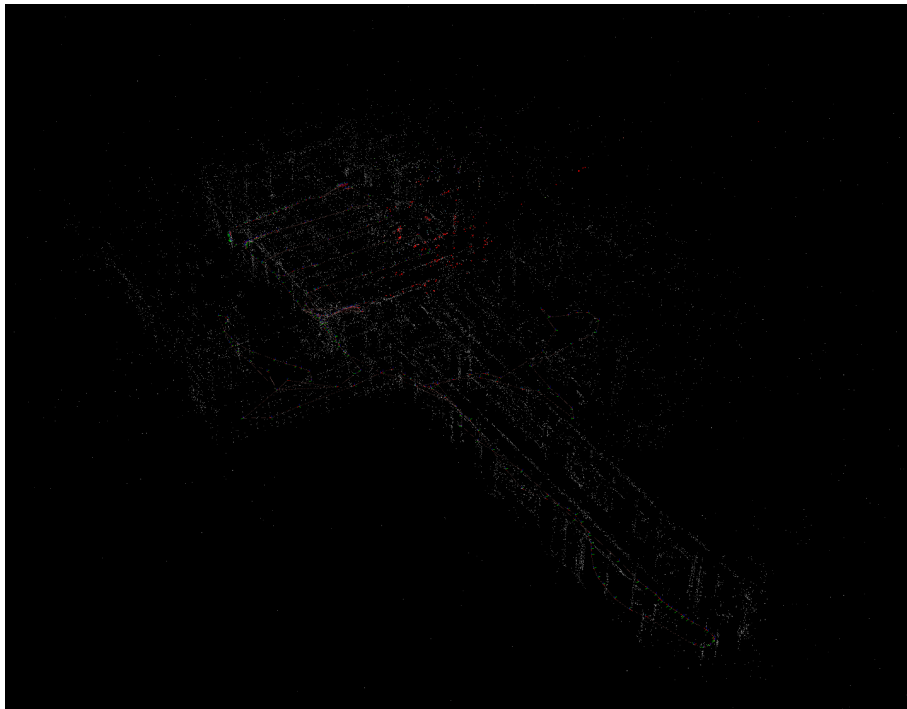


FIGURE A.11: Visual SLAM mapping side view

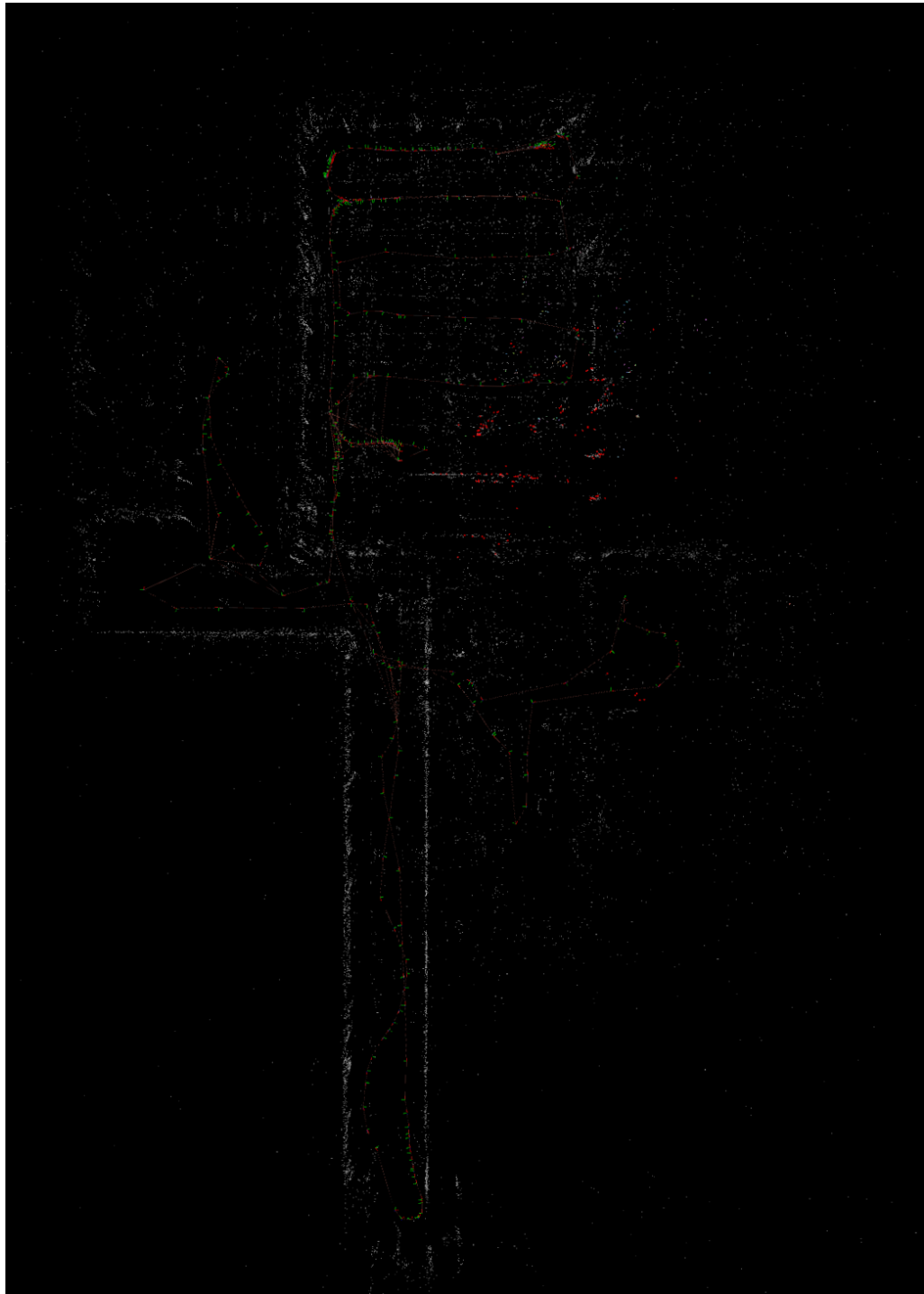


FIGURE A.12: Mapping from the whole classroom run as displayed in RVIZ

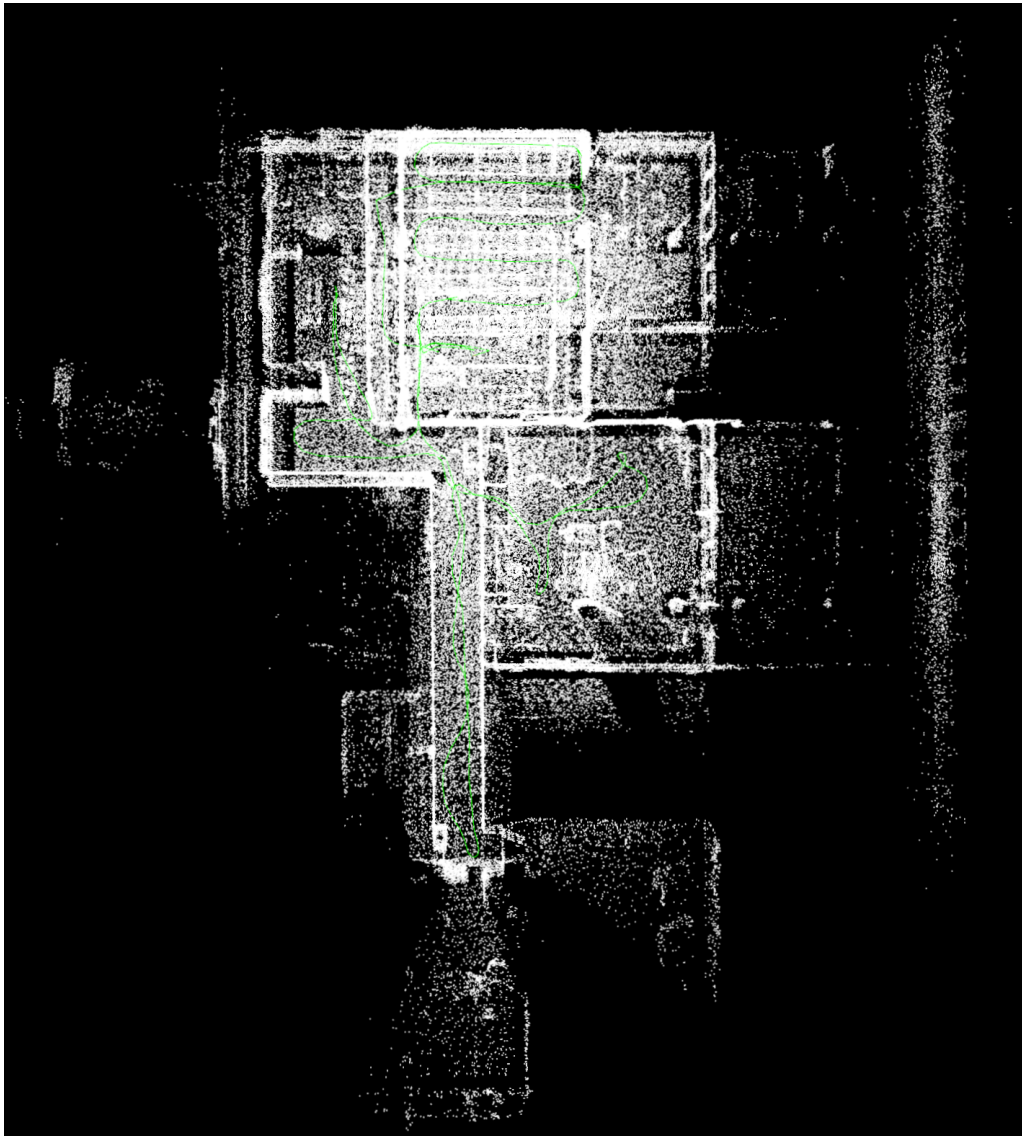


FIGURE A.13: LIDAR map top view taken from the classroom run

TABLE A.5: NVIDIA vSLAM on KITTI dataset APE sequences 11-15 (calculated)

Sequence Name	RMSE (m)	Mean (m)	Median (m)	σ (m)	Min (m)	Max (m)
11	0.895	0.680	0.494	0.582	0.000	3.156
12	11.881	11.757	11.397	1.716	0.000	15.397
13	3.693	3.401	3.622	1.439	0.000	6.413
14	0.531	0.412	0.404	0.336	0.015	1.022
15	2.189	1.852	1.522	1.168	0.000	4.189

TABLE A.6: ORB-SLAM2 on KITTI dataset APE sequences 11-15 (calculated)

Sequence Name	RMSE (m)	Mean (m)	Median (m)	σ (m)	Min (m)	Max (m)
11	2.904	1.776	0.413	2.297	0.000	8.582
12	10.247	8.577	7.062	5.606	0.000	19.221
13	4.026	3.708	3.645	1.567	0.000	7.259
14	0.450	0.298	0.105	0.337	0.000	0.968
15	1.720	1.499	1.438	0.845	0.000	3.318

TABLE A.7: SOFT2 on KITTI dataset APE sequences 11-15 (calculated)

Sequence Name	RMSE (m)	Mean (m)	Median (m)	σ (m)	Min (m)	Max (m)
11	0.549	0.487	0.526	0.252	0.000	0.911
12	4.132	3.766	4.526	1.700	0.000	5.381
13	3.517	3.238	3.193	1.374	0.000	6.218
14	0.258	0.226	0.203	0.123	0.000	0.508
15	1.193	1.000	0.817	0.652	0.000	2.256

Appendix B

Attachments

1. CAD

- Mount ZED and Livox
Description: CAD model of the ZED 2i and Livox Mid-360 mounted together.
File: ZED&LIVOX.step
Date and Time: 06/08/2024 10:10
- Tilted Mount ZED and Livox
Description: CAD model of the tilted ZED 2i and Livox Mid-360 mounted together.
File: ZED&LIVOX_30Degrees.step
Date and Time: 06/08/2024 10:10

2. KITTI_calculations

- data_nvidia
 - Sequence 11
Description: KITTI Sequence 11 from Elbrus aka. NVIDIA VSLAM [13].
File: 11.txt
Date and Time: 07/08/2024 20:52
 - Sequence 12
Description: KITTI Sequence 12 from Elbrus aka. NVIDIA VSLAM [13].
File: 12.txt
Date and Time: 07/08/2024 20:52
 - Sequence 13
Description: KITTI Sequence 13 from Elbrus aka. NVIDIA VSLAM [13].
File: 13.txt
Date and Time: 07/08/2024 20:53
 - Sequence 14
Description: KITTI Sequence 14 from Elbrus aka. NVIDIA VSLAM [13].
File: 14.txt
Date and Time: 07/08/2024 20:53
 - Sequence 15
Description: KITTI Sequence 15 from Elbrus aka. NVIDIA VSLAM [13].
File: 15.txt
Date and Time: 07/08/2024 20:53
- data_orb2
 - Sequence 11
Description: KITTI Sequence 11 from ORB-SLAM2 [48].

- File: 11.txt
Date and Time: 07/08/2024 22:10
 - Sequence 12
Description: KITTI Sequence 12 from ORB-SLAM2 [48].
File: 12.txt
Date and Time: 07/08/2024 22:10
 - Sequence 13
Description: KITTI Sequence 13 from ORB-SLAM2 [48].
File: 13.txt
Date and Time: 07/08/2024 22:10
 - Sequence 14
Description: KITTI Sequence 14 from ORB-SLAM2 [48].
File: 14.txt
Date and Time: 07/08/2024 22:10
 - Sequence 15
Description: KITTI Sequence 15 from ORB-SLAM2 [48].
File: 15.txt
Date and Time: 07/08/2024 22:10
 - data_soft2
 - Sequence 11
Description: KITTI Sequence 11 from SOFT2 [49].
File: 11.txt
Date and Time: 07/08/2024 22:15
 - Sequence 12
Description: KITTI Sequence 12 from SOFT2 [49].
File: 12.txt
Date and Time: 07/08/2024 22:15
 - Sequence 13
Description: KITTI Sequence 13 from SOFT2 [49].
File: 13.txt
Date and Time: 07/08/2024 22:15
 - Sequence 14
Description: KITTI Sequence 14 from SOFT2 [49].
File: 14.txt
Date and Time: 07/08/2024 22:15
 - Sequence 15
Description: KITTI Sequence 15 from SOFT2 [49].
File: 15.txt
Date and Time: 07/08/2024 22:15
 - Calculate APE
Description: Calculate the APE of the published KITTI data.
File: calculate.ipynb
Date and Time: 07/08/2024 22:16
3. Github repository
Description: Cloned github repo with all the code necessary to run the SLAM.

File: Code.zip

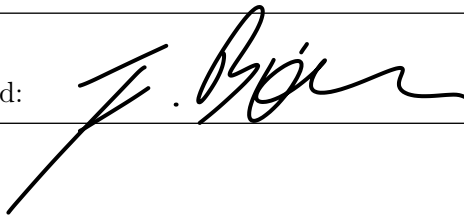
Date and Time: 09/08/2024 02:29

Declaration of Authorship

I hereby declare that I have completed this work independently and have not used any other means than the specified sources and permitted resources, including the use of artificial intelligence systems. I have indicated all passages which have been derived from outside sources either literally or in their general substance and am aware that, should I have failed to do so, the University Administration is entitled to revoke the qualification or title granted to me on the basis of my work.

Date: 09/08/2024

Signed:

A handwritten signature in black ink, appearing to be 'J. B. ...', written over a horizontal line.