Super Odometry: A Robust LiDAR-Visual-Inertial Estimator for Challenging Environments

Shibo Zhao

CMU-RI-TR-24-49

May 04, 2024

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Sebastian Scherer, *chair*
Michael Kaess
Shubham Tulsiani
Dan McGann

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my parents and grandparents.

Abstract

We propose Super Odometry, a high-precision multi-modal sensor fusion framework, providing a simple but effective way to fuse multiple sensors such as LiDAR, camera, and IMU sensors and achieve robust state estimation in perceptually-degraded environments. Different from traditional sensor-fusion methods, Super Odometry employs an IMU-centric data processing pipeline, which combines the advantages of loosely coupled methods with tightly coupled methods and recovers motion in a coarse-to-fine manner. The proposed framework is composed of three parts: IMU odometry, Visual-inertial odometry, and LiDAR-inertial odometry. The Visual-inertial odometry and LiDAR-inertial odometry provide the pose prior to constrain the IMU bias and receive the motion prediction from IMU odometry. To ensure high performance in real-time, we apply a dynamic octree that only consumes 10% of the running time compared with a static KD-tree. The proposed system was deployed on drones and ground robots, as part of Team Explorer's effort to the DARPA Subterranean Challenge where the team won 1^{st} and 2^{nd} place in the Tunnel and Urban Circuits [1], respectively. To benefit the entire robotics community, we release the SubT-MRS dataset, which is the first real-world dataset that specifically addresses failure scenarios of SLAM by incorporating a variety of degraded conditions, multiple robotic platforms, and diverse sets of multi-modal sensors.

[1]https://www.subtchallenge.com/results.html

Acknowledgments

I would like to express my heartfelt gratitude to my advisor, Professor Sebastian Scherer. I consider myself fortunate to have been selected as his student. Over the past three years, he has consistently supported me and provided invaluable guidance. I am deeply appreciative of the academic freedom he has fostered at the Airlab, as well as the resources he has generously provided to support my research.

I also want to extend my gratitude to Professor Michael Kaess and Professor Shubham Tulsiani. I gained valuable insights into SLAM from Professor Michael, while Professor Shubham introduced me to the field of learning for 3D. His course was my first exposure to using learning techniques to create a 3D world, and I greatly appreciate his exceptional teaching skills. I frequently discussed ideas with both professors, and they consistently provided insightful feedback on my experiments and valuable suggestions for my research. Thank you again for your constructive advice!

I am also deeply grateful to Professor Chen Wang, Professor Wen Shan, and Professor Ji Zhang for their suggestions on my experiments. Your insightful feedback has been instrumental in my growth as a researcher.

I am profoundly grateful to my collaborators Yuheng, Honghao, Parv, Lucas, Henry, Yuanjun, Tianhao, Jay, Damanpreet, Rushan, Haoxiang, Yuchen, Ian, John, Andrew, Xucan, Mansi, and Namya. Your support has been invaluable throughout this journey. Despite the challenges we faced during our research, I am confident that our collective efforts will lead to outstanding results. I deeply appreciate your continuous encouragement.

I dedicated nearly three years to completing this thesis, experiencing both highs and lows along the way. I remember nights when I would return home, anxious about the experiments for the following days. Despite the challenges, I am incredibly fortunate to have had friends like Yuheng, Bowen, Huiyi and Shuangcheng by my side. Your support has been invaluable, providing encouragement even during the darkest times. Your belief in me helped drive me towards achieving outstanding results. I also want to express my heartfelt gratitude to my parents for their unwavering support during the most challenging moments. Thank you all!

Funding

This research was sponsored by DARPA ($HR00111820044$). Content is not endorsed by and does not necessarily reflect the position or policy of the government or our sponsors

Contents

When this dissertation is viewed as a PDF, the page header is a link to this Table of Contents.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Ensuring robust state estimation and perception stands as one of the foremost challenges in deploying robots for safe autonomy in real-world scenarios. To address this formidable challenge, roboticists propose various solutions. To uncover a gap in current SLAM systems regarding resilient performance, we will explore two critical reasons why current SLAM systems lack robustness: 1. Limitation on Existing Sensor Fusion Methods; 2. Limitation on Existing SLAM Datasets;

1.1.1 Limitation on Existing Sensor Fusion Methods

Multi-modal sensor fusion is essential for autonomous robots to fulfill complex and dangerous missions such as perception in subterranean environments, industrial inspection, and search and rescue. In these GPS-denied scenarios, darkness, airborne obscurant conditions (dust, fog, and smoke), and lack of perceptual features are major challenges that currently hinder us from employing robotic systems for long-term autonomy. To localize in such environments, LiDAR-based odometry[76, 93, 96] seems to be a suitable choice for robots since LiDAR sensors can provide high-fidelity 3D measurements. However, in structure-less environments such as long tunnels or the presence of obscurants (e.g. fog, dust, smoke), LiDAR-based approaches fail to provide reliable motion estimation because of degeneracies and outliers. To handle these

situations, integrating with additional sensors, cameras in particular, is also required. In spite of this, the use cases of the visual camera are limited to well-illuminated environments. Therefore, we argue that both LiDAR-based [76, 93, 96], visual-based [61, 97] or LiDAR-visual-based [22, 32] SLAM methods are not the optimal choices in challenging environments. Since these sensors are environment-dependent, it is

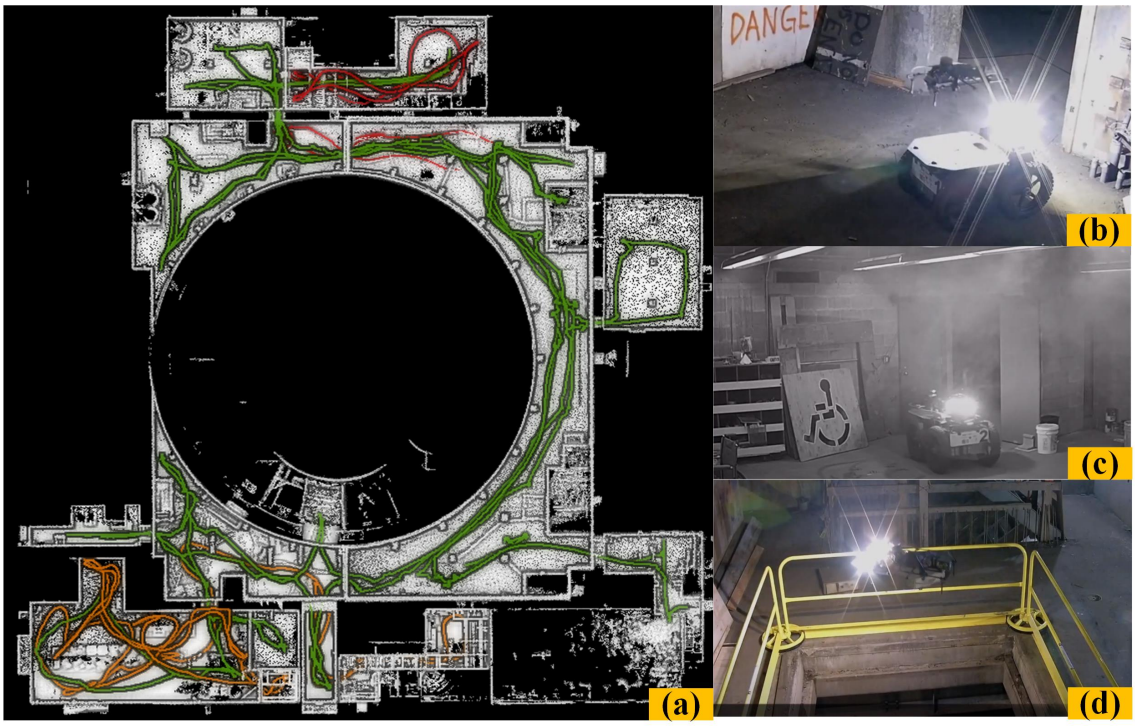


Figure 1.1: The Performance of Super Odometry in the DARPA Subterranean Challenge Alpha Course, which is a challenging environment including darkness (b), heavy fog (c), and vertical shaft(d). (a) represents the mapping result of Super Odometry in Alpha Course reconstructed by multi robots (UGV1, UAV1 and UAV2); Green, orange, and red lines are the estimated trajectory of UGV1, UAV1, and UAV2 respectively.

difficult to achieve robust performance if they are used as dominant sensors to build a SLAM system.

In the last decade, multi-modal sensor fusion has gained great popularity and can be classified as either loosely or tightly coupled methods. As shown in Table 1.1, Loosely coupled methods [57, 93, 96] have been preferred more because of their simplicity, extendibility, and low computational expense. In contrast, tightly coupled

methods [76] have proved advantageous for their accuracy and robust performance. However, tightly coupled methods are usually difficult to extend other sensors and achieve a switching scheme between sensing modalities to cope with various environments. In addition, tightly coupled methods may be vulnerable to cope with potential sensor failures since most of them only use a single estimation engine [57]. In comparison, loosely coupled methods distribute the sensor failure risks between several estimation engines and should have more robust performance. However, since most of the loosely coupled methods adopt frame-to-frame estimation, they still have limitations on accuracy and robustness.

To solve all the above challenges, we present Super Odometry [98], an efficient and extremely robust odometry system that tackles all-weather environments leveraging a novel *adaptive IMU-centric sensor fusion* framework, which supports multi-sensor, multi-locomotion, multi-robot and is resilient to multi-degradation and system failures.

1.1.2 Limitation on Existing SLAM Datasets

In addition to algorithmic enhancements, we also emphasize the importance of high-quality datasets. We found existing solutions often struggle in challenging conditions. This is because these algorithms are typically developed and evaluated using datasets from controlled environments, such as KITTI [30] and EuRoc-MAV [7], which fail to capture the complexities of real-world scenarios, hindering the development of robust SLAM solutions. To bridge this gap and push SLAM towards all-weather environments, we present an extremely challenging dataset, SubT-MRS [99], including scenarios featuring various sensor degradation, aggressive locomotions, and extreme-weather conditions.

1.2 Related Work

In this section, we will present an overview of existing sensor fusion methods in Section 1.2.1 and SLAM datasets in Section 1.2.2.

Table 1.1: The comparison of multi-modal sensor fusion algorithm

Methods	Accuracy	Resilience	Computation	Extensibility
Loosely coupled[1][2]	✗	✗	✓	✓
Tightly coupled [3][4][5][6]	✓	✓	✗	✗
Combination (Ours)	✓	✓	✓	✓

1.2.1 Efforts on Sensor Fusion

In recent years, there are a number of efforts have been made for the LiDAR-visual-inertial estimator for robust sensor fusion. These efforts typically fall into two categories: loosely coupled methods and tightly coupled methods.

Loosely Coupled LiDAR-Visual-Inertial Odometry V-LOAM algorithm [94] is proposed that adopts sequential data processing pipeline and used Visual-inertial odometry to provide motion prediction for LiDAR scan matching. However, since this pipeline still performed frame-to-frame motion estimation and the current estimate is totally based on the last estimate, it is difficult to achieve a recovery mechanism if the last estimate goes wrong. To solve this problem, Super Odometry adopts factor graph optimization and the current estimates are based on historical frames within a sliding window. Therefore, Super Odometry is fail-safe to single-point failures. To improve the robustness, some works incorporate other constraints to LiDAR-visual-inertial estimator and obtain very promising results such as adding thermal-inertial prior [37], leg odometry prior [10] or loop closure [65]. However, since these methods still performed frame-to-frame motion estimation, they have similar limitations with V-LOAM.

Tightly Coupled LiDAR-Visual-Inertial Odometry To perform joint state optimization, VIL-SLAM [77] is proposed to perform joint state optimization, it is effectively fused Visual-inertial odometry and LiDAR scan matching within a graph optimization. LIC-Fusion [102] takes a step further by introducing tightly coupled LiDAR edge features, sparse visual features, and plane features within the MSCKF framework [52]. To leverage 3D primitives effectively, a unified multi-sensor odometry approach [88] has been introduced. This method aims to jointly optimize 3D primitives, including lines and planes, enhancing the overall odometry performance.

However, since the measurements of these methods are highly tightly coupled, they are difficult to extend to other sensors. In addition, since these methods usually use a single state estimation engine, they may have a higher failure rate when the sensors are damaged.

Method Highlights In contrast with previous works, the advantages of proposed Super Odometry are:

- **Robustness and Extendibility:** Our IMU-centric sensor fusion architecture (see Fig.2) enables us to achieve high accuracy and operate with a low failure rate, since the IMU sensor is environment-independent. The system includes fail-safe mechanisms and provides an easier and flexible way to fuse multiple sensors such as GPS, wheel odometry, and etc.
- **Simple but Effective:** Super Odometry avoids complex derivations to fuse multiple sensors because the IMU has a simple and accurate probabilistic model. As long as other sensors can provide relative pose prior to constrain the IMU preintegration factor, these sensors will be fused into the system successfully. In particular, it has been extensively evaluated in challenging environments as part of Team Explorer's effort to the DARPA Subterranean Challenge.
- **Low CPU Usage and High Real-time Performance:** Since Super Odometry does not combine all sensor data into a full-blown factor graph, instead, it divided the big factor graph into several "sub-factor-graph" and each "sub-factor-graph" receives the prediction from an IMU pre-integration factor (see Fig.4.1). Therefore, the motion from each odometry factor is recovered from coarse to fine manner in parallel, which significantly improves real-time performance. Also, Super Odometry adopts a dynamic octree to organize the 3D points, which makes scan-matching very efficient.

1.2.2 Efforts on SLAM Datasets

Developing and testing SLAM systems in extreme environmental conditions is crucial for real-world reliability. However, existing SLAM datasets are often limited to single, controlled environments. For example, TUM-VI [72] and UMA-

VI [101] datasets offer indoor-outdoor visual-inertial data, but face challenges like varying illumination. EuRoC MAV [7] and UZH-FPV [19] drone racing datasets provide data on aggressive drone movements but lack variability in lighting conditions. The KITTI dataset [31] is renowned for outdoor LiDAR-Visual-Inertial data, while the Hilti SLAM dataset [35] includes dynamic lighting but prioritizes accuracy over environmental variability. TartanAir [85] covers diverse environments but is a simulation dataset, introducing a sim-to-real gap.

In contrast, we present the SubT-MRS dataset [99], featuring more than 30 challenging scenarios. These scenarios encompass a wide range of difficulties, including extreme lighting conditions such as darkness and overexposure, as well as geometrically degraded environments like featureless corridors and staircases. Moreover, the dataset incorporates perceptual challenges induced by smoke, fog, dust, and other extreme weather conditions.

1.3 Contribution

Motivated by the discussion above, we proposed an IMU-centric SLAM system combing the advantages of tightly-coupled methods with loosely coupled methods. The system design follows a key insight: An inertial measurement unit (IMU) produces smooth measurements with noise but little outliers which can make its estimate very accurate as long as the bias drift can be well-constrained by other sensors. Therefore, Super Odometry is designed around the IMU as the primary sensor. The main contributions of our work are as follows:

- We propose the first IMU-centric sensor fusion pipeline that enables accurate real-time state estimation in extreme and perceptually challenging environments.
- The proposed pipeline combines the advantages of tightly-coupled methods with loosely-coupled methods, which provides a simple but effective way to fuse multiple sensors.
- We propose to use an efficient organization of 3D points (dynamic octree) which significantly improves the real-time performance of scan matching.
- The proposed approach was deployed on multiple physical systems including drones and ground robots and has been extensively evaluated in various chal-

lenging scenarios with aggressive motion, low light, long corridors, and even in heavy dust environments.

- Also, to benefit the robotics community, we release SubT-MRS, an extremely challenging dataset. To the best of our knowledge, SubT-MRS is the first real-world dataset that specifically addresses failure scenarios of SLAM by incorporating a variety of degraded conditions, multiple robotic platforms, and diverse sets of multi-modal sensors.

Chapter 2

Background

In this chapter, we provide the theoretical background information essential for comprehending the methods proposed in this work. Section 2.1 elucidates the fundamental process of LiDAR Odometry, while Section 2.2 delineates the basic procedure of Visual Odometry. Furthermore, Section 2.3 offers an introduction to IMU and its kinematic model. Lastly, Section 2.4 delves into the concepts of loosely coupled and tightly coupled systems, highlighting their distinctions.

2.1 Introduction for LiDAR Odometry

To derive a global pose estimate $T_t \in \mathrm{SE}(3)$ at time t from each 3D scan represented as a local, egocentric point cloud $\mathcal{P} = \{p_i \mid p_i \in \mathbb{R}^3\}$, we follow a structured process consisting of three steps for LiDAR Odometry.

In Section 2.1.1, we delve into motion prediction and scan distortion, commonly known as deskewing, to mitigate distortions in the 3D data resulting from sensor movement during scanning. In Section 2.1.3, we proceed by downsampling the current scan and introducing various classical LiDAR correspondence estimation methods. Subsequently, in Section 2.1.4, we present LiDAR scan registration techniques such as scan-to-scan and scan-to-map.

2.1.1 Motion Prediction and Scan Undistortion

Scan undistortion is an important component of LiDAR odometry. A LiDAR points produces motion distortion caused by sensor movement during scanning. The utilization of distorted point clouds in matching or mapping substantially impacts the accuracy of localization and mapping. To correct such distortion, compensation methods such as the constant-velocity model, continuous-time trajectory optimization and using IMU measurements are often necessary.

Constant-velocity Model In the constant velocity model, it's assumed that the robot maintains the same translational and rotational velocities from the previous time step. Since the model doesn't require any additional sensors to remove the distortion, it is widely used in the LiDAR SLAM systems [18] [83]. For further details, readers can refer to Probabilistic Robotics [81].

Continuous-time Trajectory Another widely used approach for motion compensation is continuous-time trajectory optimization techniques, such as those utilizing B-Splines [20] [47] and the Gaussian process model[3]. Continuous-time trajectories allow pose querying at any time instants without interpolation. It can be easily used to remove the distortion of each individual point. However, the conventional continuous-time trajectory optimization is highly time-consuming and often implemented offline in LiDAR SLAM systems [47].

IMU based Motion Compensation Utilizing IMU measurements offers another effective strategy for motion compensation [75] [98]. These techniques integrate the LiDAR pose with IMU data within a frame to rectify point distortion. Thanks to the high frequency of IMU measurements (e.g., 200 Hz), IMU-based motion compensation is highly effective for common robot motions. Nonetheless, this method is still susceptible to measurement noise and bias estimation errors.

2.1.2 Point Cloud Subsampling

After correcting distortion in each LiDAR scan, it is common to downsample each scan to achieve faster convergence and higher robustness for scan registration. Down-

sampling typically employs a voxel grid, where each voxel stores only a predetermined number of points. To ensure adaptability to diverse environments, the voxel size can be adjusted dynamically [92] to accommodate varying structures while maintaining computational efficiency.

2.1.3 Correspondence Estimation

After acquiring the downsampled point cloud, LiDAR correspondences are computed. Correspondence estimation in LiDAR processing involves identifying corresponding points between consecutive or overlapping scans, essential for accurate registration and mapping.

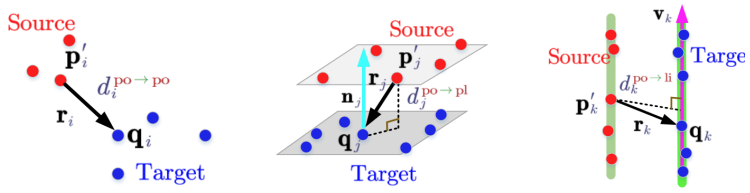
Many point cloud registration techniques follow a coarse-to-fine strategy. In this approach, a coarse registration (typically a global registration) is first applied to find an approximate rigid transformation for a pair of point clouds. Once a coarse transformation is available, a finer registration process takes place, typically employing algorithms like ICP [5], NDT [49] , or their efficient variants, to further refine the transformation. Given that global registration tends to consume considerable computational resources, for the sake of efficiency, the majority of LiDAR odometry systems typically rely on a constant velocity model or other sensors such as IMUs or wheel encoders to provide an initial coarse transformation for subsequent fine registration processes, such as ICP [5] or NDT [49].

ICP and Variants ICP algorithm [5] is the most widely adopted method for pairwise fine registration. Given q_i, p_i represent corresponding points in the source and target point clouds respectively. After applying certain rotation R and translation t of on source point cloud, the distance error r_i between source points and target points can be calculated as:

$$r_i = q_i - p_i' = q_i - (Rp_i + t). \tag{2.1}$$

In LiDAR Odometry systems, there are three types of distance metrics commonly used to get the correspondence: point-to-point, point-to-plane, and point-to-line metrics. As shown in Fig 2.2, the point-to-point (plane, line) distance can be

calculated as $d_i^{po \to po} = \|r_i\|_2$, $d_j^{po \to pl} = r_j \cdot n_j$, and $d_k^{po \to li} = \|r_k \times v_k\|_2$ respectively from the residual, normal and primary vector (r, n, v).



(a) point to point (b) point to plane (c) point to line

Figure 2.1: Overview of distance error metrics [58]

Many variations following the above concept are proposed to get better correspondence. These include integrating multi-distance metrics into the ICP process[58], incorporating continuous-time considerations into ICP [17], introducing adaptive thresholds in ICP [83]. and distribution-to-distribution matching approaches such as GICP [73] and VGICP [39].

NDT and Variants The NDT [6] splits an input point cloud into a set of voxels and fits a normal distribution to the points in each voxel. Instead of costly finding the nearest neighbor association, it takes advantage of the voxelization approach and achieves a distribution-to-distribution matching manner. Several enhancements stemming from NDT have been explored, including improvements in accuracy [51], and robustness against hyper-parameter alterations, such as multi-resolution techniques [79] and trilinear voxel smoothing [50].

Feature Based Methods LOAM [93] provides a feature-based approach to scan registration algorithms. It utilizes the curvature of the original point cloud, designating points with high curvature as edge feature points and those with low curvature as plane feature points. For edge features, the point-to-line distance serves as the distance metric, while for plane features, the point-to-plane distance is used as the distance metric. There are some other feature-based methods based on the histogram to find the correspondence such as Point Feature Histogram (FPFH)[68], and Signature of Histograms of OrienTations (SHOT) [69].

2.1.4 Scan Registration

Once we obtain reliable correspondence, how to achieve consistent registration of multiple frames of measurement (LiDAR Scan) since each scan is collectively independent and there are errors in the registration which may result in an inconsistent model. This problem is first introduced by Professor F.Lu Milios [46]. To solve this problem, the researcher starts with scan-to-scan matching [56] [40] to compute relative pose changes in LiDAR SLAM. However, the limitation of scan-to-scan matching is that it quickly accumulates errors. To overcome the accumulation of errors, scan-to-map is commonly used in both 2D LiDAR SLAM [36] and 3D LiDAR SLAM systems [93].

2.2 Introduction for Visual Odometry

In this section, we will summarize the Visual Odometry(VO) pipeline. Visual Odometry (VO) is the method employed to estimate the motion of a robot by only using the input of a single or multiple cameras. It incrementally determines the robot's pose by analyzing the pixel changes within the images.

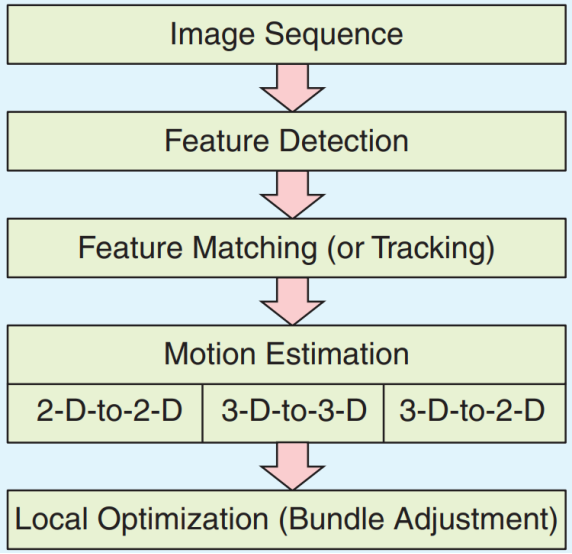


Figure 2.2: Overview of visual odometry pipeline [70]

For each new image, the initial step is **feature detection**. It identifies the most salient features such as corners or distinct points within the images and subsequently

detects and matches 2-D features with those observed in previous frames. Following this, either optical flow [74] or feature descriptors [67] are utilized in the **feature matching** process to establish correspondences between points in different frames.

The third step is **relative motion estimation** between image frames. It depends on whether the correspondences are obtained in 3D space or 2D space. If we only get the correspondence from 2D-2D space, we can only use the essential matrix to recover the motion without scale. If we get the correspondence from 3D-3D space, we will use the triangulation methods. However, If we obtain the correspondence from 2D-3D space, we will use the *perspective from n points* (PnP) method. Finally, an iterative refinement (**bundle adjustment**) can be done over the most recent frames to achieve a more precise estimation of the local trajectory.

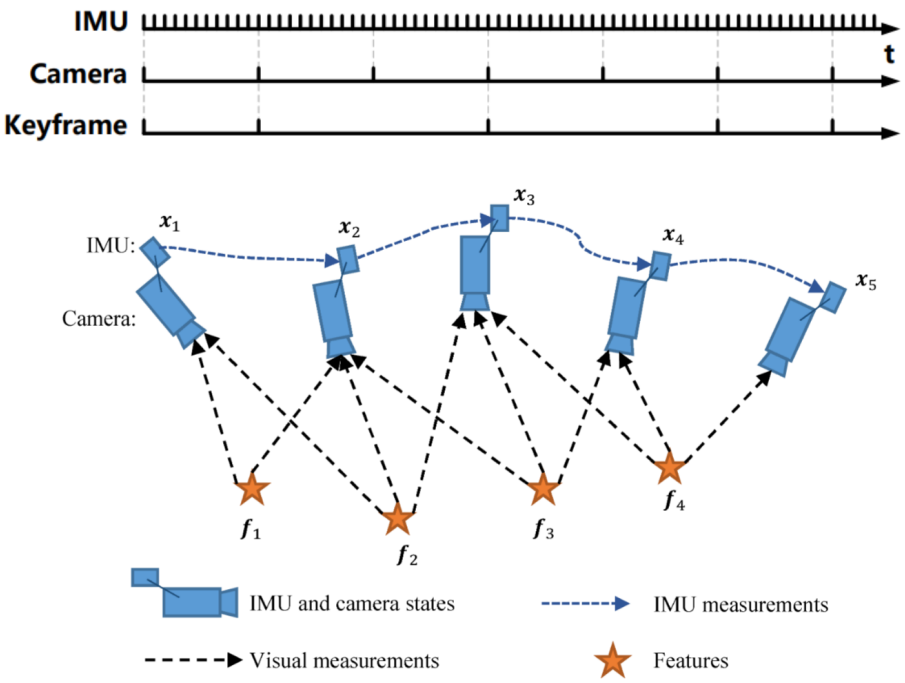


Figure 2.3: Overview of Visual Inertial Systems [89]

2.3 Introduction for IMU and its Kinematic Model

An Inertial Measurement Unit (IMU) is a critical sensor technology used extensively in navigation and motion-tracking applications across various industries, including aerospace, automotive, and consumer electronics. The primary function of an IMU is to measure and report a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers $\boldsymbol{a}_{iW}^{W}$ and gyroscopes $\boldsymbol{\omega}_{iW}^{W}$. It is widely used in Visual inertial system and LiDAR inertial systems. Taking the Visual inertial system as an example, several keyframes with low frequence and IMU measurements with high frequency are maintained in a sliding window. When a new keyframe is added, a local bundle adjustment (BA) will be employed to concurrently optimize the states of both the camera and the IMU, along with the locations of the features depicted in Figure 2.3.

2.3.1 IMU Measurement Model

The IMU measurement model describes how the true acceleration and angular velocity of a device are transformed into the measurements provided by the IMU's sensors. This model accounts for various errors inherent to the sensors, such as bias, noise, and scaling factors. The equations is shown in Eq.2.2 and Eq.2.3:

Gyroscope Measurement:

$$\tilde{\boldsymbol{\omega}}_I = \boldsymbol{\omega}_I + \mathbf{b}_g + \mathbf{n}_g \tag{2.2}$$

Accelerometer Measurement:

$$\tilde{\boldsymbol{a}}_I = \mathbf{R}_W^I \boldsymbol{a}_W + \mathbf{g} + \mathbf{b}_a + \mathbf{n}_a \tag{2.3}$$

Where:

$\tilde{\boldsymbol{\omega}}_I \in \mathbb{R}^3$: Measured angular velocity of IMU frame

$\tilde{\boldsymbol{a}}_I \in \mathbb{R}^3$: Measured acceleration of IMU frame

$\boldsymbol{\omega}_I \in \mathbb{R}^3$: True angular velocity of IMU frame

$\boldsymbol{a}_W \in \mathbb{R}^3$: True acceleration of World frame

$\mathbf{g} \in \mathbb{R}^3$: Gravitational acceleration in IMU frame

$\mathbf{n}_g \in \mathbb{R}^3$: Gyroscope measurement noise (angular random walk)

$\mathbf{n}_a \in \mathbb{R}^3$: Accelerometer measurement noise (velocity random walk)

$\mathbf{R}_I^W \in SO(3)$: Rotation matrix from world frame to IMU frame

We can also convert Hamiltonian unit quaternion to rotation matrix as:

$$q = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_z \end{bmatrix} \in \mathbb{S}^3, \quad \mathbf{R}_I^W = \begin{bmatrix} q_w^2 + q_x^2 - q_y^2 - q_z^2 & 2(q_x q_y - q_w q_z) & 2(q_x q_z + q_w q_y) \\ 2(q_y q_x + q_w q_z) & q_w^2 - q_x^2 + q_y^2 - q_z^2 & 2(q_y q_z - q_w q_x) \\ 2(q_z q_x - q_w q_y) & 2(q_z q_y + q_w q_x) & q_w^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{2.4}$$

For IMU Preintegration, please check the description in section 4.3.1.

2.4 Introduction for Multi-sensor Fusion

To achieve robust performance, most SLAM and odometry systems that integrate multiple sensor types typically employ a mix of LIDAR, cameras, and IMUs, configured in either loosely coupled or tightly coupled modes.

2.4.1 Loosely Coupled & Tightly Coupled Methods

In a loosely coupled system, the measurements from each sensor are processed independently and only fuse the final pose output of each sensor. On the other hand, a tightly coupled system processes all sensor data collectively, leveraging the big optimization and considering different constraints in batch to deliver a more robust pose estimation. The loosely coupled approach is advantageous due to its

minimal computational demand, simple and effective architecture, though it is not very accurate. On the contrary, the tightly coupled system usually takes heavy computational resources, and is challenging to extend new sensors. However, it can provide more precise state estimations.

2.4.2 Factor Graph

The Factor graph is extensively utilized in multi-sensor fusion scenarios. It comprises two key node types: variables, representing estimated parameters, and factors, embodying constraints derived from sensor measurement models. These factors represents relationships between variables. Consider the IMU Odometry factor graph as example, which features three distinct factor types: an optional prior factor, IMU factors, and visual/LiDAR relative pose factors. The prior factor describes a prior distribution concerning IMU states, stemming from the optimization cost minimization in the history sliding window. IMU factors are described by the IMU dynamic model and are expressed through IMU preintegration [27]. Visual/LiDAR pose factors are denoted by relative pose constraints. The comprehensive factor graph outlining the pose graph is depicted in Figure 2.4.

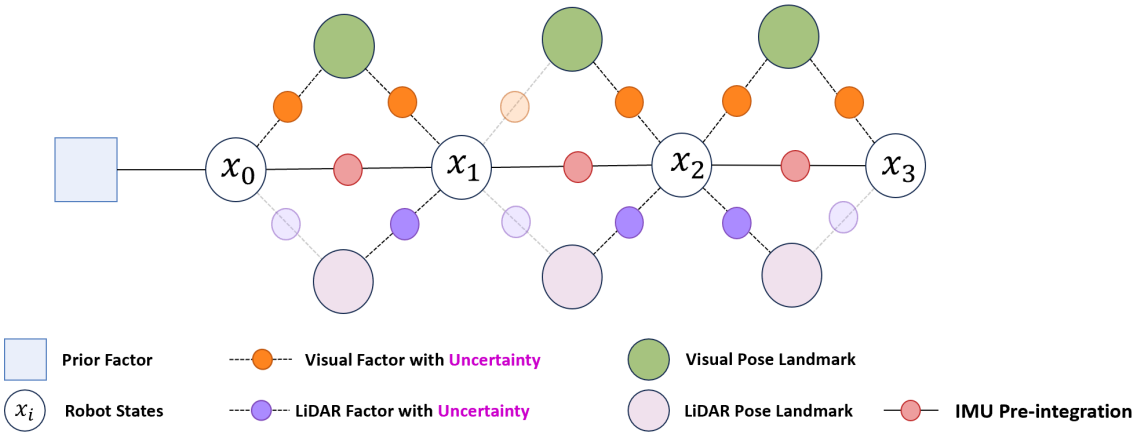


Figure 2.4: IMU Odometry Factor Graph in Multi-Sensor Fusion.

Chapter 3

SubT-MRS Dataset: Pushing SLAM Towards All-degraded Environment

3.1 Motivation

Simultaneous Localization and Mapping (SLAM) is essential in robotics since it provides foundational perception and spatial awareness, and enables machines to understand and interact with the physical world in real-time. Therefore, it has a wide range of applications such as autonomous driving and space exploration. Despite significant advancements in both geometric [9, 62, 75] and data-driven SLAM methods [80, 86], existing solutions remain fragile in challenging conditions. One main reason is that existing SLAM algorithms are often developed and evaluated with datasets from controlled environments [7, 19, 30, 35, 60, 63, 72, 101]. For example, KITTI [30, 35] dataset is mostly collected in sunny weather and EuROC-MAV dataset [7] is collected in small well-lit rooms. These datasets, unfortunately, fail to capture the challenges in real-world scenarios, hindering the development of robust SLAM solutions.

To bridge this gap and push SLAM towards all-weather environments, we present an extremely challenging dataset, SubT-MRS, including scenarios featuring various

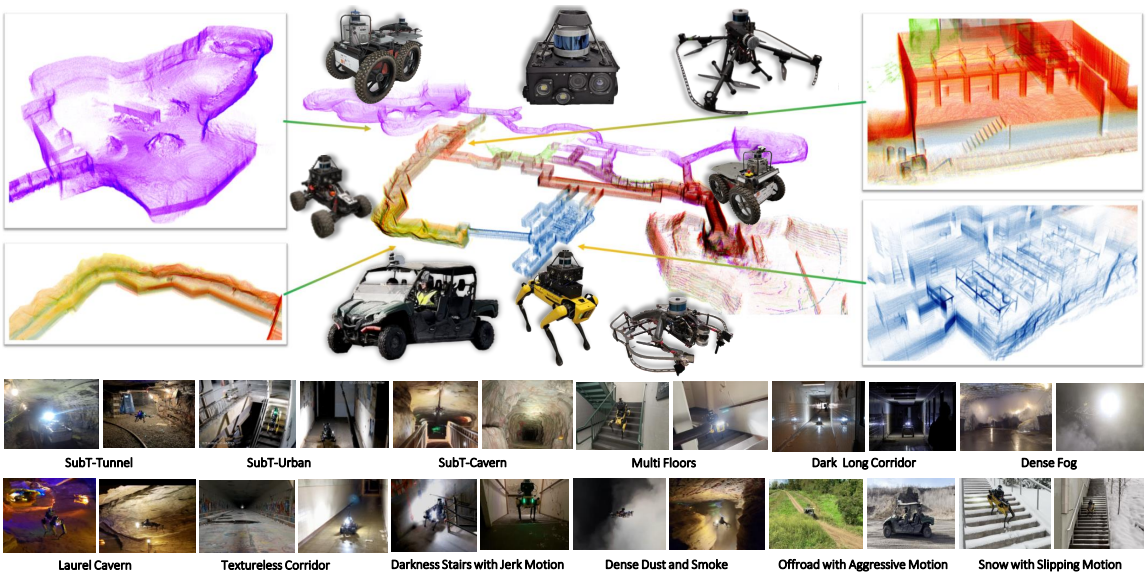


Figure 3.1: Dense reconstruction from the SubT-MRS dataset, achieved through collaboration with diverse robots equipped with multimodal sensors. Colors represent different challenging environments (tunnels, caves, urban, confined spaces) captured by various robot types (aerial, legged, wheeled). The bottom section showcases a gallery with diverse visual, LiDAR, and mixed degradations.

sensor degradation, aggressive locomotions, and extreme-weather conditions. The SubT-MRS dataset comprises 3 years of data from the DARPA Subterranean (SubT) Challenge [1] (2019−2021) and extends with an additional 2 years of diverse environments (2022−2023), containing mixed indoor and outdoor settings, including long corridors, off-road scenario, tunnels, caves, deserts, forests, and bushlands. Cumulatively, this forms a 5-year dataset encompassing over 500 hours and 100 kilometers of terrain subjected to **multimodal sensors** including LiDAR, fisheye cameras, depth cameras, thermal cameras, and IMU; **heterogeneous platforms** including RC cars, legged robots, aerial robots, and wheeled robots; and **extreme obscurant conditions** such as dense fog, dust, smoke, and heavy snow.

Additionally, we find there is no well-established metric to evaluate the robustness of a SLAM system. Existing evaluation metrics such as absolute trajectory error (ATE) [7] are not representative of the actual performance in robotic applications. We argue that to ensure the safety control of a robot, SLAM system evaluation should not only focus on poses but also its velocity. Taking robot localization as an example,

while a momentary pose error spike may only slightly undermine the overall ATE, however, it can lead to a catastrophic crash in an aerial robotic platform. To effectively gauge the performance of a SLAM algorithm, we introduce a new *robustness* metric to evaluate the reliability of the SLAM system, particularly examining smoothness and accuracy of velocity estimation.

Lastly, we perform extensive experiments using proposed degradation datasets to benchmark visual and LiDAR SLAM algorithms. These experiments identify the limitations of the existing SLAM systems and evaluate their robustness across the degradation. Our contributions include:

- **All-weather Environments** To push SLAM toward all-weather environments, we introduce SubT-MRS, an extremely challenging dataset. Spanning five years, it comprises over 500 hours and 100 kilometers of accurately measured trajectories. To the best of our knowledge, SubT-MRS is the first real-world dataset that specifically addresses failure scenarios of SLAM by incorporating a variety of degraded conditions, multiple robotic platforms, and diverse sets of multimodal sensors.
- **Robustness Metric** To evaluate the actual performance running on robots, we propose a *robustness* metric, which, to the best of our knowledge, is the first metric evaluating the reliability, safety, and resilience of SLAM.
- **SLAM Challenge** We provide a comprehensive benchmark[1] based on the SubT-MRS dataset by conducting ICCV'23 SLAM Challenge. Our evaluation reveals the limitations of state-of-the-art visual and LiDAR SLAM solutions. Furthermore, we conduct extensive experiments to verify robustness metric against various sensor degradation and length of trajectories.

3.2 Related Work

Multimodal Sensors Multimodal sensor datasets are crucial for the development of robust SLAM systems, as single sensor modalities are not comprehensive for all scenarios. Existing datasets typically focus on a limited range of modalities, such

[1]SubT-MRS dataset is completely real-world data. Simulated data in the benchmark is to ensure evaluation completeness

as monocular or stereo cameras [7, 87], LiDAR [31], event cameras [53], and depth cameras [78]. The KITTI dataset [31], although covering most modalities, is geared towards on-road scenarios and lacks thermal cameras, limiting its applicability to simple, controlled environments. Conversely, the ViViD++ dataset [41] incorporates thermal images in outdoor settings but is deficient in hardware synchronization, posing challenges for SLAM system development. The Weichen dataset [15] offers thermal images in indoor environments with accurate ground truth poses but is confined to motion capture room settings. In contrast, our SubT-MRS dataset delivers a comprehensive suite of time-synchronized multimodal sensors, including LiDAR, monocular cameras, thermal cameras, depth cameras, and IMU, along with centimeter-level ground truth, catering to a diverse range of research needs.

Heterogeneous Platforms Developing a robust SLAM system necessitates versatility in handling different motion patterns and various robotic platforms. However, most existing datasets are tailored for single-robot scenarios. While there are a few multi-robot datasets, they typically involve homogeneous platforms [14, 25, 42, 82], posing challenges in evaluating SLAM performance across varied robotic platforms. The AirMuseum dataset [21] includes drones and ground robots but lacks data from legged robots. As a comparison, SubT-MRS dataset features diverse robotic platforms including legged robots, aerial robots, and wheeled robots, operating in varied environments and sensor setups.

Extreme Environmental Conditions Developing and testing SLAM systems in extreme environmental conditions is crucial for mitigating potential real-world failures. However, most existing SLAM datasets have been primarily limited to single, controlled environments. The TUM-VI [72] and UMA-VI [101] datasets, being indoor-outdoor visual-inertial datasets, pose challenges due to varying illumination and low-texture environments. The EuRoC MAV [7] and UZH-FPV drone racing [19] datasets, popular in the SLAM community, offer data on aggressive drone movements but usually in consistently lit conditions. The KITTI dataset [31], a staple in autonomous driving research for its outdoor LiDAR-Visual-Inertial data. The Hilti SLAM dataset [35] includes both indoor and outdoor LiDAR-visual-inertial datasets with dynamic lighting and confined spaces. However, both of them focus more on

accuracy than on a variety of environmental degradations. TartanAir [85] covers most of the challenging environments but it is a simulation dataset that will pose the sim-to-real gap.

In contrast, the SubT-MRS dataset provides 30 varied scenes, encompassing a wide array of environments. These include SubT tunnels, urban areas, caverns, multi-floor structures, dark corridors, and foggy conditions. It also features textureless surfaces, dusty and smoky environments, off-road areas with aggressive motion, and snowy terrains prone to slippage. We include challenging lighting conditions such as extreme darkness and overexposure, along with geometrically challenging scenarios like featureless corridors, staircases, and self-similar cave layouts. Additionally, the dataset incorporates perceptual challenges posed by smoke, fog, dust, and various weather conditions.

Dataset	Multi-Spectrum				Multi-Degradation					Multi-Robot			
	Camera	IMU	LiDAR/Depth	Thermal	Illumination	Snow Smoke	Structureless	SubT	Aggressive Motion	Vehicle	Drone	Legged	Handheld
EuRoC MAV [7]	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
PennCOSYVIO [60]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
TUM VIO [72]	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
UZH-FPV [19]	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
College Dataset [95]	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
RobotCar [48]	✓	✓	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗	✗
UMA VI [101]	✓	✓	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
UMich [11]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗
KITTI [31]	✓	✓	✓	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗
Virtual KITTI [28]	✓	✓	✓	✗	✓	✗	✓	✗	✓	✓	✗	✗	✗
DARPA SubT [66]	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗	✗	✗
Hilti SLAM [35]	✓	✓	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✓
M3ED [12]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✓	✓	✓	✗
TartanAir [85]	✓	✓	✓	✗	✓	✓	✓	✗	✓	✗	✓	✗	✗
SubT-MRS (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3.1: Comparison of SLAM datasets on multi-sensors, multi robots, and multi degradation.

3.3 SubT-MRS Dataset

We next present the SubT-MRS dataset from three aspects including its collecting environments and settings, ground truth collection, and the new robustness metric. A detailed comparison with other datasets is listed in 3.1.

3.3.1 All-weather Environments

As mentioned in 3.2, the SubT-MRS dataset distinguishes itself through its multimodal sensor setups, heterogeneous robotic platforms, and extreme environmental conditions.

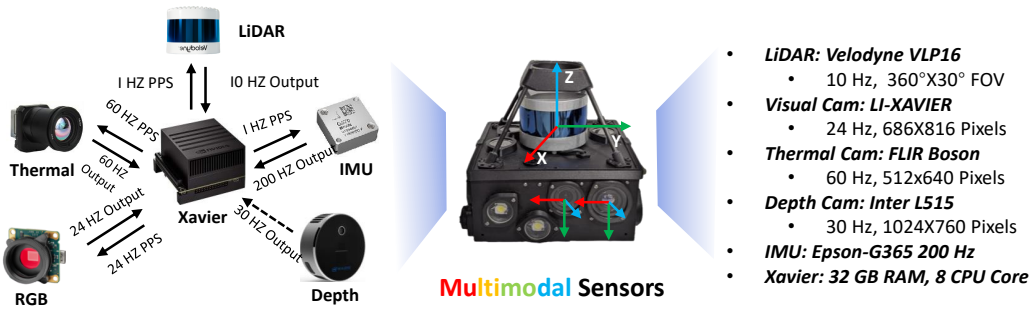


Figure 3.2: An overview of the sensor pack used in SubT-MRS dataset. It is equipped with a Xavier processing unit with hardware time synchronization for multimodal sensors including LiDAR, fisheye cameras, thermal cameras, depth cameras (option), and an IMU.

This section emphasizes this comprehensive nature pushing SLAM towards all-weather environments.

Multimodal Sensors

Multimodal sensors provide a wealth of information for SLAM systems operating in all-weather environments. This diversity improves scene understanding and strengthens the system's perception ability. Therefore, We incorporate diverse sensors to ensure system robustness.

Sensor Pack We embedded 4 Leopard Imaging RGB fisheye cameras, 1 Velodyne puck, 1 Epson M-G365 IMU, 1 FLIR Boson thermal camera, and NVIDIA Jetson AGX Xavier as a sensor pack shown in Figure 3.2.

Time Synchronization To ensure the overall consistency of the fused data, we meticulously synchronize the sensors' time using the 'pulse per second (PPS)' technique. As illustrated in 3.2, the IMU, LiDAR, and thermal camera are directly synchronized with the CPU clock, while the four RGB camera are synchronized using an FPGA board. Consequently, we can effectively manage the time synchronization gap between each pair of sensors not to exceed 3ms.

Fisheye and Thermal Camera Calibration Camera calibration plays an essential role in the efficiency of a SLAM system. For calibrating the fisheye cameras, we

Figure 3.3: The SubT-MRS datasets were collected across diverse seasons, capturing environments with perceptual challenges such as poor illumination, darkness, and water puddles, where visual sensors falter. They also include geometrically complex areas like long featureless corridors and steep multi-floor structures, challenging LiDAR systems with potential drift. Moreover, these datasets cover conditions with airborne obscurants like dust, fog, snow, and smoke in tough environments, including caves, deserts, long tunnels, and off-road areas.

employed the open-source toolkit Kalibr [64], focusing on the intrinsic and extrinsic parameters. Specifically, we use the radial-tangential distortion model to rectify the omnidirectional fisheye camera model. Calibrating thermal cameras, however, poses unique challenges, especially in gathering high-quality thermal data. To tackle this, we set up a 7×9 chessboard, heated by direct sunlight, to generate high-contrast thermal data, ensuring accuracy in the calibration process.

IMU Calibration and Extrinsic Calibration For Lidar-IMU extrinsic calibration, we utilize the CAD model to obtain the calibration parameters. In the case of Camera-IMU extrinsic calibration, we employ the Kalibr toolbox [64] to estimate the extrinsic matrix. To estimate the sensors' bias and the random walk noise of the gyroscope and accelerometer, we collected static data from the IMU and calibrated it using an Allan variance-based IMU tool [24, 29].

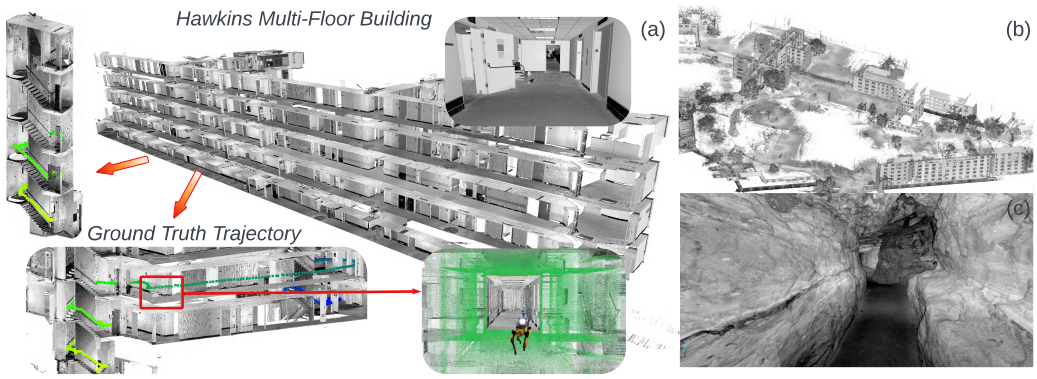


Figure 3.4: The SubT-MRS dataset facilitates the generation of high-precision ground truth maps and trajectories. Figure (a) shows the ground truth trajectory in multi-floor settings. Figure (b) displays ground truth maps for indoor and outdoor areas, encompassing long corridors, multi-floor structures, and open spaces. Figure (c) features photo-realistic scans based on our ground truth maps in cave environments.

Multi-degraded Environments

SubT-MRS includes multiple challenging environments including visually degraded environments, geometrically degraded environments, and their combination.

Visual Degradation Poor-quality visual features can significantly hinder the performance of feature extraction processes and disrupt the accuracy of feature matching. Such issues arise in low-light conditions with inconsistent brightness or image noise introduced by air obscurants. SubT-MRS encompasses these challenges and provides a wide range of visual degradation. This includes environments with limited lighting, such as hospital interiors and caves (Figure 3.3 A-F), as well as smoky or dusty conditions that cause visual obstruction (Figure 3.3 M-N), and snowy areas with reduced visibility (Figure 3.3 H, K, and L).

Geometric Degradation Lack of geometrical features poses significant challenges to LiDAR odometry. The root of this issue often lies in the limited sensing capabilities of LiDAR sensors and the constraints due to their mechanical installation. SubT-MRS captures a variety of environments that exemplify such challenges. It includes long, featureless corridors (3.3 E) and staircases (3.3 C and G). These scenarios illustrate various forms of geometric degradation, contributing to the improvement of LiDAR

odometry systems in challenging conditions.

Mixed Degradation A mixed visual and geometric degradation can further hinder the performance of SLAM systems. Examples in SubT-MRS include long, dimly lit corridors (3.3 A, E, and F), poorly illuminated staircases (3.3 C), and environments affected by snowy weather (3.3 H, K, and L). In these settings, both LiDAR odometry and visual odometry are prone to failure due to the compounded effects of mixed degradation. The inclusion of such scenarios in SubT-MRS is crucial for evaluating and improving the resilience of multimodal SLAM algorithms.

Heterogeneous Robot Platforms Most of current datasets focus on single-robot systems, limiting multi-robot SLAM development as shown in Table 3.1. To address this, we employed diverse robot platforms, including aerial, legged, and wheeled robots, navigating through various environments from urban campuses to medical facilities and natural terrains like caves. This diversity offers a range of scenarios to enhance SLAM algorithms for effective operation in challenging conditions.

Extrinsic Calibration for Multiple Robots To ensure the multi-robot system shares a common coordinate system, we perform the extrinsic calibration process [71]. It has 2 steps: First, each robot runs Super Odometry [98] to generate its local map using LiDAR data and share it with other robots through a wireless network; Second, the remaining robots identify overlapping regions between their local maps and estimate the extrinsic parameters using GICP [73].

3.3.2 Ground Truth

Ground Truth Map As depicted in 3.4, we utilized the high-precision FARO Focus 3D S120 3D scanner for creating ground-truth maps. This cutting-edge 3D scanner can measure distances up to 120m, with a maximum measurement rate of 976K points per second. The accuracy of the ground truth map is noteworthy, maintaining a range error within ±2mm. We have developed ground truth models for diverse environments, including subterranean areas like urban areas (350m×350m), caves (150m×200m), and tunnels (100×200m), as well as indoor-outdoor spaces

(200m×200m) within the CMU campus. To reduce drift, a loop closure algorithm was applied to correct the poses. Notably, 96% of the scans achieved a position uncertainty of less than 2mm. This ground-truth map is crucial for SLAM development where precision is of utmost importance.

Ground Truth Trajectory Ground truth trajectories for all sequences are generated based on the ground truth map. In this process, we establish point-to-point, point-to-plane, and point-to-line correspondences[73] between the Ground Truth map and the current LiDAR scan. Our estimated ground truth trajectory is a comprehensive fusion of various constraints derived from the ground truth map, visual odometry, LiDAR odometry, and IMU measurements. This approach ensures the robustness of our solution across diverse environmental conditions. To verify accuracy, we reconstruct LiDAR maps from ground truth trajectory and compare it with our ground truth maps.

Table 3.2: SLAM Challenge Results (Blue shadings indicate rankings of ATE and Robustness Metric; L: LiDAR I: IMU C: Camera)

#	Team	Method	Odometry Type	Device	RealTime (s)	CPU/GPU (%)	RAM (GB)	ATE↓	$R_v \uparrow / R_w \uparrow$	Sensors L	I	C
1	Liu et al	FAST-LIO2 [91], HBA [45]	Filter	Intel i7-9700K	51.310	98.667 / 0	4.052	**0.588**	**0.517/0.770**	✓	✓	
2	Yibin et al	LIO-EKF [83]	Filter	Intel i7-10700	**0.006**	52.167 / 0	0.072	4.313	0.441/0.574	✓	✓	
3	Weitong et al	FAST-LIO[2], Pose Graph[16]	Filter	Intel Xeon(R)E3-1240v5	0.125	22.63 / 0	4.305	0.663	0.473/0.747	✓	✓	
4	Kim et al	FAST-LIO2[90], Point-LIO[34], Quatro[43]	Filter	Intel i5-12500	0.268	101.108 / 0	55.64	3.825	0.479/0.615	✓	✓	
5	Zhong et al	DLO[13], Scan-Context++[38]	SW Opt	AMD Ryzen 9 5900x	0.027	**13.289** / 0	1.174	1.209	0.276/0.486	✓	✓	
1	Peng et al	DVI-SLAM [59]	Learning	Intel i9-12900	183.233	- / 149	11 (4)	**0.547**	**0.473/0.788**	✓	✓	
2	Jiang et al	LET-NET[44], VINS-Mono[62]	Hybrid	Intel i5-9400	0.064	**40.35** / 0	4.337	1.093	0.078/0.322	✓	✓	
3	Thien et al	VR-SLAM[55]	SW Opt	Intel i9-12900	0.142	176.44 / 0	9.111	3.037	0.083/0.372	✓	✓	
4	Li et al	ORB-SLAM3[8]	SW Opt.	Intel i7-10700	**0.019**	65.028 / 0	0.386	8.975	0.163/0.474	✓	✓	

3.3.3 Robustness Metric

As dicussed in 3.1, existing evaluation metrics such as absolute trajectory error (ATE) [7] have limitations in evaluating the SLAM's robustness in real-world applications. ATE primarily focuses on the accuracy of trajectory and does not consider the completeness of trajectory (recall). Also, it can not effectively capture velocity changes that have a direct impact on the robot's safety. For example, a pose spike error might not be immediately noticeable in ATE evaluation, it may result in a crash in an aerial robot control. To address these gaps, we introduce a new *robustness* metric based on estimated velocity. That is because velocity estimation from SLAM

Table 3.3: Accuracy Performance on Geometric Degradation. Red numbers represent ATE ranking. * denotes incomplete submissions.

Team	Geometric Degradation (Real World)						Simulation			Mix Degradation			Average
	Urban	Tunnel	Cave	Nuclear_1	Nuclear_2	Laurel_Caverns	Factory	Ocean	Sewerage	Long Corridor	Multi Floor	Block Lidar	
Liu et al[1]	0.307	0.095	0.629	0.122	0.235	**0.260**	**0.889**	0.757	**0.978**	1.454	**0.401**	**0.934**	**0.588**
Weitong et al[2]	**0.26**	0.096	**0.617**	**0.120**	**0.222**	0.402	0.998	0.770	1.586	**1.254**	0.577	1.056	0.663
Kim et al[3]	0.331	**0.092**	0.787	0.123	0.270	0.279	10.628	22.425	7.147	2.100	0.650	1.068	3.825
Yibin et al[4]	1.060	0.220	0.750	0.470	0.620	9.140	4.920	**0.280**	24.460	2.990	5.500	1.340	4.312
Zhong et al[5]	1.205	0.695	-	1.175	1.72	2.08	**0.889**	0.778	1.13	-	-	-	1.209*
Average	0.633	0.240	0.696	0.402	0.6134	2.432	3.665	5.002	7.060	1.950	1.782	1.099	

Table 3.4: Accuracy Performance on Visual Degradation. Red numbers represent ATE ranking. * denotes incomplete submissions.

Team	Visual Degradation (Real World)						Simulation			Average
	Lowlight 1	Lowlight 2	Over Exposure	Flash Light	Smoke Room	Outdoor Night	End of World	Moon	Western Desert	
Peng et al[1]	1.063	1.637	**0.503**	**0.44**	**0.153**	**0.827**	**0.038**	**0.195**	**0.070**	**0.547**
Thien et al[2]	1.081	2.054	1.733	1.054	10.532	7.692	0.753	1.228	1.209	3.037
Jiang et al[3]	**1.019**	**1.126**	1.911	2.341	3.757	11.821	2.154	0.604	4.010	3.193
Li et al[4]	5.768	7.834	1.757	1.295	5.370	10.766	-	30.07	-	8.98*
Average	2.232	3.163	1.476	1.282	4.953	7.776	0.982	8.024	1.763	

is crucial for the robot's control system which directly impacts stability, affecting the robot's safety. The new robustness metric is the area under the curve (AUC) of the F-1 score:

$$F_1(e) = \frac{2P(e < T)R(e < T)}{P(e < T) + R(e < T)}, \tag{3.1}$$

where the precision P quantifies the precision of the estimated velocity as a percentage: how closely the estimated velocity points lie to the ground truth point and the recall rate R quantifies the estimated velocity's completeness: to what extent all the ground-truth points are covered. A high F-1 score can only be achieved by the velocity estimation that is both accurate and complete throughout the entire run. Specifically, an estimated error e is regarded as an inlier, if it is smaller than a threshold T. To scale the threshold T within the range of $[0, 1]$, we apply exponential mapping $\exp(-10T)$ when calculating robustness metric.

Position Robustness R_p & Rotation Robustness R_r The AUC of F-1 score can be defined using both linear velocity and angular velocity, which can reflect the robustness of position and rotation estimation, respectively. Specifically, we define $R_p = \mathrm{AUC}(\mathrm{F}_1(\mathrm{v_e}))$ and $R_r = \mathrm{AUC}(\mathrm{F}_1(\omega_e))$, where v_e and ω_e are the estimated error of linear velocity and rotation velocity, respectively. Note that not all SLAM solutions can output the velocity at the desired frequency. To address this, we use B-splines

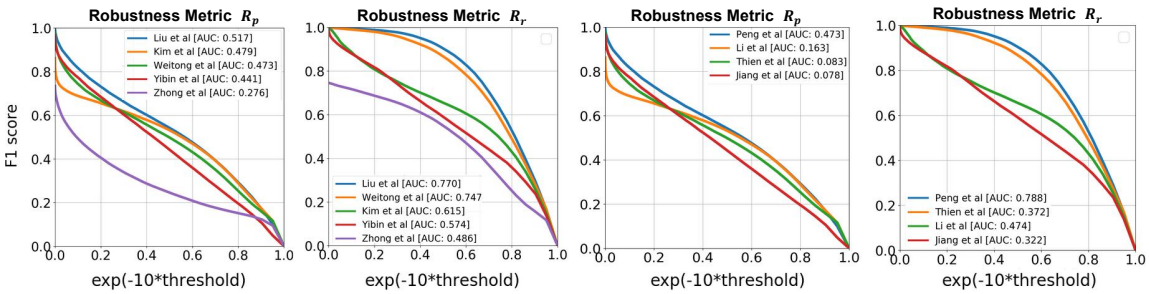


Figure 3.5: From left to right, it shows robustness metric R_p and R_r for LiDAR and visual sequences respectively. Note: This is a summary of results for all sequences, with weights based on the trajectory length. The area under the curve (AUC) represents the robustness (R_p, R_r). The x-axis shows velocity thresholds for classifying estimated velocities as inliers and the y-axis is F-1 score.

[54, 84] to obtain smooth trajectories, compute derivatives for smooth trajectories, and derive estimated linear velocity and angular velocity. For more details, please refer to the supplementary.

3.4 Open SLAM Challenge Results

In this section, we will present our works from two perspectives: accuracy evaluation and robustness evaluation.

3.4.1 Accuracy Evaluation

The results of the ICCV SLAM challenge underscore the necessity for advancements in system robustness. From 29 submissions, we identified 5 winners in the LiDAR category and 4 in the visual category. However, in the sensor fusion track, which addresses both visual and geometric degradation, no submissions met the criteria for success. This result reveals the existing SLAM systems still have lots of space to improve. Sequence characteristics will be detailed in the supplementary material. 3.2 shows the results for both the LiDAR and visual tracks. Unfortunately, there are no current solutions that can balance high accuracy and real-time performance in challenging environments. Since several environments lack geometric features or visual features, the algorithms required more processing times to achieve reasonably high accuracy. Liu et al. at the University of Hong Kong, who achieved the highest

Table 3.5: Robustness Performance on Geometric Degradation. Red numbers are robustness ranking. Larger values indicate better robustness.

Team		Geometric Degradation (Real World)						Simulation			Mix Degradation			Average
		Urban	Tunnel	Cave	Nuclear_1	Nuclear_2	Laurel_Caverns	Factory	Ocean	Sewerage	Long Corridor	Multi Floor	Block Lidar	
Liu et al[1]		**0.811**	**0.865**	0.736	0.747	0.504	**0.816**	0.157	0.135	**0.135**	0.396	**0.529**	0.371	**0.516**
Weitong et al[2]		0.773	0.838	0.690	**0.871**	**0.739**	0.152	0.153	0.149	0.081	0.407	0.456	0.345	0.471
Kim et al[4]	$R_p \uparrow$	0.747	**0.865**	**0.737**	0.870	0.689	0.777	0.014	0.001	0.010	**0.410**	0.285	0.345	0.479
Yibin et al[3]		0.650	0.783	0.557	0.721	0.490	0.481	**0.219**	**0.560**	0.019	0.175	0.260	**0.370**	0.440
Zhong et al[5]		0.567	0.683	0.204	0.680	0.426	0.447	0.103	0.084	0.122	0	0	0	0.276
Liu et al[1]		**0.888**	**0.893**	**0.816**	0.857	0.778	**0.861**	0.692	0.719	**0.652**	0.753	0.689	**0.643**	**0.770**
Weitong et al[2]		0.886	0.892	0.811	**0.893**	**0.840**	0.598	0.688	0.720	0.514	0.766	**0.706**	0.644	0.746
Kim et al[4]	$R_r \uparrow$	0.680	**0.893**	**0.816**	**0.893**	0.803	0.808	0.198	0.125	0.300	**0.802**	0.422	0.642	0.615
Yibin et al[3]		0.624	0.707	0.462	0.733	0.540	0.467	**0.862**	**0.962**	0.496	0.477	0.342	0.213	0.573
Zhong et al[5]		0.731	0.745	0.437	0.757	0.603	0.523	0.688	0.704	0.643	0	0	0	0.485
Average R_p		0.710	0.807	0.585	0.778	0.570	0.535	0.130	0.186	0.073	0.278	0.306	0.286	
Average R_r		0.762	0.826	0.668	0.827	0.713	0.651	0.626	0.646	0.521	0.560	0.432	0.428	

Table 3.6: Robustness Performance on Visual Degradation. Red numbers are robustness ranking. Larger values indicate better robustness.

Team		Visual Degradation (Real World)						Simulation			Average
		Lowlight 1	Lowlight 2	Over Exposure	Flash Light	Smoke Room	Outdoor Night	End of World	Moon	Western Desert	
Peng et al[1]		**0.357**	**0.227**	**0.264**	0.203	**0.536**	**0.270**	**0.699**	**0.893**	**0.806**	**0.472**
Thien et al[3]		0.045	0.070	0.240	0.156	0.131	0.075	0	0.031	0	0.083
Jiang et al [4]	$R_p \uparrow$	0.046	0.039	0.194	0.088	0.242	0.095	0	0	0	0.078
Li et al[2]		0.342	0.187	0.257	**0.208**	0.322	0.142	0	0.006	0	0.162
Peng et al[1]		**0.641**	**0.581**	**0.744**	0.650	**0.878**	**0.670**	**0.975**	**0.975**	**0.974**	**0.787**
Thien et al[3]		0.413	0.445	0.610	0.269	0.474	0.487	0.177	0.315	0	0.354
Jiang et al[4]	$R_r \uparrow$	0.453	0.452	0.619	0.252	0.542	0.577	0.002	0	0.157	0.339
Li et al[2]		0.642	0.574	0.657	**0.651**	0.773	0.660	0.0	0.305	0	0.473
Average R_p		0.198	0.131	0.239	0.164	0.308	0.146	0.175	0.232	0.202	
Average R_r		0.537	0.513	0.658	0.456	0.667	0.598	0.288	0.399	0.283	

accuracy, recorded an Absolute Trajectory Error (ATE) of 0.588m, but each iteration took 51.3 seconds. In the visual track, Peng et al. from the Samsung Research Center attained a leading ATE of 0.547m, but each iteration required 183 seconds.

LiDAR Track Discussion To enhance robustness in all-weather environments, a pivotal question arises: what are the limits of current LiDAR SLAM algorithms? Table 3.3 presents a summary of the ATE/RPE errors observed in real-world geometrically degraded, simulated, and mixed degradation scenarios across the top five teams. In real-world geometrically degraded environments, we observed the **first limitation:** existing LiDAR solutions struggle in confined spaces such as caves. Almost all algorithms perform worse in Cave (Figure 3.3S) and Laureal_Cavern environments (Figure 3.3N) with average ATE 0.696 and 2.432 meters respectively. One reason is that cave environments is the most confined space which may lack sufficient geometric features, compared with other environments like tunnel (Figure 3.3J) and Urban (Figure 3.3B) and Nuclear scenes (Figure 3.3U).

In simulated environments, we encountered the **second limitation:** existing

methods are tailored for fixed motion patterns like vehicles and struggle with unpredictable motion patterns. Since our simulation sequence is derived from TartanAir [85], featuring aggressive and random motion patterns, it might not follow the usual velocity distribution domain and has not been thoroughly tested. As shown in 3.3 (simulation), it resulted in significantly higher ATE errors (5.239m) compared to real-world's (0.836m).

In mixed degradation environments, the **third limitation** we found is that existing methods cannot actively select the most informative measurements to adapt to new environments. For example, in the Long Corridor sequence (3.3 A) with low lighting and a featureless environment, the scenario is intricately designed for collaborative usage of LiDAR and Visual sensors. Algorithms are expected to utilize visual information in geometrically degraded environments while disregarding it in darkness. Similarly, the Block LiDAR sequences (3.3 V) aim to simulate sensor drop scenarios frequently encountered in robotic applications. These sequences involve alternating periods of LiDAR and Visual data loss, challenging algorithms to promptly detect sensor failures and switch to the other modality for SLAM. However, the average ATE error (1.61m) in 3.3 (mixed degradation) is much higher than in geometric degradation scenarios (0.836m), suggesting the above limitations. Details are in the supplementary.

Visual Track Discussion To improve the robustness, a pivotal question arises: what are the limits of current visual SLAM? To what extent does image quality impact a visual SLAM system? 3.4 shows a summary of ATE errors from real-world and simulation scenarios from the awarded 4 teams. In real-world scenarios, **the first limitation** is the lack of anti-noise capability in current methods, especially in low-quality image settings. We assessed visual odometry accuracy in visually degraded environments, including low lighting, sunlight overexposure, flashing lights, smoke-filled rooms, and nighttime outdoor settings (Figure 3.3). Trajectories in conditions like the Smoke Room and Outdoor Night exhibit significantly higher average ATE errors (4.95m, 7.76m respectively). Smoke and night scenes challenge feature extraction and tracking and introduce sensor noise, emphasizing the need for robust anti-noise algorithms. **The second limitation** is that existing methods struggle to overcome aggressive motion. Even in simulations with relatively good

image quality, most methods still show significant average ATE errors (8.024m) on the Moon sequence (3.3 Y). See more details in supplementary.

3.4.2 Robustness Evaluation

The F-1 robustness curve provides a detailed evaluation under different error tolerances, while R_p and R_r provide an overall measure of robustness that is not dependent on a specific decision threshold. This suggests the flexibility of our new robustness metric. Figure 3.5 clearly shows the robustness performance of all teams regarding position and rotation, highlighting the differences in robustness between the methods. The values of R_p and R_r, which are the AUC under F-1 curves, are displayed in the brackets of the legends. It reveals that Liu et al. and Peng et al. are the most robust solutions for LiDAR and visual tracks, respectively.

Is the robustness metric robust? We did extensive experiments on our robustness metric against diverse sensor degradation settings. Table 3.5 and 3.6 present a summary of robustness performance in real-world, simulated, and mixed degradation scenarios for both LiDAR and visual sequences. In Table 3.5, we observe that the average values of R_p and R_r for the mixed degradation environment (0.290, 0.473) are significantly lower than geometric degradation environments (0.664, 0.754). This observation suggests that the majority of SLAM algorithms exhibit reduced robustness in mixed degradation environments as compared to geometrically degraded ones. This aligns with our expectations and verifies the effectiveness of the robustness metric.

Our robustness ranking, indicated by small red numbers, differs from those on ATE ranking shown in 3.3 and 3.4. This is because our F-1 score-based metrics R_p and R_r jointly consider precision and recall rate to provide a balanced evaluation of SLAM performance, considering both metrics across the full spectrum of thresholds. In contrast, the ATE focuses solely on precision, neglecting the trajectory's completeness (recall). We also evaluate our metric on various synthetic trajectories in the supplementary.

3.5 Discussion

We introduce SubT-MRS, a comprehensive SLAM dataset with various sensor data, locomotion patterns, and over 30 degradation types in simulation and real-world settings to push SLAM towards all-weather environments. Additionally, we introduce a new robustness metric to evaluate the reliability of SLAM systems, enhancing robot safety control. 29 teams have tested SubT-MRS in the organized SLAM challenge and we expect that it will serve as a critical benchmark for future SLAM development.

Chapter 4

IMU-centric LiDAR-Visual-Inertial Estimator

4.1 Motivation

In this section, we proposed an IMU-centric SLAM system combing the advantages of tightly- coupled methods with loosely coupled methods. The system design follows a key insight: An inertial measurement unit (IMU) produces smooth measurements with noise but little outliers which can make its estimate very accurate as long as the bias drift can be well-constrained by other sensors. Therefore, Super Odometry is designed around the IMU as the primary sensor. We will introduce how to achieve this in the following sub-sections.

4.2 System Overview

We first define frames and notation that we use throughout the paper. We denote the world frame as **W**, IMU body frame as **B**, LiDAR frame as **L** and camera frame as **C**. For simplicity, we will regard the IMU frame as our robot body frame and our goal is to estimate the position of the IMU body frame relative to a fixed world frame **W**. The robot state $\mathbf{x}_k$ can be written as:

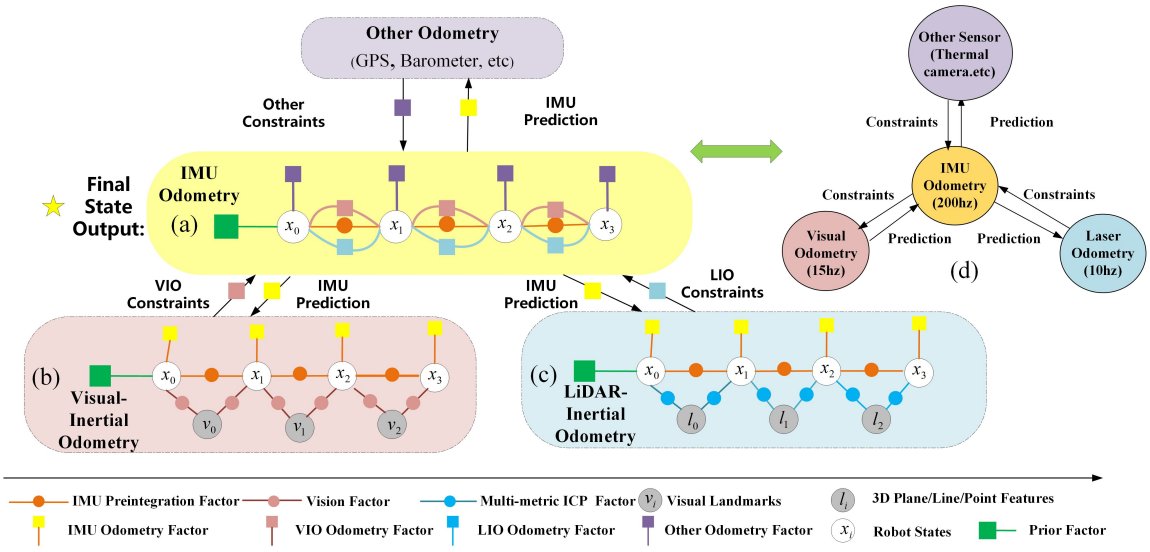


Figure 4.1: Overview of Super Odometry Alogorithm. IMU Odometry makes use of observations provided by visual odometry and LiDAR odometry to constrain the IMU bias. In return, the constrained IMU odometry provides the prediction to the visual odometry and LiDAR odometry. Meanwhile, the size of graph node can be adjusted according to the computational resource and the IMU Odometry provides the final state estimation.

$$\mathbf{x}_k = \left[\mathbf{p}^w_{b_k}, \mathbf{v}^w_{b_k}, \mathbf{q}^w_{b_k}, \mathbf{b}_a, \mathbf{b}_g\right], k \in [0, n] \tag{4.1}$$

where the k-th robot state $\mathbf{x}_k$ consists of the position $\mathbf{p}^w_{b_k}$, velocity $\mathbf{v}^w_{b_k}$, orientation $\mathbf{q}^w_{b_k}$ of IMU's center, and $\mathbf{b}_a, \mathbf{b}_g$ are accelerometer bias and gyroscope bias respectively.

An overview of the proposed system is shown in Fig.4.1. It is composed of three modules: IMU odometry, Visual-inertial odometry (VIO), and LiDAR-inertial odometry (LIO). Our system design follows a key insight: an IMU and its state estimation can be very accurate as long as the bias drift is well-constrained by other sensors. Therefore, Super Odometry uses the IMU as the primary sensor. It uses observations provided by Visual-inertial odometry (VIO) and LiDAR-inertial odometry (LIO) to constrain the accelerometer bias $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$. In return, the constrained IMU odometry provides the prediction to the VIO and LIO, which recovers motion in a coarse-to-fine manner. Moreover, the focus on the IMU makes the pipeline more robust to environment-related issues in state estimation, such as geometric or Visual degradation. Finally, the algorithm architecture allows

for an easy and flexible way to incorporate other odometry sources such as GPS and wheel odometry.

4.3 Methodology

Super Odometry is modeled using a factor graph. Here we mainly introduce three types of odometry factors for factor graph construction, namely: (a) IMU; (b) LiDAR-inertial; and (c) Visual-inertial odometry factors.

4.3.1 IMU Odometry Factors

The structure of IMU odometry factors is shown in Fig.4.1.(a), which is similar to traditional pose graph structure. However, it is different since the estimated *state* of IMU odometry not only contains position $\mathbf{p}_b^w$, velocity $\mathbf{v}_b^w$ and orientation $\mathbf{q}_b^w$ but also the accelerometer $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$. Each node in the pose graph is associated with a state. The density of nodes is determined by the lowest-frequency odometry. The edge between two consecutive nodes represents the relative body motion obtained from IMU preintegration method. Other edges could be local constraints or global constraints which depend on the type of sensors.

IMU Preintegration Factor Based on the IMU preintegration method proposed in [26], we can obtain the relative motion measurement $\Delta\mathbf{r} = \left(\hat{\alpha}_j^i, \hat{\beta}_j^i, \hat{\gamma}_j^i\right)$ between consecutive IMU body frames i and j. The IMU preintegration factor can be defined as:

$$\begin{aligned}\mathbf{e}_{ij}^{imu} &= \mathbf{z}_{ij}^b - h_{ij}^b\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &= \begin{bmatrix} \hat{\alpha}_j^i \\ \hat{\beta}_j^i \\ \hat{\gamma}_j^i \\ \mathbf{b}_{aj} \\ \mathbf{b}_{wj} \end{bmatrix} \ominus \begin{bmatrix} \mathbf{R}_w^i\left(\mathbf{p}_j^w - \mathbf{p}_i^w + \frac{1}{2}\mathbf{g}\Delta t^2 - \mathbf{v}_i^w \Delta t\right) \\ \mathbf{R}_w^i\left(\mathbf{v}_j^w + \mathbf{g}\Delta t - \mathbf{v}_i^w\right) \\ \mathbf{q}_i^{w^{-1}} \otimes \mathbf{q}_j^w \\ \mathbf{b}_{ai} \\ \mathbf{b}_{wi} \end{bmatrix}\end{aligned} \tag{4.2}$$

Where rotation matrices $\mathbf{R}$ and Hamilton quaternions $\mathbf{q}$ are used to represent rotation. $\mathbf{p}_i^w$, $\mathbf{v}_i^w$ and $\mathbf{q}_i^w$ are translation, velocity, and rotation from the body frame to the world frame. $\mathbf{g}$ is the gravity vector in the world frame. $\mathbf{b}_a$ and $\mathbf{b}_w$ are

accelerometer bias and gyroscope bias respectively. $\ominus$ is the minus operation on the IMU residuals. Note that the accelerometer bias error and gyroscope bias error are jointly optimized in the graph.

Relative Pose Factor Since the camera and LiDAR are not globally referenced and their odometry is based on the first pose of the robot, we only use their relative poses $[\Delta p_j^i, \Delta q_j^i]$ as local constraints to constrain the IMU preintegration's position $\hat{\alpha}_j^i$, and rotation $\hat{\gamma}_j^i$. The relative pose factor $\mathbf{e}_{ij}^{LIO}, \mathbf{e}_{ij}^{VIO}$ is derived as:

$$\mathbf{e}_{ij}^{LIO}, \mathbf{e}_{ij}^{VIO} = \begin{bmatrix} \Delta p_j^i \\ \Delta q_j^i \end{bmatrix} \ominus \begin{bmatrix} \hat{\alpha}_j^i \\ \hat{\gamma}_j^i \end{bmatrix} \tag{4.3}$$

IMU Odometry Optimization For each new frame, we minimize an optimization problem that consists of relative pose factor ($\mathbf{e}_{ij}^{LIO}$,$\mathbf{e}_{ij}^{VIO}$), and marginalization prior $\mathbf{E}_m$.

$$E = \sum_{(i,j)\in\mathcal{B}} \mathbf{e}_{ij}^{LIO\top} W_{ij}^{-1} \mathbf{e}_{ij}^{LIO} + \sum_{(i,j)\in\mathcal{B}} \mathbf{e}_{ij}^{VIO\top} W_{ij}^{-1} \mathbf{e}_{ij}^{VIO} + \mathbf{E}_{\mathrm{m}} \tag{4.4}$$

The relative pose factor has to be weighted with the appropriate covariance matrix W_{ij}, which can be calculated based on the reliability of the observation. For example, in visually degraded environments, $\mathbf{e}_{ij}^{VIO}$ calculated by VIO will have a lower weight. In contrast, in geometrically degraded environments, $\mathbf{e}_{ij}^{LIO}$ calculated by LIO will have a lower weight. Note that without loss of generality, the IMU odometry can also incorporate measurements from other sensors such as GPS and wheel odometry.

4.3.2 LiDAR-Inertial Odometry Factors

The structure of LiDAR-Inertial odometry factors is shown in Fig.4.1.(c). Since IMU odometry is computationally efficient and outputs at a very high frequency, its IMU preintegration factor can be naturally added into the factor graph of LiDAR-Inertial odometry. The IMU preintegration factor will be used as motion prediction for current scan-map matching and connects consecutive LiDAR frames in the factor graph.

The pipeline of the scan-map matching process can be divided into four steps, namely: (a) PCA-based Feature Extraction (b) Multi-metric Linear Square ICP (c) LiDAR-inertial Odometry Factor (d) Dynamic Octree.

PCA Based Feature Extraction When the system receives a new LiDAR scan, the LiDAR scan will be downsampled to a fixed number and then fed into the feature extraction module. The point-wise K-D tree is used to find the nearest k neighbor points within a sphere of radius r. Based on these neighbor points, the principal components analysis (PCA) method is used to analyze the local linearity $\sigma_{1\mathrm{D}}$, planarity $\sigma_{2\mathrm{D}}$, and curvatures $\sigma_{3\mathrm{D}}$ of geometrical feature. They are defined as[33]

$$\sigma_{1\mathrm{D}} = \frac{\sigma_1 - \sigma_2}{\sigma_1}, \sigma_{2\mathrm{D}} = \frac{\sigma_2 - \sigma_3}{\sigma_1}, \sigma_{3\mathrm{D}} = \frac{\sigma_3}{\sigma_1 + \sigma_2 + \sigma_3} \tag{4.5}$$

where $\sigma_i = \sqrt{\lambda_i}$ and λ_i are the eigenvalues of the PCA for the normal computation. According to the magnitude of local feature $\sigma_{1\mathrm{D}}$,$\sigma_{2\mathrm{D}}$, $\sigma_{3\mathrm{D}}$, point, line and plane features $\mathbb{F}_{i+1} = \{F_{i+1}^{po}, F_{i+1}^{li}, F_{i+1}^{pl}\}$ can be easily classified.

Multi-metric ICP Factors After extracting reliable geometric features, we use motion prediction provided by IMU odometry and transform the $\mathbb{F}_{i+1}$ from $\mathbf{B}$ to $\mathbf{W}$. Then, we find its point, line, and plane correspondence in the map by using the proposed dynamic octree. For the sake of brevity, the detailed procedures will be described in the next section.

To improve the robustness of Super Odometry, we expect to estimate the optimal transformation that jointly minimizes the point-to-point, point-to-line, and point-to-plane distance error metric, which can be computed using the following equations:

$$\begin{cases} \mathbf{e}_i^{\mathrm{po}\to\mathrm{po}} = \mathbf{q}_i - (R\mathbf{p}_i + t) & q_i, p_i \in F_{i+1}^{po} \\ \mathbf{e}_j^{\mathrm{po}\to\mathrm{li}} = \mathbf{v}_j \times (\mathbf{q}_j - (R\mathbf{p}_j + t)) & q_j, p_j \in F_{i+1}^{li} \\ \mathbf{e}_k^{\mathrm{po}\to\mathrm{pl}} = \mathbf{n}_k^\top (\mathbf{q}_k - (R\mathbf{p}_k + t)) & q_k, p_k \in F_{i+1}^{pl} \end{cases} \tag{4.6}$$

where $\mathbf{e}_i^{\mathrm{po}\to\mathrm{po}}$, $\mathbf{e}_j^{\mathrm{po}\to\mathrm{li}}$ and $\mathbf{e}_k^{\mathrm{po}\to\mathrm{pl}}$ are the point-to-point (line, plane) distance. $\mathbf{v}_j$ and $\mathbf{n}_k$ represent the corresponding principal and normal direction of features. $\mathbf{T}_{i+1} = \{\mathbf{R}, \mathbf{t}\}$ is the optimal transformation we expect to estimate.

However, in airborne obscurants environments such as dust, fog, and smoke environments, we may extract geometric features on these noise data and obtain unreliable data association. To solve this problem, we found that the quality of correspondence is depended on if the extracted features' neighbor points fit the point, line, or plane distribution. Therefore, we use distribution's quality W_l =

$\{w_i^{po\to po}, w_i^{po\to li}, w_i^{po\to pl}\}$ to define the quality of correspondence by using the following equation.

$$w_i^{po\to po} = \frac{\sigma_{3D}}{\sigma_{3Dmax}} \tag{4.7}$$

$$w_i^{po\to li/pl} = 1 - \frac{\sum_{i=1}^{k} (p_i - \bar{p})^T A (p_i - \bar{p})}{k \times d_{max}} \tag{4.8}$$

where σ_{3Dmax} is the maximum threshold of curvatures. p_i are the neighbor points of extracted features and $\bar{p}$ is their mean. For point-to-line(plane) distance error, we use normal direction $\mathbf{n}_k$ of features and define $A = I - \mathbf{n}_k^T \times \mathbf{n}_k$ and $A = \mathbf{n}_k^T \times \mathbf{n}_k$ respectively. d_{max} represents the maximum distance error for point-to-line(plane) correspondence. k is the number of neighbor points.

LiDAR-inertial Odometry Optimization The levenberg-marquardt method is then used to solve for the optimal transformation by minimizing:

$$\min_{\mathbf{T}_{i+1}} \left\{ \begin{array}{c} \sum_{p\in\mathbb{F}_i} \left\| W_l e_i^{po\to po,li,pl} \right\|^2 + \\ \sum_{(i,i+1)\in\mathcal{B}} \| W_{imu} e^{imu} \|^2 + \mathbf{E}_{imuodom}^{prior} \end{array} \right\} \tag{4.9}$$

$\mathbf{E}_{imuodom}^{prior}$ and $\mathbf{e}^{imu}$ are the predicted pose prior and IMU preintegration factor provided by IMU odometry. $\mathbf{W}_l$ and $\mathbf{W}_{imu}$ are the weight of LiDAR and imu correspondence.

When the environment is geometrically degraded, since the IMU odometry can receive the measurements from other sensors, the IMU odometry is still reliable and $E_{imuodom}^{prior}$ will be dominant in the optimization problem. Meanwhile, unreliable LiDAR constraints will be rejected or have low weight by evaluating the quality of constraints. On the other hand, when the environment is well-structured, the multi-metric ICP factor will be dominant in the optimization problem.

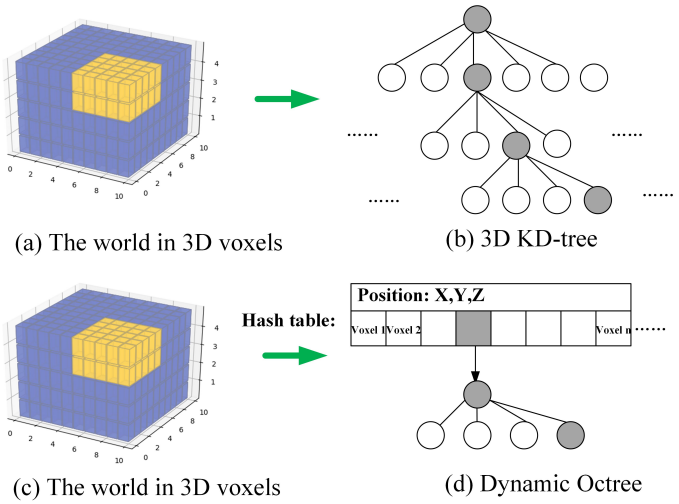


Figure 4.2: The comparison of 3D KD-tree and dynamic octree. (a,b) shows the construction process of 3D KD-tree when adding new point cloud (grey circle), which needs to change the structure of the whole tree. (c,d) shows the construction process of dynamic octree when adding new point cloud (grey circle), which only needs to change the structure of the sub-tree.

4.3.3 Dynamic Octree

Most LiDAR SLAM methods adopt the KD-tree method to organize the 3D points and fulfill data association. However, we argue that the traditional KD-tree is very time-consuming to organize the 3D points since it only uses one tree to organize all the points and needs to recreate the KD-tree every time when adding new points shown in Fig.4.2(b). To solve this problem, we propose to use a more efficient organization of 3D points named dynamic octree to facilitate data association. The dynamic octree is based on the work of Behley [4]. It stores the map as a hash table as shown in Fig.4.2(d). The world is represented by voxels and voxels can be accessed from the hash table by using their XYZ indices. Instead of only building one tree, each voxel will have its octree to organize the points and each octree can be accessed by a hash table. Since we only need to update the specific octree instead of the whole tree, the real-time performance on data association will be significantly improved.

4.3.4 Visual-Inertial Odometry Factors

To take full advantage of the fusion of vision and LiDAR sensing modalities, we track monocular visual features and LiDAR points within the camera field of view and use them to provide depth information for visual features.

Visual-inertial Odometry Optimization For each new keyframe, we minimize a non-linear optimization problem that consists of visual reprojection factor $\mathbf{e}_{reproj}$, IMU preintegration factor $\mathbf{e}_{imu}$, marginalization factor $\mathbf{E}_m$ and a pose prior from IMU odometry $\mathbf{E}^{prior}_{imuodom}$. The structure of Visual-inertial odometry factors is shown in Fig.4.1(b).

$$\min_{\mathbf{T}_{i+1}} \left\{ \begin{array}{l} \sum\limits_{i \in \mathbf{obs}(i)} \mathbf{e}^{\mathbf{T}}_{\text{reproj}} W^{-1}_{\text{reproj}} \mathbf{e}_{\text{reproj}} + \sum\limits_{(a,b) \in \mathbf{C}} \mathbf{e}^{\mathbf{T}}_{\text{imu}} W^{-1}_{imu} \mathbf{e}_{imu} \\ + \mathbf{E}_m + \mathbf{E}^{prior}_{imuodom} \end{array} \right\} \tag{4.10}$$

$\mathbf{obs}(\mathbf{i})$ represents a set that contains visual features i that is tracked by other frames. The set $\mathbf{C}$ contains pairs of visual nodes (a, b) connected by IMU factors. W_{reproj} and W_{imu} are covariance matrices for visual reprojection factor and IMU preintegration factor.

Similar to the explanation in the last section, when the environment is visually degraded, the $\mathbf{E}^{prior}_{imuodom}$ will be dominant in the optimization problem and unreliable vision factor will be rejected by analyzing the information matrix. However, when the environment has good lighting conditions, the visual factor will be dominant in the optimization problem and the IMU preintegration factor only provides the initial guess for visual feature tracking.

4.4 Discussion

In this section, we provide a concise overview of various sub-factor graphs:

(a) Visual-inertial odometry factor graph, (b) LiDAR inertial odometry factor graph, and (c) IMU odometry factor graph.

Rather than constructing a single extensive factor graph through tightly coupled methods to address a highly intricate problem, Super Odometry adopts a strategy of

dividing the overarching factor graph into smaller, more manageable sub-factor graphs. This design notably enhances the resilience of the odometry system, facilitating a smoother engine switch scheme, particularly in scenarios where certain sensors may be faulty or operating in degraded environments.

Chapter 5

Super Odometry Performance on SubT-MRS Dataset

5.1 Motivation

In this section, we evaluate the robustness of Super Odometry with other state-of-the-art algorithms in our SubT-MRS datasets. Then, we evaluate the real-time performance of the algorithms. Our experiment video can be found through this link: superodometry.com.

5.2 Dataset

We collected our test dataset with Team Explorer's DS drones (Fig.5.1(e)), deployed in the DARPA Subterranean Challenge. It has a multi-sensor setup including a Velodyne VLP-16 LiDAR, an Xsens IMU, a uEye camera with a wide-angle fisheye lens, and an Intel NUC onboard computer. The data sequences were designed to include both visually and geometrically degraded scenarios, which are particularly troublesome for camera- and LiDAR-based state-estimation.

The list of test sequences is as follows:

- **Dark-Room**: Hand-carried walk in and out of a dark room shown as Fig.5.1(c).
- **White-Wall**: Hand-carried walk with camera facing to a feature-less white

Figure 5.1: (a-d) shows the challenging environments where we collect datasets. (e) shows our multi-sensor setup.

wall shown as Fig.5.1(b).

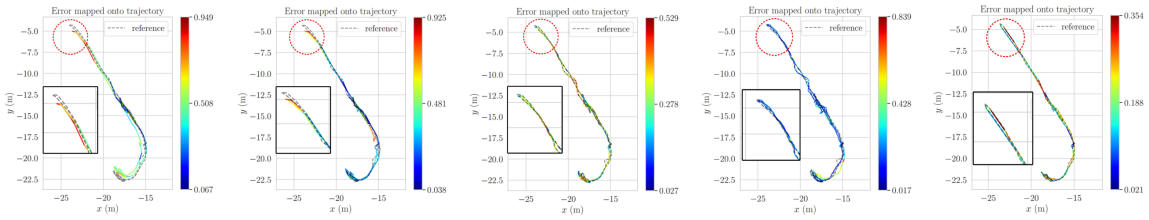
- **Constrained-Environment**: Hand-carried walk in a stair shaft shown as Fig.5.1(b).
- **Long-Corridor**: Hand-carried walk through a long corridor in an apartment building shown as Fig.5.1(a).
- **Dust**: Autonomous flying through a subterranean cave with heavy dust shown as Fig.5.1(d).
- **Urban Challenge Alpha Course**: Competition run in DARPA SubT Urban Chellange Alpha Course in an abandoned nuclear facility shown as Fig.1.1.

The trajectory ground truth for quantitative analysis on **Dark-Room**, **Long-Corridor**, **White-Wall**, and **Constrained-Environment** is generated with a Total Station (TS) and tracking prism on top of the sensor suite. Data sequences **Dust** and **Urban Challenge Alpha Course** do not have trajectory ground truth and are used for map quality and real-time performance analysis.

5.3 Robustness and Accuracy Evaluation

Our dataset contains visually and geometrically degraded scenes. Such conditions are common in underground environments, such as the ones encountered in each phase of the DARPA Subterranean Challenge. We compare our method with other LiDAR-based odometry (LOAM [93], LIO-SAM[76]) and vision-based odometry (VINS [61], Our Depth-enhanced VINS[100]). We adopt EVO [1] package to calculate the translational Absolute Trajectory Error (ATE) of each estimated trajectory against the ground truth trajectories. The maximum and RMSE results of ATEs are shown in Table 5.1. It is worth mentioning that no loop closing step is used in the experiments. We focus on comparing only the odometry part of the algorithms.

First, we consider visually-degraded environments, where camera-based algorithms are expected to give poor results. Next, we consider geometrically-degraded environments, where LiDAR-based algorithms usually suffer. Finally, we consider both types of degradation together. We show that Super Odometry has better accuracy in all cases, because of its robustness properties.



(a) VINS (b) VINS-Depth (c) LOAM (d) LIO-SAM (e) Super Odometry

Figure 5.2: Trajectories of LOAM, LIO-SAM, VINS, Depth-enhanced VINS, and Super Odometry on dark room dataset

5.3.1 Robustness Comparison in Visually Degraded Environments

Here we show the results of the **Dark-Room** sequence. In this data sequence, we hand-carried the drone and walk past a poorly lit room combined with aggressive

[1] https://github.com/MichaelGrupp/evo

motion. The image is mostly dark (Fig.5.1.(c)) and therefore challenging for feature extraction in the front-end of visual odometry.

Fig.5.2 shows the algorithm output trajectories compared with the ground truth trajectory from the Total Station. Considering the the absolute trajectory error (ATE) shown in Table 5.1, we see that purely vision-based methods have the worst performance in this situation, as expected. Our IMU-centric Super Odometry outperforms the above methods with the lowest ATE of 0.174m. We also evaluate our method on the **White-Wall** data sequence mentioned in the dataset section. The ATE of each method can be found in Table 5.1. For more details, we refer readers to the accompanying video.

5.3.2 Robustness Comparison in Geometrically Degraded Environments

In this experiment, we ran the algorithms on the **Long-Corridor** data sequence Fig.5.1.(a). This is a geometrically degenerate scenario. It is common in man-made environments such as apartment buildings, hospitals, and factories. The lack of geometric features and repeated structures may cause issues for LiDAR-based algorithms.

Fig.5.3, Fig.5.4 and Fig.5.5 show the resulting map from LOAM, LIO-SAM, and Super Odometry in this data sequence. On one hand, we observe misalignment and significant drift as shown in the yellow circle in the LIO-SAM and LOAM maps. Super Odometry, on the other hand, outperforms the previous two methods and gives an accurate map and odometry result. We also evaluate our method on similarly geometrically-degraded situations including **Long Corridor** and **Constrained-Environment** data sequence mentioned in dataset section. The ATE of each method can be found in Table 5.1. It can be seen that Super Odometry outperforms the other methods with the lowest ATE of 0.055m. For more details, we refer readers to the accompanying video.

Table 5.1: Accuracy Evaluation of LOAM, LIO-SAM, VINS, VINS-Depth and Ours Methods Operating on Challenging Environments

	ATE (in m) Transl. MAX					ATE (in m) Transl. RMSE				
Sequences	LOAM	LIO-SAM	VINS	VINS-Depth	Ours	LOAM	LIO-SAM	VINS	VINS-Depth	Ours
Constrained environments	0.779	1.914	0.907	0.972.	**0.609**	0.319	0.650	0.470	0.490	**0.259**
Long corridor	9.44	6.52	9.02	6.15	**0.35**	4.15	0.85	5.83	2.58	**0.055**
White Wall	1.013	1.159	2.801	1.733	**0.482**	0.457	0.217	1.184	0.786	**0.156**
Dark Room	0.528	0.839	0.948	0.924	**0.354**	0.263	0.246	0.527	0.429	**0.174**

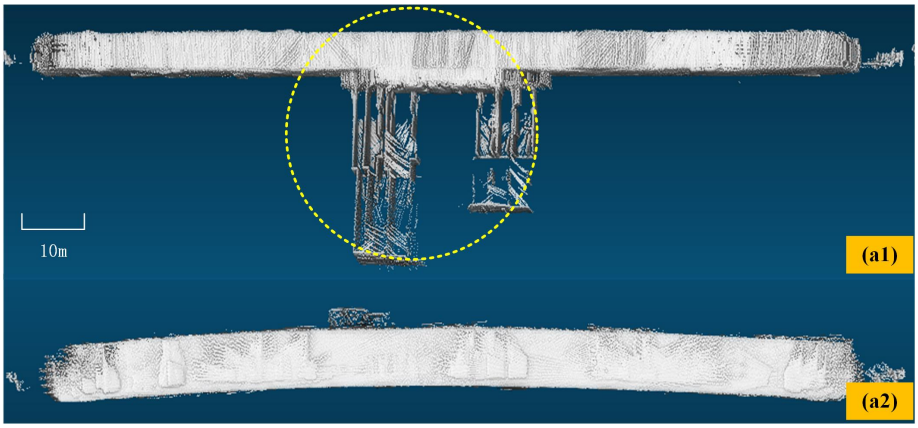


Figure 5.3: Map of LOAM in a long corridor data sequence. Upper image shows a top-down view and lower image shows the z-drift of the resulting map. We can see that Super Odometry is free of misalignment and gives a higher-quality map.

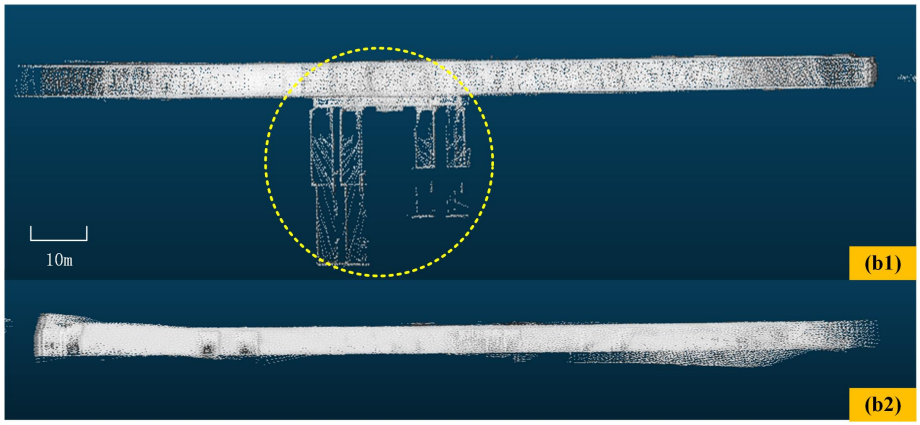


Figure 5.4: Map of LIO-SAM in a long corridor data sequence. Upper image shows a top-down view and lower image shows the z-drift of the result map. We can see that Super Odometry is free of misalignment and gives a higher-quality map.

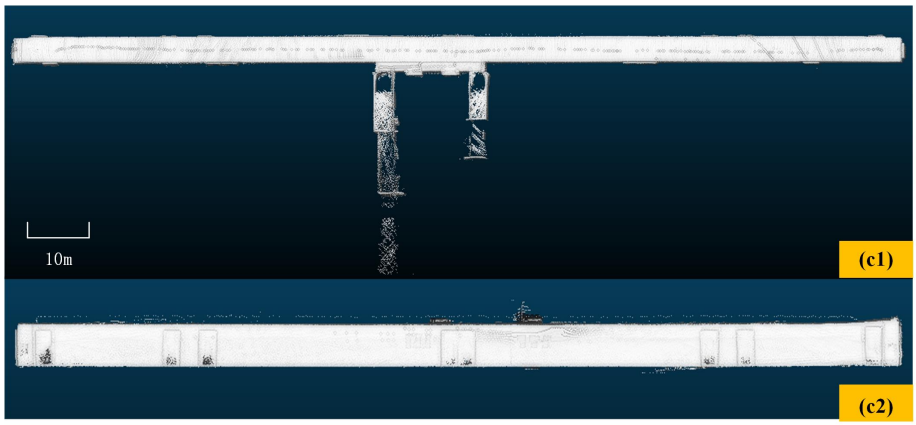


Figure 5.5: Map of Super Odometry in a long corridor data sequence. Upper image shows a top-down view and lower image shows the z-drift of the result map. We can see that Super Odometry is free of misalignment and gives a higher-quality map.

5.3.3 Robustness Comparison in Both Visually and Geometrically Degraded Environments

Here we ran the algorithms on the **Dust** dataset. This experiment shows the algorithms' abilities to overcome airborne obscurants. The dust introduces challenges to visual odometry by reducing the number of useful features in the image. It is also a problem for LiDAR odometry because it blocks the sensor field-of-view with dust particles and thus we obtain fewer geometric features. Fig.5.1(d) shows the large amount of dust generated from UAV's thrust in this experiment. Both vision-based methods (VINS-Mono and Depth-enhanced VINS) failed in this experiment as the camera provides limited information for state estimation in the dust.

From Fig.5.6 and Fig.5.7, we can see that both LOAM and LIO-SAM fail to create an accurate map of the scene. In Fig.5.6, we can observe that the map generated by LOAM has significant misalignment as shown in the yellow circles. This is mainly because LOAM relies heavily on LiDAR measurements and the IMU is only used to initialize the LiDAR scan matching. In the case of heavy dust, scan matching is likely to fail due to noisy returns. LIO-SAM performs better than LOAM, benefiting from its tightly coupled LiDAR-IMU architecture and reduced reliance on LiDAR measurements, although we can still observe map misalignment in the yellow circles in Fig.5.7. Super Odometry outperforms both LOAM and LIO-SAM in this experiment.

It gives an accurate and well-aligned map as shown in Fig.5.8. The main reason for the superior result is that Super Odometry adopts a feature extraction method based on principal component analysis and evaluates the quality of each feature. Therefore, if a feature has low quality (see Eq.(4.7,4.8)) it will be regarded as noise, and not be used in state estimation.

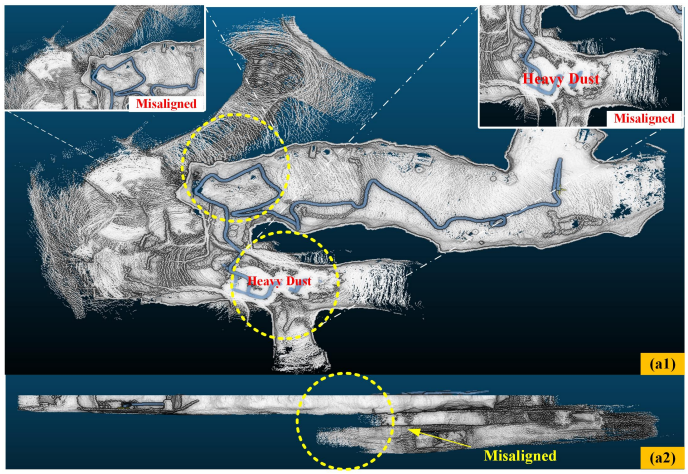


Figure 5.6: Map comparison of LOAM in the dust data sequence. Upper image shows a top-down view and lower image shows the z-drift of the result map. We can see that Super Odometry is free of misalignment and gives a higher quality map.

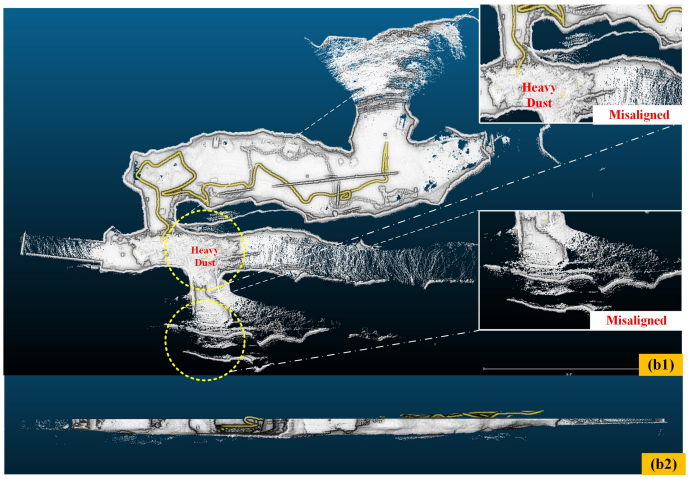


Figure 5.7: Map comparison of LIO-SAM in the dust data sequence. Upper image shows a top-down view and lower image shows the z-drift of the result map. We can see that Super Odometry is free of misalignment and gives a higher quality map.

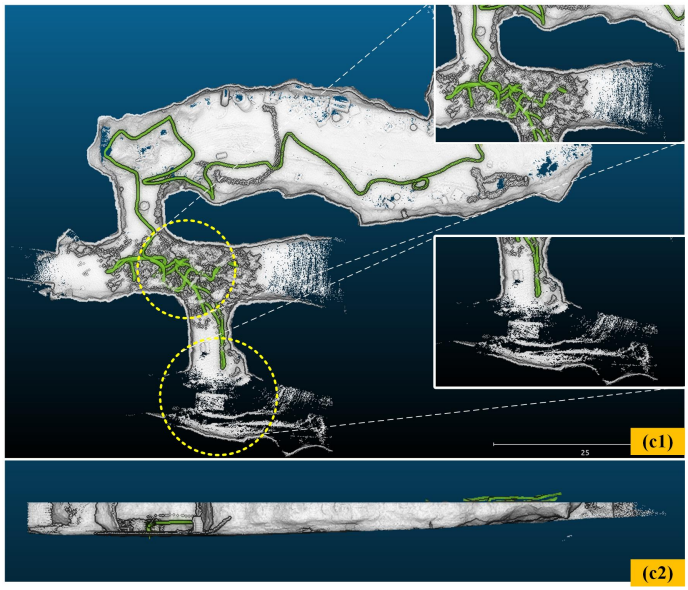


Figure 5.8: Map comparison of Super Odometry in the dust data sequence. Upper image shows a top-down view and lower image shows the z-drift of the result map. We can see that Super Odometry is free of misalignment and gives a higher quality map

5.4 Real-time Performance Evaluation

In this section, we show the real-time performance of Super Odometry and its ability to run efficiently on lightweight onboard systems. We benchmarked the algorithms on an 8-core Intel Core i7-4790K CPU. First, we evaluate the impact of using our dynamic octree instead of the traditional KD-tree. Then, we evaluate the processing time of each submodule.

5.4.1 Dynamic Octree Performance

In order to discuss the performance of our dynamic octree versus traditional KD-tree, we compared the insertion and query time of dynamic octree and KD-tree in outdoor environments.

Table 5.2: The average running time of Super Odometry in various dataset on desktop PC (with Intel Core i7-4790K CPU (ms))

Dataset	LIO	VIO	IMUODOM	Ours	LIO-SAM
Dust	21	bypass	0.8	21	102
Alpha Course	14	bypass	0.9	14	28
Dark Room	13	bypass	0.8	13	23
Long Corridor	14	28	0.8	28	24

5.4.2 Submodule Runtime

We evaluated each submodule processing time of Super Odometry and compared our LiDAR-inertial Odometry with popular LIO-SAM algorithm in various datasets shown as in Table 5.2. Overall, Super Odometry can achieve very high-frequency state estimation since all the submodules run in parallel. Instead of processing data in a sequential manner, it is important to note that super odometry achieves sensor fusion asynchronously. The optimizer of each submodule will cache the constraints and process them in small batches at its own frequency. Therefore, the total running time will be decided by the maximum submodule processing time, which significantly improves real-time performance.

IMU Odometry can directly output 1000 Hz state estimation. The LiDAR process module of Super Odometry takes a smaller amount of time than LIO-SAM. The difference is more significant in the **Dust** dataset, where LIO-SAM averages 102 ms while Super Odometry averages 21 ms. This is because in dusty environments, LiDAR scan matching is slow to converge, and Super Odometry can reject some unreliable constraints and better leverage the faster insertion and query of the dynamic octree.

5.5 Discussion

Robust state estimation in perceptually-degraded environments is very challenging due to the lack of reliable measurements. To solve this problem, we propose Super Odometry, an IMU-centric data processing pipeline, combining the advantages of a loosely coupled method with tightly coupled methods. To improve real-time performance, we propose to use the dynamic octree that significantly improves real-

time performance. The proposed method has been tested in both visually and geometrically degraded environments. The results show that our framework is robust to individual sensor failures and enables to achieve high-precision and resilient motion estimation in challenging real-world scenarios.

Chapter 6

Conclusion and Future Work

In this thesis, we push the limit of robustness of SLAM system resulting in high accuracy and efficient SLAM systems in various degraded environments. However, despite these improvements, there are still some unsolved problems for future direction.

Robust and Resilient Perception Based on robustness evaluation [99], many LiDAR SLAM systems struggle to perform effectively in degraded environments, such as structure-less corridors, or under extreme weather conditions like snow, fog, and dust. At a more fundamental level, current SLAM systems lack fundamental research in resilient algorithms and systems. Most of the SLAM systems just passively fuse all the measurements instead of actively adapting to challenging environments. The current system heavily relies on manual parameter tuning, it would be desirable to dynamically adjust depending on the scenarios. See [23] for a broader survey.

Robustness and Efficiency on Multi-sensor Fusion When integrating additional sensors with LiDAR, we inevitably acquire a large volume of sensor data, which often leads to redundancy. Striking a balance between data redundancy for robustness and maintaining lightweight computation remains an unresolved challenge. The robustness of sensor fusion, especially in the presence of inaccurate calibration and unsynchronized payloads, also significantly impacts the performance of SLAM systems. These compelling research questions still need to be addressed and improved.

Bibliography

[1] https://www.darpa.mil/program/darpa-subterranean-challenge. 3.1

[2] Chunge Bai, Tao Xiao, Yajie Chen, Haoqian Wang, Fang Zhang, and Xiang Gao. Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels. *IEEE Robotics and Automation Letters*, 7 (2):4861–4868, 2022. doi: 10.1109/LRA.2022.3152830. 3.2

[3] Tim D Barfoot, Chi Hay Tong, and Simo Särkkä. Batch continuous-time trajectory estimation as exactly sparse gaussian process regression. In *Robotics: Science and Systems*, volume 10, pages 1–10. Citeseer, 2014. 2.1.1

[4] Jens Behley, Volker Steinhage, and Armin B. Cremers. Efficient Radius Neighbor Seach in Three-dimensional Point Clouds. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015. 4.3.3

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992. 2.1.3, 2.1.3

[6] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003. 2.1.3

[7] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 2016. 1.1.2, 1.2.2, 3.1, 3.2, 3.2, **??**, 3.3.3

[8] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6): 1874–1890, 2021. 3.2

[9] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):

1874–1890, 2021. doi: 10.1109/TRO.2021.3075644. 3.1

[10] Marco Camurri, Milad Ramezani, Simona Nobili, and Maurice Fallon. Pronto: a multi-sensor state estimator for legged robots in real world scenarios. *Frontiers in Robotics and AI*, 7:68, 2020. 1.2.1

[11] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016. **??**

[12] Kenneth Chaney, Fernando Cladera, Ziyun Wang, Anthony Bisulco, M Ani Hsieh, Christopher Korpela, Vijay Kumar, Camillo J Taylor, and Kostas Daniilidis. M3ed: Multi-robot, multi-sensor, multi-environment event dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4015–4022, 2023. **??**

[13] Kenny Chen, Brett T. Lopez, Ali-akbar Agha-mohammadi, and Ankur Mehta. Direct lidar odometry: Fast localization with dense point clouds. *IEEE Robotics and Automation Letters*, 7(2):2000–2007, 2022. doi: 10.1109/LRA.2022.3142739. 3.2

[14] Long Chen, Yuchen Li, Chao Huang, Bai Li, Yang Xing, Daxin Tian, Li Li, Zhongxu Hu, Xiaoxiang Na, Zixuan Li, et al. Milestones in autonomous driving and intelligent vehicles: Survey of surveys. *IEEE Transactions on Intelligent Vehicles*, 8(2):1046–1056, 2022. 3.2

[15] Weichen Dai, Yu Zhang, Shenzhou Chen, Donglei Sun, and Da Kong. A multi-spectral dataset for evaluating motion estimation systems. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5560–5566. IEEE, 2021. 3.2

[16] Frank Dellaert, Michael Kaess, et al. Factor graphs for robot perception. *Foundations and Trends® in Robotics*, 6(1-2):1–139, 2017. 3.2

[17] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure, 2021. 2.1.3

[18] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE, 2022. 2.1.1

[19] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, 2019. 1.2.2, 3.1, 3.2, **??**

[20] David Droeschel and Sven Behnke. Efficient continuous-time slam for 3d lidar-based online mapping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5000–5007. IEEE, 2018. 2.1.1

[21] Rodolphe Dubois, Alexandre Eudes, and Vincent Frémont. Airmuseum: a heterogeneous multi-robot dataset for stereo-visual and inertial simultaneous localization and mapping. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 166–172, 2020. doi: 10.1109/MFI49285.2020.9235257. 3.2

[22] Kamak Ebadi, Yun Chang, Matteo Palieri, Alex Stephens, Alex Hatteland, Eric Heiden, Abhishek Thakur, Nobuhiro Funabiki, Benjamin Morrell, Sally Wood, Luca Carlone, and Ali akbar Agha-mohammadi. LAMP: large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments, 2020. 1.1.1

[23] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. Present and future of slam in extreme environments: The darpa subt challenge. *IEEE Transactions on Robotics*, 2023. 6

[24] Naser El-Sheimy, Haiying Hou, and Xiaoji Niu. Analysis and modeling of inertial sensors using allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, 2008. doi: 10.1109/TIM.2007.908635. 3.3.1

[25] Dapeng Feng, Yuhua Qi, Shipeng Zhong, Zhiqiang Chen, Yudu Jiao, Qiming Chen, Tao Jiang, and Hongbo Chen. S3e: A large-scale multimodal dataset for collaborative slam. *arXiv preprint arXiv:2210.13723*, 2022. 3.2

[26] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Georgia Institute of Technology, 2015. 4.3.1

[27] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual–inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016. 2.4.2

[28] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. **??**

[29] Wenliang Gao. A ros package tool to analyze the imu performance, 2018. URL `https://github.com/gaowenliang/imu_utils`. 3.3.1

[30] 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. 1.1.2, 3.1

[31] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1.2.2, 3.2, 3.2, **??**

[32] Johannes Graeter, Alexander Wilczynski, and Martin Lauer. LIMO: lidar-monocular visual odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7872–7879. IEEE, 2018. 1.1.1

[33] Timo Hackel, Jan D Wegner, and Konrad Schindler. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS annals of the photogrammetry, remote sensing and spatial information sciences*, 3:177–184, 2016. 4.3.2

[34] Dongjiao He, Wei Xu, Nan Chen, Fanze Kong, Chongjian Yuan, and Fu Zhang. Point-lio: Robust high-bandwidth light detection and ranging inertial odometry. *Advanced Intelligent Systems*, page 2200459, 2023. 3.2

[35] Michael Helmberger, Kristian Morin, Beda Berner, Nitish Kumar, Giovanni Cioffi, and Davide Scaramuzza. The hilti slam challenge dataset. *IEEE Robotics and Automation Letters*, 7(3):7518–7525, 2022. 1.2.2, 3.1, 3.2, **??**

[36] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1271–1278. IEEE, 2016. 2.1.4

[37] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis. Complementary multi–modal sensor fusion for resilient robot pose estimation in subterranean environments. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1024–1029, 2020. doi: 10.1109/ICUAS48674.2020.9213865. 1.2.1

[38] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018. 3.2

[39] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized gicp for fast and accurate 3d point cloud registration. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11054–11059. IEEE, 2021. 2.1.3

[40] Kurt Konolige, Giorgio Grisetti, Rainer Kümmerle, Wolfram Burgard, Benson Limketkai, and Regis Vincent. Efficient sparse pose adjustment for 2d mapping. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 22–29. IEEE, 2010. 2.1.4

[41] Alex Junho Lee, Younggun Cho, Young-sik Shin, Ayoung Kim, and Hyun Myung. Vivid++: Vision for visibility dataset. *IEEE Robotics and Automation*

Letters, 7(3):6282–6289, 2022. 3.2

[42] Keith YK Leung, Yoni Halpern, Timothy D Barfoot, and Hugh HT Liu. The utias multi-robot cooperative localization and mapping dataset. *The International Journal of Robotics Research*, 30(8):969–974, 2011. doi: 10.1177/0278364911398404. URL https://doi.org/10.1177/0278364911398404. 3.2

[43] Hyungtae Lim, Suyong Yeon, Suyong Ryu, Yonghan Lee, Youngji Kim, Jaeseong Yun, Euigon Jung, Donghwan Lee, and Hyun Myung. A single correspondence is enough: Robust global registration to avoid degeneracy in urban environments. page Accepted. To appear, 2022. 3.2

[44] Yicheng Lin, Shuo Wang, Yunlong Jiang, and Bin Han. Breaking of brightness consistency in optical flow with a lightweight cnn network, 2023. 3.2

[45] Zheng Liu, Xiyuan Liu, and Fu Zhang. Efficient and consistent bundle adjustment on lidar point clouds. *IEEE Transactions on Robotics*, 2023. 3.2

[46] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4:333–349, 1997. 2.1.4

[47] Jiajun Lv, Kewei Hu, Jinhong Xu, Yong Liu, Xiushui Ma, and Xingxing Zuo. Clins: Continuous-time trajectory estimation for lidar-inertial system. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6657–6663. IEEE, 2021. 2.1.1

[48] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017. ??

[49] Martin Magnusson. *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection.* PhD thesis, Örebro universitet, 2009. 2.1.3

[50] Martin Magnusson, Andreas Nuchter, Christopher Lorken, Achim J Lilienthal, and Joachim Hertzberg. Evaluation of 3d registration reliability and speed-a comparison of icp and ndt. In *2009 IEEE International Conference on Robotics and Automation*, pages 3907–3912. IEEE, 2009. 2.1.3

[51] Martin Magnusson, Narunas Vaskevicius, Todor Stoyanov, Kaustubh Pathak, and Andreas Birk. Beyond points: Evaluating recent 3d scan-matching algorithms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3631–3637. IEEE, 2015. 2.1.3

[52] Anastasios I. Mourikis and Stergios I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007. doi: 10.1109/ROBOT.2007.364024. 1.2.1

[53] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 3.2

[54] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics*, 34(6):1425–1440, 2018. 3.3.3

[55] Thien Hoang Nguyen, Shenghai Yuan, and Lihua Xie. Vr-slam: A visual-range simultaneous localization and mapping system using monocular camera and ultra-wideband sensors, 2023. 3.2

[56] Edwin Olson. M3rsm: Many-to-many multi-resolution scan matching. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5815–5821. IEEE, 2015. 2.1.4

[57] Matteo Palieri, Benjamin Morrell, Abhishek Thakur, Kamak Ebadi, Jeremy Nash, Arghya Chatterjee, Christoforos Kanellakis, Luca Carlone, Cataldo Guaragnella, and Ali-akbar Agha-mohammadi. LOCUS: a multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time. *IEEE Robotics and Automation Letters*, 6(2):421–428, Apr 2021. ISSN 2377-3774. doi: 10.1109/lra.2020.3044864. URL http://dx.doi.org/10.1109/LRA.2020.3044864. 1.1.1

[58] Yue Pan, Pengchuan Xiao, Yujie He, Zhenlei Shao, and Zesong Li. Mulls: Versatile lidar slam via multi-metric linear least square. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11633–11640. IEEE, 2021. (document), 2.1, 2.1.3

[59] Xiongfeng Peng, Zhihua Liu, Weiming Li, Ping Tan, SoonYong Cho, and Qiang Wang. Dvi-slam: A dual visual inertial slam network, 2023. 3.2

[60] Bernd Pfrommer, Nitin Sanket, Kostas Daniilidis, and Jonas Cleveland. Penncosyvio: A challenging visual inertial odometry benchmark. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3847–3854. IEEE, 2017. 3.1, ??

[61] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: a robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34 (4):1004–1020, 2018. 1.1.1, 5.3

[62] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34 (4):1004–1020, 2018. doi: 10.1109/TRO.2018.2853729. 3.1, 3.2

[63] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. The newer college dataset: Handheld lidar, inertial

and vision with ground truth. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4353–4360. IEEE, 2020. 3.1

[64] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016. 3.3.1, 3.3.1

[65] Victor Reijgwart, Alexander Millane, Helen Oleynikova, Roland Siegwart, Cesar Cadena, and Juan Nieto. Voxgraph: globally consistent, volumetric mapping using signed distance function submaps. *IEEE Robotics and Automation Letters*, 5(1):227–234, Jan 2020. ISSN 2377-3774. doi: 10.1109/lra.2019.2953859. URL http://dx.doi.org/10.1109/LRA.2019.2953859. 1.2.1

[66] John G Rogers, Arthur Schang, Carlos Nieto-Granda, John Ware, John Carter, Jonathan Fink, and Ethan Stump. The darpa subt urban circuit mapping dataset and evaluation metric. In *Experimental Robotics: The 17th International Symposium*, pages 391–401. Springer, 2021. **??**

[67] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2.2

[68] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009. 2.1.3

[69] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 2.1.3

[70] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4):80–92, 2011. (document), 2.2

[71] Sebastian Scherer, Vasu Agrawal, Graeme Best, Chao Cao, Katarina Cujic, R Darnley, R DeBortoli, E Dexheimer, B Drozd, R Garg, et al. Resilient and modular subterranean exploration with a team of roving and flying robots. *Field Robotics*, 2022. 3.3.1

[72] David Schubert, Thore Goll, Nikolaus Demmel, Vladyslav Usenko, Jörg Stückler, and Daniel Cremers. The tum vi benchmark for evaluating visual-inertial odometry. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1680–1687. IEEE, 2018. 1.2.2, 3.1, 3.2, **??**

[73] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009. 2.1.3, 3.3.1, 3.3.2

[74] Tobias Senst, Volker Eiselein, and Thomas Sikora. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1377–1387, 2012. 2.2

[75] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020. 2.1.1, 3.1

[76] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. LIO-SAM: tightly-coupled lidar inertial odometry via smoothing and mapping, 2020. 1.1.1, 1.1.1, 5.3

[77] Weizhao Shao, Srinivasan Vijayarangan, Cong Li, and George Kantor. Stereo visual inertial lidar simultaneous localization and mapping, 2019. 1.2.1

[78] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 3.2

[79] Eijiro Takeuchi and Takashi Tsubouchi. A 3-d scan matching using improved 3-d normal distributions transform for mobile robotic mapping. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3068–3073. IEEE, 2006. 2.1.3

[80] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 3.1

[81] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3): 52–57, 2002. 2.1.1

[82] Yulun Tian, Yun Chang, Long Quang, Arthur Schang, Carlos Nieto-Granda, Jonathan P How, and Luca Carlone. Resilient and distributed multi-robot visual slam: Datasets, experiments, and lessons learned. *arXiv preprint arXiv:2304.04362*, 2023. 3.2

[83] Ignacio Vizzo, Tiziano Guadagnino, Benedikt Mersch, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036, 2023. 2.1.1, 2.1.3, 3.2

[84] Chen Wang, Dasong Gao, Kuan Xu, Junyi Geng, Yaoyu Hu, Yuheng Qiu, Bowen Li, Fan Yang, Brady Moon, Abhinav Pandey, Aryan, Jiahe Xu, Tianhao Wu, Haonan He, Daning Huang, Zhongqiang Ren, Shibo Zhao, Taimeng Fu, Pranay Reddy, Xiao Lin, Wenshan Wang, Jingnan Shi, Rajat Talak, Kun Cao, Yi Du, Han Wang, Huai Yu, Shanzhao Wang, Siyu Chen, Ananth Kashyap, Rohan Bandaru, Karthik Dantu, Jiajun Wu, Lihua Xie, Luca Carlone, Marco Hutter,

and Sebastian Scherer. PyPose: A library for robot learning with physics-based optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/pdf/2209.15428.pdf. 3.3.3

[85] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual SLAM. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. 1.2.2, 3.2, ??, 3.4.1

[86] Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In *Conference on Robot Learning*, pages 1761–1772. PMLR, 2021. 3.1

[87] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020. 3.2

[88] David Wisth, Marco Camurri, Sandipan Das, and Maurice Fallon. Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry. *IEEE Robotics and Automation Letters*, 6(2):1004–1011, Apr 2021. ISSN 2377-3774. doi: 10.1109/lra.2021.3056380. URL http://dx.doi.org/10.1109/LRA.2021.3056380. 1.2.1

[89] Yao Xiao, Xiaogang Ruan, Jie Chai, Xiaoping Zhang, and Xiaoqing Zhu. Online imu self-calibration for visual-inertial systems. *Sensors*, 19(7):1624, 2019. (document), 2.3

[90] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. FAST-LIO2: fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, abs/2107.06829, 2022. 3.2

[91] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38(4):2053–2073, 2022. 3.2

[92] Chongjian Yuan, Wei Xu, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. Efficient and probabilistic adaptive voxel mapping for accurate online lidar odometry. *IEEE Robotics and Automation Letters*, 7(3):8518–8525, 2022. 2.1.2

[93] Ji Zhang and Sanjiv Singh. LOAM: lidar odometry and mapping in real-time. In *Proceedings of Robotics: Science and Systems (RSS '14)*, July 2014. 1.1.1, 1.1.1, 2.1.3, 2.1.4, 5.3

[94] Ji Zhang and Sanjiv Singh. Laser–visual–inertial odometry and mapping with high robustness and low drift. *Journal of Field Robotics*, 35(8):1242–1264, 2018. 1.2.1

[95] Lintong Zhang, Marco Camurri, and Maurice Fallon. Multi-camera lidar inertial extension to the newer college dataset. *arXiv preprint arXiv:2112.08854*, 2021. ??

[96] Shibo Zhao, Zheng Fang, HaoLai Li, and Sebastian Scherer. A robust laser-inertial odometry and mapping method for large-scale highway environments. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019. doi: 10.1109/iros40897.2019.8967880. URL http://dx.doi.org/10.1109/IROS40897.2019.8967880. 1.1.1, 1.1.1

[97] Shibo Zhao, Peng Wang, Hengrui Zhang, Zheng Fang, and Sebastian Scherer. TP-TIO: a robust thermal-inertial odometry with deep thermalpoint, 2020. 1.1.1

[98] Shibo Zhao, Hengrui Zhang, Peng Wang, Lucas Nogueira, and Sebastian Scherer. Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8729–8736. IEEE, 2021. 1.1.1, 2.1.1, 3.3.1

[99] Shibo Zhao, Yuanjun Gao, Tianhao Wu, Damanpreet Singh, Rushan Jiang, Haoxiang Sun, Mansi Sarawata, Yuheng Qiu, Warren Whittaker, Ian Higgins, Yi Du, Shaoshu Su, Can Xu, John Keller, Jay Karhade, Lucas Nogueira, Sourojit Saha, Ji Zhang, Wenshan Wang, Chen Wang, and Sebastian Scherer. Subt-mrs dataset: Pushing slam towards all-weather environments, 2023. 1.1.2, 1.2.2, 6

[100] Peng Wang Zheng Fang, Shibo Zhao. Vanishing Point Aided LiDAR-Visual-Inertial Estimator. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 5.3

[101] David Zuñiga-Noël, Alberto Jaenal, Ruben Gomez-Ojeda, and Javier Gonzalez-Jimenez. The uma-vi dataset: Visual–inertial odometry in low-textured and dynamic illumination environments. *The International Journal of Robotics Research*, 39(9):1052–1060, 2020. 1.2.2, 3.1, 3.2, ??

[102] Xingxing Zuo, Yulin Yang, Patrick Geneva, Jiajun Lv, Yong Liu, Guoquan Huang, and Marc Pollefeys. LIC-Fusion 2.0: lidar-inertial-camera odometry with sliding-window plane-feature tracking, 2020. 1.2.1