Robust Odometry and Localisation in Challenging Environments with Vision and Lidar

Lintong Zhang

St Annes College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

2024

Abstract

Odometry and localisation are two crucial modules for autonomous robot navigation. Numerous studies have demonstrated various systems operating in both indoor and outdoor environments. Despite the progress in accuracy and robustness, there remains significant potential to further enhance odometry and localisation systems, especially in challenging environments.

This thesis proposes several methods to enhance odometry and localisation systems utilising camera and lidar sensors, both individually and in combination. The aim is to address common failure modes and explore novel approaches for improvement.

The first contribution of this thesis is a multi-camera visual odometry system that integrates multiple cameras and an IMU within a factor graph framework. By tracking features across cameras and selecting a subset of features, the system achieves enhanced accuracy and robustness, able to operate in challenging environments such as underground mines and narrow spaces. The second contribution is a global lidar localisation system that leverages semantics and object instances. This work was the first to use panoptic segmentation directly on 3D lidar scans for indoor localisation. The system constructs a map with object instances and semantic classes, and estimates the lidar sensor poses online. The third contribution is a cross-modal global visual localisation system capable of operating with input maps from various 3D map representations. By rendering synthetic images from 3D maps, this system can localize monocular images and estimate 6 DoF poses. The fourth contribution is two datasets with multi-sensor modalities, aimed at promoting operations in challenging environments. The Hilti-Oxford datasets introduce a new method for obtaining millimetre-accurate ground truth poses and serve as the foundation for a major international competition.

This thesis particularly focuses on developing real-time operating systems, with each proposed algorithm capable of running on a mobile laptop. These algorithms have been extensively evaluated using real-world data and have demonstrated effective performance with both handheld devices and legged robot platforms.

Robust Odometry and Localisation in Challenging Environments with Vision and Lidar

Lintong Zhang

St Annes College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

2024

Declaration

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. I declare that this thesis is entirely my own work, and except where otherwise stated, describes my own research.

Lintong Zhang, St Anne's College

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Maurice Fallon. Your guidance and encouragement have been key to completing this thesis. Your feedback and patience helped me tackle the challenges of my research with confidence. I've really enjoyed our discussions and the camaraderie we've shared along the way. I'm especially thankful for your support in offering opportunities to attend international conferences and ensure my research could be shared outside of the ORI. I am looking forward to continuing our work together and making further contributions to the field of robotics.

I am also immensely grateful to my collaborators throughout the years: David Wisth, Marco Camurri, Tejaswi Digumarti, Georgi Tinchev, Michael Helmberger, Frank Fu, Ethan Tao, Jiarong Lin, and many more. Each of you has contributed significantly to my growth as a researcher. From brainstorming innovative ideas during late at night to tackling tough deadlines with determination, I have learned a great deal from each one of you. Your edits, reviews and diverse perspectives have enriched my work, and I cherish the professional and personal bonds we have formed.

To my DRS lab and ORI colleagues, Matias Mattamala, Ricardo Cannizzaro, Russell Buchanan, Christina Kassab, Joseph Rowell, Nived Chebrolu, and many more, your good humour and day to day support have made the last four years genuinely enjoyable. Together, we have learned new topics from the Autonomy Tea Talks, and spent hours giving feedback on each other's research. Our many social events, in addition to the DPhil work, have certainly kept me busy. I am grateful for the support network we have built and these friendships that will last many years after ORI.

I also extend my heartfelt thanks to the hardware, software, and admin teams, Wayne Tubby, Benoit Casseau, Elsa Lam, and many more friends and colleagues. You have created a friendly and vibrant environment, rich with activities and fun events, making ORI the best second home over the past years. I always enjoyed the bi-weekly ORI seminars, workshops, and afternoon tea and cake sessions. Your administrative and technical support has also been crucial in navigating the logistical challenges of my research projects, ensuring that I had the resources, assistance, and good company needed to successfully conduct my work.

My time at St Anne's College has been made wonderful not only by the comforting food but also by the numerous MCR events that have allowed me

to make friends and enrich my life outside of academia. Having access to the College, especially during Covid times from 2020-2022, helped keep my sanity and sense of community. Thank you for the opportunity also to meet so many other students and academics in a variety of fields.

I am grateful to my examiners, Tim Barboot and Daniele De Martini, for taking the time to read the thesis and providing insightful discussions on the topics.

Finally, I would like to thank my family, especially my wife, Rosanna Jackson. Your unconditional support during the toughest times has served as my rock. Your patience, understanding, and constant encouragement have given me the strength to persevere through the challenges. I could not have completed this journey without you. I also want to express my deepest gratitude to my parents. Your endless support, sacrifices, and belief in my abilities have been the foundation of my DPhil journey. Your encouragement and guidance have shaped me into the person I am today, and for that, I am always thankful.

To all of you, my heartfelt thanks.

Contents

List of Abbreviations

CNNs	Convolutional Neural Networks
DNNs	Deep neural networks
DoF	Degrees of Freedom
FoV	Field-of-View
GD	Gradient Descent
GPS	Global Positioning System
GPU	Graphics Processing Unit
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
LS	Least Squares
MAP	Maximum A Posteriori
MLPs	Multilayer Perceptrons
NeRF	Neural Radiance Fields
PnP	Perspective-n-Points
RGB	Red-Green-Blue
RGB-D	RGB-Depth
SLAM	Simultaneous Localisation And Mapping
TLS	Terrestrial Laser Scanning

1 Introduction

1.1 Motivation

Odometry and localisation are fundamental capabilities needed for mobile robots to navigate their environment and make decisions. There have been many studies of vision [44, 49], lidar [151, 145], and radar-based [2, 17] methods. Odometry is the specific problem of tracking and estimating a robot's poses with high frequency and low latency, providing an initial estimate of the trajectory. Localisation, or place recognition, can determine the robot's position in a fixed prior map, which is also known as the kidnapped robot problem. In the application of Simultaneous Localisation And Mapping (SLAM), where a robot determines its pose while building a map of its environment, localisation techniques be applied to reduce drift by recognising a previously visited location.

Despite numerous studies demonstrating pose estimation methods in controlled environments [40, 110], real-world conditions present challenges that remain under-explored. For instance, transitioning from one room to another can disrupt vision and lidar-based methods due to abrupt changes in scene observations. Sudden illumination changes can cause exposure issues, leading to a reduction in estimation accuracy. Moreover, narrow and dimly lit spaces, such as stairways and underground areas pose additional challenges to the reliability of current SLAM systems. In

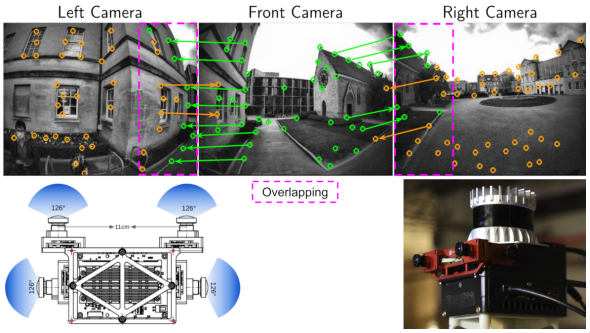
response, there is a growing focus on enhancing SLAM systems for long-term operation in challenging environments [3, 155].

While cameras and lidar sensors are commercially available and widely utilized for localisation and odometry, they function differently. Cameras are passive optical instruments that capture visual images, while lidar is an active sensor that uses laser light to measure distances. Although these technologies have been extensively tested in well-controlled environments, there is still the need to explore improved methods to enhance localisation techniques of these sensors, especially in challenging conditions [130]. This includes integrating more semantic information into scenes and prior maps, which enriches the understanding of objects within the map and potentially facilitates more accurate and resilient localisation and mapping processes.

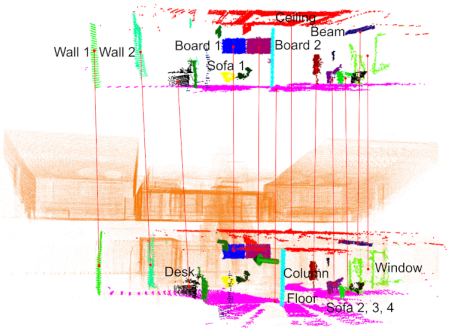
This thesis seeks to exploit the capabilities of vision and lidar sensors, either individually or in combination, to develop robust odometry and localisation solutions tailored to challenging environments. The objective is to devise and assess improved methodologies to contribute to odometry, localisation, and the broader SLAM problem. One aspect that will be explored is using multiple cameras to create redundancy and improve the accuracy of odometry. Another aspect to be examined is incorporating a deeper semantic understanding into maps, aiming to enhance the robustness and accuracy of localisation. The proposed algorithms are intended for use on both handheld devices and legged robots, with evaluations presented by testing their performance under challenging real-world conditions.

1.2 Contributions

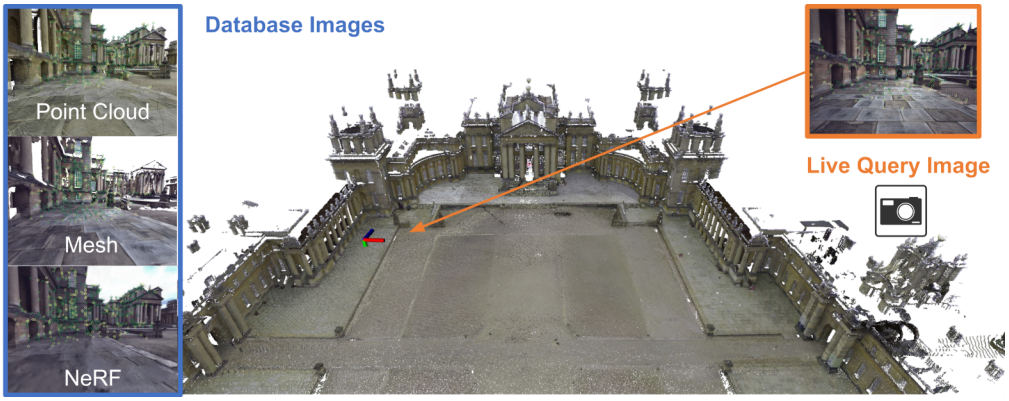
The contributions of this thesis are centred on the enhancement of modules relevant to robot odometry and localisation, specifically designed to address common failure cases and to facilitate operation in challenging real-world environments. Key contributions are highlighted in Figure 1.1. The specifics of these contributions are detailed as follows:



(a) Multi camera visual odometry in Chapter 4



(b) Learned lidar localisation from object instances in Chapter 5



(c) Visual localisation in colour 3D maps in Chapter 6

Figure 1.1: Some highlighted work in this thesis which is presented in later chapters.

- Design and implementation of a multi-camera visual-inertial odometry system that is capable of running in real-time on a laptop CPU. The system fuses as many as four separate camera streams facing in different directions, and can continue operating in challenging and degenerate scenarios. This method was demonstrated in diverse scenarios including outdoor spaces, narrow stairwells, and poorly lit underground mines.

- A learning-based lidar localisation system that can localise in a prior map with a single lidar scan through learning and matching object instances. The system can run in real-time on a mobile Graphics Processing Unit (GPU), and can also act as a loop closure mechanism in a wider lidar SLAM system.

- A cross modal global visual localisation system that can operate with input maps from three very different types of 3D map representations. The system

can localise a monocular image in colour 3D maps. It operates in real-time on a mobile laptop equipped with a GPU with a processing rate of 1 Hz.

- Multi-sensor SLAM datasets with millimetre-accurate ground truth poses, recorded in challenging environments and containing common failure cases. This dataset was an extensive undertaking that formed the basis of a major international competition jointly run with the Hilti Corporation – the HILTI SLAM Challenge 2022.

1.3 Publications

The previously described research efforts have been materialised through four first-author peer-reviewed publications developed during this project, one of which is under review at present. Additional contributions include four non-first-author peer-reviewed articles, with one article under review as well.

1.3.1 First-author Publications

1. **Zhang, L.**, Wisth, D., Camurri, M., Fallon, M., 2022. Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry. *IEEE Robotics and Automation Letters (RA-L).*

2. **Zhang, L.**, Helmberger, M., Fu, L., D. Wisth, Camurri, M.,Scaramuzza, D., Fallon, M. 2023. Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping. *IEEE Robotics and Automation Letters (RA-L).*

3. **Zhang, L.**, Digumarti, T., Tinchev, G., Fallon, M. 2023. InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning. *Robotics Science and Systems (RSS).*

4. **Zhang, L.**, Tao, Y., Lin, J., Fallon, M. 2024. Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations. *IEEE Transactions on Robotics (T-RO) - under review.*

1.3.2 Co-authored Publications

1. Tranzatto, M., Dharmadhikari, M., Bernreiter, *et al.*, including **Zhang, L.** (2023). Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned. *Field Robotics Journal.* [**Tranzatto2023**]

2. Kassab, C., Mattamala, M., **Zhang L.**, Fallon, M. (2024). Language-Extended Indoor SLAM (LEXIS): A Versatile System for Real-Time Visual Scene Understanding. *IEEE International Conference on Robotics and Automation (ICRA).* [67]

3. Rowell, J., **Zhang L.**, Fallon, M. (2024). LiSTA: Geometric Object-Based Change Detection in Cluttered Environments. *IEEE International Conference on Robotics and Automation (ICRA).* [115]

4. Wang, J., Mattamala, M., Kassab, C., **Zhang, L.**, Fallon, M. (2024) Exosense: A Vision-Centric Scene Understanding System For Safe Exoskeleton Navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - under review.* [138]

1.4 Thesis Outline

This thesis adheres to the *integrated format* prescribed by the University of Oxford. Chapters 4 through 7 comprise peer-reviewed publications, each supplemented with additional discussion and a Statement of Authorship. This statement explicitly declares the author's contributions to each work, ensuring clarity and transparency.

The structure of the overall thesis is outlined as follows:

- **Chapter 2 – Background:** Presents main definitions, theory, and methods used in the thesis.

- **Chapter 3 – Related Work:** Reviews the existing literature pertinent to the research topics of camera and lidar-based odometry and localisation methods. This chapter identifies gaps in the current research and justifies the need for the research carried out in this thesis.

- **Chapter 4 – Multi-camera Visual Odometry:** Outlines an approach to fuse multiple cameras and an Inertial Measurement Unit (IMU) in odometry estimation that brings robustness and improved accuracy in challenging environments.
- **Chapter 5 – Learned Lidar Localisation:** Bridges the gap in 3D lidar localisation for indoor environments by learning and matching object instances in a lidar scan.
- **Chapter 6 – Visual Localisation in 3D Maps:** Develops cross modal localisation by matching a monocular image with synthetic database images from colour 3D maps.
- **Chapter 7 – SLAM Dataset in Challenging Environment:** Presents two datasets that assist the development of the algorithms presented in this thesis and have also become a public benchmark for the wider SLAM community.
- **Chapter 8 – Conclusion:** Summarises the key findings of the research, and suggests areas for potential future research directions.

2
Background

This chapter introduces the foundational concepts and methodologies that underpin the research presented in this thesis. It begins with an overview of basic notation and mathematical definitions in Sec. 2.1. Sec. 2.2 delves into the principal techniques of factor graph optimisation, which is the core algorithm employed in Chapter 4. Sec. 2.3 gives the background knowledge on neural networks, which forms the foundational basis for Chapter 5 and 6. Sec. 2.4 explores a number of key topics in the camera sensor, discussing various aspects that are used throughout this thesis, including intrinsic and extrinsic properties, feature detection, and Perspective-n-Points (PnP) pose estimation algorithms. Lastly, Sec. 2.5 dives into the details of the lidar sensor, introducing various map representations, and it lays out an example pipeline of a modern deep learning network for the task of instance segmentation.

2.1 Preliminaries

2.1.1 Notation

This thesis adopts specific typographic conventions to distinguish between various mathematical objects such as vectors, matrices, and reference frames. These conventions are summarised in Tab. 2.1.

Table 2.1: Notation used for mathematical objects.

Quantity	Description	Example
Scalars	Upper/Lowercase italic	The feature match confidence score w
Vectors	Lowercase bold	The 3D point $\mathbf{p}$
Matrices	Uppercase bold	The rotation matrix $\mathbf{R}$
Sets/Manifolds	Uppercase calligraphic	The map points $\mathcal{P}$
Reference Frames	Uppercase typewriter	The *map* frame `M`
	(legacy expression)	The *world* frame $\underrightarrow{\mathcal{F}}_W$

2.1.2 Rotations and Poses

Rotation matrices and rigid-body transformations are heavily used in robotics, as they are commonly employed to describe the positioning and movement of objects in space. In this thesis, we use the formalisation of *Lie groups* to represent these transformations.

Lie groups are defined concurrently as a *group* and a *differentiable manifold*. As a group, they satisfy four fundamental criteria: (1) they possess a binary operation, (2) this operation is associative, (3) there exists an identity element, and (4) each element has an inverse. Specifically, for rotation and rigid-body matrices: (1) the binary operation is defined by matrix multiplication, which also satisfies (2); (3) the identity element is the identity matrix $\mathbf{I}$; and (4) the inverse of each element is given by its matrix inverse.

The differentiable manifold aspect of Lie groups embodies the smooth geometric constraints that delineate valid group members. For instance, a 3D orientation, which encompasses 3 Degrees of Freedom (DoF), can be represented using a 3D vector via Euler angles or an axis-angle representation [6]. However, not all vectors in $\mathbb{R}^3$ constitute valid orientations, and the specific constraints necessary for valid Euler angles are not inherently represented in these models. In contrast, while a rotation matrix—consisting of nine entries—is an over-parameterised representation, it incorporates orthonormal constraints. This structure allows for an effective portrayal of the intrinsic 3 DoF of a valid orientation in a manner that smoothly defines the manifold.

This section outlines the fundamental definitions employed throughout the thesis. Detailed explanations of additional concepts and methods are provided in the respective chapters.

Rotation Body Transform

These refer to transformations that preserve the shape and size of objects. Typically, these include rotations and translations but exclude scaling or shearing.

Rotation matrices are formally categorised within the Special Orthogonal Group, or SO(n). This group comprises all matrices $\mathbf{R}$ that adhere to the orthonormality condition $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$ and have a determinant of $\det \mathbf{R} = 1$.

The group of rigid-body matrices is identified as the *Special Euclidean Group*, or SE(n). These matrices encapsulate a relative transformation encompassing 6 DoF, representing both translation and rotation. The typical matrix structure for SE(n) is outlined as follows:

2D Case We start with a simple 2 dimensional case, where an arbitrary point (x, y) is rotated by θ and translated by $\mathbf{t}$, the SO(2) rotation matrix is the following:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{2.1}$$

The following equation represents the transformation of the point, where (x', y') is the new point after the transformation.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{t} + \mathbf{R} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.2}$$

The above relationship can be more succinctly expressed by introducing homogeneous coordinates, which involves adding a third coordinate set to 1. This allows the translation and rotation components to be combined into a single matrix.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & t_x \\ \sin\theta & \cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{2.3}$$

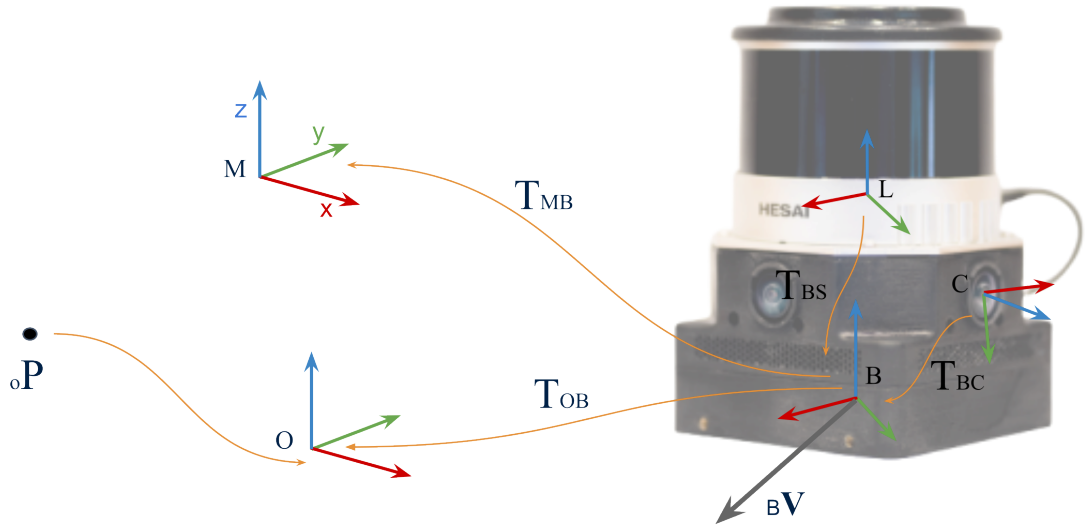


Figure 2.1: Primary reference frames and the notation associated with them which are used throughout this thesis. We adhere to the colour convention for axes, designating the x-axis in **red**, the y-axis in **green**, and the z-axis in **blue**. The odom frame `O` and the map frame `M` remain fixed, whereas the base frame `B` moves in conjunction with the robot. The camera frame `C` and lidar frame Ł is rigidly linked to `B` via the rigid-body transformation $\mathbf{T}_{\texttt{BC}}$ and $\mathbf{T}_{\texttt{BŁ}}$. The linear velocity of the base is represented by the vector ${}_{\texttt{B}}\mathbf{v}$. A fixed 3D point within the frame `I` is denoted by ${}_{\texttt{O}}\mathbf{p}$.

3D Case In three-dimensional space, rigid body transformations consist of translations along three-dimensional vectors and rotations around the coordinate axes. Specifically, when points are expressed using homogeneous coordinates, a three-dimensional transformation is represented by a 4 × 4 matrix as the following, where $\mathbf{R}$ is a 3 x 3 rotation matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2.4}$$

In this thesis, our focus is specifically on the group of 3D transformations SE(3), where $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{t} \in \mathbb{R}^3$.

2.1.3 Frames

In robotics, frames refer to coordinate systems that provide a means of expressing the position and orientation of objects and robot links in a consistent and understandable way. Each frame is typically defined by its origin and axes relative to a reference

frame. The primary reference frames used in this study are detailed in Tab. 2.2. Note that we use the conventional right-hand rule for the frames.

Table 2.2: Main reference frames used in this work.

Frame	Description
`M`	*map* fixed frame with respect to a robot's map
`O`	*odometry* fixed frame used by odometry estimators
`B`	*base* frame specified at the centre of the robot's body
L	*lidar* frame in which lidar data is captured, usually rigidly attached to the base frame
C	*camera* frame in which vision data is captured, usually rigidly attached to the base frame

We can further refine the aforementioned mathematical notation by appending specific sub-indices to denote their association with relevant reference frames. This modification is illustrated in Fig. 2.1.

2.2 Optimisation

Optimisation techniques are widely used in robotics. It is used to solve problems such as path planning, where the goal is to find the most efficient route for a robot to follow, or in motion control, where the objective is to achieve the desired movement while minimising energy consumption or maximising speed and accuracy. Algorithms such as SLAM rely heavily on optimisation to minimise a cost function of the error which links an estimated state with measurement derived from incoming sensor data and the actual state,

This section will present an overview of gradients, Hessians, Jacobians and how they are used in optimisation. Then we introduce a subcategory of optimisation, factor graph optimisation [25], and describe how this is used for pose estimation.

2.2.1 Gradient, Hessian and Jacobian

The gradient of a function is a vector that contains all the partial derivatives of the function with respect to its variables. It points in the direction of the

greatest rate of increase of the function and is denoted as ∇f. For the function $f(x_1, x_2, ..., x_n)$, the gradient is:

$$\nabla f = [\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n}]^T \tag{2.5}$$

The gradient is used in various gradient-based optimisation methods such as Gradient Descent (GD).

The Hessian matrix is a square matrix of second-order partial derivatives of a function. It describes the local curvature of the function, helping to understand the degree of convexity properties around a point. For the function $f(x_1, x_2, ..., x_n)$, the Hessian matrix is:

$$H_f = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 x_n} \\ \frac{\partial^2 f}{\partial x_2 x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n x_1} & \frac{\partial^2 f}{\partial x_n x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{2.6}$$

In optimisation, particularly when dealing with functions of multiple variables, the gradient and the Hessian matrices are critical tools for analysing the function's behaviour and finding minima or maxima. The Hessian is particularly important in second-order optimisation techniques, such as Newton's method. It is used not only to determine the direction in which to move to minimise the objective function but also to calculate the size of the steps to take towards the optimum. The Hessian's properties (such as being positive definite or negative definite) at a critical point (where $\nabla f = 0$) can determine whether the point is a local minimum, local maximum, or a saddle point.

The Jacobian matrix is the generalisation of the gradient to vector-valued functions (functions with *multiple outputs*). It is a matrix consisting of all first-order partial derivatives of a vector function. Each row of the Jacobian matrix represents the gradient of one component of the vector function. For a vector function $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^m$ defined by $\mathbf{F}(x_1, x_2, ..., x_n) = [f_1(x_1, ..., x_n), f_2(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)]$, the Jacobian is:

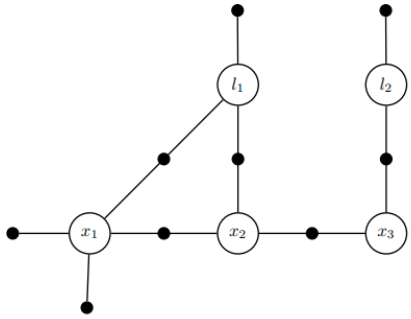


Figure 2.2: A factor graph consists of nodes (circles) linked by factors (black dots). The image is sourced from work by Dellaert and Kaess [25].

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_m} \end{bmatrix} \tag{2.7}$$

When optimising functions with multiple variables, Newton's method can be generalised using the Jacobian matrix to find zeros or to solve systems of nonlinear equations efficiently. This method uses the inverse of the Jacobian to iteratively approach an optimal solution.

Note that this thesis uses the Jacobian concept in Chapter 4 Sec.VI to construct the joint Jacobian matrices in order to select important image features.

2.2.2 Factor Graph Optimisation

Factor graph optimisation is a powerful framework for representing the factorisation of a probability distribution function and formulating complex estimation problems that arise in various fields such as robotics, computer vision, and network science. At its core, a factor graph is a bipartite graph representing the variables of interest and the constraints or measurements that link them. This graphical model, with an example shown in Fig. 2.2, is highly effective in depicting the relationships and dependencies between the variables, making it easier to understand and represent high-dimensional optimisation problems.

One of the key advantages of factor graph optimisation is its flexibility and generality. It can accommodate a wide range of problem types, from linear to

nonlinear, discrete to continuous, and even those with a mixture of different types of variables and constraints. Additionally, many factor graphs can be efficiently solved using sparse linear algebra techniques, as many real-world problems represented by these graphs are sparse in nature.

Another significant benefit is the modularity of the factor graph representation. This allows for the easy incorporation of new data or measurements into the graph, making it particularly suitable for online or incremental optimisation problems where information arrives sequentially over time.

The optimisation process in factor graphs involves adjusting the values of the variables to best fit the constraints or measurements represented by the factors. These algorithms seek to minimise or maximise an objective function that quantifies the error or likelihood of the measurements given the variable states.

For the factor graph, the Maximum A Posteriori (MAP) probability estimate can be written from the product of each factor's likelihood function $\phi_i(\mathcal{X})$ [6]:

$$\mathcal{X} = \arg\max_{\mathcal{X}} \left(\prod_{i=1} \phi_i(\mathcal{X}) \right) \tag{2.8}$$

This is equivalent to the standard sum of the Least Squares (LS) optimisation problem where we assume each factor is in the presence of white Gaussian noise,

$$\mathcal{X} = \arg\max_{\mathcal{X}} \left(\sum_{i=1} (\| \, h_i(\mathcal{X}) \ominus z_i \, \|^2_{\sum_i}) \right) \tag{2.9}$$

where $h_i(\mathcal{X})$ is the measurement model, $\sum_i$ is the covariance associated with each measurement and the operator $\ominus$ represents the generalised difference between two points on a manifold. The $\ominus$ is required to properly deal with differences between rotations over the SO(3) manifold.

2.2.3 Factor Graph for Pose Estimation

Our goal is then to use the factor graph framework to optimise the posterior probability of the state $\mathcal{X}_k$ based on the inputs $\mathcal{Z}_k$.

$$\mathcal{X}_k^* = \arg\max_{\mathcal{X}_k} p(\mathcal{X}_k|\mathcal{Z}_k) \propto p(\mathcal{X}_0)p(\mathcal{Z}_k|\mathcal{X}_k) \tag{2.10}$$

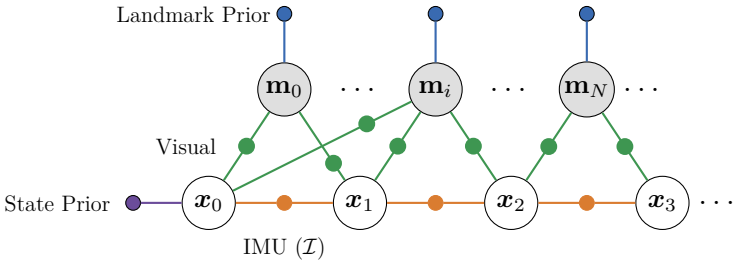


Figure 2.3: A factor graph corresponds to a visual-inertial odometry problem and consists of state and landmark nodes linked by prior, IMU, and landmark factors.

In this context, the last component of (2.10) represents the likelihood function. This function is directly proportional to the posterior probability, allowing it to serve as an effective cost function for optimisation purposes. When the assumption is made that measurements are conditionally independent and influenced by zero-mean Gaussian noise, this optimisation (2.10) get simplified into a LS problem.

$$\mathcal{X}^* = \arg\min_{\mathcal{X}_k} \sum_{\mathcal{T}} \sum_{\forall i \in \mathsf{K}_k} \|\mathbf{r}_{\mathcal{T}_i}\|^2_{\Sigma_{\mathcal{T}_i}} \tag{2.11}$$

As an example, we can model the pose estimation problem of a visual-inertial system as shown in Fig. 2.3 where the pose states are X_0 to X_3, and they can be linked by pre-integrated IMU measurements or wheel odometry measurements. The optimisation becomes the following:

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \sum_{j \in M} \|\mathbf{r}_{\mathbf{m}_{j,0}}\|^2_{\Sigma_{\mathbf{m}_{j,0}}} + \sum_{i \in \mathsf{K}_k} \|\mathbf{r}_{\mathcal{I}_{\Delta i}}\|^2_{\Sigma_{\mathcal{I}_{\Delta i}}} + \|\mathbf{r}_0\|^2_{\Sigma_0} + \sum_{j \in \mathsf{M}_i} \|\mathbf{r}_{\mathbf{m}_j}\|^2_{\Sigma_{\mathbf{m}_j}} \tag{2.12}$$

where the residuals are from landmark priors, IMU, state priors, and visual landmarks respectively. These factors will be used to create the factor graph structure shown in Fig. 2.3.

The definition of each residual is further explained in Section IV, Chapter 4.

2.3 Deep Neural Networks

Deep neural networks (DNNs) are advanced neural networks with multiple layers that can model complex patterns in data. They consist of an input layer, several

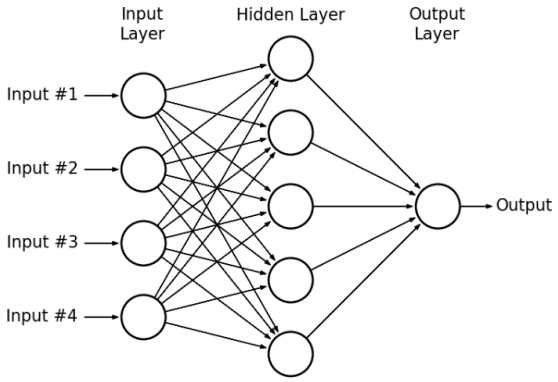


Figure 2.4: Illustration of a Multilayer Perceptrons neural network. Note the hidden layers would be more than one.

hidden layers, and an output layer, where each neuron processes input and passes it on. Activation functions like rectified linear unit (ReLU) introduce non-linearity, helping the network learn intricate patterns. DNNs are trained on large datasets using optimisation algorithms like GD, and the back propagation step adjusts the weights to minimise errors. Regularisation techniques like dropout prevent overfitting, ensuring the model works well on new data. DNNs excel in tasks like image recognition, speech recognition, and natural language processing.

Different types of neural networks are tailored to specific applications. Multilayer Perceptrons (MLPs), the simplest form, consist of fully connected layers and are used for classification and regression tasks. Convolutional Neural Networks (CNNs) excel in image and video recognition by capturing spatial hierarchies in data. Recurrent Neural Networks (RNNs), along with variants like Long Short-Term Memory (LSTM) networks, are designed for sequential data tasks such as language modelling and time series prediction. Each network type employs distinct architectures and learning mechanisms to tackle specific challenges in deep learning applications.

2.3.1 Multilayer Perceptrons

MLPs are a type of neural network that consists of multiple layers of nodes, or neurons, connected in a feedforward manner. It includes an input layer, one or more hidden layers, and an output layer. Each neuron in a layer is connected to

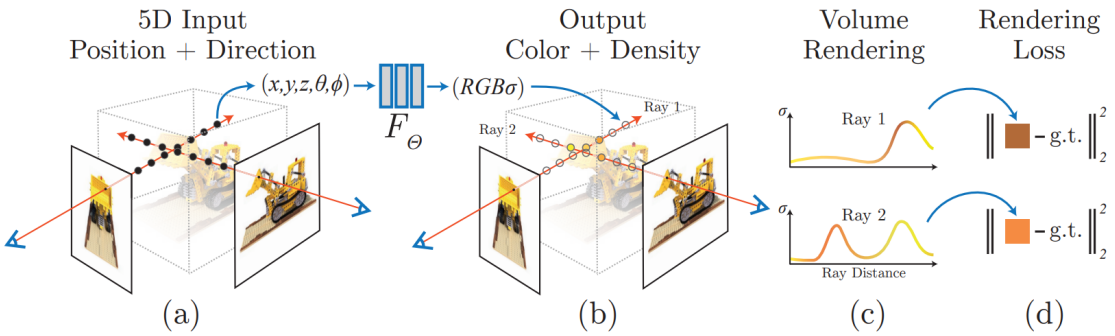


Figure 2.5: Overview of the Neural Radiance Fields (NeRF) scene representation pipeline [97], showing (a) synthesising images from input: position and direction, (b) feeding into an MLPs to produce colour and density, (c)volume rendering to form an image, (d) optimisation over reconstruction loss.

every neuron in the next layer, making the network "fully connected." An example of this is shown in Fig. 2.4.

MLPs are capable of learning complex functions and patterns in data by adjusting the weights of the connections between neurons during training. They use activation functions, such as ReLU or sigmoid, to introduce non-linearity into the model, enabling it to capture intricate relationships in the data. MLPs are commonly used for tasks like classification and regression, where the goal is to predict an output based on given inputs. A disadvantage of MLPs is that the total number of parameters can become extremely large, as it is the product of the number of perceptrons in each layer (layer 1 multiplied by layer 2, multiplied by layer 3, and so on). This leads to inefficiencies due to redundancy in such high dimensions. Additionally, another drawback is that MLPs disregard spatial information since they take flattened vectors as inputs.

A recent advancement with MLPs is the NeRF, a framework that represents a 3D scene using weights stored in the network. An overview of how NeRF works is shown in Fig. 2.5. NeRF can synthesise novel views of complex scenes, providing photo-realistic renderings from unseen camera angles. In Chapter 6, we integrate the NeRF rendering technique for visual localisation in 3D colour maps.

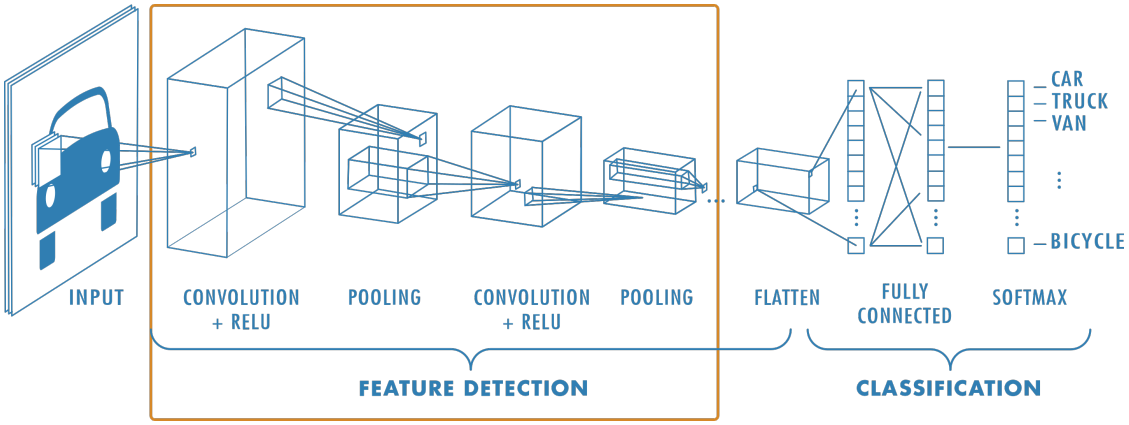


Figure 2.6: An example overview of CNNs for image classification [63]. An image of a car goes through several layers of filtering and pooling and gets classified as a class of transport object.

2.3.2 Convolutional Neural Networks

CNNs are another type of neural network model specifically designed to process and analyze visual data. As shown in Fig. 2.6, CNNs employ a series of convolutional layers that apply filters to the input data to detect patterns and features such as edges, textures, and shapes. These filters slide over the input, performing convolution operations that preserve the spatial relationships between pixels. Additionally, CNNs include pooling layers to reduce the spatial dimensions of the data, which helps decrease computational complexity and control overfitting. The hierarchical structure of CNNs allows them to build increasingly abstract representations of the input data, making them highly effective for tasks such as image and video recognition. Their ability to automatically and adaptively learn spatial hierarchies of features has made CNNs the backbone of many computer vision algorithms, including object detection and segmentation.

Compared to MLPs, CNNs can account for local connectivity, where each filter scans the entire image based on its size and stride, enabling the filter to detect and match patterns regardless of their location in the image. This results in smaller shared weights, making CNNs easier to train than MLPs. CNNs are also more effective and can go deeper, as their layers are sparsely connected rather than fully connected. This means that each node does not connect to every

other node, allowing CNNs to take both matrices and vectors as inputs while maintaining efficiency and performance.

Many modern image processing network architectures are built upon CNNs. In this thesis, we use SuperPoint [28] and NetVLAD [4] networks to detect local and global image features. The detailed architectures for SuperPoint and NetVLAD are explained in Sec. 2.4.1.

2.4 Camera

The camera sensor plays a pivotal role in robotics, from navigation to manipulation. The imaging sensor captures light intensity on a photosensitive surface, which can distinguish specific spectral ranges and provide greyscale, colour, or other more exotic images such as thermal or hyperspectral images.

A camera operates by capturing light and recording it to create an image, utilising components like the lens, aperture, shutter, and image sensor. The lens focuses light onto the camera sensor or film, with the aperture controlling the amount of light entering the camera. The shutter dictates the duration of light exposure, impacting the image's brightness and clarity. In digital cameras, the sensor converts light into electrical signals, which are processed into a digital image, while in traditional film cameras, light-sensitive film captures the image through a chemical reaction.

Pinhole Camera

The simplest camera model is the pinhole camera model, which consists of a light-tight box with a small hole on one side and a photosensitive surface on the opposite side. Light from the scene passes through the pinhole, projecting an inverted image on the photosensitive surface. The size of the pinhole affects image sharpness and light exposure, with a smaller pinhole creating a sharper image but requiring longer exposure times. The distance between the pinhole and the photosensitive surface, known as the focal length f, influences the image size and Field-of-View (FoV). Pinhole cameras, which in theory provide an infinite depth of field. The pinhole projection of a point in 3D $\mathbf{P}(x, y, z)$ can be written as:

$$\mathbf{p}' = \mathbf{K}_i, \mathbf{P} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} f_x X + c_x Z \\ f_y Y + c_y Z \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2.13}$$

Where $\mathbf{K}_i \in \mathbb{R}^{4\times4}$ is the *intrinsic calibration* matrix. f_x and f_y are the camera focal lengths, while c_x and c_y are the coordinates of the centre of the image.

To obtain the point on the image plane, we need to normalise the image coordinates by dividing $\mathbf{p}'$ by its third column:

$$\mathbf{p} = \frac{\mathbf{p}'}{\mathbf{p}'_{[3]}} = \frac{1}{z} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \\ 1 \\ 1/z \end{bmatrix} = \begin{bmatrix} u \\ v \\ 1 \\ d \end{bmatrix} \tag{2.14}$$

Here (u, v) are the pixel coordinates in the image plane and d is the inverse depth, also known as disparity. Similarly, a pixel in the image plane can be projected back into the 3D world by following the reverse order of the above equations. The pinhole camera model will be used throughout this thesis, particularly in Chapters 4 and 6.

Fisheye Lens Camera

A fisheye camera is a type of camera lens designed to capture ultra-wide, hemispherical images. This lens produces strong visual distortion, intended to create a wide panoramic or hemispherical image. Different from a pinhole camera which uses a simple aperture to project 3D points onto an image plane, the fisheye lens camera is more complex, and bends the light to achieve its ultra-wide-angle effect.

The intrinsic parameters for a fisheye camera can be defined as the following. Given the same point in 3D $\mathbf{P}(x, y, z)$, the pinhole projection of point $\mathbf{P}$ is $[a, b]$, where,

$$a = x/z \quad \text{and} \quad b = y/z \tag{2.15}$$

$$r^2 = a^2 + b^2 \tag{2.16}$$

$$\theta = a \tan(r) \tag{2.17}$$

The fish eye distortion can be described using the *generic fisheye camera model* [66], where,

$$\theta_d = \theta(1 + k_1\theta^2 + k_2\theta^4 + k_3\theta^6 + k_4\theta^8) \tag{2.18}$$

Hence the distorted image pixel coordinates $[u, v]$ of the projected 3D point are,

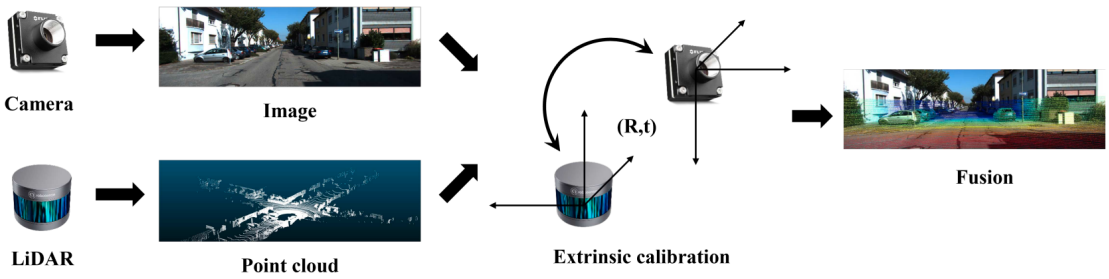
$$u = (\theta_d/r)a \quad \text{and} \quad v = (\theta_d/r)b \tag{2.19}$$

Achieving accurate intrinsic calibration is essential for precisely projecting any point from the 2D image plane to the 3D world. As the camera moves through the environment, it is possible to track certain feature points on the image plane and continuously estimate their positions in the world. Inaccurate intrinsic parameters can lead to unstable 3D point estimation, which can negatively impact the sensor's pose estimation accuracy. Additionally, inaccurate calibration can cause incorrect constraints to be passed to a VIO system optimisation, resulting in instability. This thesis primarily uses fisheye cameras for visual odometry and localisation due to their wider FoV. Particularly in narrow locations or scenarios with significant motion, wide FoV cameras help maintain feature tracking.

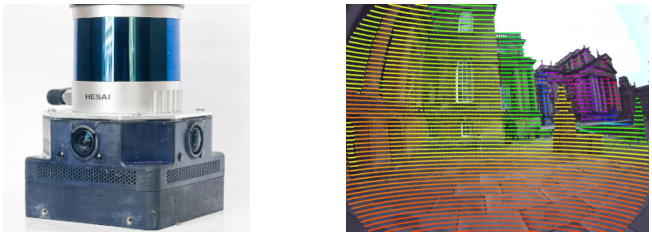
Extrinsic Calibration

Besides the intrinsic calibration mentioned in the pinhole camera model, there is also extrinsic calibration. Intrinsic calibration is concerned with the camera's internal characteristics, such as the focal length and lens distortion. Extrinsic calibration, on the other hand, determines the spatial relationship between the camera and another reference frame within the system, such as an IMU or lidar.

Both types of calibration are crucial when using camera measurements in pose estimation. For instance, calibration between a camera and an IMU is essential for visual-inertial systems, as it can either align the two trajectories or cause their trajectories to diverge over time. Furthermore, in 3D colour reconstruction, accurate intrinsic parameters of the camera and extrinsic calibration between the camera and lidar are necessary. Fig. 2.7 illustrates the camera lidar overlay with accurate



(a) Autonomous driving example of a lidar camera overlay image, sourced from [87].



(b) Frontier device used in this thesis and its lidar camera overlay image.

Figure 2.7: Extrinsic calibration describes the spatial transformation between two sensors. In these two examples, calibration can affect the depth image overlay and subsequent 3D reconstruction.

extrinsic calibration. Conversely, inaccurate extrinsic calibration can lead to lidar points being assigned the incorrect colour from the camera's pixels, an issue that becomes more pronounced with the increasing distance of objects from the camera.

2.4.1 Local and Global Features

Local features are typically derived from small patches or regions of an image. They are designed to be invariant to changes in illumination, viewpoint, and other factors affecting the image's appearance. Each feature can be represented and described with a descriptor providing distinct and informative characteristics of an image or video frame. Common local feature extraction techniques include Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Oriented FAST, and Rotated BRIEF (ORB).

In this work, we make use of both classical local feature detectors, such as the corner detector FAST [114] in Chapter 4, and learned feature detectors, such as SuperPoint [28] in Chapter 6. Since FAST is a well-known classic feature detector, we will give a brief description of FAST but present more explanation

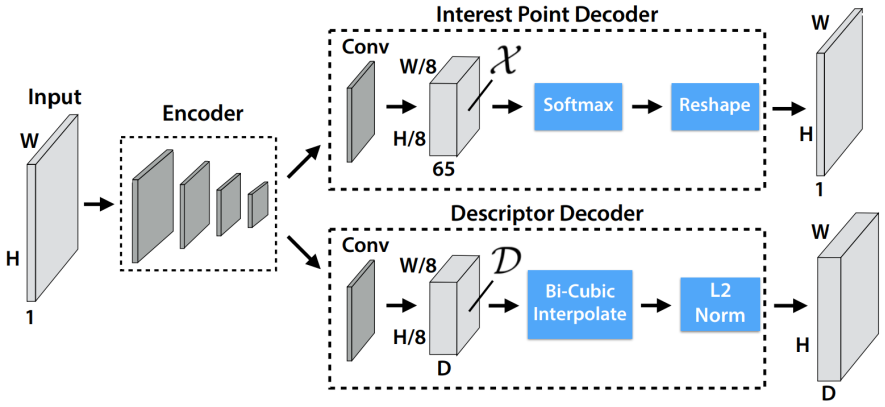


Figure 2.8: Network architecture of SuperPoint.

on SuperPoint. FAST is an efficient corner detection algorithm, that identifies key points in an image by comparing the intensity of a pixel to a circle of surrounding pixels. FAST is known for its computational speed and simplicity, making it particularly suitable for real-time applications. Despite its speed, FAST may not provide as much robustness against noise and varying image conditions compared to more complex feature detectors.

SuperPoint first trains a feature detector called Magic Point based on synthetic images made with primitive shapes. It then produces pseudo ground truth labelling on real images with Magic Point to detect interest points through Homographic Adaptation. This adaption is a series of random homography procedures on the images, such as translation, scaling, wrapping, and rotations, to obtain all interest points. Finally, it conducts joint training on image pairs warped from one another to further train the detector and descriptor.

Fig. 2.8 shows the architecture of SuperPoint. The image input goes through a VGG style encoder to reduce the dimensionality. This encoder is constructed with convolutional layers, pooling layers, and a standard non-linear activation function. The interest point and descriptor decoder are the key components. For the interest point decoder, each pixel of the output represents the likelihood of that pixel being a feature point. The descriptor decoder conducts bicubic interpolation on the descriptor, followed by L2 normalisation of the activations to ensure they are of

unit length. Note that there are no deconvolution layers, to reduce computation time and achieve real time performance.

Conversely, global features, also known as global descriptors, are high-level, holistic characteristics of an image or video frame computed using the entire image or video frame. Global features are typically derived by aggregating local features in some way, such as by computing statistics over the local feature descriptors or by constructing a histogram of the local feature orientations. Global features can be used to represent an image's content and perform tasks such as image classification, scene recognition, and video segmentation. SOTA global feature descriptors are frequently utilised in image retrieval tasks and form an integral part of the place recognition pipeline. One of the most popular classic approaches is Bag of Words (DBoW)[46]. This approach treats images as "documents" and local features as "words." It first extracts local features using algorithms like SIFT or ORB, clustering these features to create a visual vocabulary, and mapping image features to the nearest visual word. Each image is then represented as a histogram of visual word occurrences. By comparing these histograms, the DBoW approach efficiently recognises and matches places.

Recent advancements in learning-based global feature descriptors include models such as NetVLAD [4], R2D2, and D2Net, among others. NetVLAD is built with CNNs and a special VLAD layer. The central element of the VLAD layer is inspired by the image representation technique known as the Vector of Locally Aggregated Descriptors. VLAD is similar to techniques such as Bag of Words, but captures statistical information of the local descriptor over the image. The output of the VLAD image representation is a matrix of size $K \times D$. K is the number of clusters, and is represented as c_k and D is the dimension of the local descriptors, represented as $\mathbf{x}_i$. The (j,k)-th element of $\mathbf{V}$ is calculated as below:

$$\mathbf{V}(j,k) = \sum_{i=1}^{N} a_k(\mathbf{x}_i)(x_i(j) - c_k(j)) \tag{2.20}$$

where $x_i(j), c_k(j)$ is the j-th dimension of i-th descriptor and k-th cluster centre. Function a_k produces 1 when the descriptor belongs to cluster k, it is otherwise

0. However, Eq. 2.20 is not differentiable because of a_k, but it can be rewritten to be used in network training, details can be found in [4].

NetVLAD is very powerful and generates descriptors capable of accurately identifying the location of the query image, even amidst significant clutter (e.g., people, cars), variations in viewpoint, and stark differences in illumination, including day and night conditions.

In this thesis, we use classic local feature detectors in Chapter 4, and learning-based feature detectors in Chapter 6.

2.4.2 Perspective-n-Points Pose Computation

Local features detected in individual cameras can aid in estimating camera poses when combined with the 3D point projection method mentioned in 2.4. By identifying and matching these features across multiple views, we can establish correspondences that enable the reconstruction of the camera's relative positions and orientations. We begin by addressing the two-camera view problem, where local features are detected and matched in two images, and the corresponding point positions are known in the 3D space. The problem is to determine the relative pose between the two camera sensors, also known as the PnP problem.

Given a set of image feature correspondences between two cameras, such as the orange image features highlighted in Fig. 2.9. Image features in camera 2 can be projected into 3D space as point $\mathbf{P}_{c_2} = (x_{c_2}, y_{c_2}, z_{c_2})$, in the camera 2 frame. With the perspective projection model and camera intrinsic parameters matrix $\mathbf{K}$, we can formulate the relative pose estimation problem in Eq. 2.21 and 2.22,

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \mathbf{K}\, \Pi\, \mathbf{T}_{c_1 c_2} \begin{bmatrix} x_{c_2} \\ y_{c_2} \\ z_{c_2} \\ 1 \end{bmatrix} \tag{2.21}$$

$$\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \mathbb{I} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{c_2} \\ y_{c_2} \\ z_{c_2} \\ 1 \end{bmatrix} \tag{2.22}$$

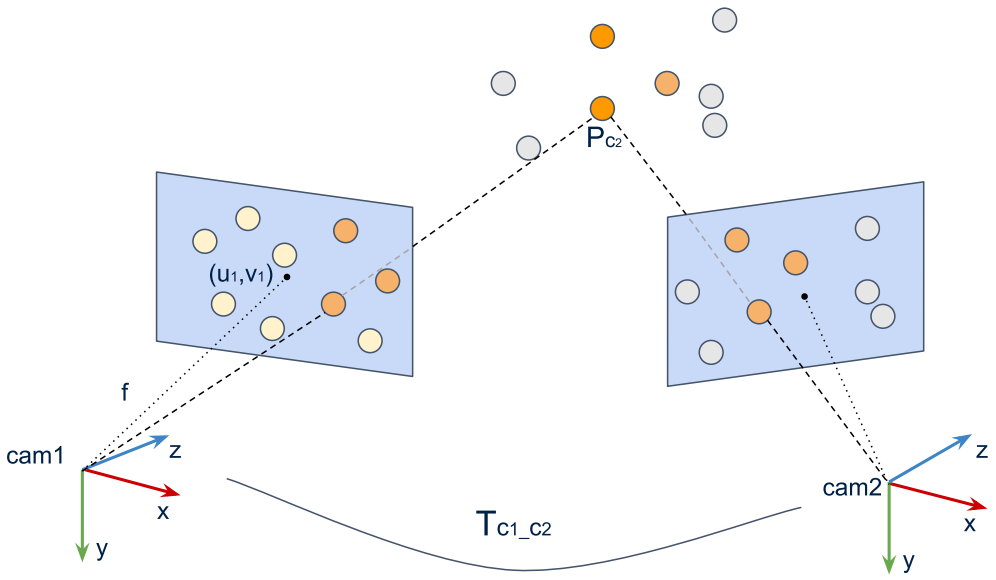


Figure 2.9: PnP algorithm: estimate $\mathbf{T}_{c_1c_2}$ given a set of 3D points in camera 2 frame and local feature correspondences.

where (u_1, v_1) is the feature location in the camera 1 image, f_x, f_y is the focal length, c_x, c_y is the optical centre. The goal is to work out $\mathbf{T}_{c_1c_2}$. This becomes a pose refinement problem with multiple 3D points and corresponding image feature points, formulated as:

$$\arg\min \sum_i ||(u_i, v_i) - \Pi(\mathbf{T}_{c_1c_2}, {}_{c_2}\mathbf{P}_i)||^2 \tag{2.23}$$

This can be solved using the non-linear Levenberg-Marquart (LM) method in an iterative fashion. The initial solution for co-planar feature points requires at least four points and non-planar feature points require at least six points using the direct linear transformation (DLT) algorithm. When there are larger amounts of incorrect image features to 3D point correspondences, RANSAC can be applied to make the solver more robust to these outliers.

In Chapter 6, the system detects local features in images and applies the pinhole model and the PnP algorithm described above to localise the camera pose.

2.5 Lidar

Lidar, an acronym for Light Detection and Ranging, is a remote sensing technology that uses laser light, as opposed to radar which employs radio waves. The core

				
Lidar Type	Ouster os1-64	Ouster os0-128	Hesai XT32	Hesai QT64
Num of Beams	64	128	32	64
Vertical FoV (deg)	45	90	31	104
Range * (m)	90	35	80	20
Accuracy (cm)	2.5	2.5	1	3
Point Rate (pps)	2,621k	5,242k	640k	384k
Power (W)	14-20	14-20	10	8

Table 2.3: An overview of lidars used in this work. *: range at 10% reflectivity. pps: point per second.

principle of lidar is relatively straightforward: it emits a laser pulse towards a target surface and captures the reflected light with a receiver sensor. By measuring the time it takes for the laser to travel to the object and return, the device accurately calculates the distance between itself and the target. This measurement process can occur millions of times per second, creating a detailed and precise spatial representation of the area in the form of a point cloud.

In this study, we use various 3D mobile lidar systems, as summarised in Table 2.3. Initially, our research, detailed in Chapters 4 and 5, began with the use of lidars produced by Ouster. However, we soon found that while Ouster lidars are capable, their accuracy was not as high as Hesai lidars, and they exhibited relatively high power consumption for robotic platforms. Consequently, as discussed in Chapter 7, we transitioned to the Hesai XT32 for our dataset, selecting it for its superior range accuracy. Nonetheless, the XT32's limitation of 32 beams and a 31° FoV posed challenges in capturing floor and ceiling details, which are critical for indoor localisation and mapping. To mitigate this issue, we have also used the Hesai QT64 in some of our projects, offering more beams. While beneficial, this adjustment was at the cost of reduced ranging accuracy.

Figure 2.10: Terrestrial Laser Scanners from left to right: Leica BLK350, Z+F 5016, Leica RTC360.

Terrestrial Laser Scanner

Terrestrial Laser Scanning (TLS) is static tripod-based lidar technology that captures high-resolution, 3D information about the shape and surface characteristics of objects and landscapes. Mounted on a tripod, modern TLS systems can capture colour data by integrating Red-Green-Blue (RGB) values from a built-in camera, enhancing the visual quality of the 3D models created from the scans. The precision of TLS and its capacity to record fine details make it particularly valuable in fields where precise measurements are crucial, such as structural deformation analysis, construction progress monitoring, and forensic investigations.

The main difference between TLS and mobile lidar lies in the mobility and accuracy of the scanning device. TLS operates from a stationary position, conducting scans from a fixed point and targeting specific areas. TLS typically achieves higher accuracy, making it an ideal technology for conducting detailed surveys of buildings, structures, and archaeological sites. In contrast, mobile 3D lidar systems are designed to scan extensive areas from moving platforms, such as drones, vehicles, or aircraft, but with lower accuracy.

In this work, we employ the TLS scanners to create ground truth models for localisation and mapping purposes. As illustrated in Fig. 2.10, we use the Leica BLK360 for the Newer College dataset, and the Z+F 5016 for the Hilti-Oxford dataset, as well as the Leica RTC360 for the datasets discussed in Chapter 6. The BLK sensor is an entry level sensor while the other two are professional grade.

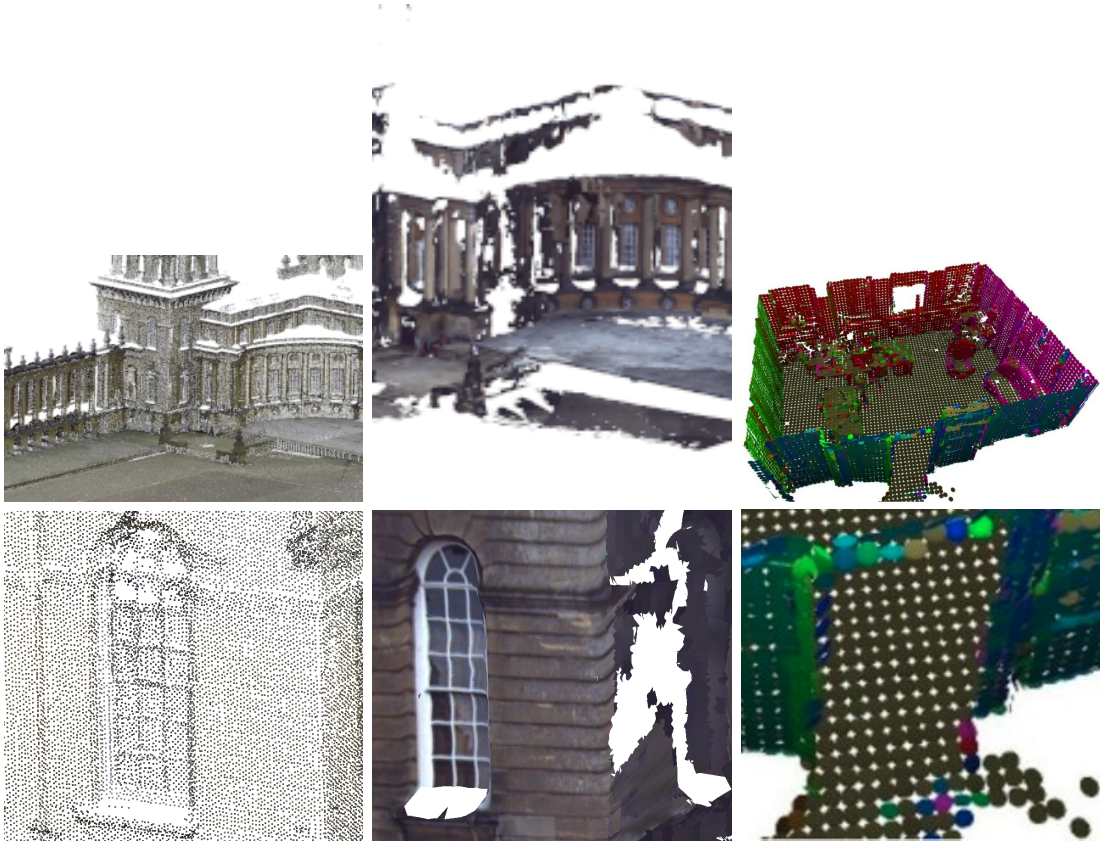


Figure 2.11: From left to right: examples of 3D reconstruction from lidar in point cloud, mesh and surfel representations. The bottom row shows close up views.

The Leica BLK360 has a range of 60 m and a range accuracy of 8 mm at 20 m, while the Leica RTC360 offers a range of 130 m with a range accuracy of 2.9 mm at 20 m. The Z+F 5016 boasts a measurement range of up to 365 m, with an accuracy of less than 2.9 mm at 20 m.

2.5.1 Point Cloud, Mesh and Surfel

When using lidar for 3D reconstruction, the choice of representation largely depends on the specific requirements of the reconstruction task and its application.

A point cloud is a collection of data points in 3D space, representing the x, y, and z coordinates of individual points on an object's sampled surface. Point clouds are typically generated by 3D scanners, as illustrated on the left in Fig. 2.11. They serve as the raw data for 3D reconstruction but do not provide information about how these points interconnect or any continuity between the surfaces they

represent. The visualisation of point clouds often produces a "starry night" effect, where only discrete points are visible.

A mesh, on the other hand, is a 3D representation that includes vertices, edges, and faces, which are often triangular. By connecting these vertices, meshes form a continuous surface that accurately portrays an object's shape, as shown in the middle of Fig. 2.11. Meshes are crucial for applications requiring a solid, realistic depiction of 3D forms, such as 3D animation and video games. This format requires more storage due to the detailed information about the surface structure it contains.

A surfel, or surface element, represents a small surface patch and carries additional data such as normal vectors and sometimes colour and radius, exemplified on the right in Fig. 2.11. Unlike a zero-dimensional point, each surfel represents part of a surface. Surfel-based models are particularly useful for rendering and simulation, where detailed surface characteristics are crucial. They provide greater flexibility than meshes in managing complex geometries or surfaces that are challenging to triangulate effectively.

Point clouds are simpler and quicker to process but lack detailed information and direct insight into surface connectivity. Meshes offer a comprehensive and precise representation, ideal for high-quality visualisations and simulations. Surfels strike a balance, offering more detailed surface information than point clouds but without the full commitment to an interconnected structure seen in meshes. This flexibility is beneficial in scenarios where detailed local surface properties are critical. However, both mesh and surfel may struggle in natural environments when representing objects such as plants and grass due the their complex profile.

In Chapter 5, we use point cloud representation for lidar localisation. In Chapter 6, we employ both mesh and point cloud representations and conducted a comparative analysis of their localisation performance based on the rendered images from these two types of reconstructions.

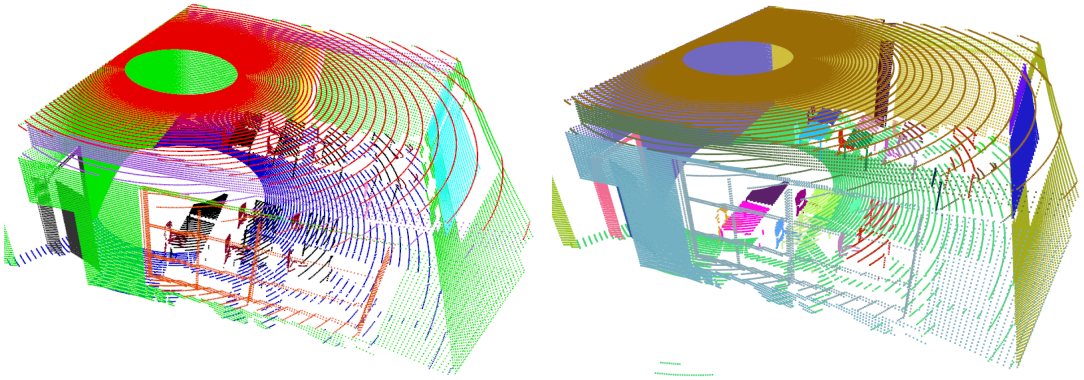


Figure 2.12: The left image shows semantic segmentation coloured according to classes, e.g. walls are green and the tables are black. The right image shows instance segmentation where every object instance has its own colour.

2.5.2 Semantic and Instance Segmentation

Semantic segmentation and instance segmentation are two research topics used to analyse and interpret point clouds, each serving specific purposes based on different concepts.

Semantic segmentation involves labelling each point in a point cloud with a class that defines what object or material that point corresponds to. The classes, such as walls, ceilings, tables, and chairs, are predefined and ought to be consistently applied across the entire dataset. An example of semantic segmentation can be seen on the left side of Fig. 2.12. The objective of this method is to understand the scene at a categorical level, identifying areas that correspond to different types of objects or materials. However, it does not distinguish between individual objects of the same type. In semantic segmentation, every point in the point cloud is assigned a class label, but points belonging to different instances of the same class are not differentiated.

Instance segmentation, also known as panoptic segmentation, extends beyond categorical labelling by differentiating between individual instances within each category. This method is crucial for tasks requiring precise object localisation and counting, such as distinguishing between two chairs situated next to each other in a room. The output includes a labelling of each point with both a class and an instance

identifier, enabling the identification of distinct entities within the same class. For example, as shown on the right side of Fig. 2.12, two walls in the point cloud would be labelled as "Wall 1" and "Wall 2," facilitating individual recognition and analysis.

For both types of segmentation, state-of-the-art methods are predominantly based on deep learning approaches. In Chapter 5, we use the SoftGroup [137] method for instance segmentation. In the following sections, we provide a more detailed overview of this method, outlining its mechanics and applications in the context of point cloud processing.

Instance Segmentation Network

Many existing methods for instance segmentation typically begin with semantic segmentation, where each point in a point cloud is assigned to a specific class based on fixed predictions. However, these fixed classifications can be inaccurate, adversely affecting the subsequent grouping stage. Such inaccuracies can lead to a lack of sufficient overlap between the predicted instances and the actual ground truth, resulting in a high number of false positives.

To address these challenges, SoftGroup employs a novel approach that integrates a bottom-up method for soft grouping with a top-down refinement process. This strategy allows SoftGroup to be more flexible in its handling of point classifications. Unlike other methods that rigidly assign each point to a single class, SoftGroup allows each point to be associated with multiple classes. This flexibility helps to mitigate errors stemming from initial semantic predictions. By enabling a more nuanced association of points to potential classes, SoftGroup effectively reduces the instances of false positives by more accurately distinguishing true object points from background. This approach not only improves the accuracy of instance segmentation but also enhances the overall robustness of the segmentation process in complex environments.

In the first part of the bottom-up grouping stage of SoftGroup, the U-Net backbone extracts features from point data, as illustrated in the system architecture depicted in Fig. 2.13. Following feature extraction, two key processes occur:

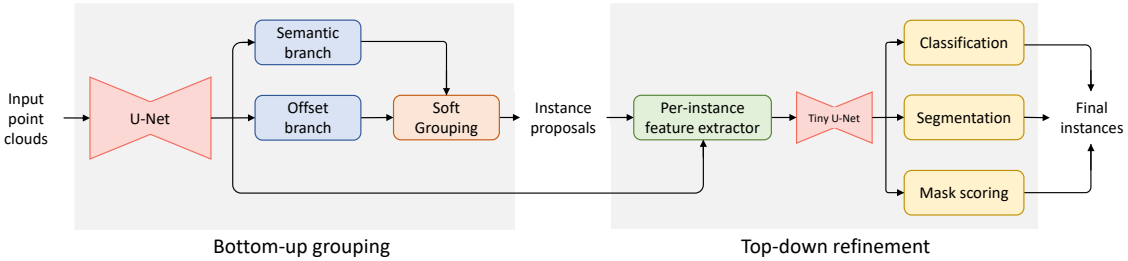


Figure 2.13: Overview of the SoftGroup instance segmentation network architecture.

the semantic and offset branches compute semantic scores and offset vectors, respectively. The semantic branch outputs semantic scores for M points over N classes $S = \{s_1, ..., s_N\} \in \mathbb{R}^{M \times N_{class}}$. The offset branch generates offset vectors for each point, mapping them to the geometric centre of the instance to which the point is assigned. Represented as $O = \{o_1, ..., o_N\} \in \mathbb{R}^{M \times N_{class}}$, these vectors enable the shifting of points toward the centre of their respective instances, facilitating more effective grouping in subsequent processes. Finally, a soft grouping module proposes instance clusters.

For training the semantic and offset branches, they employ cross-entropy loss and L_1 regression loss, respectively, as detailed in Eq. 2.24 and 2.25,

$$L_{semantic} = \frac{1}{M} \sum_{1}^{M} \mathbf{CE}(\mathbf{s}_k, s_k^*) \tag{2.24}$$

$$L_{offset} = \frac{1}{\sum_1^M \mathbb{1}_{\{\mathbf{o}_k\}}} \sum_{1}^{M} \mathbb{1}_{\{\mathbf{p}_k\}} \parallel \mathbf{o}_k - \mathbf{o}_k^* \parallel_1 \tag{2.25}$$

where $\mathbf{s}^*$ is the semantic label, $\mathbf{o}^*$ is offset label representing the vector from a point to the geometric centre of the instance. $\mathbb{1}_{\{.\}}$ is the indicator function indicating whether the point p_i belongs to any instance.

In the second stage of the top-down refinement process, features are extracted again for each instance and fed into a tiny U-net, followed by three heads that give classification, segmentation and masking scores.

The classification branch predicts the classification scores $C = \{c_1, ..., c_j\} \in \mathbb{R}^{J \times (N_{class}+1)}$, where J is the number of instances. The segmentation branch predicts an instance mask m_j for each instance j. The mask scoring branch estimates

the intersection over union (IoU) of a mask with ground truth $E = \{e_1, ..., e_j\} \in \mathbb{R}^{J \times N_{class}}$. The training loss of all branches is a combination of cross entropy loss in Eq. 2.26, binary cross entropy in Eq. 2.27, and L^2 regression losses in Eq. 2.28.

$$L_{class} = \frac{1}{K} \sum_{1}^{J} \mathbf{CE}(\mathbf{c}_k, c_k^*) \tag{2.26}$$

$$L_{mask} = \frac{1}{\sum_1^J \mathbb{1}_{\{\mathbf{m}_k\}}} \sum_{1}^{J} \mathbb{1}_{\{\mathbf{m}_k\}} \mathbf{BCE}(\mathbf{m}_k - \mathbf{m}_k^*) \tag{2.27}$$

$$L_{mask-score} = \frac{1}{\sum_1^J \mathbb{1}_{\{\mathbf{e}_k\}}} \sum_{1}^{J} \mathbb{1}_{\{\mathbf{e}_k\}} \parallel \mathbf{e}_k - \mathbf{e}_k^* \parallel_2 \tag{2.28}$$

Here, $\mathbf{c}^*, \mathbf{m}^*, \mathbf{e}^*$ are the classification, segmentation, and mask scoring targets, respectively. $\mathbb{1}_{\{.\}}$ indicates whether the proposal is a positive sample.

Finally, the total loss is,

$$L = L_{semantic} + L_{offset} + L_{class} + L_{mask} + L_{maskscore} \tag{2.29}$$

The whole network can be trained from end to end with supervised ground truth from a labelled dataset.

In Chapter 5, we use semantic and instance segmentation of the point cloud to extract object instances for indoor localisation. The SoftGroup model, explained here, provides key results in instance segmentation, enabling accurate identification and differentiation of objects within the environment.

3
Literature Review

This chapter provides a literature review on the various technologies and methodologies used in pose estimation for robotics. Sec. 3.1 offers an overview of different sensors, such as cameras, lidar, IMU, and Global Positioning System (GPS). Sec. 3.2 reviews visual and LiDAR-based odometry techniques, while Sec. 3.3 examines visual and LiDAR-based localization methods. Sec. 3.4 explores different SLAM systems, discussing various algorithms and their performance. Finally, Sec. 3.5 discusses the datasets used for developing and benchmarking robotic navigation algorithms.

3.1 Sensor Overview

Determining the location of a robot through accurate pose estimation is a key challenge and a fundamental capability for autonomous navigation. Within the field of robotics, various sensors are applied to provide accurate pose estimation, including GPS, lidar, camera, sonar, and radar. This section will give an overview of how different sensors have been adopted for pose estimation and their respective advantages. Examples of these sensors are shown in Fig. 3.1.

Camera

Pose estimation with a typical visual spectrum camera is the most widely studied method. Cameras can capture rich scene information although they do not directly

Figure 3.1: From the left to right: Sony IMX camera, Hesai lidar, Navtech radar and Ublox GPS.

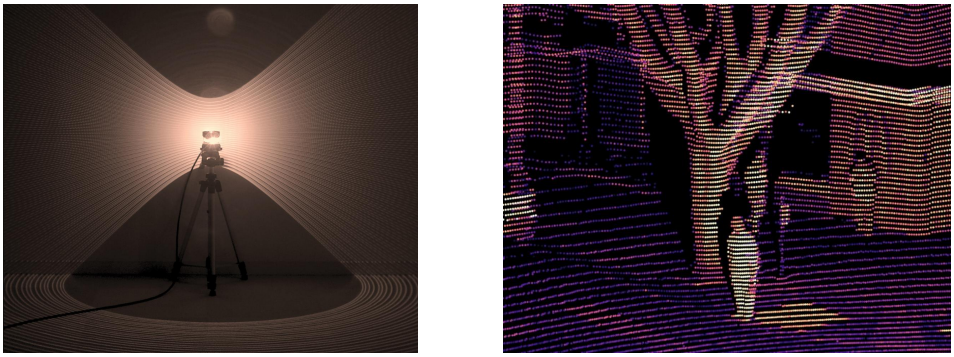


Figure 3.2: *Left*: Ouster OS0 lidar with a 360° surrounding view and a 90° vertical field of view. *Right*: An example of a detailed scan from a modern lidar sensor.

give the relative location of the objects with respect to the sensor. Hence, geometric information needs to be extracted from the images to deduce the location of the camera. Nevertheless, visual-based pose estimation can work in an environment with minimal physical structures and many odometry systems have demonstrated good performance with low drift rate and robustness in varied environments [39, 82, 15]. However, visual-based methods are prone to error and fail under changing illumination or the appearance of dynamic objects. While addressing these issues has been an active field of study, [96, 155, 148], robust and consistent pose estimation under challenging environments still remains an active research problem.

Lidar

Lidar is often chosen as a suitable pose estimation sensor since it can provide precise and direct measurements of its surrounding area. Lidar technology has progressed from the 1D point lidar to the 2D single beam, and now 3D multi-beam lidar with 360° horizontal and up to 90° vertical fields of view[1], as shown in Fig. 3.2.

[1] https://ouster.com/products/os0-lidar-sensor/

Lidar measurements typically provide unstructured points, known as a point cloud. These individual isolated points provide accurate 3D locations but lack distinctive information compared to a set of RGB image patches.

Despite the higher cost of lidar, it is still the preferred sensor modality for field robotics tasks such as conducting obstacle avoidance and providing odometry. Many lidar odometry methods [151, 122, 145] have demonstrated impressive results on pose estimation, less than 1 m drift per 100 m travel under good conditions. While lidar can be resilient to lighting changes, the main drawback is that estimation tends to degrade when there are a lack of distinguishable physical structures, e.g. in a long corridor or large open spaces. However, the recently developed Doppler lidar provides velocity data for each lidar point, which has been demonstrated to improve odometry in geometrically challenging environments [144].

Radar, GPS and Sonar

There has also been a burgeoning interest in radar pose estimation since the introduction of Frequency-Modulated Continuous Wave (FMCW) radar. Radar has a different signal propagation behaviour compared to lidar, as the large wavelength of radar allows it to propagate around and past objects. This enables radar to function effectively through dust, rain, and fog up to hundreds of meters, making it useful for navigation in various weather conditions. However, radar is a 2D scanning sensor in the horizontal plane and also has large measurement uncertainty in the vertical plane, making it challenging to work in uneven terrain or serve as a collision detection sensor. Additionally, radar is physically larger compared to other sensors. Recent studies have shown that radars can be particularly beneficial in wide open spaces with scarce features [3, 17]. Radar localisation [116] and odometry [3] have also shown promising initial results, suggesting that radar could provide robust pose estimation in the near future.

Furthermore, sensors such as GPS have been deployed in specific environments for pose estimation [80]. Modern real time kinematics (RTK) GPS can receive signals from satellite systems along with a correction stream from nearby base

stations, which can achieve centimetre position accuracy. While GPS works well in outdoor open spaces [50], it loses satellite reception once the sensor moves indoors or into a narrow outdoor space. Recently, it has become popular to use GPS as the prior for localisation [83, 105].

Marine sonar sensors are commonly used on autonomous underwater vehicles (AUVs) [36, 60]. This is because sonar sensors send signals with a long wavelength, behaving similarly to the FMCW radar that can travel through water to reach much longer distances than many other sensors. However, when used in the air, the sonar sensor has a short range and measurements are very noisy, so it is typically limited to tasks such as obstacle avoidance [126, 37], for example on robot vacuum cleaners due to the affordability of sonar sensors.

3.2 Odometry

In this section, we will focus the literature review on one type of pose estimation method: odometry systems, specifically visual and lidar odometry systems. For visual odometry, we will study visual-inertial odometry methods, followed by a more in-depth review of multiple-camera visual-inertial odometry, with the latter being one focus of this thesis. We will then discuss lidar odometry, dividing it into three categories: registration-based, feature-based, and deep learning-based lidar odometry. Finally, we will explore methods that learn features from lidar point clouds, as this is another relevant aspect of this thesis.

3.2.1 Visual Odometry

As discussed in Sec. 3.1, cameras contains concentrated and detailed scene information, which can be packed into one single image. There have been major advancements in camera related technology. Modern image sensors can offer higher resolution, better dynamic range, automatic exposure control and a large FoV. There has been extensive research into visual state estimation, ranging from the most minimal setup of a single camera to more complex setups with multiple cameras.

Depending on how information is extracted from the images, visual odometry can be divided into direct and indirect methods.

For indirect methods, also known as feature-based, the common approach is to extract features from images and to track them in subsequent frames. Features, which represent the key information inside an image, are typically pixels with high intensity contrast. These features can be triangulated and estimated together with the camera pose by minimising reprojection error, as shown in the front-end odometry system introduced by Campos *et al.* [15]. On the other hand, direct methods make use of intensity values from the whole image to minimise photometric errors and, thus, calculate the camera motion [104, 34]. In addition, there are methods existing in-between direct and indirect methods, which use a combination of sparse features and direct image alignment, as shown in the SVO system by Forster *et al.* [39].

Besides the traditional passive image-based sensor, RGB-Depth (RGB-D) cameras can directly estimate depth by projecting structured light or infrared signals into the environment. There has been some work using RGB-D cameras to provide odometry [70, 71, 62], but the main drawback of the RGB-D sensor is the relatively short range and high noise of the depth estimation.

Another very recent alternative is the event camera for vision-based state estimation. Operating differently from image sensor based cameras, event-based cameras capture per-pixel brightness changes as a stream of asynchronous events. The distinctive features of event cameras are low latency (microseconds), a high dynamic range, and little to no motion blur. For a detailed survey on event cameras, please refer to [45]. However, given the unique data stream that event cameras produce, most classic computer vision algorithms need to be redesigned [150, 156]. Although event camera-based algorithms are still in the early stages of development, some promising applications of event cameras have been demonstrated using unmanned aerial vehicles in tracking and control tasks [98, 73], particularly in high-speed motions and dynamic environments [135, 29].

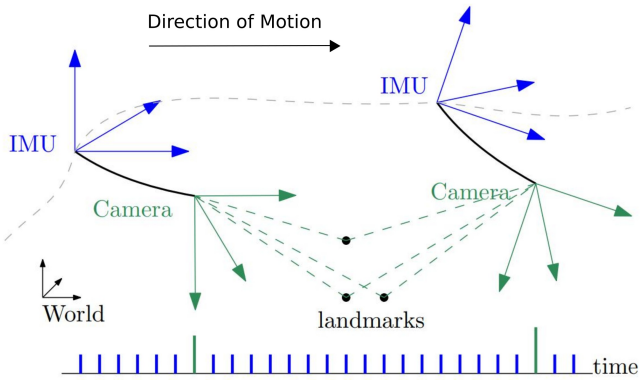


Figure 3.3: An illustration of a typical Visual-Inertial Odometry (VIO) framework: A camera (green) and an IMU (blue) are rigidly attached to one another (the black line). At two timestamps along the trajectory, the camera views observe the same landmarks.

Visual-Inertial Odometry

Visual-inertial odometry (VIO) is a specific odometry technique that integrates the visual motion tracking research described in the previous section with motion sensing achieved by gyroscopes and accelerometers within an IMU. This method has achieved significant progress among vision-based pose estimation techniques. VIO systems aim to compute high-frequency camera motion in real time, as illustrated in Fig. 3.3. The pose estimation output from a VIO system can assist high-frequency robotic tasks such as control and obstacle avoidance. It has become the standard technique for odometry in Micro Aerial Vehicles (MAVs) [12, 123, 38]. VIO is also adaptable to a variety of platforms, including handheld devices, quadrupeds, and vehicles, with applications extending to both indoor and outdoor navigation. In VIO frameworks, cameras track scene information at the camera frame rate, while IMUs provide high-frequency motion estimates between keyframes.

Extensive VIO research has been conducted using both monocular and stereo cameras. VIO algorithms are generally divided into optimisation-based or filter-based approaches. In recent years, optimisation-based methods like VINS-Mono [110], ORB-SLAM3 [15], and OKVIS [82] have gained popularity due to their improved accuracy in optimising the sensor pose trajectory. As an example, Semi-Direct Monocular Visual Odometry (SVO) [39], employs a semi-direct approach that tracks and triangulates pixels with high image gradients while simultaneously

optimising sparse features to estimate both the environmental structure and the robot's motion. On the other hand, a contrasting line of research based on Extended Kalman Filters (EKF) approaches marginalise previous states to maintain efficiency [99, 10]. Some contemporary filtering methods have reached competitive levels of performance by leveraging stochastic cloning [49]. For further insights, readers are encouraged to review detailed comparisons in [27] and [56].

Multi Camera Visual Inertial Odometry

Although VIO frameworks are generally effective, they are susceptible to failure in degenerate scenarios such as over/underexposure, textureless surfaces, or aggressive motion. Reliance on a minimal setup, typically just a monocular camera, introduces a single point of failure. Even with integrated IMU measurements, the system can quickly diverge, typically within 1-2 seconds, following a feature tracking failure. However, incorporating information from additional cameras can significantly enhance robustness and estimation accuracy, albeit at the cost of increased complexity and computational demand. A key advantage of Multi-Camera VIO (MC-VIO) is redundancy; for instance, if one camera experiences a significant reduction in tracked features, the other cameras can still effectively estimate motion.

MC-VIO has been less extensively studied due to the complexities associated with hardware and potentially high computational demands. Early work by Tribou *et al.* [131] proposed a parallel tracking and mapping system with non-overlapping FoV cameras. They modified the MCPTAM [53] system to initialise with non-overlapping FoV cameras. The system did require more specific initialisation procedures and the results showed that the non-overlapping camera configuration performed less accurately than the overlapping FoV configuration, especially in more challenging scenarios.

Recent work from Jaekel *et al.* [64] fused two pairs of stereo cameras and accounted for the uncertainty in extrinsics in both their front- and back-end systems to enhance performance. However, their experiments were limited to a few real-world scenarios and lacked ground truth comparisons. A VIO system by Liu *et*

al. [90], using three pairs of stereo cameras, estimated sensor motion by minimising photometric errors. This work was developed and tested in autonomous driving scenarios, where the authors demonstrated the benefits of a multi-camera system during nighttime driving. However, the reliance on a constant velocity motion model renders the method unsuitable for handheld sensing or drones with highly dynamic motions. Another unique omnidirectional setup, proposed by Seok *et al.* [121], included four very large FoV cameras. Although the overlapping image regions were treated as four stereo cameras, the system did not fully leverage the camera setup to track features across the full set of camera sensors. Focusing on aerial robotics, Muller *et al.* [101] implemented two pairs of stereo cameras on an MAV, oriented upwards and downwards. The images from their wide-angle cameras were divided and processed through four separate stereo VIO systems running on a Field Programmable Gate Array (FPGA). The VIO outputs, combined with an IMU, were then fused using an EKF in a loosely coupled manner. However, the experimental results were confined to an indoor office setting. Additionally, Eckenhoff *et al.* [33] utilised a Multi-State Constraint Kalman Filter (MSCKF) to integrate up to six cameras with multiple IMUs. The MSCKF's low computational demands facilitated asynchronous camera keyframe processing and real-time operation. While detailed simulation results were provided, real-world testing was restricted to a small laboratory environment.

This overview of MC-VIO indicates opportunities to leverage cameras arranged to have overlapping FoV. Such configurations allow for the potential tracking of a specific feature across multiple cameras for as long as it remains visible in any camera which could potentially reduce drift rate. There have not been many studies investigating cross-camera feature tracking. In this work, we are particularly motivated to address the common challenges and failure scenarios in monocular and stereo VIO; areas that have not been the focus of recent research.

3.2.2 Lidar Odometry

Lidar sensors behave quite differently to camera sensors. They can accurately determine the range to objects by emitting laser light toward an object and measuring the time taken for the reflected light to return to the receiver. Lidar is widely used in surveying, geomatics, atmospheric physics, and many other fields. Pose estimation with lidar has been studied since the 1990's including a seminal work from Thrun *et al.* [127]. Lidar continues to be the dominant sensor for accurate mapping, localisation, and obstacle avoidance in the application of real time robotics. Lidar odometry has progressed from using 2D single beam to full multi-beam 3D sensors. With the recent push to commercialise autonomous driving, the cost of lidar is getting lower and more configurations are being introduced to the market. For example, there are lidars with long distance ranges of up to 250 m and wide vertical FoV of 90°. Overall, lidar odometry can be divided into three categories: point-wise registration-based, feature-based and deep learning-based methods. We will go into detail to discuss the relevant research done in each category.

Registration-Based Lidar Odometry

The registration-based method generally involves aligning the input scan with another scan or previously accumulated scans, as illustrated in Fig. 3.4 (left). The Iterative Closest Point (ICP) algorithm is commonly used to align scans in a point to point fashion, characterising the registration-based method as "dense". A recent implementation of this concept is Continuous-Time ICP [26], where each scan undergoes elastic deformation to match the map (previous accumulated scans) through the simultaneous optimisation of two poses at the beginning and end of the scan, interpolated based on the lidar point timestamp. This method results in continuous-time odometry that aligns scans with the map. CT-ICP has demonstrated increased robustness to fast motions and improved precision in registration. Another approach, KISS-ICP [136], uses a point-to-point ICP algorithm with an adaptive threshold for correspondence matching to effectively estimate odometry. The key advantage of KISS-ICP is its minimal complexity, yet it achieves

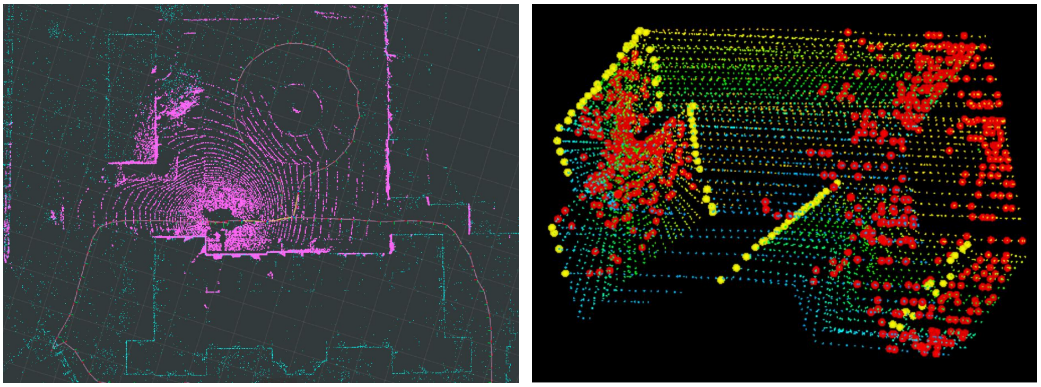


Figure 3.4: *Left*: registration-based lidar localisation, where purple denotes the current input scan and cyan shows the downsampled prior map. *Right*: feature based lidar odometry from LOAM [151], with an example of extracted edge points (yellow) and planar points (red) from a lidar cloud taken in a corridor.

performance comparable to state-of-the-art methods. An alternative to ICP-based registration, such as FastLIO2 [145], uses an Iterated Kalman Filter to compute the residual between the scan and the local map for registering the lidar pose. FastLIO2 is a computationally efficient and tightly coupled lidar inertial odometry framework. It is versatile and capable of supporting lidar sensors with frequencies up to 100 Hz.

Sparse Feature-Based Lidar Odometry

Similar to sparse feature extraction in vision-based methods, pose estimation with lidar can track handcraft features and segments in the 3D space and calculate displacement between successive scans. We can consider this approach as a "sparse" method. One of the first influential real time odometry systems is LOAM [151], where the authors built a real time system composed of two parts. The first part contains a high frequency odometry module that tracks edge and planar points to estimate the velocity of the lidar, as shown in Fig 3.4 (right). The second part entails a lower frequency module for point cloud fine matching and registration. Building on top of LOAM, LeGO-LOAM [122] can achieve similar results with reduced computational power. This was achieved by removing the ground plane with further segmentation and filtering, meaning that feature tracking and pose optimisation required lower computational complexity. However, LeGO-LOAM relies on the assumption of a wheeled robot travelling on the ground, so the ground

plane would always need to be present. In more recent work, VILENS [141] extracts and tracks geometric primitives, specifically planes and lines, across consecutive lidar scans. These planes and lines are formulated as factors and jointly optimised in a factor graph. The VILENS method is further integrated with visual, IMU and legged odometry to provide the state estimation for legged robots.

Deep Learning-Based Lidar Odometry

Deep learning-based approaches have been gaining traction, following their success in the computer vision domain. The recently proposed methods, PointNet and PointNet++ [109], demonstrated the efficiency of DNNs in addressing problems related to 3D point clouds. Although still in the early stages of research, deep learning-based pose estimation is progressing in two distinct directions. One approach adopts an end-to-end fashion to regress a pose transform between two raw input point clouds. The other approach aims to replace specific components within traditional pipelines that could benefit from deep learning enhancements.

For instance, Deep-LO [19] is an end-to-end network that processes each raw input point cloud to form vertex and normal maps. Features are extracted through dedicated vertex and normal networks and then fed into a pose estimation network to determine the relative motion. Similarly, LO-Net [85] is another end-to-end lidar odometry system, which includes a feature to mask dynamic objects within the scan. By leveraging the normals of the point cloud, LO-Net optimises a geometric consistency loss function to predict a dynamic mask. A competing method, DeepICP [92], works by first utilising PointNet++ to extract features from the lidar point cloud. Feature descriptors are then computed using a mini PointNet, enabling the corresponding point generation layer to establish point associations. Despite these advancements, end-to-end deep learning-based methods do not yet rival the performance of classical approaches.

Feature Learning from Lidar Points

A significant challenge in extracting features from lidar point clouds has been the reliance on handcrafted features. This challenge primarily arises due to the difficulty

in designing networks that can learn from unstructured point clouds, especially when these clouds are sparse. Vision-based approaches have demonstrated that learned features can outperform classical handcrafted features [94, 93], particularly under challenging conditions such as varying illumination, and change of viewpoints. In the early stages of lidar feature learning research, it was unclear how to replicate these improvements, as learning networks were not initially designed to support continuous 3D point data. Recently, however, new approaches to 3D point cloud learning have emerged. These primarily fall into two categories: a voxelised spatial representation or an unstructured set of points representation.

The voxelised spatial representation was first introduced because it can be easily adapted for convolutional filters. A typical convolutional filter works efficiently on 2D images. For 3D voxels, the convolution filter can still be applied but would result in a much higher dimensionality. Choy *et al.* [21] used this approach to show some promising results when applied to object reconstruction. The downside of this approach is the significant computational overhead of dealing with voxel space. When a 32x32x32 voxel grid was used, the dimensionality turned into 32,768. Despite high computational requirements, the voxelised representation can only retain a coarse structure, losing details on small objects such as chair legs.

The use of an unstructured set of points as input for neural networks was popularised by PointNet [109]. This network learns a symmetric function to aggregate information from unordered inputs, ensuring invariance to permutation. In contrast to voxelised representations, a raw point cloud with 1,024 points has a dimensionality of 3,072, which is ten times smaller than that typically found in voxelisation approaches. Another major advancement is PointCNN [88], which simultaneously weights and permutes the input features before applying a specialised convolution known as X-Conv. PointCNN is particularly advantageous because it can account for the shape of points while being invariant to their order. For certain applications, such as object classification, PointCNN proves more suitable than other networks designed specifically for point clouds, including PointNet.

Recent advancements have seen the introduction of sparse tensor networks, specifically designed to operate directly on point clouds. Notable work such as Choy *et al.* [20] employs sub-manifold sparse convolutions within its neural network architecture. These convolutions are more efficient as they preserve a greater degree of sparsity compared to traditional sparse convolutions [52]. Typical lidar scans, which are comprised of point clouds with substantial spatial sparsity, can thus be effectively represented as sparse tensors. The application of these advanced techniques allows for the development of deeper networks that require less memory and computational power, thereby facilitating practical real-time capabilities. This advancement has spurred a wave of research focused on point cloud data applications, including semantic and instance segmentation [137, 124], object detection [18], and learning-based lidar localisation [86, 61].

3.3 Localisation

Related to but distinct from odometry, localisation is defined as the process by which a system determines its 6 DoF pose with a pre-existing 3D map or database. Within this framework, place recognition plays a crucial role, aiming to match a given image or lidar scan with the most similar entry in a pre-established database, although it is not primarily focused on precise pose estimation. The research on localisation and place recognition, informed by data from vision and lidar sensors, can be organised into four main categories. These are 1) visual localisation that uses a visual database, 2) lidar-based localisation against a lidar map, 3) a hybrid approach integrating both visual and lidar data against a unified database, 4) cross-modal localisation, which involves using visual data to localise within a lidar map or vice versa.

3.3.1 Visual Localisation

Among the above listed categories, visual-only localisation has attracted the most research interest. Visual localisation techniques strive to determine a 6 DoF pose by localising against a map constructed from numerous images. Typically, these maps

are created using Structure-from-Motion (SfM) pipelines [119], which facilitate the recovery of 3D structures from multiple viewpoints.

In the context of localisation, visual place recognition is treated as an image retrieval challenge. Early methods like FAB-MAP [22] and DBoW [46] addressed this issue using local features. However, more contemporary approaches such as NetVLAD [4], PatchNetVLAD [54], and EigenPlaces [9] utilise neural architectures to derive global descriptors for retrieval purposes. Following retrieval, metric pose estimation is conducted using local feature matching, employing techniques such as SIFT [91], SuperPoint [28], or R2D2 [113]. This approach is demonstrated in methods like HLoc by Sarlin et al. [117] and the system from Sattler *et al.* [118], which utilise a hierarchical strategy for large-scale visual localisation.

In addition to the methods previously discussed, alternative strategies have been pursued for visual localisation. For instance, MeshLoc [106] demonstrates the feasibility of constructing a dense 3D mesh model from multi-view stereo point clouds and rendered synthetic images can then be used to create a visual database. Another novel approach is presented in [146], where the focus shifts from using a global descriptor for query images to detecting instances of buildings in outdoor environments. This method retrieves the best matches from a database of buildings, showing enhancements in long-term and large-scale localisation datasets compared to traditional hierarchical frameworks.

3.3.2 Lidar Localisation

Similar to visual localisation, lidar localisation also maintains a database consisting of 3D lidar scans to facilitate map-based navigation. Early works first build a complete prior map of the environment, then during online operation, incoming scans are statistically matched inside the prior map. As an example, Baldwin and Newman [5] fuses two 2D push-broom lidars, one vertically and the other horizontally. The vertical lidar builds a dense map while the horizontal lidar estimates linear and rotational velocity. As the vehicle moves, these lidar scans are built into a 3D swathe. For each 3D swathe, an objective function minimises the

Kullback-Leibler divergence of distribution of projected 3D points against the prior map. Another work [142] uses 3D scene structure and ground plane reflectivity to localise a vehicle in a prior map. The method models the world as a mixture of Gaussians over 2D grid cells. By characterising the distribution of z-height and reflectivity in each cell, they show their results are robust to diverse environments, including appearance changes on the road.

Recently, more research has been proposed to achieve global lidar localisation, which can also act as a loop closure component for lidar-based odometry, as discussed in 3.2.2. The approach to retrieval and matching is often hierarchical, centring on the creation of global descriptors for each scan. Both handcrafted and learned descriptors have been employed to enhance localisation accuracy. He *et al.* [55] proposed an approach projects the 3D point cloud into multiple 2D planes, generates descriptors for each of the planes based on their point density, and combines them into a global descriptor. Another example, ScanContext [72] projects the whole 3D point cloud into the bird's eye view and voxelise into polar and Cartesian coordinates. The descriptors are generated based on voxel cell height. ScanContext is designed specifically for the noisy and sparse point cloud acquired in an outdoor environment. For learning-based descriptors, Logg3dNet [134] and PointNetVlad [132] apply deep learning networks to generate strong global descriptors, showing better performance for large scale outdoor environments. Additionally, some techniques enhance localisation robustness by learning semantics within the lidar point clouds for outdoor environments. For example, Kong *et al.* [76] propose a place recognition methodology based on semantic graphs that preserves the topological integrity of the point cloud. This semantic approach was demonstrated to offer greater resilience to environmental changes.

Segment-Based Lidar Localisation

Instead of using the whole point cloud, segment-based localisation methods extract clusters of points - *segments* [30] - from the raw input point cloud, and use them to localise within a prior map. The segment is not required to be an object

with semantic meaning. It could be a portion of a flat wall or a part of an object. Generated descriptors from a whole lidar scan can retain all the contextual information in the scene, such as landmarks and objects, but this method often suffers from occlusions or changes in the scene. Furthermore, processing the whole point cloud leads to heavier computation. Segments, being "semi-dense", are large enough to capture repeatable and reliable features while also being robust to temporal changes and occlusions. Note that this method was made possible by using dense lidars with multiple beams or accumulating multiple lidar scans.

Segment-based lidar localisation was pioneered by SegMatch [32]. SegMatch is able to extract more reliable feature matches and outperformed the previous state-of-the-art 3D point descriptors. In later work [31], a SegMap module was introduced into a SLAM system where meaningful segment features were extracted using a data-driven descriptor. The approach proposed that learned features from a voxelised input space are suitable for semantic descriptions.

Inspired by SegMatch, [128] extends the feature descriptor to additionally consider segment geometry properties and point distributions. Their work focused on solving localisation in unstructured environments such as forests, as well as in structured environments, such as urban settings. The revised hand-crafted feature descriptors based on SegMatch improved the detection performance in unstructured environments. The main drawback of this approach is the high computational complexity as it uses large feature descriptors in the pruning stage, making it intractable for real time operation. However, in their following work [129], the authors address the computational issue by defining a learning architecture to learn descriptors directly from the segmented point cloud. The Efficient Segmentation and Matching (ESM) algorithm runs on a CPU in real time and was tested in urban and rural environments. In more recent work, Locus [133] combines global descriptors with segments and spatiotemporal high-order pooling to enhance place recognition, proving particularly effective in overcoming challenges such as changes in viewpoint and occlusions.

3.3.3 Combined Visual Lidar Localisation

In scenarios where mobile robotic platforms are equipped with multiple sensors, several methods have been developed to integrate data from both cameras and lidars for localisation against a unified database of images and lidar scans. OneShot [112] features a network tailored to concurrently process lidar segments and their corresponding camera images, creating unique descriptors for each. This method improves pose estimation by segmenting lidar scans and aligning their descriptors with a database, resulting in superior retrieval rates when visual data is combined with lidar information. Another alternative technique by Bernreiter *et al.* [8] utilises a spherical format to develop descriptors for paired lidar scans and images. This method stands out due to its flexibility, accommodating various configurations of camera and lidar sensors during both training and querying phases. Recent advances also include Adafusion [78], which employs an attention network to implement adaptive weighting to the image and lidar pairs, forming descriptors. This adaptive approach allows for different contributions from each sensor type, leading to enhanced retrieval rates and greater robustness against environmental changes. Additionally, the newly proposed LC^2 [79] converts 2D images and 3D point clouds into 2.5 D depth images to minimise modality differences. By training a network to generate joint descriptors, this method has shown to be particularly effective in localising under varying lighting conditions.

3.3.4 Cross-Modal Localisation

In the field of cross-modality, a common strategy involves using cameras to localise within a lidar point cloud map. Early initiatives, such as the one described by Borges *et al.* [11], involved extracting edges from the lidar point cloud and aligning them with lines derived from camera images to localise the camera within the point cloud map. To mitigate the issue of localisation drift, Yu *et al.* [149] introduced an enhanced method that extracts edges and lines to link camera images with the lidar map. This method leverages a prior pose obtained from a visual odometry (VO) system, effectively reducing drift by aligning live images with the lidar map.

However, these techniques typically excel in structured environments where lines and edges are prominent features.

With autonomous driving gaining more research interest, there is a growing focus on developing effective localisation methods using widely available and cost-effective camera sensors. For instance, Wolcott *et al.* [143] developed a technique to generate a greyscale synthetic view from a mesh map built by a survey vehicle equipped with a 3D lidar. By minimising the normalised mutual information between live camera view and synthetic view, they were able to accurately estimate a pose within the existing lidar map. Similarly, Pascoe *et al.* [108] synthesised colour images from a textured mesh, accommodating changes in camera calibration.

Recent developments have further enhanced cross-modal image-to-lidar registration techniques. For example, Zuo *et al.* [157] align dense stereo depth images with a lidar map to correct visual odometry drift. Meanwhile, Liang *et al.* [89] integrate semantic scene understanding by using the lidar point cloud map and its semantic labels to generate a prior semantic map. During operation, semantic cues from live images are used for localisation within this prior map. Both approaches fall under the broader category of camera-lidar registration, which typically requires an initial prior for localisation.

3.4 SLAM Systems

SLAM is an essential capability to enable an autonomous robot or system to build a map of its unknown environment. We would think SLAM as the joint application of odometry and place recognition. The odometry module estimates the device's incremental motion over time using sensors like cameras, lidar, or IMUs, updating the local map and position incrementally. However, odometry naturally accumulates errors, leading to drift. The loop closure module addresses this by recognising previously visited locations and correcting accumulated errors, which is often a mode of the localisation method. Here we will first give an overview of the SLAM system modules with a focus on backend optimisation module, and then review some well known visual and lidar SLAM systems.

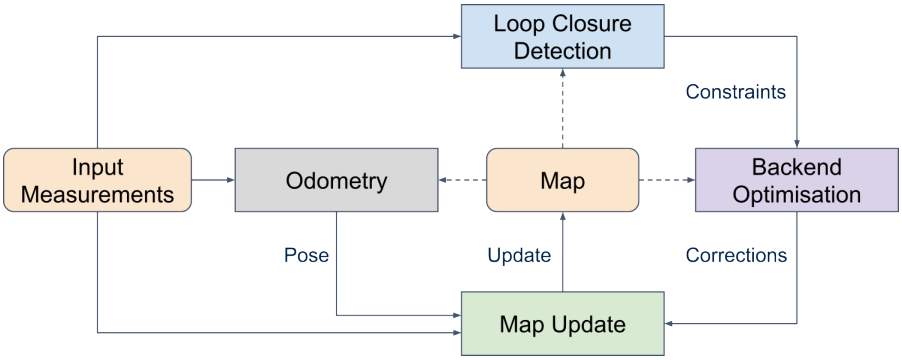


Figure 3.5: Overview of a SLAM system.

3.4.1 System Overview

As illustrated in Fig. 3.5, a typical SLAM system is comprised of the following modules: a front-end odometry (discussed in Sec. 3.2), a loop closure detection (discussed in Sec. 3.3), a map update, and a backend optimisation module. Measurements such as lidar scans, images and IMU data are fed into the odometry module, where initial poses are estimated. The map is continuously updated with new inputs. When a loop closure is detected, all poses are optimised, and the trajectory is corrected, which in turn, triggers a map update.

The backend optimisation techniques can be categorised into two types. The *full batch smoother* re-estimates all the states and variables at every time stamp. The *fixed lag smoother* maintains a section of past states and variables, keeping a fixed-size sliding window of the last N states. This method strikes a balance between speed and accuracy, making it suitable for online applications. The work by Leutenegger *et al.* [81] integrates visual measurements with an IMU to maintain a bounded-size optimisation window. In another study, iSAM2 [65] introduces a novel data structure, the Bayes Tree graphical model, for factor graph-based optimisation, achieving efficient real-time performance in SLAM. (For more details on optimisation and pose graph optimisation, please refer back to Sec. 2.2.) Most backend optimisation techniques use an external optimisation library. Some of the most popular libraries include General Graph Optimisation (g2o) [77], Ceres [1], and Georgia Tech Smoothing and Mapping (GTSAM) [24].

Note that an alternative to the optimisation technique is the *filtering* approach, where only one previous state is kept when estimating at each time stamp. This includes methods like the EKF, the Unscented Kalman Filter (UKF), and MSCKF [100].

3.4.2 Visual SLAM Systems

Early significant contributions to visual SLAM include the seminal work by Davison *et al.* [23], which demonstrates real-time monocular SLAM with robust localisation and mapping capabilities. Another foundational work by Klein and Murray introduces Parallel Tracking and Mapping (PTAM) [74], which decouples the tracking and mapping tasks for improved efficiency. More recent advancements, such as ORB-SLAM [102], leverage ORB feature-based methods to enhance robustness and real-time performance, achieving reliable results in diverse environments. Dense visual SLAM approaches, such as LSD-SLAM by Engel et al. [35] and KinectFusion by Newcombe *et al.* [103], further extend capabilities to reconstruct detailed 3D environments.

Significant progress has been made in recent years with visual SLAM systems like ORB-SLAM3 [15], which incorporates stereo and RGB-D sensors for enhanced accuracy and robustness, offering both SLAM and localisation modes [46]. Works such as ElasticFusion [140] have utilised RGB-D cameras for dense visual SLAM in small indoor environments. Similarly, Fusion++ [95] focuses on object-level mapping while performing localisation in indoor office scenes.

The integration of deep learning techniques has also shown promise. Li *et al.* [84] and Gao *et al.* [47] utilise DNNs to improve feature extraction and loop closure detection. CNN-SLAM by Tateno *et al.* [125] demonstrates that using depth prediction with CNNs for monocular images can achieve more robust performance where traditional monocular SLAM approaches tend to fail. Additionally, recent research such as DS-SLAM [148] leverages a semantic segmentation network with a moving consistency check method to reduce the impact of dynamic objects, thereby improving SLAM accuracy in dynamic scenes.

3.4.3 Lidar SLAM systems

Many lidar-based SLAM systems typically consist of both an odometry and a mapping module. Here, we focus our discussion on several examples of lidar SLAM systems that incorporate both odometry and place recognition modules.

Cartographer [59] is an open-source SLAM system developed by Google. It registers each scan to a submaps first. Then when the system revisits a previously mapped area, it detects loop closures by comparing the current scan against the entire map using a branch-and-bound approach, then it uses the generalised ICP algorithm to find transformations that align the scans. All constraints are then integrated into a global pose graph, which is optimised to minimise the overall error, thus refining both the map and the trajectory estimates. Similarly, HDL-Graph-SLAM [75] registers individual lidar scans by applying normal distribution transform to obtain the odometry. It detects loop closures based on the geometric distance threshold, followed by global pose graph optimisation to smooth the trajectory and to distribute the drift across the trajectory.

In ART-SLAM [41], odometry is obtained through a keyframe-based approach that uses scan-to-scan matching. The system uses a three-phase algorithm for loop detection. Initially, potential loop closures are detected based on the overlap of lidar scans rather than relying solely on positional data, which can be affected by drift. The system projects lidar scans into spherical coordinates and calculates the overlap between pairs of scans using a neural network model, which gives the loop closure candidates. Finally, it conducts a global pose graph optimisation.

Chanoh *et al.* [107] combine continuous-time trajectory optimisation with map-centric loop closure techniques to improve the accuracy of SLAM in large-scale environments. This method uses a combination of multi-resolution surfel maps and dense surfel fusion techniques. When a loop closure is detected, the system identifies the misalignment between the current map and the global map, applying non-rigid deformation to correct these discrepancies. Similarly, Wildcat [111] also uses the continuous time trajectory representation where the odometry module operates in a sliding window fashion and continuously aligns the surfel map generated by the

lidar. When significant overlap between submaps is detected, the Wildcat system adds LC constraints to the global pose graph optimisation. Note that the Wildcat system won the Hilti SLAM Challenge 2022 and please refer to [152] for details.

3.5 Datasets

Datasets are pivotal resources in the development and evaluation of robot navigation algorithms, which are fundamental to the operation of autonomous robots and augmented reality systems. These datasets typically include sensor data from lidars, cameras, and IMUs, providing diverse environmental representations crucial for testing the robustness and accuracy of odometry, localisation or mapping algorithms. Notable datasets like the KITTI [48], TUM [120], EuRoC [14], and Oxford RobotCar [7] datasets offer varied scenarios ranging from urban driving to indoor navigation. They enable researchers to benchmark against established results and foster innovations by providing real-world conditions that challenge and refine algorithmic approaches. Such datasets are indispensable for advancing the field, as they not only facilitate the comparison of competing methodologies but also help in identifying new challenges and areas for improvement.

Benchmark datasets for robot navigation can be categorised according to their operational domains, sensory data and quality of ground truth measurements.

Vision and IMU Datasets

In the field of VIO, the research community frequently uses two datasets: EuRoC and TUM VI. The EuRoC dataset consists of 11 sequences of data captured from a micro aerial vehicle outfitted with a Skybotix stereo visual-inertial sensor. This setup includes hardware time-synchronised stereo camera images and IMU measurements. Half of the sequences are captured with a motion capture system which gives millimetre-level accurate 6 DoF ground truth poses, but the other half only has 3 DoF position ground truth.

Expanding the range of environments, the TUM VI dataset includes both outdoor and indoor sequences. This dataset was collected using a hand-held device equipped

with a global shutter stereo camera and an IMU, also achieving hardware time synchronisation. With 28 sequences that extend into outdoor settings, the TUM VI dataset starts and ends in an indoor motion capture setup, ensuring accurate ground truth poses are available for these parts of the sequences. This feature provides a broader array of scenarios for benchmarking visual-inertial navigation systems.

Vision, Lidar and IMU Datsets

Datasets such as KITTI [48], Oxford Radar RobotCar [7], Boreas [13], WoodScape [147], and UMich [16] are specifically designed for the autonomous driving domain and include IMU, camera images, and lidar data. WoodScape and UMich feature 360° cameras, whereas KITTI utilises a linear array of stereo cameras, including both colour and grayscale. The UMich dataset, recorded using a Segway scooter, spans both indoor and outdoor environments. In contrast, KITTI and WoodScape, which employ larger vehicles, are restricted to outdoor settings.

In addition to vision, lidar and IMU sensors, the Oxford Radar RobotCar [7] and the Boreas [13] datasets include a 360° Navtech radar and each covers several hundred kilometres of driving distance.

All datasets provide accurate ground truth trajectories using GPS/GNSS-INS measurements, giving trajectories accurate to several centimetres.

Datasets for Challenging Environments

As navigation systems evolve, there is an increasing demand for datasets that present more challenging conditions, emphasising common failure cases and complex real-world scenarios. Here we discuss a few recent datasets that are focused on multimodal and challenging environments [68, 58].

For instance, the 2021 Hilti Challenge dataset [58] features an AlphaSense five fisheye camera module, providing a 270° continuous FoV. For comprehensive lidar coverage, it utilises the Ouster OS0-64 lidar, which offers a 90° FoV and ±3 cm range accuracy. The 2021 Hilti dataset includes sequences that navigate through large, feature-poor environments like an empty car park, covering a variety of indoor and outdoor settings, as well as aggressive movement in tight spaces. In addition,

the Boreas dataset [13], collected over a year by driving repeated along one route, captures significant seasonal variations and adverse weather conditions such as rain and snow. This dataset offers a unique opportunity to compare algorithms based on multi-season performance. Another compelling dataset by Giubilato *et al.* [51] was recorded on Mt. Etna, Sicily, to simulate environments similar to Mars and the Moon, presenting unique challenges for navigation algorithms. This dataset was captured using a mobile rover-like platform equipped with a DLR planetary stereo camera, a solid-state lidar, and an IMU. The challenging conditions include severe visual aliasing and a paucity of distinct structures, which significantly test the capabilities of both visual and lidar-based navigation techniques.

4
Multi-Camera Visual Odometry

In addressing the front-end odometry module of a SLAM system, we note that that while VIO frameworks can be effective in benign circumstances, they encounter difficulties in degenerate scenarios such as over/under exposure, textureless surfaces, or during aggressive motion. The typical approach of integrating IMU measurements with a monocular camera setup introduces a single point of failure. The system can diverge rapidly, usually within 1-2 seconds, when the feature tracking fails. To enhance robustness and estimation accuracy, it can be beneficial to integrate information from additional cameras, albeit at the expense of increased complexity and computational demand. One significant advantage of Multi-Camera VIO (MC-VIO) is its redundancy; if one camera experiences a drop in tracked features, the other cameras can continue to estimate motion effectively. Furthermore, the use of multiple cameras in a SLAM system often take advantage of overlapping views, which can facilitate feature tracking across multiple cameras. However, this configuration introduces the challenge of increased computational demands. To manage this, one effective approach is to selectively process only a subset of the most informative features. This strategy aims to optimise computational resources by focusing on the most critical data, thereby enhancing the system's efficiency without compromising the accuracy of motion estimation. In this chapter, we

present a factor graph formulation that combines multiple fisheye cameras as well as demonstrations of the system operating in many challenging scenarios.

4.1 Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry

The following article was published in the *IEEE Robotics and Automation Letters (RA-L)* and it was also presented at the *IEEE International Conference on Robotics and Automation (ICRA)* 2022 [154]. An accompanying video is available online at: https://youtu.be/cLWeAT72e0U.

Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry

Lintong Zhang[1], David Wisth[1], Marco Camurri[1], and Maurice Fallon[1]

***Abstract*—We present a multi-camera visual-inertial odometry system based on factor graph optimization which estimates motion by using all cameras simultaneously while retaining a fixed overall feature budget. We focus on motion tracking in challenging environments, such as narrow corridors, dark spaces with aggressive motions, and abrupt lighting changes. These scenarios cause traditional monocular or stereo odometry to fail. While tracking motion with extra cameras should theoretically prevent failures, it leads to additional complexity and computational burden. To overcome these challenges, we introduce two novel methods to improve multi-camera feature tracking. First, instead of tracking features separately in each camera, we track features continuously as they move from one camera to another. This increases accuracy and achieves a more compact factor graph representation. Second, we select a fixed budget of tracked features across the cameras to reduce back-end optimization time. We have found that using a smaller set of informative features can maintain the same tracking accuracy. Our proposed method was extensively tested using a hardware-synchronized device consisting of an IMU and four cameras (a front stereo pair and two lateral) in scenarios including: an underground mine, large open spaces, and building interiors with narrow stairs and corridors. Compared to stereo-only state-of-the-art visual-inertial odometry methods, our approach reduces the drift rate, relative pose error, by up to 80 % in translation and 39 % in rotation.**

***Index Terms*—Visual-Inertial SLAM; Omnidirectional Vision; Localization**

I. Introduction

STATE estimation is a fundamental capability required for autonomous robot navigation in real-world scenarios. Motion tracking using cameras is very popular due to their low weight, small form factor, and low hardware cost. Specifically, Visual-Inertial Odometry (VIO) methods, which fuse feature tracking from a camera with estimation from an Inertial Measurement Unit (IMU), have now become the standard for odometry on Micro Aerial Vehicles (MAVs) [1]. These systems can also be deployed on a range of platforms, such as handheld rigs, quadrupeds, and vehicles with applications ranging from indoor and outdoor navigation to underground exploration [2] (Fig. 1).

Manuscript received: September, 9, 2021; Revised November, 30, 2021; Accepted December, 6, 2021.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This research was part funded by the EU H2020 Projects THING and Innovate UK-funded ORCA Robotics Hub (EP/R026173/1), a Royal Society University Research Fellowship (Fallon) and a Google DeepMind studentship (Wisth). It has been conducted as part of the ANYmal research community.

[1]Oxford Robotics Institute, Department of Engineering Science, University of Oxford, UK lintong, davidw, mcamurri, mfallon@robots.ox.ac.uk

Digital Object Identifier (DOI): see top of this page.

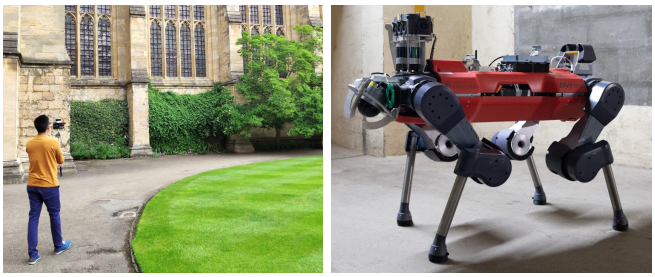


Fig. 1. We tested our multi-camera odometry algorithm with data from a handheld sensor rig in New College, Oxford (*left*) and the ANYmal quadruped [3] during preparation for the DARPA SubT Challenge [2] (*right*). Video: https://youtu.be/cLWeAT72e0U.

When performing VIO, cameras can track scene information at the frame rate while the IMU can provide high frequency motion estimates between keyframes. Although VIO frameworks are generally effective, they are prone to failure in degenerate scenarios such as over/under exposure, textureless surfaces, and when there is aggressive motion. Another issue with the minimal setup composed of just a monocular camera is that this system results in a single point of failure. Even if IMU measurements can be integrated, the system can quickly diverge within a few seconds of a feature tracking failure.

Fusing information from additional cameras can greatly improve robustness and estimation accuracy at the cost of extra complexity and computational burden. The most obvious benefit of Multi-Camera VIO (MC-VIO) is redundancy, e.g., if one of the cameras suffers from a sudden drop in tracked features, the other cameras can still estimate motion. Furthermore, if the cameras are arranged to have overlapping Fields of View (FoV), a feature could potentially be tracked across the cameras for as long as it is visible in any camera. Multi-camera solutions have been presented in the past [4], [5], [6], however, no proposal has studied cross camera feature tracking.

In this paper, we explore how to effectively track features across all the cameras and how to select the best subset of features to keep the computational time bounded.

A. Contribution

The main contributions of this work are the following:

- A novel factor graph formulation that tightly fuses tracked features from any number of stereo and monocular cameras, along with IMU measurements, in a single consistent optimization process.
- A simple and effective method to track features across cameras with overlapping FoVs to reduce duplicate landmark tracking and improve accuracy.

- A submatrix feature selection (SFS) scheme that selects the best landmarks for optimization with a fixed feature budget. This bounds computational time and achieves the same accuracy compared to using all available features.
- Extensive experimental evaluation across a range of scenarios demonstrating superior robustness, particularly when VIO with an individual camera fails.

The proposed algorithm, VILENS Multi-Camera (VILENS-MC), builds upon our previous VILENS estimation system [7], [8], by fusing multiple cameras and improving front-end feature processing.

II. Related Work

There has been extensive research into monocular and stereo camera VIO. These methods can be categorized into either optimization or filter based approaches. In recent years, optimization based methods such as VINS-Mono [9], ORB-SLAM3 [10], and OKVIS [11] have become popular due to their ability to optimize a trajectory of poses. This is in contrast with methods based on Extended Kalman Filters (EKF) which marginalize all previous states. However, some modern filtering methods have achieved competitive performance via stochastic cloning [12]. For a more detailed review, readers can refer to comparisons in [13] and [14].

MC-VIO has been less extensively studied because of hardware complexity and potentially high computational requirements. Recent work from Jaekel et al. [4] fused two pairs of stereo cameras and accounted for the uncertainty in the extrinsics in both their front- and back-end systems to improve performance. However, they had limited real-world experiments and no ground truth comparisons.

A VIO system by Liu et al. [15], with three pairs of stereo cameras, estimated poses by minimising photometric errors. This work was developed and tested in autonomous driving scenarios where the authors showed the benefit of a multi-camera system when driving at night. However, since a constant velocity motion model was used, their method is not suitable for handheld sensing or robots with highly dynamic motions.

A unique omnidirectional setup was proposed by Seok et al. [5] with four large FoV cameras. The four overlapping image regions were treated as four stereo cameras, but the system did not fully take advantage of the camera setup to track features across the camera pairs.

With a focus on aerial robotics, Müller et al. [16] used two pairs of stereo cameras on an MAV (looking up and down). The images from their wide-angle cameras were split in half and fed into four separate stereo VIO systems running on a Field Programmable Gate Array (FPGA). The VIO outputs and an IMU were then fused together using an EKF in a loosely coupled fashion. However, the experimental results were limited to an indoor office building.

Kuo et al. [17] introduced a more general design for multi-camera Simultaneous Localization and Mapping (SLAM), which involved an adaptive initialization scheme, keyframe selection, and map management. Their system was based on SVO [18] and their approach required minimal parameter tuning. However, their real-world experiments showed little improvement in accuracy when using a multi-camera setup.

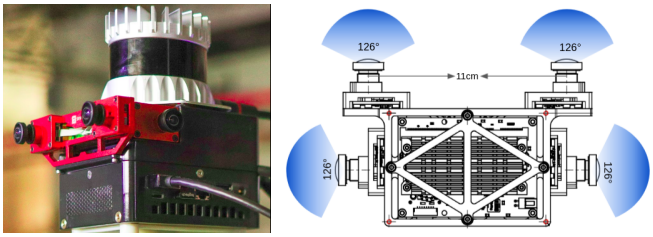


Fig. 2. VILENS-MC is evaluated using a custom built multi-camera handheld device. See details in Section VII-A.

Meanwhile, using a Multi-State Constraint Kalman Filter (MSCKF), Eckenhoff et al. [6] fused six cameras with multiple IMUs. The MSCKF's low computational requirements allowed for asynchronous camera keyframe processing and real-time operation. They presented detailed simulation results, however, real-world experiments were limited to a small lab environment.

We introduce a MC-VIO framework to fuse an arbitrary number of stereo and monocular cameras, with a focus on multi-camera feature selection and tracking.

We are particularly motivated to address common challenges and failure cases in mono and stereo VIO, which have not been a focus of recent studies. Hence, our system is designed and tested in a variety of challenging environments, covering fast and abrupt motions, severe illumination changes, indoor/outdoor scenes, and dark underground environments.

III. Problem Statement

We aim to estimate the position, orientation, and velocity of a mobile platform equipped with an IMU and multiple hardware-synchronized cameras. Fig. 2 shows the multi-camera device used in for the majority of the experimental results.

The relevant reference frames are as follows: the earth-fixed world frame $\mathtt{W}$, the platform-fixed base frame $\mathtt{B}$, the IMU frame $\mathtt{I}$, and n individual camera frames $\mathtt{C}_c$ where $c \in \{1, ..., n\}$.

Unless otherwise specified, the position ${}_{\mathtt{W}}\mathbf{p}_{\mathtt{WB}}$ and orientation $\mathbf{R}_{\mathtt{WB}}$ of the base (with ${}_{\mathtt{W}}\mathbf{T}_{\mathtt{WB}} \in \mathrm{SE}(3)$ as the corresponding homogeneous transform) are expressed in world coordinates; the linear velocity is in the base frame ${}_{\mathtt{B}}\mathbf{v}_{\mathtt{WB}}$, and IMU biases ${}_{\mathtt{I}}\mathbf{b}^g$, ${}_{\mathtt{I}}\mathbf{b}^a$ are expressed in the IMU frame.

A. State and Measurements Definition

The state of the sensor rig at time t_i is defined as follows,

$$\boldsymbol{x}_i \triangleq [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i^g \ \ \mathbf{b}_i^a] \in \mathrm{SO}(3) \times \mathbb{R}^{12} \tag{1}$$

where: $\mathbf{R}_i$ is the orientation, $\mathbf{p}_i$ is the position, $\mathbf{v}_i$ is the linear velocity, and $\mathbf{b}_i^g$, $\mathbf{b}_i^a$ are, respectively, the usual IMU gyroscope and accelerometer biases. In addition to the states, we track point landmarks $\boldsymbol{m}_\ell$ as triangulated visual features.

The objective is to estimate the optimized trajectory $\mathcal{X}_k$ of all states $\boldsymbol{x}_i$ and landmarks $\boldsymbol{m}_\ell$ visible up to the current time t_k within a fixed lag smoothing window.

The measurements from the set of multiple cameras $\mathcal{C}$ and an IMU $\mathcal{I}$ are hardware synchronized but received at different frequencies. We define $\mathcal{Z}_k$ as the full set of measurements received within the smoothing window.

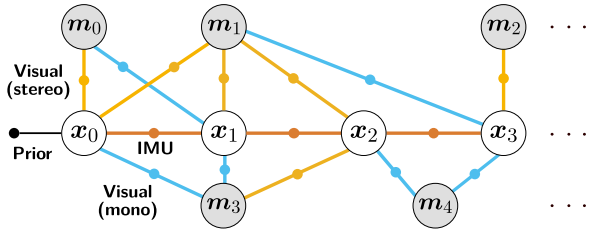


Fig. 3. Sliding window factor graph structure, showing prior, visual, and preintegrated IMU factors. Landmarks can be tracked across both stereo and mono cameras to create longer feature tracks and improve accuracy.

B. Maximum-a-Posteriori Estimation

We maximize the likelihood of the measurements $\mathcal{Z}_k$, given the history of states $\mathcal{X}_k$,

$$\mathcal{X}_k^* = \arg\max_{\mathcal{X}_k} p(\mathcal{X}_k|\mathcal{Z}_k) \propto p(\mathcal{X}_0)p(\mathcal{Z}_k|\mathcal{X}_k) \tag{2}$$

The measurements are formulated as conditionally independent and corrupted by white Gaussian noise. Therefore, Eq. (2) can be expressed as the following least squares minimization [19],

$$\mathcal{X}_k^* = \arg\min_{\mathcal{X}_k} \|\mathbf{r}_0\|^2_{\Sigma_0} + \sum_{i\in \mathsf{K}_k} \left(\|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\Sigma_{\mathcal{I}_{ij}}} + \sum_{\ell \in \mathsf{M}_i} \|\mathbf{r}_{\boldsymbol{x}_i,\boldsymbol{m}_\ell}\|^2_{\Sigma_{\boldsymbol{x}_i,\boldsymbol{m}_\ell}} \right) \tag{3}$$

where $\mathcal{I}_{ij}$ are the IMU measurements between t_i and t_j and K_k are all the keyframes indices in the sliding window up to t_k. Each term is the residual associated to a factor type, weighted by the inverse of its covariance matrix. The residuals include prior, IMU, and visual landmark factors.

IV. Factor Graph Formulation

In this section, we describe the measurements, residuals, and covariances of the IMU and visual factors shown in Fig. 3.

A. Preintegrated IMU Factors

We follow the standard manner of IMU measurement preintegration from [1] to constrain the pose, velocity, and biases between two consecutive nodes of the graph, providing high frequency state updates between nodes. The residual has the following form,

$$\mathbf{r}_{\mathcal{I}_{ij}} = \left[\mathbf{r}^\mathsf{T}_{\Delta\mathbf{R}_{ij}}, \mathbf{r}^\mathsf{T}_{\Delta\mathbf{v}_{ij}}, \mathbf{r}^\mathsf{T}_{\Delta\mathbf{p}_{ij}}, \mathbf{r}_{\mathbf{b}^a_{ij}}, \mathbf{r}_{\mathbf{b}^g_{ij}} \right] \tag{4}$$

For a detailed definition of the above residuals, see [1].

B. Stereo and Mono Landmark Factors

We define a visual landmark in Euclidean space as $\boldsymbol{m}_\ell \in \mathbb{R}^3$. Given the platform pose $\mathbf{T}_i$ (for simplicity, we omit the fixed transform between base and camera), we can project $\boldsymbol{m}_\ell$ onto the image plane with the function $\pi : \mathrm{SE}(3) \times \mathbb{R}^3 \mapsto \mathbb{R}^2$, resulting in the projected coordinates $(u_\ell, v_\ell) \in \mathbb{R}^2$ on the image plane (orange/green circles in Fig. 4). Thus, the residual at state $\boldsymbol{x}_i$ for landmark $\boldsymbol{m}_\ell$ is defined as [7],

$$\mathbf{r}_{\boldsymbol{x}_i,\boldsymbol{m}_\ell} = \begin{pmatrix} \pi^L_u(\mathbf{T}_i, \boldsymbol{m}_\ell) - u^L_{i,\ell} \\ \pi^R_u(\mathbf{T}_i, \boldsymbol{m}_\ell) - u^R_{i,\ell} \\ \pi_v(\mathbf{T}_i, \boldsymbol{m}_\ell) - v_{i,\ell} \end{pmatrix} \tag{5}$$

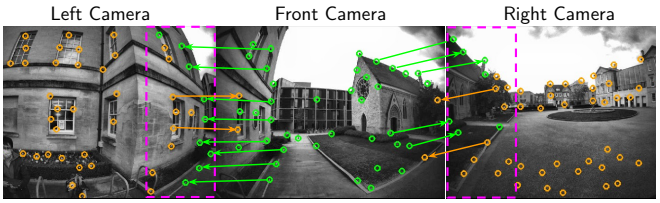


Fig. 4. VILENS-MC takes advantage of any overlapping image regions (purple rectangles) in a multi-camera setup to track features across cameras. This increases feature track length and avoids tracking the same feature independently in different cameras. The arrows indicate features being tracked from one image to another.

where $(u^L, v), (u^R, v)$ are the pixel locations of the detected landmark. $\Sigma_{\boldsymbol{m}}$ is computed using an uncertainty of 0.25 pixels. We also account for lens distortion using the covariance warping method from [8]. For monocular landmarks, only the first and the last rows of Eq. (5) are used.

The location of the landmarks detected by the stereo camera pair is initialized using stereo triangulation. For landmarks detected in monocular cameras, we triangulate the feature location over the last N_{obs} frames using the Direct Linear Transform (DLT) algorithm from [20].

V. Cross Camera Feature Tracking

Most multi-camera systems do not take advantage of the overlapping FoVs of their setups, which produce common image regions that allow for cross-camera feature matching.

We propose a simple and effective method to continuously track features across different cameras which we call Cross Camera Feature Tracking (CCFT). CCFT can avoid the optimization of redundant landmarks by tracking the same feature in different cameras simultaneously. As shown in Fig. 3, we continually add constraints between states $\boldsymbol{x}_i$ and landmarks $\boldsymbol{m}_\ell$ even as they are tracked across cameras. For example, landmark $\boldsymbol{m}_1$ is first tracked in a stereo camera at $\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2$, then tracked in a monocular camera at $\boldsymbol{x}_3$.

A common approach to perform this feature matching would be via feature descriptors. However, this would be computationally expensive for real-time, high-frequency VIO on a mobile platform. For example, the combination of feature extraction and stereo matching in ORB-SLAM2 [21] takes ~24.8 ms which is significant for a 30 Hz image stream. Instead, we make use of the known camera extrinsics and the preintegrated IMU measurements between states to associate features geometrically.

When feature depth is available at the current image time t_k (e.g., from stereo camera triangulation), then we can directly project the visual landmark $\boldsymbol{m}_\ell$ from camera $\mathtt{C}_i$ into camera $\mathtt{C}_j$ via the extrinsic transformation between cameras $\mathbf{T}_{\mathtt{C}_i\mathtt{C}_j}$,

$$(u, v)_{\ell,c_j,t_k} = \pi_{c_j}\left(\mathbf{T}_{\mathtt{C}_j\mathtt{C}_i}\, {}_{\mathtt{C}_i}\boldsymbol{m}_{\ell,t_k}\right) \tag{6}$$

Alternatively, if the feature depth is not directly available at the current timestamp (e.g., if the feature is only tracked in a monocular camera), then we instead use the preintegrated IMU measurements to estimate the landmark location at the current time,

$$(u, v)_{\ell,c_j,t_k} = \pi_{c_j}\left(\mathbf{T}_{\mathtt{C}_j\mathtt{C}_i}\, \hat{\mathbf{T}}_{t_k,t_{k-1}}\, {}_{\mathtt{C}_i}\boldsymbol{m}_{\ell,t_{k-1}}\right) \tag{7}$$

where $\hat{\mathbf{T}}_{t_k,t_{k-1}}$ is the estimated transform between the previous camera pose at t_{k-1} and the current time t_k.

This process is completed for each feature in the overlapping image regions. The projected feature location is refined by matching it to the closest image feature in camera c_j, using a Euclidean distance metric. This is a highly effective method in practice, assuming there is a good extrinsic calibration. Any incorrect associations are handled by the optimizer using robust cost functions.

Fig. 4 shows an example of the feature matches using CCFT. The purple areas highlight the overlapping image regions where features can be tracked across cameras. The CCFT method typically reduces the number of landmarks added into the optimization back-end and gives improved estimation accuracy. This is discussed further in Section IX-A.

VI. Submatrix Feature Selection

In general, increasing the number of features tracked in a VIO system improves the estimation accuracy [15]. However, it also increases computation, eventually reaching a point where the algorithm can fail due to computational constraints. This is a particular problem in multi-camera systems, where more features can be tracked than the optimizer can handle. Thus, it is important to track and optimize only the best features.

Specifically, our SFS algorithm is based on [22], where the authors presented a feature selection algorithm to reduce the computational cost of active map-to-frame feature matching. We instead apply this algorithm to MC-VIO and implement changes to improve numerical stability. The aim is to maintain accuracy while reducing the optimization time of MC-VIO.

In this section, we first outline the construction of the joint feature Jacobian and covariance matrices. Then, we describe the *Max-logDet* algorithm to select the most representative subset of tracked features to be optimized.

A. Construct Joint Feature Jacobian Matrix

The optimization aims to minimize the cost function defined in Eq. (3). Since we are interested in selecting the best features in each frame k, we focus on improving the conditioning of the landmark residual, where $\boldsymbol{z}_\ell$ is the feature location of (u, v),

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \sum_{\ell \in \mathsf{M}} \|\mathbf{r}_{\boldsymbol{x},\boldsymbol{m}_\ell}\|^2_{\Sigma_{\boldsymbol{m}_\ell}} \tag{8}$$

$$= \arg\min_{\mathcal{X}} \sum_{\ell \in \mathsf{M}} \|\pi(\mathbf{T}, \boldsymbol{m}_\ell) - \boldsymbol{z}_\ell\|^2_{\Sigma_{\boldsymbol{m}_\ell}} \tag{9}$$

Stacking $\boldsymbol{m}_\ell$, $\boldsymbol{z}_\ell$ into column matrices $\mathbf{M}$, $\mathbf{Z}$ we get,

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \|\Pi(\mathbf{T}, \mathbf{M}) - \mathbf{Z}\|^2_\Sigma \tag{10}$$

Applying the method from [22], we perform a first order linearization around the initial value $\mathbf{T}_g$,

$$\|\Pi(\mathbf{T}, \mathbf{M}) - \mathbf{Z}\|^2_\Sigma \approx \|\Pi(\mathbf{T}_g, \mathbf{M}) + \mathbf{H}_\mathbf{T}(\mathbf{T} \ominus \mathbf{T}_g) - \mathbf{Z}\|^2_\Sigma \tag{11}$$

where $\mathbf{H}_\mathbf{T}$ is the pose Jacobian linearized about the value $\mathbf{T}_g$, and $\ominus$ is the Lie group difference operator between poses [8].

To minimize Eq. (11), we can use Gauss-Newton optimization to iteratively update the pose estimate,

$$\mathbf{T}_{g+1} = \mathbf{T}_g - \mathbf{H}^+_\mathbf{T} \left(\Pi(\mathbf{T}_g, \mathbf{M}) - \mathbf{Z}\right) \tag{12}$$

where $\mathbf{H}^+_\mathbf{T}$ is the Moore-Penrose pseudo-inverse of $\mathbf{H}_\mathbf{T}$. This method has converged (i.e., reached minimal error) when the update step is approximately zero, i.e.,

$$\boldsymbol{\epsilon}_\mathbf{T} = \mathbf{H}^+_\mathbf{T} \left(\Pi(\mathbf{T}_g, \mathbf{M}) - \mathbf{Z}\right) \approx 0 \tag{13}$$

Performing another first order linearization around $\mathbf{M}_g$,

$$\boldsymbol{\epsilon}_\mathbf{T} \approx \mathbf{H}^+_\mathbf{T} \left(\Pi(\mathbf{T}_g, \mathbf{M}_g) - \mathbf{Z} + \mathbf{H}_\mathbf{M}(\mathbf{M} - \mathbf{M}_g)\right) \tag{14}$$

$$\boldsymbol{\epsilon}_\mathbf{T} = \mathbf{H}^+_\mathbf{T} \left(\boldsymbol{\epsilon}_\boldsymbol{Z} - \mathbf{H}_\mathbf{M}\boldsymbol{\epsilon}_\mathbf{M}\right) \tag{15}$$

where $\mathbf{H}_\mathbf{M}$ is the landmark-to-image projection Jacobian. We define the measurement error as $\boldsymbol{\epsilon}_\boldsymbol{Z} = \Pi(\mathbf{T}_g, \mathbf{M}_g) - \mathbf{Z}$, and the landmark error as $\boldsymbol{\epsilon}_\mathbf{M} = \mathbf{M} - \mathbf{M}_g$.

Without loss of generality, the measurement error can be modeled as $\boldsymbol{\epsilon}_{\boldsymbol{z}_\ell} \sim \mathcal{N}(0, \Sigma_{z_\ell})$ which is influenced by image processing parameters and noise. However, if biased landmark errors exist, then the mean of error distribution is non-zero. This could be caused by batch perturbation of the landmark positions by incorrect camera poses, or a slight scale error. Hence the landmark error is modeled as non-zero-mean i.i.d. Gaussian, $\boldsymbol{\epsilon}_{\boldsymbol{m}_\ell} \sim \mathcal{N}(\mu_{\boldsymbol{m}_\ell}, \Sigma_{\boldsymbol{m}_\ell})$, such that the expected value of $\boldsymbol{\epsilon}_\mathbf{T}$ is,

$$\mathbb{E}[\boldsymbol{\epsilon}_\mathbf{T}] = \mathbb{E}\left[\mathbf{H}^+_\mathbf{T} \left(\boldsymbol{\epsilon}_\boldsymbol{Z} - \mathbf{H}_\mathbf{M}\boldsymbol{\epsilon}_\mathbf{M}\right)\right] \tag{16}$$

$$\mathbb{E}[\boldsymbol{\epsilon}_\mathbf{T}] = \mathbf{H}^+_\mathbf{T}\mathbf{H}_\mathbf{M}\mathbf{1}_n\mu_\mathbf{M} \tag{17}$$

$$\mathbf{H}^+_\mathbf{M}\mathbf{H}_\mathbf{T}\mathbb{E}[\boldsymbol{\epsilon}_\mathbf{T}] = \mathbf{1}_n\mu_\mathbf{M} \tag{18}$$

where $\mathbf{1}_n$ is a tall matrix made of identity matrices to transform the dimension of $\mu_\mathbf{M}$. $\mathbf{H}_\mathbf{M}$ consists of n diagonal blocks made up of individual landmark Jacobians $\mathbf{H}_{\boldsymbol{m}_\ell} \in \mathbb{R}^{2\times 3}$, where n is the number of landmarks with known 3D positions. $\mathbf{H}_\mathbf{T}$ consists of pose Jacobians for each landmark $\mathbf{H}_{\mathbf{T},\boldsymbol{m}_\ell} \in \mathbb{R}^{2\times 6}$.

To obtain a joint Jacobian matrix $\mathbf{H}_J$, we add an additional row of zeros to $\mathbf{H}_{\mathbf{T},\boldsymbol{m}_\ell}$ and a row of $[0, 0, 1]$ to $\mathbf{H}_{\boldsymbol{m}_\ell}$. This makes $\mathbf{H}_{\mathbf{T},\boldsymbol{m}_\ell}$ invertible without changing the structure of the least squares problem [23]. Performing block-wise multiplication results in the matrix $\mathbf{H}_J$,

$$\mathbf{H}_J = \begin{bmatrix}\mathbf{H}^{-1}_{\boldsymbol{m}_0}\mathbf{H}_{\mathbf{T},\boldsymbol{m}_0} & \dots & \mathbf{H}^{-1}_{\boldsymbol{m}_{n-1}}\mathbf{H}_{\mathbf{T},\boldsymbol{m}_{n-1}}\end{bmatrix}^\mathsf{T} \tag{19}$$

from which the simplified pose covariance matrix follows,

$$\Sigma_\mathbf{T} = \mathbf{H}^+_J(\mathbf{H}^+_J)^\mathsf{T} = (\mathbf{H}^\mathsf{T}_J\mathbf{H}_J)^{-1} \tag{20}$$

B. Submatrix Selection

Since the overall pose error depends on the spectral properties of $\mathbf{H}_J$, we aim to find a submatrix $\mathbf{H}_{sub}$ (i.e., subset of features) that best preserves this distribution. This can be understood as finding row blocks in $\mathbf{H}_J$ to maximize the norm of the selected submatrix.

We adapt the stochastic sampling method presented in [22] for feature selection. We choose the *Max-logdet* metric since it best approximates the original full feature set in visual odometry [23], [24]. Additionally, the combination of greedy (deterministic) and stochastic sampling (randomized acceleration) gives the best approximation ratio of any polynomial time algorithm [25].

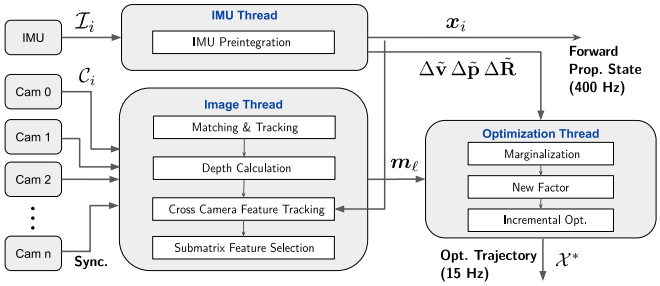


Fig. 5. Overview of the VILENS-MC architecture. The IMU and cameras are handled in separate threads by front-end measurement handlers. The back-end produces both a high frequency forward-propagated output and a lower frequency optimized output.

We note that since $\mathbf{H}_{J_i} \in \mathbb{R}^{2\times 6}$ is not a full rank matrix, the determinant of its square $\mathbf{H}_{J_i}^{\mathsf{T}}\mathbf{H}_{J_i}$ would be zero. In contrast to [22], we add a small diagonal matrix $\lambda\mathbf{I} \in \mathbb{R}^{6\times 6}$ to the $\log\det$ calculation,

$$\mathbf{H}_{J_i} = \arg\max \left(\log\det(\mathbf{H}_{J_i}^{\mathsf{T}}\mathbf{H}_{J_i} + \mathbf{H}_{sub}^{\mathsf{T}}\mathbf{H}_{sub} + \lambda\mathbf{I})\right) \quad (21)$$

This makes the squared matrix full rank with a non-zero determinant while preserving its spectral properties. An alternative method would be projecting $\mathbf{H}_{J_i}^{\mathsf{T}}\mathbf{H}_{J_i}$ onto a full rank subspace, which we will consider in future work.

We apply this algorithm to each camera individually in our multi-camera setup. An ablation study showing how this method maintains high accuracy and improves optimization time is presented in Section IX-B.

VII. Implementation

An overview of the system architecture is shown in Fig. 5. Three parallel threads perform IMU preintegration, camera image processing, and optimization, respectively. The system outputs the optimized state estimate from the factor graph at camera keyframe frequency (typically 15 Hz) for navigation or mapping, while a forward-propagated state at IMU frequency is also produced at 400 Hz for high frequency tasks.

The factor graph is solved using fixed lag smoothing based on the efficient incremental optimization solver iSAM2 from the GTSAM library [19]. For all of our experiments, we used a lag time of 3.5 s. To reduce the effect of outliers, the visual factors are added to the graph using the Dynamic Covariance Scaling (DCS) robust cost function [26].

A. Hardware and Calibration

The sensor used for our experiments is the Alphasense multi-camera development kit from Sevensense Robotics AG, shown in Fig. 2. An onboard FPGA synchronizes the IMU and four grayscale fisheye cameras – a frontal stereo pair with an 11 cm baseline and two lateral cameras. Each camera has a FoV of $126^\circ \times 92.4^\circ$ and a resolution of 720×540 px. This configuration produced an overlapping FoV between the front and side cameras of about 36°. The cameras and the embedded cellphone-grade IMU were operated at 30 Hz and 200 Hz, respectively. Camera-IMU extrinsic and intrinsic calibration was conducted offline using the Kalibr [27] toolbox. The Ouster OS0-128 lidar shown in the figure was used only for ground truth generation and all sensors were hardware synchronized with an Intel NUC via Precision Time Protocol (PTP).

Fig. 6. Test datasets: *Top:* Newer College Dataset [28]. *Bottom Left:* Oxford Mathematical Institute. *Bottom Right:* A dark underground limestone mine.

B. Initialization

We initialize the IMU biases by averaging the first 1 s of data at system start up (assuming the IMU is stationary).

To solve the scale initialization problem, which is often present in monocular visual odometry systems, we combine preintegrated IMU measurements and depth from the stereo camera pair. Notably, the CCFT method allows features from the stereo camera to flow into the monocular camera, speeding up the depth initialization process.

C. Visual Feature Tracking

To get an even feature distribution across each image, we split them into 3×3 tiles before applying the FAST feature detector to each segment. These features are tracked between frames using the KLT feature tracker and across cameras using the CCFT method described above. Outliers are rejected using RANSAC-based methods. Thanks to the incremental optimization and multi-threading, every second frame is used as a keyframe, achieving a 15 Hz nominal output.

VIII. Experimental Results

We evaluated our algorithm on a variety of challenging indoor and outdoor datasets, varying from narrow corridors to large open spaces. We compared our algorithm to state-of-the-art methods, evaluating both quantitative and qualitative performance. We also include ablation studies showing how our proposed contributions reduced Relative Pose Error (RPE) and decreased computation time. Finally, we demonstrated the versatility of this approach by applying VILENS-MC to a quadruped robot operating in an unlit mine.

A. Datasets

Fig. 6 gives an overview of the datasets used for evaluation. These include additional multi-camera experiments collected for the Newer College Dataset (NCD) [28], operation in an

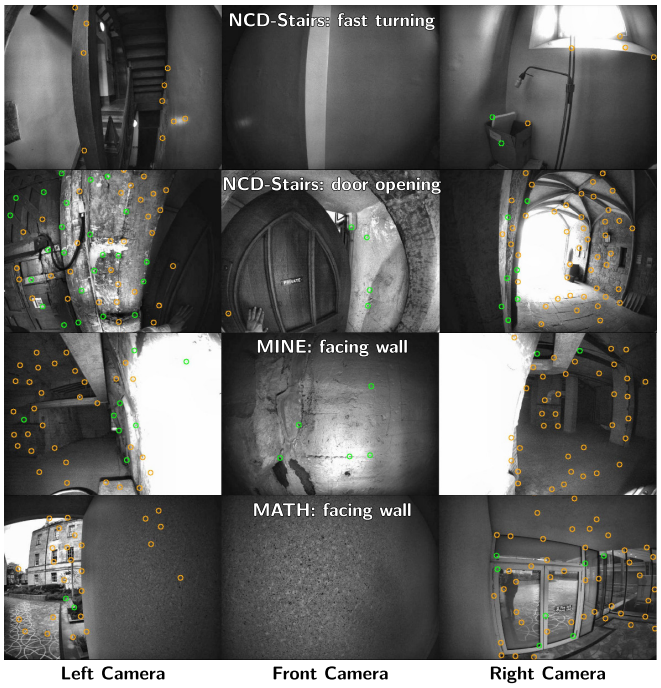


Fig. 7. Examples of challenging scenes where it was difficult to track features. Orange and green circles are features tracked by monocular and stereo cameras, respectively (some have been tracked across images). Note that stereo features were mostly absent or rejected in these examples.

unlit underground limestone mine (MINE), and a large circuit around the Oxford Mathematical Institute (MATH).

These datasets were collected using the handheld multi-camera device described in Section VII-A. They were specifically chosen to test the limits of VIO systems and include challenges such as aggressive shaking of the device (up to 5.5 rad/s), severe illumination changes, and loss of visual feature tracking in one or more cameras (see Fig. 7). All datasets were collected at a fast walking pace of ∼1.5 m/s. The specific experiments are:

- **NCD-Quad:** Moving between narrow corridors and a large open college quadrangle, with strong illumination changes caused by bright sunlight (244 m, 3 min).
- **NCD-Stairs:** Traversing a very narrow staircase, including a sequence where a door is opened directly in front of the camera (59 m, 2 min).
- **MINE:** A medium scale, dark underground environment with regular illumination changes due to onboard lighting (236 m, 3 min).
- **MATH:** A large scale outdoor environment with aggressive shaking of the device up to 5.5 rad/s (329 m, 4 min).

The multi-camera experiments which extend NCD (NCD-Quad, NCD-Stairs) have been publicly released. [1]

To generate the ground truth, Iterative Closest Point (ICP) registration was used to align the current lidar scan to detailed prior maps collected from a commercial laser scanner. The high frequency motion estimate from the IMU was used to carefully remove lidar motion distortion [7]. For an in-depth discussion on ground truth generation, refer to [29].

[1]Available at https://ori-drs.github.io/newer-college-dataset/.

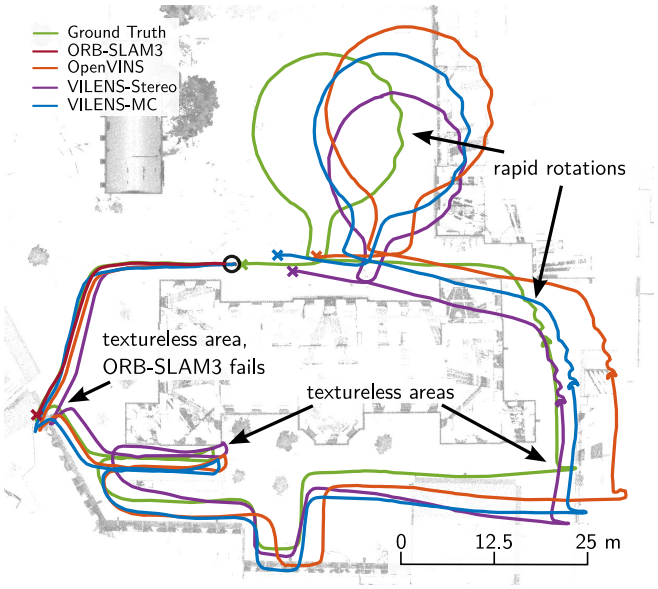


Fig. 8. Top-down view of **MATH** dataset comparing the estimated trajectory of ORB-SLAM3, OpenVINS, and VILENS-MC against the ground truth. A black circle marks the start of the trajectory, while colored crosses indicate the last pose of each trajectory. Textureless locations, where cameras were facing walls, are shown in Fig. 7, row 4.

10 m Relative Pose Error (RPE) – Translation [m] / Rotation [°]				
	OpenVINS	**ORB-SLAM3**	**VILENS-Stereo**	**VILENS-MC**
Cameras	2 (Stereo)	2 (Stereo)	2 (Stereo)	4
NCD-QUAD	1.01 / 1.26	**0.23** / 0.91	0.30 / 1.24	0.31 / **0.82**
NCD-STAIRS	0.33 / 3.05	0.20* / 3.04	0.43 / 5.85	**0.16** / **2.56**
MINE	0.98* / 2.10	Fail	0.41* / 3.57	**0.20** / **1.65**
MATH	0.65* / 1.69	Fail	0.56* / 1.57	**0.26** / **1.03**

TABLE I
PERFORMANCE COMPARISON BETWEEN DIFFERENT ALGORITHMS
* = ONE OR MORE INSTABILITIES OCCURRED

B. Quantitative Analysis

Table I shows the RPE over 10 m for OpenVINS [12], ORB-SLAM3 [10] and VILENS [8] (which our system builds upon). We refer to the configuration of VILENS which uses only stereo and IMU input as *VILENS-Stereo*. Note we were not aware of any open-source MC-VIO algorithm for comparison.

Compared to our baseline (VILENS-Stereo), we reduced mean RPE by 45 % / 50 % in translation/rotation, while tracking fewer features due to SFS. VILENS-MC also outperformed other state-of-the-art stereo VIO algorithms by up to 80 % / 39 % in translation/rotation.

In benign VIO conditions, such as those found in the NCD-Quad dataset, the performance of VILENS-MC was similar to state-of-the-art systems. In particular, ORB-SLAM3 appeared to outperform the other algorithms by taking advantage of its local map matching SLAM system to correct for failure events when the camera revisited a previous location.

However, in the MINE and MATH datasets, where there were few or no stereo features, VILENS-MC performed well, even when the other stereo VIO systems failed or had significant drift. This highlights a key benefit of the tight fusion of multiple cameras, the natural robustness to scene degradation.

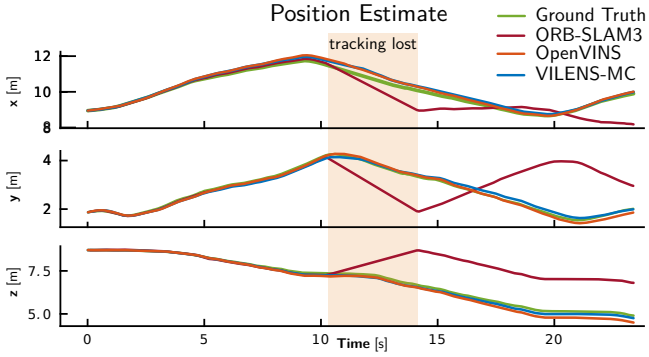


Fig. 9. ORB-SLAM3 lost tracking and failed to estimate odometry for 4 s in NCD-Stairs (scenario shown in Fig. 7 Row 3). Note that elevation (z axis) is properly tracked afterwards.

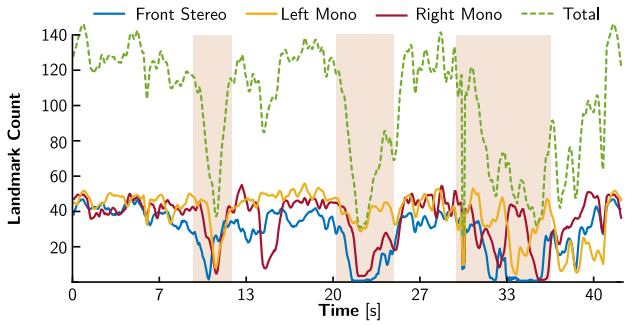


Fig. 10. A 40 s section from the **MINE** dataset showing landmark count per each camera: front stereo pair (solid blue), left mono camera (solid yellow), right mono camera (solid red). The total number of landmarks is dashed green.

C. Degeneracy and Failure Analysis

1) MATH: A top-down view of the trajectories in the MATH dataset is shown in Fig. 8. In challenging scenarios where stereo camera feature tracking failed (marked as "textureless areas"), stereo VIO algorithms continued to operate but relied on the IMU only, causing the RPE to quickly increase, while VILENS-MC maintained robust estimation by utilizing multiple cameras looking in different directions.

2) NCD-Stairs: Fig. 9 shows an extract from the NCD-Stair dataset. Note how ORB-SLAM3 suffered from significant drift when no features were detected in the stereo camera for 4 s until it re-initialized (orange area). Even though the average RPE over the entire NCD-Stair sequence remained low, a gap in estimations would be undesirable for an autonomous robot.

3) MINE: Fig. 10 shows the number of tracked landmarks during a 40 s interval in the MINE dataset. During this time, several instances occurred where feature tracking failed in either one or two cameras. Failures occurred when the device entered a dark room (∼10 s), when it approached a wall and turned on the spot (∼20 s, shown in Fig. 7 row 3), and when it entered a dark corner where only one camera could observe features at a time (∼30 s). By tracking features across multiple cameras, VILENS-MC maintains accurate estimation throughout these events.

	Relative Pose Error – Translation [m] / Rotation [°]				
	Baseline	**CCFT**	**Baseline** (90 feat.)	**Baseline** (150 feat.)	**SFS** (90 feat.)
NCD-QUAD	**0.27** / **0.77**	0.34 / 0.88	0.37 / 1.02	0.34 / 0.88	**0.31** / **0.82**
NCD-STAIR	0.17 / 3.60	**0.16** / **2.85**	0.28 / 5.06	**0.16** / **2.85**	0.18 / 3.03
MINE	0.25 / 1.91	**0.21** / **1.64**	0.34 / 2.70	0.21 / **1.71**	**0.20** / 1.75
MATH	0.42 / 1.01	**0.28** / **1.01**	0.33 / 1.16	0.28 / **1.01**	**0.26** / 1.03

TABLE II
ABLATION STUDIES: CROSS CAMERA FEATURE TRACKING (CCFT) AND SUBMATRIX FEATURE SELECTION (SFS)

IX. ABLATION STUDY

A. Cross Camera Feature Tracking

In this section, we evaluate the benefit of the CCFT method (Section V). CCFT enables continuous feature tracking over longer periods and adds additional constraints to landmarks.

Table II summarizes the effect this method had on mean RPE (over 10 m), showing that on average CCFT decreased both translational and rotational RPE on all datasets except for translation in NCD-QUAD. We attribute this to the system's susceptibility to extrinsic multi-camera calibration errors.

Additionally, CCFT can reduce the number of landmarks in the optimization by ∼10 %. Since our hardware setup (ref. Section VII-A) had only a small overlapping FoV between frontal and lateral cameras, we expect to see a greater benefit with a larger FoV overlap.

B. Submatrix Feature Selection

This ablation study demonstrates the benefit of SFS. The baseline approach simply spreads features evenly across the image based on their FAST score and mutual pixel distance.

A comparison between the baseline and proposed SFS methods is shown in Table II. The first two columns show the baseline with 90 and 150 features tracked and optimized across all cameras, while the SFS method selected only 90 out of 150 tracked features to optimize.

By selecting only the most representative features, SFS outperformed the baseline (90 features) by up to 36 %, even though both methods optimized the same number of features.

Crucially, SFS lowers computation without sacrificing accuracy. Comparing the baseline (150 features) to SFS (90 features), the latter achieves similar accuracy with a 20 % reduction in optimization time (53.7 ms to 43.0 ms). The standard deviation of the computation was also reduced by 24.3 %. This demonstrates that SFS can achieve similar accuracy with lower and more consistent computational requirements.

X. DEMONSTRATION ON LEGGED ROBOTS

To demonstrate the versatility of our approach, we tested VILENS-MC on the ANYmal C quadruped robot equipped with an Alphasense for the DARPA Subterranean Challenge [2]. In the trajectory shown in Fig. 11 the robot autonomously explored the same environment as the MINE dataset. VILENS-MC achieved almost the same accuracy as for the handheld dataset (0.27 m, 2.65° RPE), indicating that performance was similar across the different platforms.

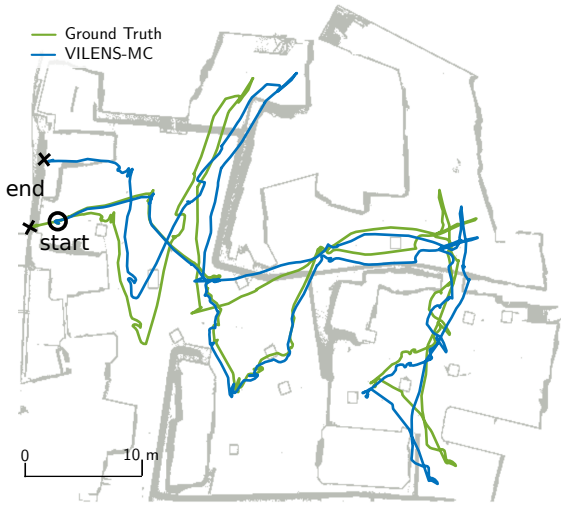


Fig. 11. State estimate on an ANYmal C quadruped robot autonomously exploring an underground mine over ∼10 min. Fig. 6 shows the robot's on-board lighting which caused strong illumination changes in the images.

XI. Conclusion

We presented a novel factor graph formulation for state estimation that tightly fuses an arbitrary number of cameras. The joint optimization of IMU measurements with monocular and camera constraints enables graceful handling of degenerate scenarios without requiring hard switching between cameras. This also simplifies the initialization of the individual monocular cameras. We have demonstrated comparable tracking performance to state-of-the-art VIO systems in benign conditions and better performance in extreme situations, such as in the case of aggressive motions, loss of tracking in one or more cameras, or abrupt light changes.

We proposed two algorithmic components specific to multi-camera odometry: cross camera feature tracking and submatrix feature selection. Ablation studies showed that these components improve accuracy and reduce optimization time.

Overall, we have presented a robust MC-VIO system that is capable of handling various challenging real-world environments while consistently achieving accurate state estimation. Future work would consider including camera extrinsics into the optimization.

References

[1] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-Manifold Preintegration for Real-Time Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[2] M. Tranzatto, *et al.*, "CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge," *Field Robotics*, 2021.

[3] M. Hutter, *et al.*, "ANYmal – A Highly Mobile and Dynamic Quadrupedal Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 38–44.

[4] J. Jaekel, J. G. Mangelson, S. Scherer, and M. Kaess, "A Robust Multi-Stereo Visual-Inertial Odometry Pipeline," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[5] H. Seok and J. Lim, "ROVINS: Robust Omnidirectional Visual Inertial Navigation System," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6225–6232, 2020.

[6] K. Eckenhoff, P. Geneva, and G. Huang, "MIMC-VINS: A Versatile and Resilient Multi-IMU Multi-Camera Visual-Inertial Navigation System," *IEEE Transactions on Robotics*, pp. 1–20, 2021.

[7] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1004–1011, 2021.

[8] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *arXiv preprint arXiv:2107.07243*, 2021.

[9] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[10] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, pp. 1–17, 2021.

[11] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[12] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 4666–4672.

[13] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 2502–2509.

[14] M. He, C. Zhu, Q. Huang, B. Ren, and J. Liu, "A review of monocular visual odometry," *The Visual Computer*, vol. 36, no. 5, pp. 1053–1065, 2020.

[15] "Towards Robust Visual Odometry with a Multi-Camera System," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018.

[16] M. G. Müller, *et al.*, "Robust visual-inertial state estimation with multiple odometries and efficient mapping on an MAV with ultra-wide FOV stereo vision," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3701–3708, 2018.

[17] J. Kuo, M. Muglikar, Z. Zhang, and D. Scaramuzza, "Redesigning SLAM for Arbitrary Multi-Camera Systems," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 2116–2122.

[18] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[19] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics*, vol. 6, pp. 1–139, 2017.

[20] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[21] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[22] Y. Zhao and P. A. Vela, "Good Feature Matching: Toward Accurate, Robust VO/VSLAM with Low Latency," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 657–675, 2020.

[23] ——, "Good Feature Selection for Least Squares Pose Optimization in VO/VSLAM," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 1183–1189.

[24] L. Carlone and S. Karaman, "Attention and Anticipation in Fast Visual-Inertial Navigation," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2019.

[25] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy," *Proceedings of the National Conference on Artificial Intelligence*, vol. 3, pp. 1812–1818, 2015.

[26] K. MacTavish and T. D. Barfoot, "At All Costs: A Comparison of Robust Cost Functions for Camera Correspondence Outliers," in *Conference on Computer and Robot Vision*, 2015, pp. 62–69.

[27] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 4304–4311.

[28] L. Zhang, M. Camurri, and M. Fallon, "Multi-Camera LiDAR Inertial Extension to the Newer College Dataset," *arXiv preprint arXiv:2112.08854*, 2021.

[29] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry
Publication Status	x Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Lintong Zhang, David Wisth, Marco Camurri, and Maurice Fallon, ``Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry," in IEEE Robotics and Automation Letters, 2022.

Student Confirmation

Student Name:	Lintong Zhang		
Contribution to the Paper	• Contributed to development of the idea • Implemented novel contributions mentioned in the paper • Executed closed-loop field experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	April 16, 2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Maurice Fallon		
Supervisor comments Lintong was the lead instigator, implementer and researcher on this project and wrote the bulk of the paper		
Signature 	Date	May 3, 2023

This completed form should be included in the thesis, at the end of the relevant chapter.

4.2 Discussion

Since our initial exploration of the MC-VIO system, subsequent research has expanded upon and refined our methodology. He *et al.* [57] introduced a VIO algorithm that manages multiple monocular cameras without overlapping FoV. Kaveti *et al.* [69] adopted a generic camera model to represent any number of cameras as a single imaging device, utilising cross-camera matched features, similar to our approach, to reduce the total number of landmarks. Additionally, Wang *et al.* [139] employed the same multi-camera hardware as our system and extended the VIO system to a complete SLAM system. Their approach improved the pre-integration IMU formulation and notably won the Hilti Slam Challenge 2023. These developments show the growing interest in multi-camera VIO systems and its effectiveness in addressing common failure cases in challenging environments.

A VIO system often acts as the front-end module within the SLAM framework, so it is inevitable that the VIO system will accumulate drift over time. To mitigate this issue, it is crucial to integrate a visual loop closure module within the SLAM system to reduce overall drift. This concept has been successfully implemented in our follow up work [67], led by my colleague Christina Kassab. With MC-Vilens as the odometry front end, the visual SLAM system, named LEXIS, can effectively close the loops and minimise drift. We utilised CLIP, a vision language model, and demonstrated that it could effectively close loops and semantically label different spaces, enhancing the robustness and semantic understanding of the SLAM system.

During calibration using Kalibr [43], the two side-facing monocular cameras and the front-facing stereo cameras are individually calibrated relative to the IMU. However, we observe that inter-camera calibration is prone to errors and variability over long periods, even when mounted on a rigid platform. One straightforward solution is to incorporate intrinsic and extrinsic calibration as factors with small covariance in the factor graph optimisation. Additionally, due to the small overlap in the FoV between the front and side-facing cameras, we could leverage the lidar mounted on top of the device to perform joint camera-lidar calibration offline using DiffCal, developed by my colleague Frank Fu *et al.* [42]. This step would help

verify the extrinsic obtained from Kalibr, potentially improving cross-camera feature tracking accuracy and, consequently, enhancing pose estimation accuracy.

For potential future work, it would be beneficial to provide pose covariance in the MC-VIO system. We can directly obtain the marginal posterior density matrix from the GTSAM backend optimiser. Although it is a costly operation in terms of timing to extract covariances at a high frequency, we can extract the covariances at a lower frequency and re-estimate at a higher frequency based on other system metrics, such as through the number of landmarks in each camera and landmark pixel movements between keyframes.

However, compared to vision-based systems, lidar is known to provide superior accuracy in both front-end odometry and as a complete SLAM system [152]. In the following chapters, we first investigate lidar-only solutions, addressing common challenges encountered in indoor environments. Then we explore hybrid localisation methods that integrate both vision and lidar, leveraging the strengths of each sensor to enhance the overall system performance.

5
Learned Lidar Localisation

Place recognition is a key component in SLAM systems, used to establish loop closures that help correct the drift accumulated in the odometry system. Currently, the majority of lidar localisation research focuses on outdoor environments, particularly in autonomous driving contexts. However, there is a significant gap in research targeting the unique challenges of indoor environments. Indoor spaces are typically narrower and more constrained, filled with a variety of differently sized objects, which lead to occlusions caused by clutter.

This chapter introduces InstaLoc, a novel solution for localisation designed for indoor environments. InstaLoc operates by localising individual lidar scans within a pre-existing map. InstaLoc draws inspiration from human navigational skills, where positioning is often based on the recognition of distinctive layouts and objects. It identifies and matches specific object instances in a scanned scene with those in a prior map. This approach is further enhanced by the use of panoptic segmentation to facilitate direct inference on 3D lidar scans, demonstrating an innovative method for indoor localisation.

5.1 InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning

The article [153] summarising this project was presented in *Robotics: Science and Systems (RSS) 2023*. An accompanying video is available online at: https://youtu.be/2votkTLJFFU.

Lintong Zhang, Tejaswi Digumarti, Georgi Tinchev, and Maurice Fallon, "InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning" in Robotics: Science and Systems (RSS), July, 2023.

Robotics: Science and Systems 2023
Daegu, Republic of Korea, July 10-July 14, 2023

InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning

Lintong Zhang[1], Tejaswi Digumarti[1], Georgi Tinchev[2], and Maurice Fallon[1]
[1] University of Oxford [2] Amazon Research

***Abstract*—Localization for autonomous robots in prior maps is crucial for their functionality. This paper offers a solution to this problem for indoor environments called InstaLoc, which operates on an individual lidar scan to localize it within a prior map. We draw on inspiration from how humans navigate and position themselves by recognizing the layout of distinctive objects and structures. Mimicking the human approach, InstaLoc identifies and matches object instances in the scene with those from a prior map. As far as we know, this is the first method to use panoptic segmentation directly inferring on 3D lidar scans for indoor localization. InstaLoc operates through two networks based on spatially sparse tensors to directly infer dense 3D lidar point clouds. The first network is a panoptic segmentation network that produces object instances and their semantic classes. The second smaller network produces a descriptor for each object instance. A consensus based matching algorithm then matches the instances to the prior map and estimates a six degrees of freedom (DoF) pose for the input cloud in the prior map. InstaLoc utilizes two efficient networks, requires only one to two hours of training on a mobile GPU, and runs in real-time at 1 Hz. Our method achieves between two and four times more detections when localizing, as compared to baseline methods, and achieves higher precision on these detections.**

I. Introduction

Localization is a fundamental capability needed for mobile robots to navigate their environment and make decisions. There have been many studies on vision, lidar, and radar-based localization. The parent problem of Simultaneous Localisation and Mapping (SLAM) concerns a robot determining its pose while building a map of its environment concurrently. Localization, or place recognition, can contribute to SLAM by helping to *close loops*, or to determine the robot's position in a fixed prior map - the *kidnapped robot* problem.

Many popular localization methods using visual and lidar sensors have been proposed. Among visual-based approaches, visual teach-and-repeat [16, 17] is one of the most popular methods, where a robot first constructs a visual prior map and then localizes on its repeat phase. Compared to image-based camera solutions, modern 3D lidar sensors are view-invariant, robust to lighting changes, and can operate when the path traveled is offset from the original path. Given that lidar is a precise and long-range sensor, lidar localization has been heavily researched in outdoor environments, especially in the context of autonomous driving [21, 14, 27]. However, there are fewer approaches for indoor environments because these environments contain more complex structures and clutter, hence fewer clear separations between objects in lidar scans. In an indoor environment, there are many different classes of objects, with one dataset proposing 13 semantic classes

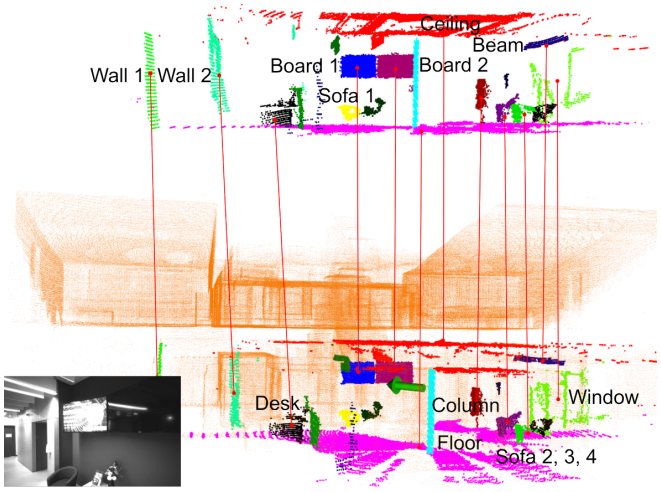


Fig. 1: An illustration of the *InstaLoc* method, where a 'live' lidar scan (top) is localized inside the orange colored prior map (bottom) by matching semantically segmented objects (red lines). The green arrow shows the estimated position and the left corner image is the corresponding camera view.

[2]. The indoor scene varies greatly: from bare box-shaped rooms with four walls to narrow corridors with two long walls. Room surfaces are often covered with objects such as electronics, hanging art, ceiling lights, bookcases, and various decorative objects. Localization algorithms cannot rely on flat ground assumptions as there is often an incomplete view of the floor. In addition, there are changes in levels, with steps and staircases. Nevertheless, it is important to localize in these indoor scenarios to enable robots to operate robustly in complex office buildings, construction sites, warehouses, and other commercial environments.

In this paper, we draw inspiration from how humans perceive the world and reach the "I know where I am" moment. By memorizing and recognizing the distinctive structures and unique objects inside a space, humans can spatially locate themselves in the environment. Based on the same principle, InstaLoc makes use of individual object instances to localize. Different from existing approaches that rely on primitive shapes or other handcrafted features, InstaLoc learns to segment and match individual objects to the prior scene. These scenes include both fixed objects (walls, ceilings, beams) and movable objects (chairs, desks), in order to tolerate active dynamics and longer term scene changes. To train the segmentation network with accurate class labels, we leverage a

simulator to synthesize and automatically annotate every point — thus avoiding onerous point cloud labeling. To overcome the challenge of imperfect instance segmentation, we designed a sparse convolutional descriptor network that infers many instances simultaneously and tolerates mild changes in the instance point cloud.

To summarize, our contributions are:

- A novel learning-based lidar localization approach for indoor environments that can process dense lidar scans on a mobile GPU in real time.
- An improved panoptic segmentation network that works with single lidar scans.
- A fast and efficient descriptor network to learn object instances with a variable number of input points.
- State-of-the-art performance on indoor localization compared to other segment-based methods, achieving two to four times more detection.

II. Related Work

In this section, we describe recent work on segment-based lidar localization and its applications to urban, natural, and industrial environments. We review approaches that use semantic segmentation for outdoor localization. Finally, we discuss methods that rely on the geometry of the scene and algorithms which can localize in maps made with other sensor modalities (co-localization).

A. Outdoor Segment-based localization

Scan segment matching was first introduced by Douillard et al. [6], where segments were considered as a midway point between local and global approaches for describing a scene. The approach was initially applied to lidar localization by Dubé et al. [7] where segments were extracted directly from raw point clouds and described with a descriptor based on the geometry of the segment (such as its eigenvalues and proportions). Later, Dubé et al. [8], Tinchev et al. [20] described segments using a neural network to provide a richer and more meaningful descriptions. Building on this research, Ratz et al. [18] showed that lidar segments fused with visual data further improve the performance of global localization algorithms. Cramariuc et al. [5] fused both colour and semantic information from images to create an enriched point cloud that was later segmented and used for localization. We use this segment-direct concept as the basis of InstaLoc, however in contrast to these approaches, InstaLoc does not use engineered segmentation methods, nor images, to extract the semantic information but directly learns to predict the per-point instance annotation.

There are several relevant outdoor lidar localization methods that make use of semantics and segments. Vidanapathirana et al. [21] used global descriptors with segments and spatiotemporal high-order pooling for place recognition. Kong et al. [14] presented a semantic graph-based approach to place recognition, where the topological information of the point cloud is preserved. Zhu et al. [27] extracted common semantic classes, such as vehicles, trunks, and poles from the raw point cloud for loop closure detection. The above methods are designed primarily for outdoor scenarios and are inadequate for an indoor setting. However, they demonstrate the value that semantic information brings to place recognition. To extend this line of research, we leverage a panoptic segmentation method that predicts both the semantic mask and instance label of each point.

B. Indoor localization

Specifically focusing on indoor localization, the state-of-the-art methods often focus on planar floors or geometric features which describe corners and intersections as landmarks for localization. For example, Wei et al. [24] used planar floor assumption to constrain the vertical pose drift of a robot in a multi-floor parking lot. [26, 10] used planar surfaces to efficiently align two lidar scans for loop closure detection. Li et al. [15], Wang et al. [23] used floor plan features such as corners and wall intersections for localization. Bae et al. [3] proposed to use semantic features to detect and match corners of doors and walls. Other works rely upon a predefined map of the world such as a BIM model or a floor plan. Hendrikx et al. [12], Yin et al. [25] built a map from a subset of semantic entities and their associated geometries drawn from a BIM model of the world. They used a spatial database to query the position of the robot within a graph-based localization approach. They impose a prior to use static features for localization. In comparison to these approaches, we do not rely on planes or any other explicit structure to constrain our localization performance. Instead, our approach is to learn to segment semantically meaningful objects and match them between different observations of the scene.

III. Methodology

In this section, we first formulate the research problem, then present the entire pipeline as shown in Fig.2: the panoptic segmentation network, the instance description module, and the matching and pose estimation module,

A. Overview

The problem is defined as localizing a single query lidar scan $\mathcal{Q} = \{\boldsymbol{q}_i \in \mathbb{R}^3\}$ within a prior map $\mathcal{M} = \{P_1, P_2, \dots P_{t_{i-1}}\}$. We seek to determine the pose of the lidar at time t_i defined as follows,

$$\boldsymbol{x}_i \triangleq [\mathbf{t}_i, \mathbf{R}_i] \in \mathrm{SO}(3) \times \mathbb{R}^3 \tag{1}$$

where $\mathbf{t}_i \in \mathbb{R}^3$ is the translation, $\mathbf{R}_i \in \mathrm{SO}(3)$ is the orientation of $\mathcal{Q}$ in $\mathcal{M}$. The map $\mathcal{M}$ is a collection of registered lidar scans, $P_t = \{\boldsymbol{m}_{i,t} \in \mathbb{R}^3\}$, accumulated over time.

We approach the problem at the level of objects and compute the pose $\boldsymbol{x}_i$ by matching object instances identified in the query scan $\mathcal{Q}$ with those previously identified in the map $\mathcal{M}$.

The first step is to partition the map scans into meaningful object instances. Prior approaches have used planes or region growing methods to segment objects with a scan. This segmentation approach works well in outdoor environments such as in the case of autonomous vehicle localization. This is because

sizes of outdoor objects and the separation between them is many times greater than the average inter-point distance in a lidar scan. However, in indoor environments, due to the close proximity of objects with each other, space partitioning and region growing approaches perform poorly. On the other hand, there are many distinguishable objects such as furniture, doors and windows; because of this we can use semantic object segmentation to partition the environment into these object segments.

Objects observed in a query scan will often be quite different from those in the prior map. This could be due to observing the object from a different viewpoint in the query scan than from which it was observed in the prior map. Partial observations from different viewpoints, occlusions by other objects, or different point sampling densities (if the query and map scans were taken from different ranges) all contribute to variation in the reconstruction of an object instance. Due to this variation in the observations, finding matches by aligning a 3D point cloud of the objects between the query and the map will result in poor pose estimation. To overcome this issue, we use object level descriptors that capture the distinguishing features of each object. Estimating pose by matching these descriptions provides some robustness against the variations which occur due to differing viewpoints and partial observations. Descriptor matching also requires lower computational and memory resources as the dimensions of the descriptor are typically smaller than the number of points in each object.

After this step, descriptors of the objects segmented from the query scan need to be matched against the database of objects with descriptors in the map to determine correspondences between the query scan and the map. We use the approach from [1] to group descriptors based on their similarity and to find correspondences. Finally, we use RANSAC on a subset of correspondences to estimate the 6-DOF pose of the lidar sensor by aligning the matched objects between the two scans.

In InstaLoc both the instance segmentation module and the instance description module are modeled using deep neural networks which work directly on 3D point cloud data. Typical lidar scans are point clouds with large amounts of spatially sparse data. We use sparse tensors to represent this data and designed both networks in our framework using the Spatially Sparse Convolution Library (SpConv) [4] which uses sub-manifold sparse convolutions in its neural network implementation. Sub-manifold convolutions have the advantage that they maintain a greater degree of sparsity than other sparse convolutions by overcoming the issue of sub-manifold dilation [11]. As a result, deeper networks with lower memory and computational requirements, and practical real-time capabilities can be constructed to work with large amounts of sparse data. Furthermore, Graham et al. [11] also showed that sub-manifold sparse convolutions are more efficient than alternate approaches that use spatial partitioning schemes.

B. Instance Segmentation

The instance segmentation module is a point-wise panoptic segmentation network. Given a lidar scan, i.e. a set of N 3D points, $P = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \dots \boldsymbol{p}_N | \boldsymbol{p}_k \in \mathbb{R}^3\}$ as input, the network predicts for each point $\boldsymbol{p}_k$ a semantic label s_k corresponding to the object class that the point belongs to (e.g. chair, table, wall, ceiling) and an instance label i_k representing the unique object that the point corresponds to (e.g. chair1, chair14 or chair42). We use the state-of-the-art *Softgroup* [22] network architecture to construct this module. This architecture consists of three stages; (1) a U-Net based point-wise prediction network that generates semantic scores and an offset vector representing the distance from the point to the instance it belongs to, (2) a soft-grouping step where points are grouped by similarity of their semantic scores and their spatial distance to generate instance proposals and (3) a refinement network that extracts features for every instance proposal and then uses a tiny U-Net based network to refine the proposals.

A fixed distance threshold used to group the points in step (2) of the *Softgroup* architecture works well for point clouds with uniform sampling density. In lidar scans, the sampling density is much lower along the vertical axis as compared to the density in the horizontal axis and points become further separated as the sensing range increases. If a fixed distance threshold is used for grouping, then the number of instance proposals would be overestimated in regions further away from the sensor; this can lead to incorrect object segmentation. We counter this issue by using an adaptive radius threshold proportionate to the vertical distance between two beams. Typically, 3D lidars rotate 360° horizontally and have a vertical field of view of θ radians. For a point $\boldsymbol{p}_i(x_i, y_i, z_i)$ resulting from a lidar beam in a point cloud, with the sensor origin O, its radius threshold ρ_i is:

$$\rho_i = \alpha \cdot d(\boldsymbol{p}_i, O) \cdot \tan(\frac{\theta}{N_{beam}}) = \alpha \cdot d(\boldsymbol{p}_i, O) \cdot \frac{\theta}{N_{beam}} \quad (2)$$

where

$$d(\boldsymbol{p}_i, O) = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad (3)$$

and N_{beam} is the number of lidar beams and α is a constant scale factor.

The output of panoptic segmentation network is a set of M object instances $\mathcal{I} = \{I_1, I_2, \dots, I_M\}$ where each object instance is a set of N_j points representing the 3D coordinates of the point and the semantic label s_j of the object, i.e. $I_j = \{\boldsymbol{h}_{k,j} | k = \{1, 2 \dots N_j\}, \boldsymbol{h}_{k,j} = (\boldsymbol{p}_{k,j}, s_j)\}$.

C. Instance Description

After the object instances are segmented, the next step is to generate descriptors for each of the instances. An overview of the network is shown in Fig. 3. The network is designed to be small and fast: it can take all instances in one batch with varying number of points as input. This is done using the instance descriptor network, which consists of four sub-manifold sparse convolutional layers of increasing feature size followed by three fully connected layers, with a dropout layer before the final fully connected layer. The input to the network is a set of object instances $\mathcal{I} = \{I_1, I_2, \dots, I_M\}$, the output of the instance segmentation network, with each instance I_j containing N_j points (with N_j varying for each object). The descriptor network output for each object instance I_j is an

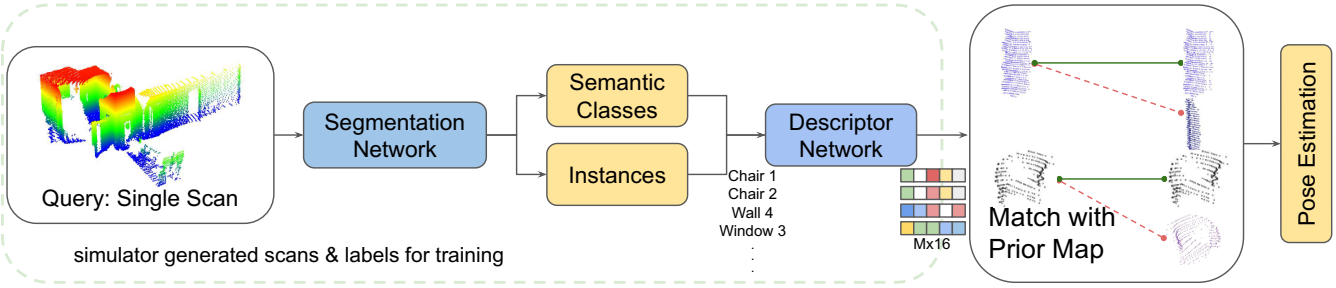


Fig. 2: Overview of the proposed learned lidar localization system

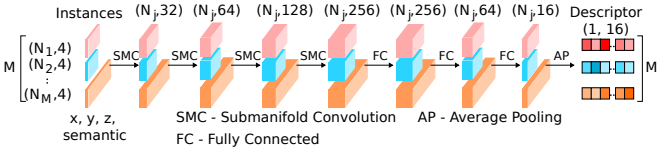


Fig. 3: Instance descriptor network architecture. The input is a set of object instances with a variable number of points per instance, with each point representing the 3D coordinates of the point and the semantic label of the object. The network consists of layers of sub-manifold convolutions with increasing feature size followed by fully connected layers, with dropout before the final fully connected layer. Finally an average pooling layer computes a single descriptor for each object instance.

$N_j \times D$ tensor where every row is a descriptor of length D for one point in the object instance. Finally, an average pooling layer computes the average of the N_j descriptors to create a single descriptor of length D for each object instance. This results in an output vector of dimensions $M \times D$, where M is the number of object instances.

The network is trained using triplet loss. If $a, p, n \in \mathbb{R}^D$ are the descriptors for an anchor, the corresponding positive element and a negative element respectively, then the triplet loss $\mathcal{L}_{triplet}$ can be calculated as

$$\mathcal{L}_{triplet}(a, p, n) = \max\{d(a, p) - d(a, n) + m, 0\} \quad (4)$$

where

$$d(x, y) = ||x - y||_2$$

is the pairwise distance between the descriptors; and the margin m is set to 1. The average loss over all the samples in a mini-batch is considered as the loss during training.

D. Matching

For each instance in the query scan, we first obtain its N closest descriptors from the database of instances of the prior map. This generates a list of instance-to-instance correspondences. A correspondence grouping method from [1] is used to find the correct correspondence. We start from a seed correspondence $c_n = \{I_n^Q, I_n^M\}$, where I_n^Q and I_n^M are two instances from the query scan and the prior map respectively. We then loop through all candidate correspondences, so another correspondence $c_m = \{I_m^Q, I_m^M\}$ can be grouped with c_n if:

$$||I_n^Q - I_m^Q|| - ||I_n^M - I_m^M|| < \epsilon \quad (5)$$

ϵ is the parameter that restricts how strictly the grouping algorithm behaves. The accepted consensus group has to contain a minimum of τ instances. Finally, for the 6 DoF pose estimation, we apply a RANSAC step on the subset of correspondences to align the query scan with the prior map, with τ and ϵ.

IV. Implementation

A. Simulated Lidar Data

Training deep learning algorithms requires large amounts of data. To bypass the need to do time-consuming manual labeling, we constructed several indoor environments in the Unreal Engine simulator to take advantage of automatic labeling. As well as being automatic, it eliminates errors in human labeling and can be easily extended to other environments. We created about 20 unique rooms and assembled them into six room networks which contained a total of ∼1500 objects. As an example, two of the six networks are shown in Fig. 4. We used the Airsim plugin [19] to capture over 90 scans from these spaces.

The simulator allowed us to configure the lidar settings — including frequency, range, the field of view, and the number of lidar beams. The simulated lidar configuration we used was modelled on the Ouster OS-128 lidar [1], which has ∼50 m range, 90° field of view, and 128 lidar beams. Note, that this is a wide field of view and dense lidar coming on the market. Similarly to the existing indoor point cloud dataset, Stanford 3D Indoor Scene Dataset (S3DIS) [2], we used 13 object classes: ceiling, floor, column, beam, wall, table, chair, bookcase, sofa, window, door, board, and clutter.

1) Labeled Data for Instance Segmentation: Each simulated lidar beam that intersects with an object would result in a range measurement and a unique object ID. Using the object ID, we can assign a semantic class and an instance number. These labels are used in the supervised training of the two networks. Overall, each point has five fields: (X, Y, Z) coordinates, semantic class, and object instance number.

[1] https://ouster.com/products/scanning-lidar/os0-sensor/

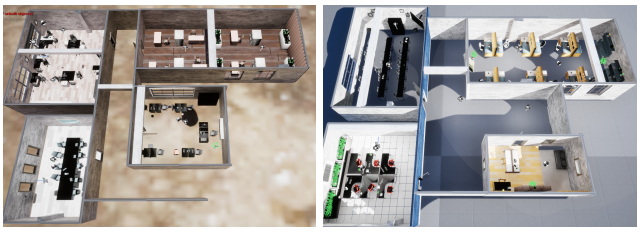


Fig. 4: Two indoor office networks constructed using the Unreal Engine to simulate lidar scans with semantic labels.

2) Triplets for Descriptor Network: To train the descriptor network, we need to generate object instances as triplets - with anchor, positive and negative instances. First, we generate two scans that are 2m apart and a 10° rotation in the simulator. Given that every object in the lidar scan is labeled, we classify the same objects in these two scans as the anchor and positive instance. We then randomly selected another object as the negative instance. Because the anchor and positive instances have mild viewpoint differences, the objects scanned in the point clouds may have slight changes. These slight appearance changes contribute to algorithm robustness. In total, about 9900 triplet object instances were generated for training, validation, and testing.

B. Training

As mentioned in Sec. III-A, both networks are built with a sparse tensor framework and were trained on a 4GB mobile GPU, NVIDIA Quadro T2000.

1) Instance Segmentation Network: We use a pre-existing Softgroup model (trained on S3DIS) as a warm start. The voxel size was set to 2 cm and the minimum number of points in each instance was set to 50. The network was trained for 50 epochs which took about one hour.

2) Instance Descriptor Network: The network is trained from scratch with a triplet loss function (Eq. (4)). Compared to a whole scan (which usually contains over 100,000 points) each triplet instance is only a small fraction of a whole scan; because of this we could increase the batch size to allow parallel input. The descriptor network was trained for 90 epochs, and took around 90 minutes.

Both networks were trained with an Adam optimizer with a learning rate of 0.001.

V. Experiment and Results

In this section, we describe experiments conducted on instance segmentation and descriptor networks. This is followed by real world experiments using InstaLoc as a complete localization system. Lastly, we demonstrate that the algorithm is robust to a changing number of prior map scans which indicates robust performance.

A. Experimental Setup

We use a fully labeled simulated dataset to train the instance segmentation and descriptor network. The dataset also holds 113 test scans for the instance segmentation network and 2123 test triplet instances for the descriptor network.

For the localization experiment, we collected an indoor environment dataset using a Ouster lidar OS-128 in small, medium, and large scale buildings. The dataset includes sequences in office rooms, meeting rooms, and social spaces as well as lecture theatres, staircases, and hallways. Fig. 5 shows the prior maps built with a lidar SLAM system. The SLAM poses are 0.7 m apart so there are 147, 192, and 384 individual scans which form the final map for George, Thom, and IEB buildings. As an indication of size, the estimated map floor area for each building is around 500 m^2, 1100 m^2, and 2000 m^2 respectively. However, in our localization experiments, the prior map is made up of a subset of registered scans that are spaced 2.1 m apart. As the lidar sensor was running at 10 Hz, the localization system is triggered every ten scans - once per second.

Tab. II presents specific details for each building. For example, the prior map of George Building consists of 32 scans, and the trajectory length is 96 m. In total, 106 scans were queried. A detection is classified as being correct when the estimated pose is within 0.2 m and the orientation is within 10° of the ground truth pose. Please note, there is no point cloud alignment step, such as Iterative Closest Point (ICP) refinement, and the pose estimation is from the instance correspondence matching.

B. Results

1) Instance Segmentation Results: Fig. 6 shows two illustrations of instance segmentation results. The left side image of a scan is from the simulated dataset. In this classroom environment, each object instance has been assigned a random color. Chairs, tables, and wall surfaces are individually segmented. Note that the door (colored in red) is partially segmented from the blue wall. In another example result, the right side image of a scan was captured in a hallway in George Building with the Ouster lidar. The hallway connects several rooms with lidar beams scanning into those rooms, which resulted in several partially scanned walls. The ceiling is accurately segmented (colored in orange), but the blue wall is mixed with one light green and one black segment. The imperfection in segmentation is expected as the current state-of-the-art instance segmentation method, SoftGroup, achieves an average precision (AP) of around 54.5 % on indoor datasets such as the S3DIS dataset [2].

Unlike the S3DIS dataset, there is no visual color information for lidar points in our simulator synthesized data. In addition, our data contains a larger variety of spaces and objects than S3DIS, and some objects in the scans are scanned partially. Hence after applying the default SoftGroup on our synthesized data, it reaches 39 % average precision across 13 classes. Our proposed improvement of incorporating lidar properties in (2) improved the Average AP from 39 % to 41 %, shown in Tab. I. Larger objects such as ceilings, floors, and walls have higher AP. Objects such as boards, windows, and doors, which are gathered under ”other1” and ”other2” in Tab. I have much low AP, less than 20 %.

One key design consideration for our localization method to be able to deal with imperfect segmentation is to use all

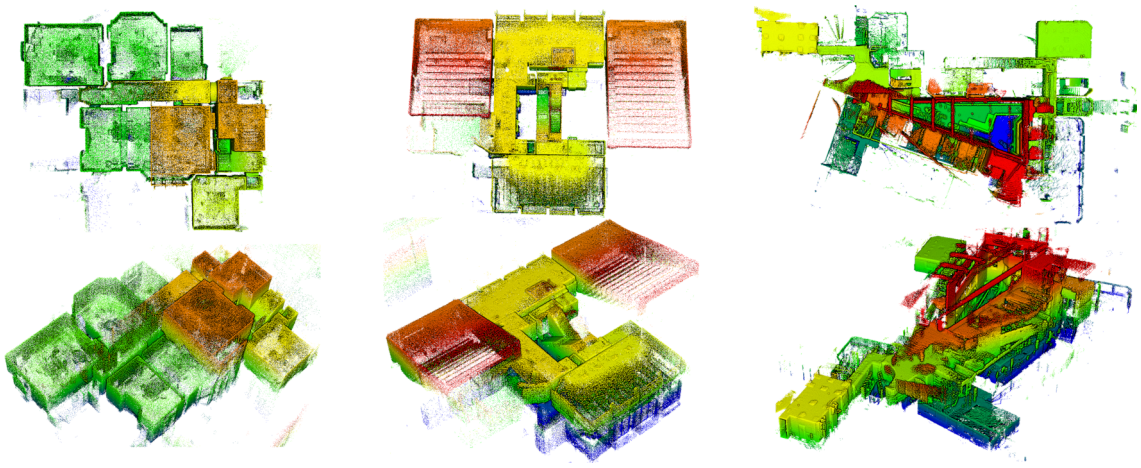


Fig. 5: Our indoor datasets. The height direction is indicated by color: blue is the lowest level and red is the highest level. *Left:* Small size George building. *Middle:* Medium size Thom building. *Right:* Large size Information Engineering Building.

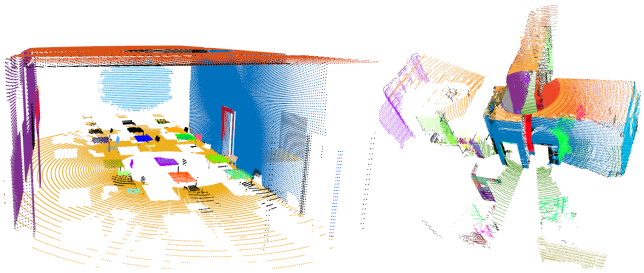


Fig. 6: Instance segmentation results. Left: Result with a simulated lidar scan. Right: Result with a real Ouster lidar scan. Random colors are assigned to each instance.

Class	AP	Class	AP	Class	AP
ceiling	0.923	floor	0.838	wall	0.565
column	0.632	beam	0.367	chair	0.723
sofa	0.402	others1	0.144	others2	0.163
Average AP				41.3	

TABLE I: Average precision for each object class. others1 is the mean value of table, board, and window, others2 is the mean value of door, bookcase, and clutter.

available instances with descriptors that can tolerate incomplete object point clouds.

2) Instance Descriptor Results: In 3D point cloud learning, data representation and augmentation have a significant impact on achieving on best matching or labeling performance. We experimented with several approaches and found that centering individual instances and applying random rotations during data preparation can optimize learning results. We randomly eliminate 20 % of the points in each instance and add random noise to lidar point positions during data preparation to improve descriptor robustness.

Fig. 7 (right) presents descriptor pairwise distances between the anchor, positive and negative instances in a subset of 120 test triplets. The blue lines correspond to smaller, positive distances (as desired). There is a clear separation between the

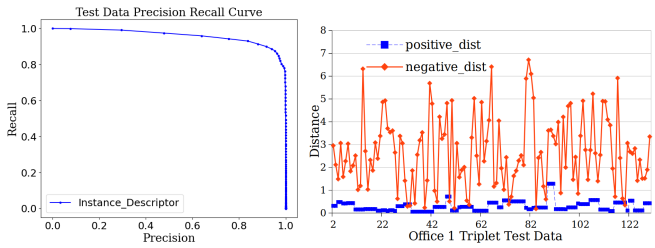


Fig. 7: *Left:* A precision/recall curve for all test data in the descriptor network. *Right:* A subset of the test data showing in blue the distance between the anchor and the positive instances. In red, the line shows the distance between the anchor and the negative instances.

typical positive and negative distances.

The graph in Fig. 7 (left) shows the precision and recall curve for the 2000 test triplets. At a descriptor distance ($\mathcal{L}_2$ norm) threshold of 0.56, the model can classify the instances with 91.4 % precision and 88.1 % recall. Here we purposely choose a smaller distance threshold to have higher precision as false positives are more detrimental to the localization system. Our network is fast and efficient. For comparison, we tested on the Thom Building dataset. Averaged across all scans, ESM [20] descriptor processed 30 segments in a scan in 72 ms, while InstaLoc descriptor network processed 30 instances in 21 ms. In addition, our descriptor network can operate on any number of input points but ESM needs to downsample a segment to 256 points and SegMap uses fixed 3D voxel grid dimension of $32 \times 32 \times 16$ which compromises on detail.

3) Localisation Results: Fig. 1 shows a lidar scan that has been successfully localized within the ground floor of Thom building. Several object instances have been matched including walls, sofas, and flat planes (TV screens). The top section shows the matched instances within the query scan, and the bottom section shows the matched instance within the larger prior map. The estimated pose is indicated with a gold arrow.

InstaLoc and two state-of-the-art baselines were tested with datasets from George, Thom, and Information Engineering

Data Building	Length (m)	Scan Number Map	Query	ESM* [20] Detect	Recall	Precision	SegMap* [8] Detect	Recall	Precision	InstaLoc (Ours) Detect	Recall	Precision
George	96	32	106	12	11 %	75 %	28	26 %	81 %	**56**	**49 %**	**93 %**
Thom	121	45	137	36	26 %	**92 %**	28	30 %	83 %	**88**	**58 %**	91 %
IEB	253	98	211	29	14 %	93 %	27	13 %	56 %	**94**	**42 %**	**95 %**

TABLE II: Numerical summary table showing the performance of InstaLoc compared to two state-of-the-art benchmarks. The prior map is made of N scans and the query scan is the total number of scans queried. *: both methods have been adapted for better performance.

Building (IEB). InstaLoc successfully detected 48 out of 106 scans in the prior map of George building, and all detections were correct according to the ground truth. Hence the recall rate is 45 % and the precision is 100 %. For Thom building, the recall rate was 56 % but the precision was lower at 86 %. The lower precision was largely due to the two near identical lecture theatre halls on the two sides of Thom building, shown in Fig. 5. This caused confusion in the localization system. Over the three sequences, the average recall was around 47 %, and the average precision was about 94 %. Note that all three datasets are for test, the segmentation and description networks have not seen them as they are trained on simulated lidar scans.

We selected two segment-based localization methods as comparative baselines, as they are most similar to our method. We modified the two algorithms to the best of our efforts to offer a fair comparison in indoor environments. In the Efficient Segmentation and Matching (ESM) paper[20], the authors used the Euclidean cluster extraction (ECE) method to segment the lidar scans. As it was originally designed for outdoor environments, objects in the scan are expected to be distinctly separated, especially after removing points corresponding to the ground. However, in an indoor environment, the ECE method cannot separate objects efficiently as walls and ceilings often become one segment. To mitigate this, we first calculate the curvature and remove high curvature points so there are distinct gaps between structured objects. After this, the ECE method can produce more reasonable segments.

The second algorithm we test is SegMap [8]. We first simplified the system by removing the lidar accumulating through the odometry system, as the lidar scans in our test are from a 128-beam lidar so it is very dense compared to 16 or 32-beam lidar used in their paper. More importantly, we used the incremental region growing method [9] for segmentation, which computes local normals and curvatures for each point and uses these to extract flat or planar-like surfaces. After these two modifications, the system can operate in real time and have better segmentation performance.

However, even as we improved the segmentation method in both systems, there is still a limitation in their descriptor network. One factor is that it does not use sparse tensor networks, and as a result only a small and fixed number of points can be used as input.

A table presenting comparison results is shown in Tab. II, our approach outperforms the baseline methods by a factor of between two and four times in recall, and also achieved higher precision. In general, these systems tend to be tuned to prefer higher precision - for accurate and trustworthy localization.

Data Building	Fewer Scans Map	R/P %	Default Density Map	R/P %	More Scans Map	R/P %
George	22	30 / 94	32	45 / 100	48	49 / 84
Thom	33	45 / 97	45	56 / 86	60	54 / 86
IEB	68	30 / 100	98	41 / 97	125	42 / 93

TABLE III: Ablation study: varying the number of scans used for the prior map. The same number of query scans are used as Tab. II. R and P are the recall and precision values respectively.

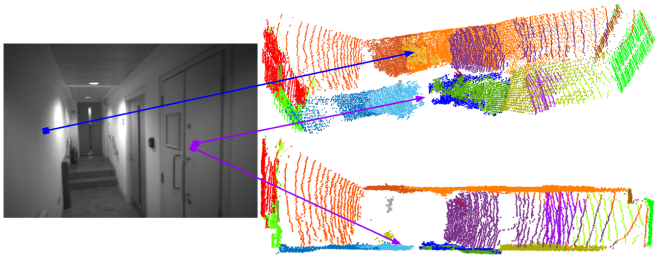


Fig. 8: An example of inferior instance segmentation within a corridor from two different viewpoints. In the side view scan, both the left and right walls are being over-segmented. In the top view scan, the points near the sensor origin ($<$1.0 m) have much higher noise, resulting in uneven wall surfaces.

We also considered ScanContext [13] for comparison, but its descriptor is too rudimentary to work in tight indoor spaces, as opposed to the road networks it was designed for.

4) Varying the Size of the Prior Map: As a robustness test, we conducted experiments to see how the number of individual scans in a prior map can affect localization results. Intuitively, as the number of prior map scans is reduced, the localization detection rate should reduce. Shown in Tab. III, the middle column is the baseline which has the same configuration as II, and the columns left and right of it have either an increased or decreased number of prior map scans. With the decreased number of prior map scans, there is a slight reduction in recall values, but no negative effect on precision. This demonstrates the robustness of our system to changes in the number of prior map scans.

C. Limitations

As mentioned in Sec. V-B1, the precision of the instance segmentation can directly impact the performance of the localization system. Since the descriptor network has already reached high precision and recall values, good instance segmentation is a key way of improving overall localization recall

performance. In our experiments, the instance segmentation network performs well in structured and enclosed spaces such as theatres, classrooms, offices, etc. However, it performs much more poorly in corridors and staircases, especially when there are embedded small objects inside the walls, such as handrails.

As shown in the camera image in Fig. 8 (left), there were fire extinguishers, radiators, and cupboards along the corridor walls. We did specifically use examples of hallways and corridors in our training dataset. While that did improve performance, the results still have room for improvement. This might be due to the inconsistent point cloud density on the walls and the noisier lidar measurements from the Ouster lidar at close distances. An example of this issue is shown in Fig. 8 (right) in a corridor. This is a topic for future work.

VI. Conclusion

In this paper, we proposed a fast and accurate lidar localization approach. InstaLoc learns to segment and describe different object instances in a scene. It consists of two networks, joined together to recognize and describe individual objects. InstaLoc can localize between two to four times as many matches as two state-of-the-art baseline methods while retaining high levels of precision. In future work, we want to improve the localization performance in hallways and corridor spaces. Moreover, we intend to combine visual information with lidar measurements for instance segmentation. Equally important, we aim to extend InstaLoc to be independent of the type of operating environment. Lastly, we will add flexibility to InstaLoc to work with sparse lidar scans from different lidars.

Acknowledgments

This research was partly funded by the Horizon Europe project DIGIFOREST (Grant ID 101070405), UKRI/EPSRC ORCA Robotics Hub (EP/R026173/1), and a Royal Society University Research Fellowship (Fallon).

References

[1] Aitor Aldoma, Federico Tombari, Luigi di Stefano, and Markus Vincze. A global hypotheses verification method for 3d object recognition. In *European Conference on Computer Vision*, 2012.

[2] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D semantic parsing of large-scale indoor spaces. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1534–1543, 2016.

[3] Sang-Hyeon Bae, Sung-Hyeon Joo, Jun-Hyun Choi, Hyun-Jin Park, and Tae-Yong Kuc. Localization system through 2D LiDAR based semantic feature for indoor robot. In *International Conference on Ubiquitous Robots*, pages 338–342. IEEE, 2022.

[4] SpConv Contributors. SpConv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022.

[5] Andrei Cramariuc, Florian Tschopp, Nikhilesh Alatur, Stefan Benz, Tillmann Falck, Marius Bruehlmeier, Benjamin Hahn, Juan I. Nieto, and Roland Y. Siegwart. SemSegMap – 3d segment-based semantic localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1183–1190, 2021.

[6] Bertrand Douillard, Alastair Quadros, Peter Morton, James Patrick Underwood, Mark De Deuge, S Hugosson, M Hallström, and Tim Bailey. Scan segments matching for pairwise 3d alignment. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3033–3040. IEEE, 2012.

[7] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMatch: Segment based place recognition in 3D point clouds. In *International Conference on Robotics and Automation (ICRA)*, pages 5266–5272. IEEE, 2017.

[8] Renaud Dubé, Andrei Cramariuc, Daniel Dugas, Juan Nieto, Roland Siegwart, and Cesar Cadena. SegMap: 3d segment mapping using data-driven descriptors. In *Robotics: Science and Systems (RSS)*, 2018.

[9] Renaud Dubé, Mattia G. Gollub, Hannes Sommer, Igor Gilitschenski, Roland Siegwart, Cesar Cadena, and Juan Nieto. Incremental-segment-based localization in 3-d point clouds. *IEEE Robotics and Automation Letters*, 3(3):1832–1839, 2018.

[10] Patrick Geneva, Kevin Eckenhoff, Yulin Yang, and Guoquan Huang. LIPS: Lidar-inertial 3D plane slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 123–130. IEEE, 2018.

[11] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9224–9232, 2018.

[12] Bob Hendrikx, Pieter Pauwels, Elena Torta, Herman Bruyninckx, and Marinus van de Molengraft. Connecting semantic building information models and robotics: An application to 2D LiDAR-based localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11654–11660, 2021.

[13] 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, 2018.

[14] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. Semantic graph based place recognition for 3d point clouds. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8216–8223, 2020.

[15] Zhikai Li, Marcelo H Ang, and Daniela Rus. Online localization with imprecise floor space maps using stochastic gradient descent. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8571–8578. IEEE, 2020.

[16] Kirk MacTavish, Michael Paton, and Timothy D Barfoot. Selective memory: Recalling relevant experience for long-term visual localization. *Journal of Field Robotics*, 35(8):1265–1292, 2018.

[17] Matias Mattamala, Milad Ramezani, Marco Camurri, and Maurice Fallon. Learning camera performance models for active multi-camera visual teach and repeat. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 14346–14352, 2021.

[18] Sebastian Ratz, Marcin Dymczyk, Roland Y. Siegwart, and Renaud Dubé. OneShot global localization: Instant LiDAR-visual pose estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5415–5421, 2020.

[19] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. *International Symposium on Field and Service Robotics*, 2017.

[20] Georgi Tinchev, Adrian Penate-Sanchez, and Maurice Fallon. Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a CPU. *IEEE Robotics and Automation Letters*, 4(2): 1327–1334, 2019.

[21] Kavisha Vidanapathirana, Peyman Moghadam, Ben Harwood, Muming Zhao, Sridha Sridharan, and Clinton Fookes. Locus: LiDAR-based place recognition using spatiotemporal higher-order pooling. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5075–5081, 2020.

[22] Thang Vu, Kookhoi Kim, Tung Minh Luu, Xuan Thanh Nguyen, and Chang-Dong Yoo. Softgroup for 3d instance segmentation on point clouds. In *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2698–2707, 2022.

[23] Xipeng Wang, Ryan J Marcotte, and Edwin Olson. GLFP: Global localization from a floor plan. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1627–1632. IEEE, 2019.

[24] Xin Wei, Jixin Lv, Jie Sun, Erbao Dong, and Shiliang Pu. GCLO: Ground constrained lidar odometry with low-drifts for gps-denied indoor environments. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2229–2235. IEEE, 2022.

[25] Huan Yin, Zhiyi Lin, and Justin KW Yeoh. Semantic localization on BIM-generated maps using a 3D LiDAR sensor. *Automation in Construction*, 146:104641, 2023.

[26] Lipu Zhou, Shengze Wang, and Michael Kaess. π-LSAM: LiDAR smoothing and mapping with planes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5751–5757. IEEE, 2021.

[27] Yachen Zhu, Yanyang Ma, Long Chen, Cong Liu, Maosheng Ye, and Lingxi Li. GOSmatch: Graph-of-semantics matching for detecting loop closures in 3d lidar data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5151–5157. IEEE, 2020.

Appendix: Further Implementation Details

This appendix discusses further implementation details of the InstaLoc method and acts as an extension to Sec. IV of the main paper. In particular, we will discuss the training procedures, data generation and augmentation, and our triplet mining strategy.

1 Panoptic Segmentation Network

For panoptic segmentation of the lidar scan, we adopted SoftGroup method of Vu, Thang et al[1], a state-of-the-art method for 3D point cloud instance segmentation. SoftGroup offers a model pertained on the S3DIS dataset. Each scan in the S3DIS dataset is a high accuracy static 3D scan of a room with millions of points. Hence we need to adapt the network and model to better suit the sparser point clouds from 3D automotive lidars such as Ouster or Velodyne. Details of the algorithmic adaptation are in Sec III B. The radius threshold ρ_i in Equa. (2) is added into CUDA processing for each point.

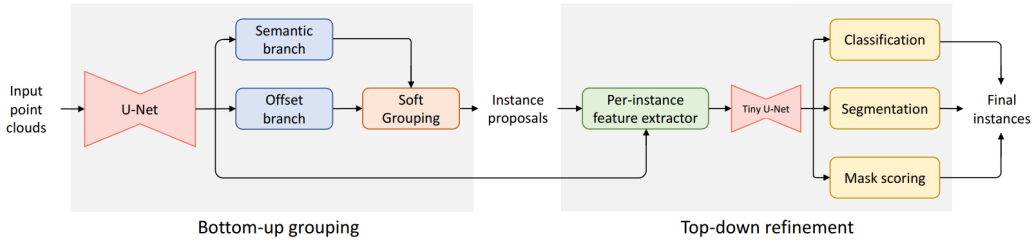


Figure 1: SoftGroup panoptic segmentation architecture

1.1 Generating Simulated Scans

Extending the description presented in Sec. IV A, we purchased the 3D model assets [2] from the Unreal Engine marketplace and built a six-room building space. Within this space we simulated the lidar scans shown in Fig. 4. We enabled complex collision properties in the Unreal Engine to obtain object details such as object instance and class for each simulated lidar points - not just just their bounding boxes. Similarly to the S3DIS dataset, we assign the objects to 13 classes. Tab. 1 shows the number of objects in each class.

- Movable: board, bookcase, chair, sofa, table
- Fixed: beam, ceiling, column, door, floor, wall, window

Note that objects are classed as clutter, which could be both movable or fixed, but are typically movable.

Level	beam	board	bookcase	ceiling	chair	clutter	column	door	floor	sofa	table	wall	window	Total
Level 1	4	4	10	1	34	119	3	8	4	0	37	18	9	251
Level 2	4	10	11	1	39	113	4	8	4	1	32	20	7	254
Level 3	4	12	7	1	35	109	4	8	6	4	33	30	7	253
Level 4	1	16	8	1	35	109	3	11	6	6	26	20	9	251
Level 5	5	14	3	1	59	65	6	9	5	2	50	16	13	248
Level 6	7	5	4	1	41	98	8	7	11	5	38	16	13	254

Table 1: Number of objects in each class in the simulated environments

1.2 Data Augmentation

Similarly to many point cloud segmentation methods, we augmented the scans by adding Gaussian noise of 1cm along each of the three dimensions, randomly mirror flipping each scan, and applying random yaw rotations.

[1] Vu, Thang et al. "SoftGroup for 3D Instance Segmentation on Point Clouds." 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022): 2698-2707.

[2] https://www.cgtrader.com/3d-models/interior/interior-office/low-poly-interior-13-office

1.3 Training

We captured 270 simulated scans for training, validation, and testing, and split them into 90 scans each. The training step aims to fine-tune the pre-trained S3DIS model to adapt to the sparser lidar scans. The original S3DIS dataset contains 271 scans so we use a smaller number of simulated scans for fine-tuning training.

To fine-tune the pre-trained model, we adapted a two-step training process. First, we froze the modules in the top-down refinement section (highlighted in color) and only trained the U-Net, semantic branch, and offset branch modules. Second, we froze the aforementioned modules and fine-tuned the modules in the tiny U-Net, classification, segmentation, and masking scoring. In this way, we can maintain all the weights in the pre-trained model and adjust them for sparse 3D lidar scans.

2 Instance Descriptor Network

2.1 Generate Triplet Data

Extending the description in Sec IV A, we obtain instance triplets from the simulated lidar scans in 1.1. So as to achieve view invariance in the instance descriptor, we generate pairs of Lidar scans from nearby poses in each room, $2\,\mathrm{m}$ apart with a 10° rotation around the yaw axis. The algorithm to generate triplets is described below. In total, we generated 9900 triplets to train the descriptor network.

Algorithm 1: Generate triplets for a pair of lidar scans

Result: N number of triplets
Segment objects in each scan by using their ground truth object instance ID;
Bin the objects based on their semantic label;
while *each semantic label* **do**
  `anchor`: Pick an object from scan 1;
  `positive`: Pick the same object from scan 2, using the instance ID;
  `negative`: Randomly pick 4 other objects: from the same class in scan 1 and 2; from a different class in scan 1 and 2;
  return 4 x (anchor, positive, negative)
end

2.2 Data Augmentation

For each instance in the simulated triplet data, we follow the same procedure as Appendix 1.2. We noticed the simulated lidar scans have very even spacing between points and little noise in the scan line pattern compared to real lidar scans, so we randomly eliminate $20\,\%$ of the points in each instance to mimic such effect.

2.3 Training

The train, validate and test split for the triplet network was chosen to be around 60, 20, and $20\,\%$. As the size of each triplet is relatively small compared to the original point cloud, we train and test 8 triplets in a batch on our mobile GPU. Note that during inference we combine all instances from each point cloud as one batch for the forward pass.

3 Sim to Real Transfer

In general, the sim-to-real domain gap between lidar scans is smaller than with images. Images depend on lighting, reflection, camera angle, and object texture making transfer much more challenging. By contrast, 3D lidar scans only contain geometric information (XYZ point coordinates) which is much more amenable to direct transfer and training with a smaller amount of samples. In addition, we applied data augmentation in Appendix 1.2 and 2.2 to narrow the gap between the simulated point clouds and the real point clouds.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning
Publication Status	x Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Lintong Zhang, Tejaswi Digumarti, Georgi Tinchev, and Maurice Fallon, "InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning," *Robotics: Science and Systems*, 2023.

Student Confirmation

Student Name:	Lintong Zhang		
Contribution to the Paper	• Contributed to development of the idea • Implemented novel contributions mentioned in the paper • Executed closed-loop field experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	April 16, 2024

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Maurice Fallon		
Supervisor comments Lintong was the lead instigator, implementer and researcher on this project and wrote the bulk of the paper		
Signature 	Date	May 03, 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

5.2 Discussion

5.2.1 Time Complexity

Although the system processes one scan at a time to build the prior map database, the representation of the database uses the KDTree structure to store the descriptor for objects in the scans. Hence, the size of the prior map is not directly proportional to the number of scans, but the number of objects in the scans. Inserting a new descriptor in the balanced KDTree takes $\mathcal{O}(\log n)$ time. During online inference, we purposely design the segmentation module to only keep large objects in the scan such as walls, ceilings, and tables. Each object is queried through the KDTree to find its nearest neighbour in the database, which typically has a complexity of $\mathcal{O}(\log n)$ time, n being the number of objects in the database. For m objects in the query scan, the final complexity would be $\mathcal{O}(m \log n)$. Hence our system is scalable to large areas given that the overall complexity is influenced by the KDTree retrieval. However, InstaLoc uses two neutral networks to conduct segmentation and generate descriptors. So the time cost of the KDTree is insignificant compared to the network inference time.

5.2.2 Pose Confidence

Throughout the system, we applied tight thresholds to achieve high precision for pose estimation. During the step of matching with the prior map in Sec. D, the closest neighbour of each object in the database is thresholded by the Euclidean distance of the descriptors. Furthermore, the ϵ in correspondence grouping, Eq. 5, has a relatively small value. The system requires a minimum number of objects to match before estimating a pose. So if the sensor is positioned outside the prior map, the system is unlikely to produce any false positives.

If we were to produce a pose covariance metric, there are a few metrics that can be combined into a pose uncertainty value: descriptor distances between all matched objects, the number of matched objects, and the physical distance between the centroid of matched objects.

In Sec. V experimental evaluation, we stated that a detection is classified as being correct when the estimated pose is with 0.2 m and the orientation is with 10° of the ground truth pose. This metric criteria is selected because an additional ICP step should easily converge from an initial guess meeting these criteria.

Since the publication of our initial research, there have been more developments in the field of 3D scene understanding and instance segmentation. These advancements offer inspiration and highlight several areas for potential improvement in our research.

Firstly, the ScanNet dataset offers a wider variety than the S3DIS dataset. It contains indoor scenes that more closely mimic real-world lidar scans. By switching the pre-trained model from S3DIS to ScanNet, we may be able to improve the segmentation module to generalise across a broader range of indoor environments. This shift could streamline our processes by eliminating the need for additional fine-tuning on synthetic lidar data from Unreal Engine, while also harnessing the increased accuracy provided by the real-world data captured in ScanNet.

The joint processing of RGB images with lidar scans, facilitated by a camera-lidar calibration method, can assign RGB information to each point in the lidar scan. It is crucial to ensure precise alignment between the camera and lidar sensors, allowing for accurate colour projection into the point cloud. The integration of RGB data would further enhance instance segmentation results, as most modern instance segmentation methods utilise colour information.

In addition, we could refine our lidar prior maps by transitioning to an instance-based map that accumulates and merges different semantic labels for each object across multiple scans. This will allow us to create a prior map where each object is distinctly labelled, allowing more detailed and accurate scene representation. By merging instance segmentation inferred across multiple scans within a room, we can correct any inaccurate labels, achieving the most precise per-point label

classification. With the object-instance-based prior map, the development of an object graph could improve how we interact with and analyse 3D spaces. For instance, querying specific objects such as chairs in a given scene could become more intuitive and give more accurate results.

Since the publication, we released follow on work, LiSTA [115] using the descriptor network proposed in this chapter. In LiSTA, we proposed a change detection algorithm that allows us to segment and classify discrete objects that are repositioned between multiple missions. We further introduced a correspondence grouping method and a confidence metric that offers a solution to quantify uncertainty when classifying the changed objects.

Nevertheless, despite the above proposed improvement, the reliance on bulky, high-power, and expensive lidar technology for localisation still remains a drawback. An alternative could be using cameras for localisation within the same 3D lidar prior map. However, bridging the gap between these two sensors remains a challenge, due to their inherent differences in data capture and interpretation. Exploring ways to integrate these modalities will be a primary focus in the next chapter.

6
Visual Localisation in 3D Maps

In the pursuit of more cost effective and accessible localisation solutions within point cloud maps, this chapter explores a cross-modal solution involving camera and lidar data. While lidar maps, obtainable through 3D scanners or SLAM missions, are highly accurate, the weight, power consumption and cost of the lidar sensor are major drawbacks compared to camera-based systems. This significant cost disparity has motivated our exploration of alternative methods.

Building on our previous work, which effectively uses lidar for localisation in maps also generated with lidar, this chapter focuses on substituting the costly lidar component for localisation with a single camera. This transition not only mitigates cost concerns but also broadens the potential for adoption and reuse of a lidar 3D prior map. Additionally, we explore the possibility of directly localising with just a camera within a lidar map, utilising existing SLAM maps or point cloud scans. This approach could facilitate multi-robot operations in a shared map environment, offering a cost-effective camera localisation solution over a resource-intensive full lidar SLAM pipeline.

Moreover, as there are various 3D lidar map representations, this chapter will also explore how these formats operate to enhance our understanding and optimise our localisation methods.

6.1 Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations

The following article has been submitted to the *IEEE Transaction on Robotics (TRO)*. An accompanying video is available online at: https://youtu.be/rkUKaF7i6oc.

Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations

Lintong Zhang[1], Yifu Tao[1], Jiarong Lin[2], Fu Zhang[2], Maurice Fallon[1]

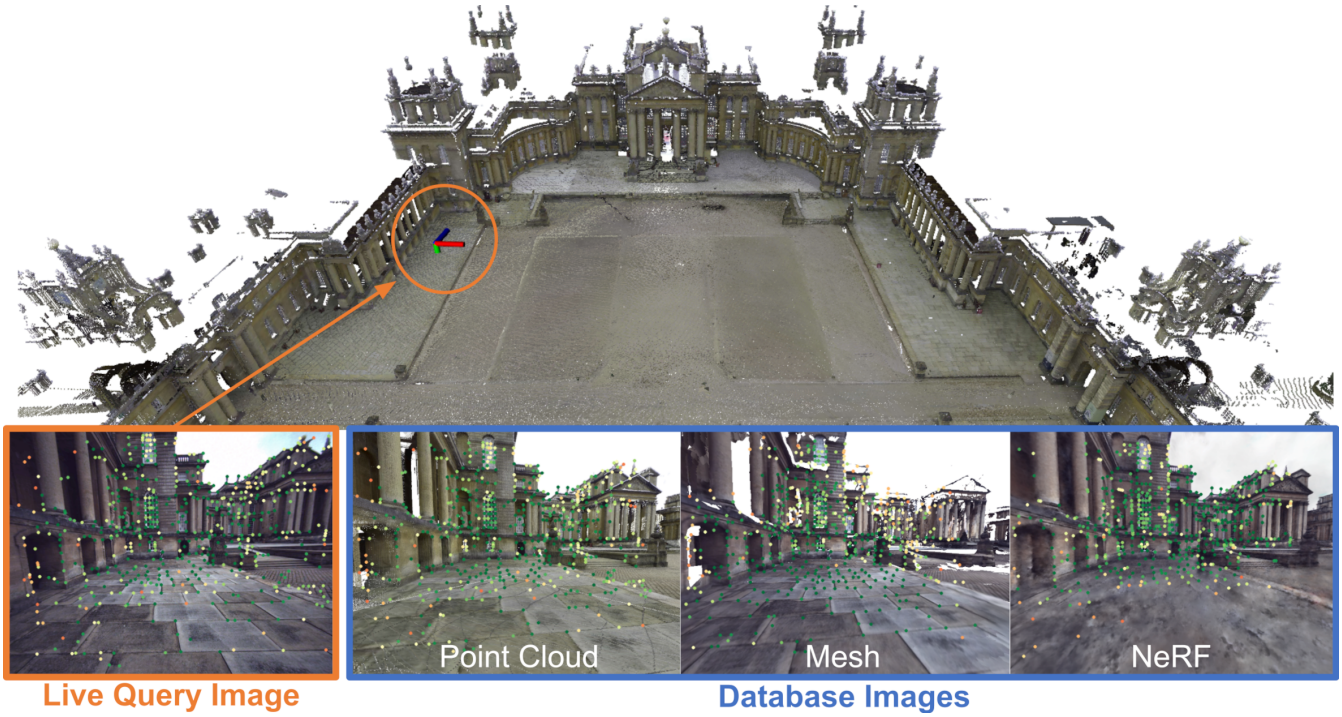


Fig. 1: Localization of a single query image to a database of images that are synthesized from either point cloud, mesh, or NeRF representations of Blenheim Palace. After identifying a matching image in the database, features are extracted with SuperPoint (as shown above) and matched with SuperGlue. (Displayed point cloud is not directly used.)

***Abstract*—This paper introduces and assesses a cross-modal global visual localization system that can localize camera images within a color 3D map representation built using both visual and lidar sensing. We present three different state-of-the-art methods for creating the color 3D map — namely point clouds, meshes, and neural radiance fields (NeRF). Our system constructs a database of synthetic RGB and depth image pairs from these representations. This database serves as the basis for global localization. We present an automatic approach that builds this database by synthesizing novel images of the scene and exploiting the 3D structure encoded in the different representations. Next, we present a global localization system that relies on the synthetic image database to accurately estimate the 6 Degrees of Freedom (DoF) camera pose of monocular query images. Our localization approach relies on different learning-based global descriptors and feature detectors which enable robust image retrieval and matching despite the domain gap between (real) query camera images and the synthetic database images. We assess the system's performance through extensive real-world experiments in both indoor and outdoor settings, in order to evaluate the effectiveness of each map representation, the advantages of learning-based features and descriptors, and the benefits against traditional structure-from-motion localization approaches. Our results show that all three map representations can achieve consistent localization success rates of $55\,\%$ and higher across various environments. NeRF synthesized images show superior performance, localizing query images at an average success rate of $72\,\%$. Furthermore, we demonstrate that our synthesized database enables global localization even when the map creation data and the localization sequence are captured when travelling in opposite directions. Our system, operating in real-time on a mobile laptop equipped with a GPU, achieves a processing rate of $1\,\mathrm{Hz}$.**

***Index Terms*—Localization, Mapping, Sensor Fusion, RGB-D Perception**

Manuscript submitted May, 2024;
[1]Oxford Robotics Inst., Dept. of Engineering Sci., Univ. of Oxford, UK.
[2]Dept. of Mechanical Engineering, Univ. of Hong Kong, China

I. Introduction

Recognizing a previously visited place and estimating a robot/sensor's accurate metric pose is fundamental to the problem of robot localization. Localization not only addresses the question of determining where a robot is, but it also enables other tasks such as loop detection in Simultaneous Localization and Mapping (SLAM), multi-robot or multi-session mapping, and augmented reality. Robot localization can be achieved using different sensor modalities, with camera [1], [2], [3] and lidar-based approaches [4], [5], [6] the most commonly studied. While cameras offer a compact form

factor and are often low-cost, visual localization and mapping systems may struggle with lighting or seasonal changes. Conversely, lidar systems are physically bigger, have higher power consumption and increased cost — factors which pose challenges for their widespread deployment on mobile robots.

For any localization system, a prior representation of the scene—a prior map—is required. The prior map is usually built using the same sensor modality that will be used for later deployment. For example, autonomous industrial inspection has been demonstrated using legged platforms equipped with lidar [7], where a prior map is built by accumulating laser scans, and localization is performed against the map using methods such as ICP [8]. Similarly, purely visual localization methods often build a map using a structure-from-motion pipeline (SfM) such as COLMAP [9], and then localize hierarchically using visual place recognition followed by pose estimation methods such as perspective-n-points (PnP) [1].

Achieving cross-modal localization is typically a more challenging problem due to the inherent differences in sensing modalities and map representations. While lidar-based localization often utilizes point cloud maps, some applications may only have access to a colored mesh-based building model [10]. In contrast, visual localization methods rely on a database of 3D points with associated descriptors, specifically tailored to the feature detectors used during the map construction [3], [2]. Moreover, emerging scene representations such as NeRF [11] and Gaussian Splatting [12] are also dependent on the specifics of the original sensing setup. The variety of map representations poses challenges when applying existing localization approaches.

Our approach creates a unified map database for each of the three candidate map representations we study. The visual database is made up of RGB and depth images at known poses and is automatically synthesized from color point clouds, meshes, or NeRF maps. These maps are built using vision and lidar data acquired from either industrial-grade Terrestrial Laser Scanners (TLS) or hand-held mapping devices. Using this unified representation, we achieve scalable global localization resilient to scene changes and lighting variations with state-of-the-art visual localization components. An overview of the system operation is shown in Fig. 1.

The key contributions of our work include:

- A versatile and unified approach for global visual localization of a single camera using a database of synthetic RGB and depth images generated from point cloud, mesh, or NeRF 3D color maps.
- A strategy for automatically determining the poses within the color 3D maps from which the synthetic images should be generated.
- An extensive evaluation across both indoor and outdoor environments showcasing the effectiveness and robustness of our proposed method.
- A performance comparison between the aforementioned multi-modal localization system and two (purely) visual localization systems.

Our streamlined system operates on a mobile laptop at 1Hz — and is thus suitable for real-time operation.

II. Related Work

We define localization as the process of estimating the 6 DoF pose of a sensor within a 3D prior map or database. This is commonly achieved by means of a place recognition stage—which finds place candidates in the prior map—, and a registration step—that estimates a precise 6 DoF pose using the candidates. Given two sensing inputs of vision and lidar, the current literature can be categorized into four categories — based on sensing modality. These include (a) visual localization within a visual map, (b) lidar localization within a lidar prior map, (c) combined visual and lidar localization within a joint visual and lidar map, and (d) cross-modality visual localization within a lidar prior map or vice versa.

A. Visual Localization

Visual localization methods aim to obtain a 6 DoF pose by localizing against a map built from multiple images. In general, such maps are built using Structure-from-Motion (SfM) pipelines [9], which are able to recover the 3D structure from multiple views. Given that this reconstruction is accurate up to a scale factor, additional prior information, such as knowledge of a stereo baseline or direct depth sensing, is required for metric localization.

For localization, visual place recognition is implemented as an image retrieval problem. Methods such as FAB-MAP [13] and DBoW [14] solve the problem using local features, but more recent methods such as NetVLAD [15], PatchNetVLAD [16], EigenPlaces [17] rely entirely on neural architectures to obtain global descriptors for retrieval. Metric pose estimation is then performed via local feature matching using approaches such as SIFT [18], SuperPoint [19], or R2D2 [20]. This type of pipeline has been implemented in methods such as HLoc [1] and [2], which employ a hierarchical approach to large-scale visual localization.

In addition to the aforementioned methods, alternative approaches have been explored in visual localization. MeshLoc [21] shows it is feasible to construct a dense 3D mesh model from multi-view stereo point clouds and to render synthetic images via an OpenGL pipeline to create a visual database. In another study, [22] opts for a different retrieval approach. Rather than using a global descriptor for a query image, they detect instances of buildings in outdoor environments and retrieve the best match from a database of buildings. They show improvement in long-term and large-scale localization datasets over classic hierarchical frameworks.

B. Lidar Localization

Similar to visual localization, lidar localization also maintains a map database—in this case of 3D lidar scans. A hierarchical approach to retrieval and matching typically relies on generating global descriptors from each lidar scan. Early work started with handcrafted descriptors. Himstedt et al. [23] draw inspiration from the bag-of-words approach and transform 2D lidar scans into a histogram representation for place recognition. He et al. [24] project the 3D point cloud into multiple 2D planes, generating descriptors for each of

the planes based on point density, and combining them into a global descriptor. More recent work has moved towards learned global descriptors. Vidanapathirana et al. [5] and Uy et al. [6] both leverage deep learning networks and are able to produce robust global descriptors for loop closure in outdoor large-scale environments.

Recent work in lidar localization techniques has increasingly integrated semantics and extracted segments within the lidar point cloud. Kong et al. [25] introduce a place recognition approach based on a semantic graph, preserving the topological information of the point cloud. They show that by working on a semantic level, their method can be more robust to environmental changes. Aiming to address the same challenge, Segmap [26] introduces a segment-based map representation and generates descriptors for these segments using a learned network. These descriptors serve multiple purposes: localization, map reconstruction, and semantic information extraction. Building upon the segment concept, Locus [27] further extends the approach by combining descriptors of all segments with spatio-temporal high-order pooling to generate fixed-size global descriptors, which are effective for place recognition. This method was demonstrated to achieve robustness in challenging scenarios such as viewpoint changes and occlusions. However, there are unique challenges for indoor environments, as objects can clutter the room in confined spaces, leading to more occlusions. InstaLoc [28] demonstrated good performance for indoor localization by extracting object instances and generating a descriptor for each object before globally matching these objects to estimate the pose.

C. Combined Visual/Lidar Localization

In scenarios where multiple sensors are available on a mobile robotic platform, some approaches fuse information from both cameras and lidars for localization within a combined image and lidar database. Oneshot [29] customizes a network to generate descriptors from both lidar segments and their corresponding images from a camera. It estimates the sensor pose by extracting segments from a lidar scan and matching their descriptors to a database. The study demonstrates an enhanced retrieval rate when integrating visual information compared to lidar data alone. Bernreiter et al. [30] utilizes a spherical representation to generate descriptors for associated lidar scans and images. A notable advantage of their approach is the flexibility in accommodating different camera and lidar sensor configurations during both training and querying stages.

In recent work, Adafusion [31] introduces a method that employs adaptive weighting on image and lidar pairs to generate descriptors through an attention network. An adaptive weight allows for different contributions from each sensor, and the results show improved retrieval rates and robustness to changing environments. LC2 [32] is an alternative method wherein 2D images and 3D lidar point clouds are both converted into 2.5D depth images. Then, a network is trained to extract global descriptors from disparity and range images. By reducing modality discrepancy, their method can perform well in very different lighting conditions across multiple missions.

D. Cross-Modal Localization

For cross-modal localization, most previous work uses a camera to localize in a lidar map. This is the problem we also focus on in this paper. EdgeMap [33] extracts straight lines from the point cloud map to build a 3D edge map, and applies an edge filter for the camera image. Based on a particle filter, the likelihood of each pose hypothesis is calculated through the filtered camera image and the edge map. Building upon the line extraction idea, Yu et al. [34] extract lines from both the camera image and the lidar map. With a predicted pose from a visual odometry (VO) system, the camera pose is iteratively optimized by minimizing projection error based on 2D-3D line correspondence. They demonstrate that their method can greatly reduce drift by registering live images to the lidar map. However, these methods primarily apply to structured environments where lines and edges are prominent features.

Within the context of autonomous driving, researchers have also been developing localization approaches utilizing inexpensive and readily available camera sensors. Wolcott et al. [35] propose a method to generate a greyscale synthetic view from a mesh map through shading. By minimizing the normalized mutual information between live and synthetic images, this method estimates a pose within the lidar prior map. Similarly, Pascoe et al. [36] synthesize color images from a textured mesh and can further compensate for camera calibration changes.

Very recently, more improvements have been made to enhance cross-modal image-to-lidar registration methods. Approaches like Zuo et al. [37] register dense stereo depth to a lidar map to correct for visual odometry drift. Based on semantic scene understanding, [38] utilizes the point cloud map and its semantic labels to generate a prior semantic map. During runtime, semantic cues are extracted from live images to localize within the prior semantic map. Both methods fall within the context of camera-lidar registration, which require an initial guess—a relative pose prior. In our work, we focus on global localization without any pose prior information.

III. METHOD

A. Problem Definition

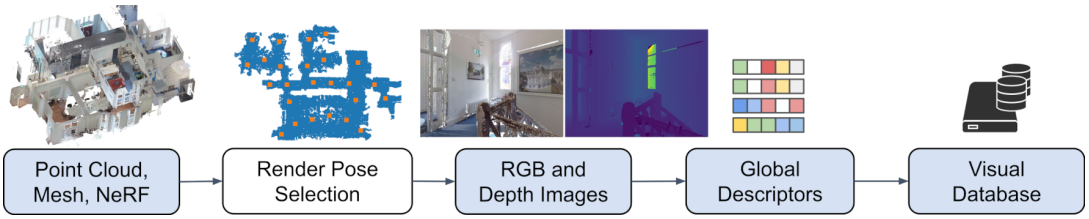
Our goal is to estimate the position and orientation of a single live RGB camera in a prior map, $\mathcal{M} = \{\mathcal{I}_1, \mathcal{D}_1, \mathcal{I}_2, \mathcal{D}_2, ..., \mathcal{I}_n, \mathcal{D}_n\}$, where $\mathcal{I}_i, \mathcal{D}_i$ is the i-th synthesized RGB and depth image pair, that are rendered at $\mathbf{T}_{\mathtt{WM}_i}$. The relevant frames are the earth-fixed world frame $\mathtt{W}$, and the map image frame $\mathtt{M}_i$.

Unless specified otherwise, the position $\mathbf{p}_{\mathtt{WM}_i}$ and orientation $\mathbf{R}_{\mathtt{WM}_i}$ of the map image are expressed in world coordinates, with $\mathbf{T}_{\mathtt{WM}} \in \mathrm{SE}(3)$ as the corresponding homogeneous transform.

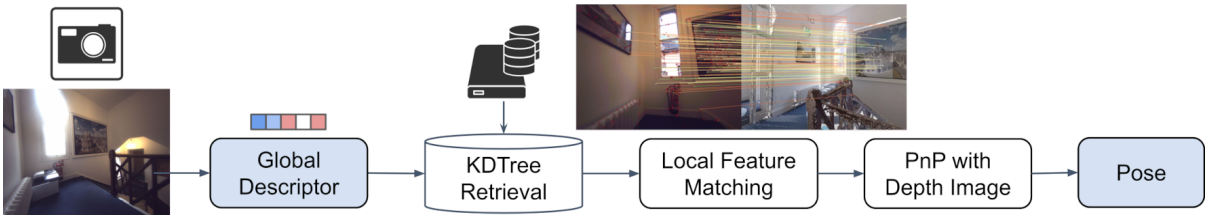
We aim to determine the pose of the live camera image $\mathcal{C}$ at a given time relative to a prior map image, defined as follows:

$$\mathbf{T}_{\mathtt{CM}_i} \triangleq [\mathbf{t}_i, \mathbf{R}_i] \in \mathrm{SO}(3) \times \mathbb{R}^3 \tag{1}$$

where $\mathbf{t}_i \in \mathbb{R}^3$ is the translation and $\mathbf{R}_i \in \mathrm{SO}(3)$ is the orientation of $\mathcal{C}$ relative to $\mathcal{M}_i$. Given $\mathbf{T}_{\mathtt{WM}_i}$ is known, we can find the live camera image pose in the world frame $\mathbf{T}_{\mathtt{WC}}$.



(a) Step 1: Constructing a visual database from a color 3D map.



(b) Step 2: A live camera image being matched against a rendered image to obtain a relative pose estimate

Fig. 2: Overview of the system showing localization of a camera in the 3D prior map. The blue boxes represent data, and the white boxes represent algorithms.

B. System Overview

We propose a method that integrates learning-based approaches with classical visual geometry to achieve image localization within a color lidar 3D map. The system design is illustrated in Fig. 2, which shows that the system uses a color 3D map (either point cloud, mesh or NeRF based) to generate synthetic RGB and depth images (Step 1). Using these images, we then construct a database of global descriptors. During live operation (Step 2), the system receives a query image and generates a descriptor. It then retrieves the closest match in the database. Learned local features are extracted from the images and matched. After retrieving the corresponding depth image, the camera pose can then be estimated based on the matched image features and their positions in the world frame.

C. Pose Selection when Rendering a Visual Database

A key step is how to decide the set of poses in the 3D prior map from which the artificial images should be synthesized. The goal is to render a reasonably low number of images (to keep the map database small) while achieving coverage across the point cloud or mesh. This step is crucial because it can directly affect both the database size and the available camera views, which influences the overall localization performance. Optimal place recognition can be compromised if the virtual camera is positioned in an unsuitable location—where the map is incomplete, or the camera does not visualize during localization. Our approach involves automatically identifying free space within the map and strategically selecting rendering poses along a "free path corridor".

To address this, we propose a straightforward yet effective method that utilizes geometry and image processing to select rendering poses.

For multi-floor building scans, we calculate normals using data from the TLS scans or the SLAM point cloud map. Planes are then extracted by filtering based on normal directions.

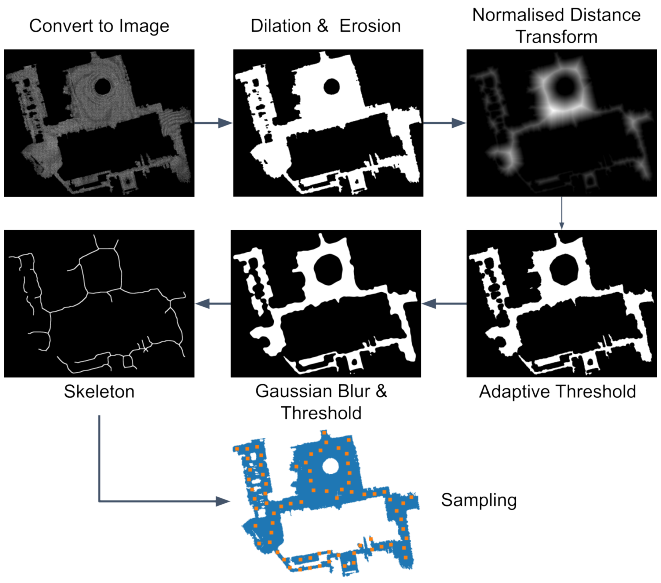


Fig. 3: Steps to establish a set of plausible rendering positions within a 3D map, which we call a "free path corridor". The map is split into floors and top-down images are rendered containing all the upward-facing points (selected using their normals). The orange points indicate the final selected positions.

Each floor plane can be isolated by constructing a histogram counting the number of points with normals facing upwards by point height. Subsequently, we convert each floor plane point cloud into a top-down image and subject this image to a sequence of image processing techniques to determine the reasonable walking area, as shown in Fig. 3. Dilation and erosion operations are applied to close small holes, followed by a normalized distance transform to identify the center of

all the free spaces. A smoothed version of the primary "free path corridor" is produced using thresholding and Gaussian blurring. Finally, a thinning step is implemented to derive a skeleton representing the path of all the walkable space, which is then sampled to obtain a set of rendering positions. We generate four RGB/depth camera image pairs for every position along the path — forward, back, and two side-facing views. This method is used for all of the 3D prior map representations: point clouds, meshes, and NeRF representations.

D. Generating Synthetic Images

We aim to generate synthetic RGB and depth image pairs rendered from selected poses using one of our three prior map representations. This section describes the construction of each map representation which in turn facilitates the generation of synthetic images for localization.

1) Point Clouds — using Terrestrial Lidar Scanning: Point clouds are point samples represented within a 3D coordinate system, sampling an object's or scene's external surface. Point clouds are commonly produced by technologies such as lidar scanning and TLS. In our study, we do not utilize point clouds derived from a SLAM system (such as FastLIO [39]), we instead use our TLS-generated point clouds as they are also used to generate precise ground truth maps with millimeter-level measurement accuracy. Further details about the TLS scanning process are provided in Sec. IV-C.

To generate RGB and depth images from the point cloud, we utilize the rendering capabilities of Open3D [40], which leverages an OpenGL framework[1]. Optimal renders are achieved by strategically positioning several light sources around the scene to facilitate specular reflections and shininess. Cameras are placed and maneuvered within the point cloud according to the predefined rendering poses (from the previous subsection). For each pose, 3D points are projected into the camera's field of view using a pinhole projection model, which allows the pixel size of each point to be adjusted.

Positioning the camera close to a wall causes gaps between points when using a small fixed pixel size, leading to a see-through effect in the rendered image, as illustrated on the left side of Fig. 4. To address this issue, the pixel size of rendered points is varied according to an inverse depth strategy , as detailed in Eq. (2). This approach ensures that the pixel size remains within the designated maximum and minimum splatting sizes. $\mathbf{z}_{(u,v)}$ represents the depth of a point in 3D space relative to the camera frame.

$$\rho_{(u,v)} = \min[\max[\frac{\rho_{max}}{\mathbf{z}_{(u,v)}}, \rho_{min}], \rho_{max}] \tag{2}$$

Due to a limitation imposed by OpenGL, adjusting individual point sizes dynamically is not feasible, as the parallel processing architecture requires a uniform point size for effective splatting. To circumvent this limitation, we applied Eq. (2) by rendering multiple frames, each with a point size determined by the maximum and minimum point depths within the scene. These frames are then combined to compose the scene's final RGB and depth image. This is illustrated in Fig. 5, showcasing how four images rendered with different point sizes are merged into a single RGB image.

[1]OpenGL, Khronos Group, https://www.opengl.org/

Fig. 4: Images from the left and right sides illustrate the before and after of rendered images from the point cloud when utilizing an adjustable point size based on the distance to the virtual camera.

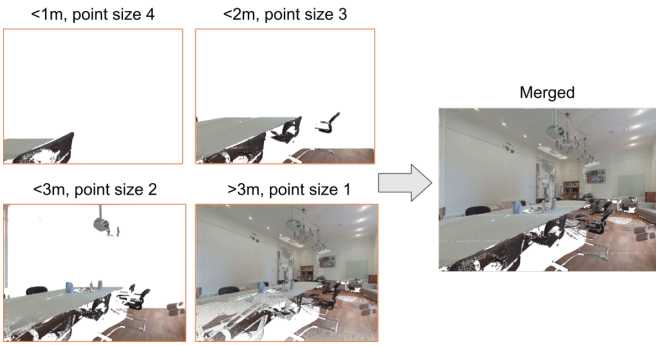


Fig. 5: Illustration of merging point cloud rendered images of different point sizes to get the final RGB and depth images.

2) Textured Meshes — using ImMesh: Meshes are widely used in robotics as dense representations for 3D scenes and objects. A mesh is a set of point vertices and the edges between them which together form a set of polygonal faces. Textured meshes are a popular method for representing static 3D scenes [41], [42], as they allow surfaces to be textured with color images, accurately capturing the scene's appearance.

In our experiments, we use a state-of-the-art mesh reconstruction framework, ImMesh [43], to reconstruct the triangular mesh of scenes captured using lidar scans. ImMesh initiates this process by estimating the pose of the lidar, followed by registering each scan to a global map. This map is partitioned into fixed-sized volumetric voxels. Mesh reconstruction is carried out online using an incremental, voxel-wise meshing algorithm. Initially, all points within a voxel are projected onto an estimated principal plane to reduce dimensionality. Subsequently, the triangular mesh is reconstructed through a series of voxel-wise operations: mesh pull, commit, and push steps. For more detailed information about the mesh

construction process, readers are referred to Section VI of [43]. ImMesh is designed to efficiently reconstruct the mesh of large-scale scenes in real time while maintaining high geometric accuracy.

However, further modifications were necessary to color and texture the mesh reconstruction. While ImMesh constructs a mesh of a scene with adjacent triangle facets connected by edges, these facets lack color information. To address this issue, we use color images captured by the visual camera to texture the facets. The camera poses and image exposure time can be estimated using R^{3}LIVE++ [44]. To enhance the smoothness and natural appearance of the mesh texture, for each triangle facet $\boldsymbol{\mathcal{T}}_i$, we blend images captured by the n nearest viewing cameras to form its texture $\mathbf{I}_i$ (in our experiment, n is set to 5). The pixel value $\mathbf{I}_i\left(u,v\right)$ at position (u,v) is blended as follows:

$$\mathbf{I}_i\left(u,v\right)=\frac{1}{\bar{\sigma}}\sum_{j=1}^{n}\left(\sigma_j\cdot\mathbf{I}_j\left(\boldsymbol{\pi}(\mathbf{R}_j,\mathbf{t}j,\mathbf{P})\right)\right) \tag{3}$$

$$\bar{\sigma}=\frac{1}{n}\sum_{j=1}^{n}\sigma_j,\quad \mathbf{P}=\mathbf{A}^{-1}(u_i,v_j)\in\mathbb{R}^3 \tag{4}$$

where $\boldsymbol{\pi}(\mathbf{R}_j,\mathbf{t}_j,\mathbf{P})$ is the camera projection function that projects a 3D point $\mathbf{P}$ (on the triangle facet $\boldsymbol{\mathcal{T}}_i$) to the image plane using the estimated camera pose $(\mathbf{R}_j,\mathbf{t}_j)$. σ_j denotes the estimated exposure for the j-th image, and $\mathbf{A}(\cdot)$ is the wrap transform that maps $\mathbf{P}$ on facet $\boldsymbol{\mathcal{T}}_i$ to (u,v) in texture coordinates, which can be written as $(u,v)=\mathbf{A}(\mathbf{P})$.

After reconstructing the textured mesh, we generate synthetic images of the captured scene. This process is accomplished by rendering the mesh into RGB and depth images using OpenGL with the predetermined rendering poses (the approach described in III-C).

3) Neural Radiance Fields — using SiLVR: NeRF is an emerging 3D representation that is particularly effective for photorealistic novel view synthesis. NeRF uses an implicit representation, usually a Multi-Layer Perceptron (MLP) [45], to model a radiance field. The scene is represented as a function whose inputs include a 3D location $\mathbf{x}=(x,y,z)$ and a 2D viewing direction (θ,ϕ). The output of the function is a color $\mathbf{c}=(r,g,b)$ and volume density σ.

The optimization of a NeRF is based on regulating the rendered image using the input image as the ground truth. Given the extrinsics and intrinsics of the camera, an image can be represented as a collection of rays $r(t)=o+td$. The expected color of each ray can be computed from the points sampled along the ray using volumetric integration with quadrature approximation [46], [47] as $\hat{c}_u=\sum_{i=0}^{N}w_ic_i$, where:

$$w_i=\exp\left(-\sum_{j=1}^{i-1}\delta_j\sigma_j\right)(1-\exp\left(-\delta_i\sigma_i\right)). \tag{5}$$

Early NeRF formulations were very computationally intensive — with training typically taking days until convergence. Explicit representations such as voxels [48], [49] and 3D Gaussians [50] have been shown to accelerate training and rendering substantially. Nerfstudio [51] is a well-supported open-source project which incorporates these representations. It also includes features that are effective when working with real-world data, namely scene contraction [52] which can better represent unbounded scenes, and appearance encoding [53] which can model per-image appearance changes including lighting conditions and weather.

As with traditional visual 3D reconstruction systems such as Multi-view Stereo, it is difficult for NeRF to reconstruct textureless surfaces or locations with limited multi-view input. However, accurate depth is important for retrieval and matching, as described in Sec. III-E. In addition, training a single NeRF for a large-scale scene is difficult due to relative small model size and limited computational hardware.

In this work we use our previous work on visual-lidar NeRF mapping called SiLVR (Scalable Lidar-Visual Reconstruction) [54]. SiLVR builds upon the vision-based Nerfstudio pipeline by adding strong geometry regularization using the lidar depth measurements. With volumetric rendering, one can render both the expected depth and surface normal. SiLVR uses depth regularization [55] to encourage the ray distribution to follow a narrow normal distribution by minimizing the KL-Divergence between them:

$$\mathcal{L}_{\text{Depth}}=\sum_{\mathbf{r}\in\mathcal{R}}\text{KL}[\mathcal{N}(\mathbf{D},\hat{\sigma})\|w(t)] \tag{6}$$

Additionally, SiLVR adds surface normal regularization to enhance the surface reconstruction. We use the same loss function as MonoSDF [56], but instead of learning-based surface normal prediction for supervision, we estimate surface normals directly using the lidar range image. The rendered surface normal is computed as the negative gradient of the density field and is supervised using the following loss function:

$$\mathcal{L}_{\text{Normal}}=\sum_{\mathbf{r}\in\mathcal{R}}\|\hat{N}(\mathbf{r})-\bar{N}(\mathbf{r})\|_1+\left\|1-\hat{N}(\mathbf{r})^\top\bar{N}(\mathbf{r})\right\|_1 \tag{7}$$

When training the NeRF model, there are several other considerations. Training large-scale scenes (such as Blenheim Palace in Fig. 1) usually leads to inferior results compared to a smaller scene. One reason for this is the limited model size — the representation power of our learnt model is determined partially by the number of parameters that can be optimised. A relatively small model trained with a large dataset can lead to under-fitting. Additionally, loading thousands of images from a large-scale scene into RAM memory is not always practical. Therefore, SiLVR adopts a submapping system to divide the scene into smaller overlapping submaps. Specifically, the submaps are created by applying Spectral Clustering [57] to divide the full global trajectory into smaller sections. The submaps overlap — with part of the global trajectory reused at the submap boundaries. This allows a smooth transition from one submap to another.

E. Retrieval and Matching

Upon completion of the synthetic image generation process and the subsequent creation of corresponding depth images, the resulting image database is then input into the NetVLAD

[15] network. This network is built with a convolutional neural network and a special VLAD layer. The central element of VLAD layer is inspired by the image representation technique known as the Vector of Locally Aggregated Descriptors. The network generates descriptors capable of accurately identifying the location of the query image, even amidst significant clutter (e.g., people, cars), variations in viewpoint, and stark differences in illumination, including day and night conditions. The descriptors of all the rendered RGB images are stored in the KDTree data structure for future indexing.

Upon receiving an incoming live camera image, our system initially undergoes a distortion correction process before being subjected to the same NetVLAD network. This step produces a descriptor for the live camera image. Afterward, the KD-Tree retrieves the virtual image that exhibits the closest match based on the descriptor values.

For the pair of matched cameras and virtual images obtained in the aforementioned step, local features are extracted using the SuperPoint algorithm. Due to the large domain gap between the synthetic-to-real images, we observe traditional, not learning-based feature detectors, such as SIFT [18], fail to detect common features. SuperPoint is first pre-trained on a synthetic dataset, boosting the domain adaptation performance. Then SuperPoint is further trained with a multi-scale, multi-transform augmentation, enabling self-supervised training of interest point detectors. This gives SuperPoint the properties of repeatable feature detection across different view angles.

Subsequently, these local features are matched through SuperGlue [58]. The SuperGlue network constitutes a Graph Neural Network coupled with an Optimal Matching layer, which is trained specifically to conduct matching on two sets of sparse image features.

Finally, with access to the matched local features and the depth image, the Perspective-n-Point (PnP)[59] algorithm is used to estimate the relative pose between the live camera image and the rendered reference image. Based on the known pose of the reference image, we can triangulate the live query image in the color 3D map. The pseudo algorithm describing the details of the matching process is shown in Algo. 1; note that we utilize the confidence score from the SuperGlue network to filter out bad matches.

Algorithm 1 Query pose estimation with reference image.

Input: camera intrinsic f_x, f_y, c_x, c_y, depth image D, reference image pose in world frame T_{WR}
Output: T_{WQ} (query image camera pose in world frame)

for $i = 1, 2, \ldots, N$ matched keypoint **do**
  $q_u, q_v \leftarrow$ query image keypoint[i]
  $r_u, r_v \leftarrow$ reference image keypoint[i]
  depth $P_z \leftarrow D[v][u]$
  $P_x \leftarrow (r_u - cx) * z/fx$
  $P_y \leftarrow (r_v - cy) * z/fy$
  if match confidence score $\omega > 0$ **then**
    3d points $\leftarrow [P_x, P_y, P_z]$
    image points $\leftarrow [q_u, q_v]$
  end if
end for
if image points ≤ 6 **then**
  T_{QR} , success $\leftarrow$ perspective N points with (3D points, image points)
  if success **then**
    Return $T_{WQ} = T_{WR} * T_{QR}^{-1}$
  end if
end if

F. Implementation Details

For NeRF 3D reconstruction, we train each SiLVR map over 150,000 iterations, with 4096 rays sampled from the training set in each iteration. This takes around 5 hours on average for all experiments on a single Nvidia RTX 4090 GPU.

For the localization system, the input RGB images have 720×540 pixels resolution, and we use off-the-shelf pre-trained networks. Specifically, we use a pre-trained NetVLAD backbone based on VGG16, trained on the Pitts30K [60] dataset. Additionally, the SuperGlue network we use is pre-trained on ScanNet data [61] for indoor scenes and MegaDepth data [62] for outdoor environments.

The localization part of the system (Step 2) is designed for real-time applications. Upon initialization, all rendered images are loaded, global descriptors are extracted for KDTree indexing, and local features are stored in memory. All algorithms are implemented on a Dell laptop with an Intel Core i7-9850H CPU running at 2.60GHz and a Nvidia Quadro T2000 GPU with 4GB of memory. The online localization pipeline processes each input image in approximately 0.5 seconds, with the following upper limits for each step: global descriptor generation (200ms), KDTree retrieval (1ms), local feature detection (60ms), local feature matching (100ms), and pose estimation (60ms). These performance metrics enable the system to comfortably operate at 1 Hz.

IV. Experiment Setup and Hardware

A. Dataset

We collected two recordings at each location: one was used to build the prior map and to render the database images, and the second was used for localization testing. The recordings were collected with a walking speed of around 1m/s. The test locations had the following distinct characteristics:

- ORI [indoor] was collected at the Oxford Robotics Institute. The ORI dataset encompasses indoor environments such as office rooms, staircases, and a kitchen. This dataset is characterized by faster camera rotation movements than the outdoor dataset with the content in the images changing quickly as the recording device moves from room to room.
- Math [outdoor, medium scale] was collected outside the Oxford Mathematics Institute on a summer's day. This dataset is characterized by substantial lighting variations, with transitions between areas of direct sunlight and shadows cast by the buildings. The presence of bushes, trees, and lawns increases the complexity of the image rendering.

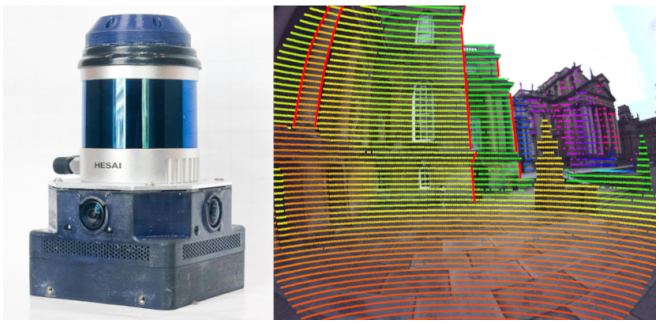


Fig. 6: Frontier multi-sensor unit consists of a 3D lidar, 3 orthogonal cameras and an IMU (left image). A fisheye image overlaid with lidar points (right image), demonstrating accurate intrinsic and extrinsic calibration (red lines are manually added to highlight the depth change).

- Blenheim [outdoor, large scale] was collected at Blenheim Palace, a 300-year-old country house in Woodstock, Oxfordshire. It captures walking sequences within the palace courtyard, partially extending into the palace's main hall. This dataset is challenging as images often contain significant portions of ground and sky, while the geometric symmetry of the courtyard further adds complexity.

The test recording, `ORI`, was collected across 7 office rooms spread over 2 floors with a total trajectory length of $125\,\mathrm{m}$. The `Math` trajectory is $451\,\mathrm{m}$ long, and the `Blenheim` trajectory is 386m long.

B. Hardware - Frontier

All experimental data was collected using a *Frontier*, Fig. 6, a self-developed multi-sensor unit consisting of a Sevensense Alphasense Multi-Camera kit with 3 orthogonal cameras of 1440×1080 pixels resolution. The device contains a Hesai QT lidar with 64 beams and 104.2° vertical field of view. Note that the Hesai lidar is used to build the mesh and NeRF maps and generate ground truth poses.

The extrinsic calibration between the lidar and camera is established following the methodology outlined in [63], ensuring sub-millimeter accuracy in translation and sub-degree precision in rotations. Accurate calibration is crucial, as shown in Fig. 6, particularly for accurately overlaying lidar scans onto color images when reconstructing color 3D maps.

C. Ground Truth Maps and Trajectories

To generate ground truth point cloud maps, we used a (professional grade) Leica RTC360 Terrestrial Laser Scanner (TLS) to scan the `Math` and `Blenheim` sites and an (entry-level) Leica BLK360[2] to scan the `ORI` site. Views of these maps are shown in the left section of Tab I. TLS scans are captured around $3\,\mathrm{m}$ apart for indoor `ORI` and $15\,\mathrm{m}$ to $20\,\mathrm{m}$ apart for outdoor `Math` and `Blenheim`. Note that ultra-dense distance between TLS scans is impractical due to the considerable time required to scan and process point clouds.

[2]https://leica-geosystems.com/products/laser-scanners

Each Hesai lidar scan was registered to the ground truth point cloud to obtain a 6 DoF pose. We refer readers to our previous paper, where we explain the details of the ground truth generation process in more detail [64], which we claim achieves centimeter precision.

These point clouds were also used to render the synthetic images for the point cloud map representation, as explained in Sec III-D1. Before rendering, we voxel-filtered the `ORI` point cloud to $5\,\mathrm{mm}$ resolution and the `Math` and `Blenheim` clouds to $1\,\mathrm{cm}$.

V. Experiment Results

A. Comparison of Rendered Images

First we present a set of illustrative examples of rendered images obtained using the TLS point cloud, NeRF, and mesh reconstruction pipelines. Fig. 7 illustrates examples drawn from the three datasets. Each pipeline has the ability to generate reasonably lifelike synthetic images. However, we note that each also exhibits distinct limitations.

For the indoor `ORI` dataset, object occlusion and the relatively sparse coverage of the TLS scanner results in some visible gaps in the point cloud, leading to white regions in the rendered images. Mesh reconstruction faces challenges in accurately representing smaller objects or intricate details, primarily due to the fixed size of the triangular faces. This method is more suitable for objects with planar surfaces than objects with complex profiles. For example, the handrails are difficult to reconstruct with a mesh, as shown in (f). Conversely, NeRF demonstrates its superiority in outdoor environments, successfully reconstructing fine details such as bicycle rails where mesh reconstruction struggles, as shown in (b). However, NeRF often fails to capture intricate ground floor patterns, as shown in (g) and (h), while mesh reconstruction and point cloud techniques, which employ direct image projection, can easily accomplish this.

B. Full System Evaluation

In this section, we quantify the localization performance of the three different rendering pipelines, benchmarking these against two entirely vision-based localization systems, as detailed in Tab. I. The systems we compare are HLoc and COLMAP, both based on Structure-from-Motion (SfM).

Tab. II presents a high-level analysis of the suitability of these systems for key robotics tasks. The SfM approaches are non-metric, meaning their pose estimates are not scaled to real-world dimensions. This limitation necessitates other information to be available to allow effective deployment in robotic applications. These methods are also not real-time, as the SfM bundle adjustment process would take a couple of hours to a day, depending on the image or map size.

In the following section, we will discuss the configurations of HLoc and COLMAP in our experiments and compare them with our methods of localizing using point cloud, mesh, and NeRF-based map representations. For a fair comparison, the primary performance metrics we use are *retrieval rate*, the proportion of images correctly retrieved as a percentage of the total number of queried images, and *localization rate*, the

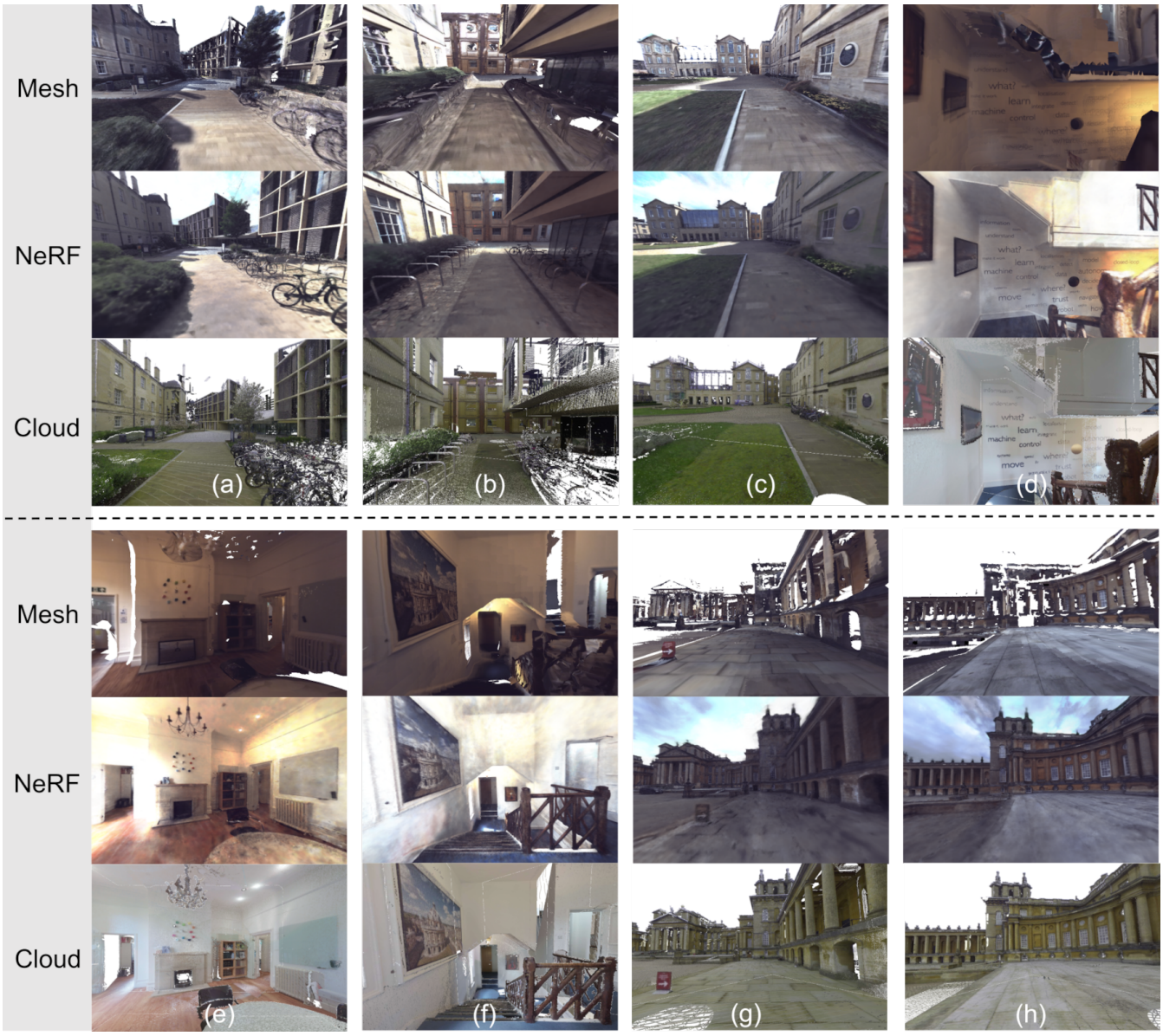


Fig. 7: A comparison of rendered images from mesh, NeRF, and point cloud in `Math` (a-c), `ORI` (d-f), and `Blenheim` (g-h).

proportion of images correctly retrieved and localized within a threshold as a percentage of the total number of queried images. As we are evaluating global localization systems, the threshold is within $1\,\mathrm{m}$ and $30°$ of ground truth poses for indoors `ORI`. For outdoor `Blenheim` and `Math`, the threshold is within $2\,\mathrm{m}$ and $30°$.

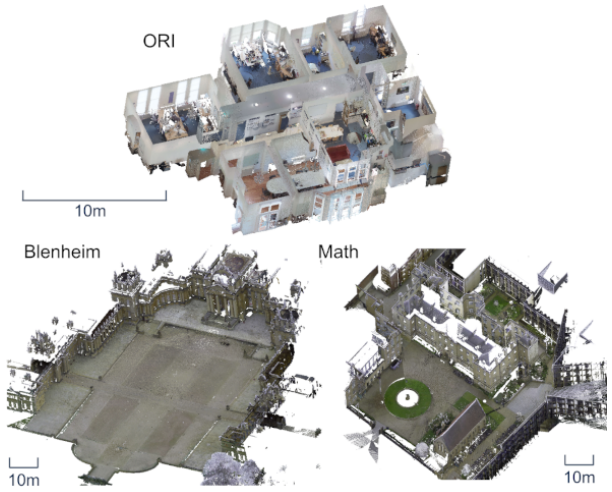
1) Our Methods – Cloud, Mesh and NeRF: For a fair comparison, we render an equal number of images for each dataset at the same locations. For `ORI`, render poses are $2\,\mathrm{m}$ apart, generating 130 images. Since `Math` and `Blenheim` cover larger areas than `ORI`, render poses are $4\,\mathrm{m}$ apart, and 300 and 210 images were produced, respectively.

When testing our approach of localizing with a database of rendered images, we found consistent performances for each of our three reconstruction techniques (point cloud, mesh, and NeRF) across three distinct test locations. On average, the localization rates for the point cloud and mesh approaches were $56\,\%$ and $58\,\%$, respectively, while the NeRF-based approach achieved the best performance at $72\,\%$. The relatively lower localization rates with point cloud and mesh systems can be attributed to the projection of color and textures from multiple camera views onto the 3D space, occasionally resulting in occluded regions and color inconsistencies.

Fig. 8 illustrates the performance of pose estimation for query images localized in `Math` against a NeRF-rendered image database. The ground truth pose trajectory is drawn in blue, the estimated poses in orange dots, and the estimated poses that exceeded the threshold are highlighted in purple.

The notable advantage of point cloud and mesh mapping systems is their ability to generate color 3D maps online. The point cloud and mesh results are available immediately following a SLAM run, making them useful for real-time

TABLE I: Performance comparison of the different localization methods. (”DB size” is the number of images in the database. ”Ret. Rate” refers to the retrieval rate. ”Loc. Rate” means the localization rate. COLMAP (*) required all the images from all three cameras to reconstruct the prior map accurately for the indoor ORI recording.)



Dataset	Method		DB Size	Ret. Rate	Loc. Rate
ORI	HLoc	Sparse	130	N/A	Fail
		Dense	254	N/A	86
	COLMAP	Dense	3020*	97	**96***
	Ours	Cloud	130	68	54
		Mesh		81	61
		NeRF		86	**67**
Math	HLoc	Sparse	503	N/A	20
		Dense	1019	N/A	39
	COLMAP	Dense	1569	66	**61**
	Ours	Cloud	300	89	56
		Mesh		85	57
		NeRF		97	**75**
Blenheim	HLoc	Sparse	210	N/A	42
		Dense	449	N/A	55
	COLMAP	Dense	449	N/A	**78**
	Ours	Cloud	210	77	57
		Mesh		77	56
		NeRF		86	**73**

TABLE II: Comparative analysis of localization methods for use in different robotics tasks. ✓ means suitable, ✓̸ means partly suitable, and ✗ means unsuitable.

Method	Metric Pose	Real Time	Planning
HLoc COLMAP	✓̸	✗	✓̸
Ours-Cloud Ours-NeRF Ours-Mesh	✓	✓	✓

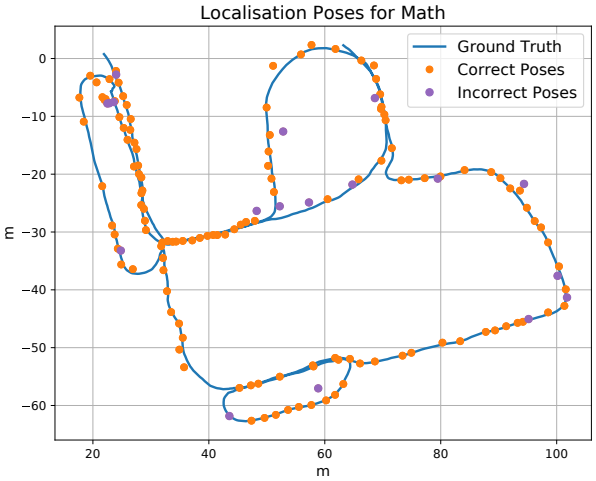


Fig. 8: Estimated poses in `Math` from the NeRF synthesized image database. The purple dots show estimated poses exceeding the 2 m and 30° threshold compared to the ground truth poses.

applications that cannot support rendering delays. Conversely, NeRF reconstruction offers the most photorealistic rendering outcomes but requires training after data collection. Additionally, synthetic image generation with the NeRF reconstruction is executed through an MLP inference step, which also requires a slightly longer time (e.g. $\sim$ 1 s per image of resolution 720×540 on a Nvidia RTX 4090 GPU). In contrast, the OpenGL-based rendering utilized in the point cloud and mesh method takes a few milliseconds per image.

2) HLoc: HLoc [1] is a 6 DoF visual localization toolkit. It employs a hierarchical localization approach that integrates image retrieval and feature-matching techniques. We first configured HLoc to build an SfM model using all database images and then used it to register query images to the SfM model. For an objective comparison, we used the same pre-trained networks as used in our localization method — NetVLAD for image retrieval, SuperPoint for local feature detection, and SuperGlue for feature matching.

For HLoc and COLMAP, the experiments started with the same number of database images as our system. We increased the number of images until their performance improvement plateaued. In general, visual localization systems require a larger number of images to construct an accurate SFM.

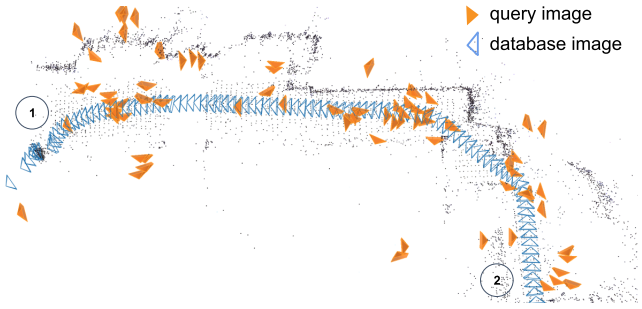
To obtain query image pose estimates (up to the scale metric), we first refine the pose estimates by eliminating any erroneous matches and then use Evo[3] to align the query image poses with the ground truth poses to recover the scaling factor and the alignment transformation. Finally, we apply this scaling factor to all estimated poses. We can then calculate the localization rate.

During the initial SfM reconstruction step, HLoc performed well with indoor data but encountered difficulties in accurately constructing the larger outdoor environments of `Math` and `Blenheim`. We found that constructing a robust SfM solution

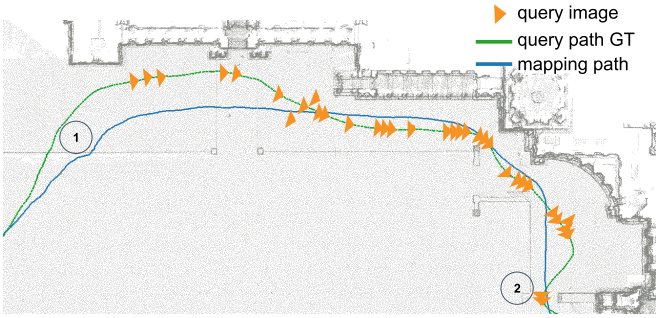
[3]https://github.com/MichaelGrupp/evo

TABLE III: Localization when traveling in the opposite direction to the mapping trajectory with novel view synthesis.

Method	Camera	Images	Ret. Rate %	Loc. Rate %
COLMAP	front	98	N/A	0
HLoc	front	98	N/A	4
NeRF	front	30	10	7
Mesh	front	30	37	**33**
COLMAP	front + side	196	N/A	26
HLoc	front + side	196	N/A	39
NeRF	front + side	30	88	**75**
Mesh	front + side	-	-	-
Point Cloud	omni	30	98	**88**



(a) HLoc map: Although the SfM reconstruction is accurate, only four camera views could be correctly localized around the bend.



(b) Mesh map: 30 out of 90 query images could be correctly localized.

Fig. 9: Localizing when traveling in a direction opposite to the mapping phase. The mapping direction (blue) is from 1 to 2, and the query direction (green) is from 2 to 1. Orange triangles represent the localized camera view results.

requires a dense set of images with substantial overlap and minimal lighting changes. For ORI, an image database of 130 images (as used for map rendering pipelines) was insufficient as the images were too sparse for the SfM solver to achieve sufficient overlap. Increasing the number of images (to 254 and later 500) enabled the creation of a complete reconstruction, albeit with some observable drift in the staircases connecting the two floors of the building. For MATH, the best results were achieved with 1019 images, and increasing the database (to 2040 images) did not noticeably improve the SfM model. Lastly, for Blenheim, HLoc achieved its best performance with 449 images, and doubling the image count did not improve the overall SFM accuracy.

Overall, HLoc worked well in the indoor setting, but the reconstruction was less successful in large outdoor scenes.

3) COLMAP: This system [9] is a widely-used SfM and Multi-View Stereo pipeline with graphical and command-line interfaces. It can construct visual maps using ordered and unordered image collections. In our experiments, we adopted the recommended approach of utilizing SIFT as the feature detector and a vocabulary tree for image retrieval. Our results show that COLMAP requires a higher number of database images than HLoc. It achieves its best performance when incorporating images from the three cameras of the Frontier (front, left, and right facing) within the dataset. This was particularly the case for the indoor ORI dataset, where the camera motion is more abrupt/jerky. For ORI, COLMAP required all images from all three cameras — facing left, forward, and right, a total of 3020 images.

Once the SfM solution is accurately estimated, localizing new images is straightforward, especially if a query image closely resembles the database images. For MATH, the query and map images were captured in quite different lighting conditions. Hence, the performance of image retrieval is lower with COLMAP as it uses the non-learning-based image retrieval method. Across all three locations, COLMAP gave good localization performance, but constructing the SFM solution takes double the amount of time compared to HLoc, e.g., for MATH, COLMAP takes around 4.5 hours with 1569 images, and HLoc takes 2 hours for 1019 images.

C. Localization in Directions Unseen during Mapping

Our method has the powerful capability of generating images from any viewpoint. This is very useful when attempting to localize when traveling in directions opposite to that used to create the map. For instance, in the scenario where the map is constructed by traversing from left to right, as shown in Fig. 9, we can synthesize images from all four orthogonal directions. This capability facilitates localization even when traveling in the reverse direction, a challenge often encountered in conventional visual localization methodologies, which typically requires storing images from both travel directions in the database.

The initial recording was used to reconstruct a mapping database with only the front-facing camera, as shown in Fig. 9a. The map was generated by traversing from point 1 to point 2, where 98 images were captured (illustrated in blue). We then attempted to localize to the map during a reverse traversal from points 2 to 1, with the results depicted using orange-colored camera views. Of the 87 query images, only 4 were successfully localized (when turning around the corner) due to minimal overlap in the image views. A similar pattern was observed with COLMAP; although the localization module was given a perfect reconstruction, it failed to localize any query images. HLoc performed slightly better than COLMAP due to its learning-based feature matching approach.

NeRF also demonstrated suboptimal performance in generating images from the reverse perspective, likely due to insufficient training views. In contrast, the mesh-base pipeline demonstrated good performance. It could effectively generate rendered images facing in the un-mapped direction (albeit with

(a) Scenes in the original maps.

(b) Changes after three months.

Fig. 10: Ablation study: assessing changes in `Math` and `ORI` 2-3 months apart. The top row displays images from the original dataset, while the bottom row shows corresponding images captured months later which reflect the typical movement of objects within the scene. The images on the right-half are from `Math`, while the bottom right ones were captured on a dark day. Please note that some images have been brightened for easier interpretation here.

imperfections resulting in holes in the rendered images). Despite these visual limitations, mesh-rendered images achieved a localization rate of approximately 33 %.

In a second experiment, we expanded the mapping database to include images from two cameras, the side and front camera. As expected, this enhanced the performance for COLMAP and HLoc, with side camera views available for reverse image matching. NeRF exhibited much better localization performance, matching 75% of the images. Note that the mesh reconstruction implementation only works with one camera.

To provide a broader comparison, we show that point clouds captured with an omnidirectional camera (both front and backward facing) can achieve a very high localization rate.

To conclude, this section demonstrates that our system can generate synthetic images in novel views, which aids the localization in a direction that is unseen during the mapping. In particular, mesh-based map representation can render images for localization even when there are very limited views of the environment, while NeRF can render better images for localization with a moderate number of views.

D. Ablation Study: Changes in Environment

When using a prior map for localization, one would expect that changes to the scene would affect algorithm performance. In indoor environments, furniture and decorations are often reconfigured. Meanwhile, outdoor environments are affected by seasonal and lighting changes, while dynamic objects such as parked cars or bicycles may be added or removed. Our method must be resilient to such changes.

To demonstrate the robustness of our approach to scene change, we tested our camera localization system with data collected 2-3 months after the initial construction of the maps. The right-hand side of Fig. 10 shows two images from `MATH` with changes (shown with red circles) in the vegetation color, the location of parked cars, and the configuration of the bicycle stand. On the left-hand side, images from `ORI` show rearranged furniture and equipment.

TABLE IV: Localization performance amid scene changes in prior maps.

Image Source	Dataset	Time	Ret. Rate	Loc. Rate
Point Cloud	ORI	Original	68	**54**
		3 Months later	**70**	48
NeRF	Math	Original	**97**	**75**
		2 Months later	91	51
Mesh	Math	Original	**85**	**56**
		2 Months later	85	49

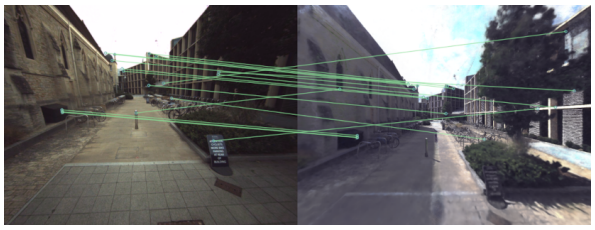
In both locations, lighting variation between the 2-3 month period was clearly observed. The detailed results of localization performance are presented in Tab. IV. All three map representations show a small drop in retrieval and location rates. However, they can still localize around 50 % of query images, demonstrating the robustness of matching against one unified synthetic image database, aided by learning-based visual feature detector SuperPoint and matcher SuperGlue.

E. Ablation Study: Feature Detection and Matching

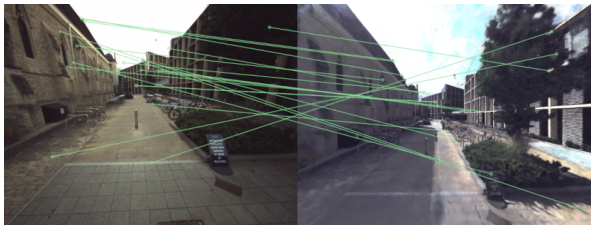
As shown in Fig. 7, there is a significant reality gap between images captured by real cameras and synthetically rendered images. Each map representation has its own limitations when rendering the synthetic images. Images rendered from TLS

TABLE V: Average performance of different feature detector combinations for matching a set of ten real and synthetic image pairs. An example is shown in Fig. 11. Note that many of the features matched for Akaze and SIFT are incorrect.

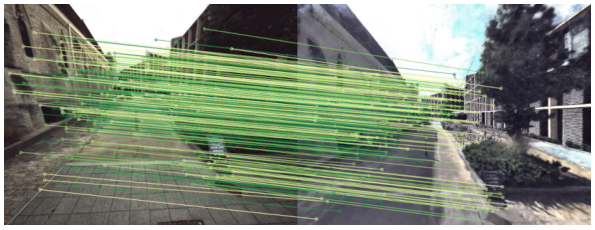
Method	Feature Detected	Matched
Akaze [65] + NN	410	18
SIFT [18] + KNN	841	21
R2D2 [20] + NN	1024	69
D2Net [66] + NN	2932	132
SuperPoint [19] + SuperGlue [58]	1024	201



(a) Detector: OpenCV Akaze, Matcher: Nearest Neighbor



(b) Detector: OpenCV SIFT, Matcher: Flann KNN



(c) Detector: SuperPoint, Matcher: SuperGlue

Fig. 11: Example of matching a camera image to a NeRF rendered image with different feature detectors.

point clouds often lack photorealism, exhibit a surreal quality, and have visible gaps corresponding to unscanned regions. Conversely, mesh-rendered images can contain areas with fragmented reconstruction and struggle to faithfully represent small and intricate geometric structures. For the NeRF-generated images, when the rendering poses differ largely from the training data viewpoints, one can often see a fog-like effect in the generated images. Perhaps learning-based feature detectors and matching techniques could help to bridge the domain gap between synthetic and real images and alleviate these issues.

In this section, we conduct a comparative analysis of different feature detectors and matchers using a representative dataset of ten image pairs drawn from across the three datasets. One image is a live camera image, and the other is a rendered image from either the point cloud, mesh, or NeRF pipelines. Fig. 11 shows one scene with common challenges: vegetation, low-textured ground, and thin bike rails. Fig. 11c shows how learning-based approaches such as SuperPoint and SuperGlue can effectively identify and match numerous correct features. In contrast, traditional feature detectors like Akaze (Fig. 11a) and SIFT (Fig. 11b) perform poorly in this scenario.

Tab. V presents quantitative results for this test, comparing several learning-based and classic feature detection and matching techniques. Among the learning-based feature detectors, SuperPoint, R2D2, and D2Net show promising performance by matching many features. However, traditional methods like SIFT and Akaze match fewer features, many of which are incorrect. Note that R2D2 produces descriptors of size 128, and D2Net generates descriptors of size 512. Given that the SuperGlue matcher is pre-trained with descriptors of size 256, we use the Nearest Neighbour (NN) matcher with a ratio threshold test for R2D2 and D2Net. SuperPoint and R2D2 are capped at 1024 features, while D2Net, a dense image feature extractor, naturally yields more feature points.

In summary, leveraging learning-based feature detectors and matching techniques is crucial to mitigate the domain gap between camera-captured and synthetic images, enabling robust feature detection and matching across diverse image representations.

VI. Conclusions and Future Work

A. Conclusions

In this study, we introduced a cross-modal localization system tailored to the task of localizing a single camera image within a color 3D map. We demonstrated a pipeline to construct 3D prior maps using three distinct representations: point clouds, meshes, and NeRF. Each representation could then be used to synthesize RGB and depth images. Utilizing the color 3D map as input, our visual localization pipeline estimates the camera pose of a query image through a retrieval and matching process, leveraging learning-based descriptors and feature detectors. We conducted a comprehensive analysis of localization performance across these representations and discussed their respective merits. Additionally, we offered a benchmark comparison with two purely vision-based localization systems to situate our results within the wider field of visual localization. Notably, both point cloud and mesh representations achieve a localization accuracy of $55\,\%$ for query images, while the NeRF representation surpasses these, achieving a localization rate of $72\,\%$.

B. Limitation and Future Work

1) Scene Change: While we have studied the effect of scene changes within the context of a 3D color map, we acknowledge that there would likely be a performance decrease during gradual scene transitions over a period of months and years. In future research, we aim to devise a detection algorithm capable of promptly identifying these changes in real time, thus facilitating the identification of out-of-date regions of the map. Additionally, in scenarios where the device is equipped with a lidar sensor, we propose to update the

existing map following a remapping process to ensure its continued accuracy and relevance.

2) Synthetic Image: All three map representations encounter specific challenges in accurately rendering scenes with minimal texture details. This difficulty is particularly notable in 3D mapping and reconstruction, where effectively capturing the RGB texture details of ground and vegetation proves challenging. Furthermore, during the online localization of our system, images that predominantly contain low-texture objects often yield insufficient features for matching, resulting in inaccurate pose estimations. Addressing these challenges remains a significant area for improvement.

REFERENCES

[1] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From Coarse to Fine: Robust hierarchical localization at large scale," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] T. Sattler, B. Leibe, and L. Kobbelt, "Efficient & effective prioritized matching for large-scale image-based localization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1744–1756, 2017.

[3] B. Zeisl, T. Sattler, and M. Pollefeys, "Camera pose voting for large-scale image-based localization," in *Intl. Conf. on Computer Vision (ICCV)*, 2015, pp. 2704–2712.

[4] G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[5] K. Vidanapathirana, M. Ramezani, P. Moghadam, S. Sridharan, and C. Fookes, "LoGG3D-Net: Locally guided global descriptor learning for 3D place recognition," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 2215–2221.

[6] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "ANYmal in the Field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics*. Springer Singapore, 2021, pp. 247–260.

[8] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[9] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] H. Yin, Z. Lin, and J. K. Yeoh, "Semantic localization on BIM-generated maps using a 3D lidar sensor," *Automation in Construction*, vol. 146, p. 104641, 2023.

[11] J. Liu, Q. Nie, Y. Liu, and C. Wang, "NeRF-Loc: Visual localization with conditional neural radiance field," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2023, pp. 9385–9392.

[12] H. Matsuki, R. Murai, P. H. J. Kelly, and A. J. Davison, "Gaussian Splatting SLAM," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Intl. J. of Robot. Res.*, vol. 27, no. 6, pp. 647–665, 2008.

[14] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[15] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, 2018.

[16] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14 141–14 152.

[17] G. Berton, G. Trivigno, B. Caputo, and C. Masone, "EigenPlaces: Training viewpoint robust models for visual place recognition," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, October 2023, pp. 11 080–11 090.

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Intl. J. of Computer Vision*, vol. 60, pp. 91–110, 2004.

[19] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 224–236.

[20] J. Revaud, P. Weinzaepfel, C. R. de Souza, and M. Humenberger, "R2D2: Repeatable and reliable detector and descriptor," in *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*, 2019.

[21] V. Panek, Z. Kukelova, and T. Sattler, "MeshLoc: Mesh-based visual localization," in *Eur. Conf. on Computer Vision (ECCV)*, 2022.

[22] F. Xue, I. Budvytis, D. O. Reino, and R. Cipolla, "Efficient large-scale localization by global instance recognition," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 327–17 336.

[23] M. Himstedt, J. Frost, S. Hellbach, H.-J. Böhme, and E. Maehle, "Large scale place recognition in 2D lidar scans using geometrical landmark relations," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 5030–5035.

[24] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 231–237.

[25] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3D point clouds," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 8216–8223.

[26] R. Dubé, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *Intl. J. of Robot. Res.*, vol. 39, no. 2-3, pp. 339–355, 2020.

[27] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, and C. Fookes, "Locus: Lidar-based place recognition using spatiotemporal higher-order pooling," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 5075–5081.

[28] L. Zhang, S. Tejaswi Digumarti, G. Tinchev, and M. Fallon, "InstaLoc: One-shot global lidar localisation in indoor environments through instance learning," in *Robotics: Science and Systems (RSS)*, 2023.

[29] S. Ratz, M. Dymczyk, R. Y. Siegwart, and R. Dubé, "OneShot Global Localization: Instant lidar-visual pose estimation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 5415–5421.

[30] L. Bernreiter, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "Spherical multi-modal place recognition for heterogeneous sensor systems," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1743–1750.

[31] H. Lai, P. Yin, and S. Scherer, "AdaFusion: Visual-lidar fusion with adaptive weights for place recognition," *IEEE Robot. Autom. Lett. (RA-L)*, vol. 7, no. 4, pp. 12 038–12 045, 2022.

[32] A. J. Lee, S. Song, H. Lim, W. Lee, and H. Myung, "$(LC)^2$: Lidar-camera loop constraints for cross-modal place recognition," *IEEE Robot. Autom. Lett. (RA-L)*, vol. 8, no. 6, 2023.

[33] P. Borges, R. Zlot, M. Bosse, S. Nuske, and A. Tews, "Vision-based localization using an edge map extracted from 3D laser range data," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 4902–4909.

[34] H. Yu, W. Zhen, W. Yang, J. Zhang, and S. Scherer, "Monocular camera localization in prior lidar maps with 2D-3D line correspondences," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pp. 4588–4594, 2020.

[35] R. W. Wolcott and R. M. Eustice, "Visual localization within lidar maps for automated urban driving," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 176–183.

[36] G. Pascoe, W. Maddern, and P. Newman, "Direct visual localisation and calibration for road vehicles in changing city environments," in *IEEE Intl. Conf. on Computer Vision Workshop (ICCVW)*, 2015, pp. 98–105.

[37] X. Zuo, W. Ye, Y. Yang, R. Zheng, T. Vidal-Calleja, G. Huang, and Y. Liu, "Multimodal localization: Stereo over LiDAR map," *J. Field Robot.*, vol. 37, no. 6, pp. 1003–1026, 2020.

[38] S. Liang, Y. Zhang, R. Tian, D. Zhu, L. Yang, and Z. Cao, "SemLoc: Accurate and robust visual localization with semantic and structural constraints from prior maps," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 4135–4141.

[39] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Trans. Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[40] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[41] J. Arvo, *Graphics Gems II*. Morgan Kaufmann Publishers Inc., 1991.

[42] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy, *Polygon Mesh Processing*. A K Peters, 2010.

[43] J. Lin, C. Yuan, Y. Cai, H. Li, Y. Ren, Y. Zou, X. Hong, and F. Zhang, "ImMesh: An immediate lidar localization and meshing framework," *IEEE Trans. Robotics*, 2023.

[44] J. Lin and F. Zhang, "R^3LIVE++: a robust, real-time, radiance reconstruction package with a tightly-coupled lidar-inertial-visual state estimator," *arXiv preprint arXiv:2209.03666*, 2022.

[45] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *Eur. Conf. on Computer Vision (ECCV)*, 2020.

[46] N. Max, "Optical models for direct volume rendering," *IEEE Trans. on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[47] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *SIGGRAPH*, vol. 18, no. 3, pp. 165–174, 1984.

[48] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5501–5510.

[49] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," in *SIGGRAPH*, 2022.

[50] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian splatting for real-time radiance field rendering," *ACM Trans. on Graphics*, vol. 42, no. 4, Jul 2023.

[51] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *SIGGRAPH*, 2023.

[52] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022.

[53] R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth, "NeRF in the Wild: Neural radiance fields for unconstrained photo collections," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[54] Y. Tao, Y. Bhalgat, L. F. T. Fu, M. Mattamala, N. Chebrolu, and M. Fallon, "SiLVR: Scalable lidar-visual reconstruction with neural radiance fields for robotic inspection," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2024.

[55] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 882–12 891.

[56] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[57] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[58] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neuralnetworks," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2020.

[59] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: A hands-on survey," *IEEE Trans. on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.

[60] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual place recognition with repetitive structures," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.

[61] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2018.

[63] L. F. T. Fu, N. Chebrolu, and M. Fallon, "Extrinsic calibration of camera to lidar using a differentiable checkerboard model," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023, pp. 1825–1831.

[64] L. Zhang, M. Camurri, D. Wisth, and M. Fallon, "Multi-Camera LiDAR Inertial Extension to the Newer College Dataset," *arXiv:2112.08854*, 2021.

[65] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *British Machine Vision Conf. (BMVC)*, 2013.

[66] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: A trainable cnn for joint detection and description of local features," in *IEEE Intl. Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019.

Lintong Zhang (Graduate Student Member, IEEE) received an M.Eng. degree in Electrical and Electronics Engineering from Imperial College London, UK, in 2015. From 2015 - 2020, he worked as a robotic engineer at Dyson and Oxa on applications such as domestic robots and autonomous vehicles. He is currently pursuing the DPhil degree in Engineering Science from the University of Oxford, UK. His research interests include probabilistic and machine-learning methods for localization, mapping, and scene understanding.

Yifu Tao (Graduate Student Member, IEEE) received an M.Eng. degree in Engineering Science from the University of Oxford, UK, in 2020. He is currently pursuing the DPhil degree in Engineering Science from the University of Oxford, UK. His research interests include 3D reconstruction using visual and lidar sensors and deep learning methods.

Jiarong Lin (Graduate Student Member, IEEE) received a B.S. degree in Optical Information Science and Technology from the University of Electronic Science and Technology of China (UESTC) in 2015. He is currently a Ph.D. candidate in the Department of Mechanical Engineering, the University of Hong Kong (HKU), Hong Kong, China. His research interests include light detection, and ranging (lidar) mapping and sensor fusion.

Fu Zhang (Member, IEEE) received the B.E. degree in automation from the University of Science and Technology of China (USTC), Hefei, China, in 2011 and the Ph.D. degree in controls from the University of California at Berkeley, Berkeley, CA, USA, in 2015. He joined the Department of Mechanical Engineering, University of Hong Kong (HKU), Hong Kong, as an Assistant Professor in 2018. His current research interests include robotics and controls, with a focus on UAV design, navigation, control, and LiDAR-based SLAM.

Maurice Fallon (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from University College Dublin, Dublin, Ireland, in 2004 and the Ph.D. degree in acoustic source tracking from the University of Cambridge, Cambridge, U.K., in 2008. From 2008 to 2012, he was a Postdoc and a Research Scientist with MIT Marine Robotics Group working on SLAM. Later, he was the Perception Lead of MIT's team in the DARPA Robotics Challenge. Since 2017, he has been a Royal Society University Research Fellow and an Associate Professor with the University of Oxford, Oxford, U.K. He leads the Dynamic Robot Systems Group, Oxford Robotics Institute. His research interests include probabilistic methods for localization, mapping, multisensor fusion, and robot navigation.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Localization of a Single Camera in 3D Maps: A Comparison Between Point Cloud, Mesh, and NeRF Representations
Publication Status	☐ Published ☐ Accepted for Publication x Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Lintong Zhang, Yifu Tao, Jiarong Lin, and Maurice Fallon, "Localization of a Single Camera in 3D Maps: A Comparison Between Point Cloud, Mesh, and NeRF Representations", *Robotics: Science and Systems*, 2023.

Student Confirmation

Student Name:	Lintong Zhang		
Contribution to the Paper	• Contributed to development of the idea • Implemented novel contributions mentioned in the paper • Executed closed-loop field experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	April 16, 2024

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Maurice Fallon		
Supervisor comments Lintong was the lead instigator, implementer and researcher on this project and wrote the bulk of the paper.		
Signature 	Date	May 03, 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

6.2 Discussion

In this chapter, we introduce a unified localisation pipeline that can be used to produce a visual database of RGB and depth images at known poses from different types of map representation. This approach enables effective visual localisation across diverse map representations, including point clouds, meshes, and NeRF. We have demonstrated the advantages of our system over traditional visual localisation methods and found that NeRF-based rendering achieves the highest level of localisation performance.

Although our system has undergone extensive ablation studies, further evaluation could be beneficial, particularly regarding how the quantity of synthesised database images influences localisation rates.

One notable limitation encountered is that some synthesised images do not contribute effectively to localisation due to their poor quality. For instance, images rendered near the boundary of the map often capture an empty view looking out of the mapped space. Additionally, some images capture scenes within the map that are low in texture or lack distinctive features, leading to poor image retrieval and feature matching. To address this, we could develop an algorithm to identify and eliminate such ineffective images and instead render new images in more suitable areas of the map. Currently, approximately a quarter of rendered images exhibit these issues, leading to inefficiencies and unnecessary memory usage in the database.

6.2.1 Time Complexity

During online localisation, a global descriptor is produced for each incoming image and the nearest neighbour is retrieved with a KDTree data structure. Similar to the previous chapter, the complexity for query is $\mathcal{O}(\log n)$, n being the number of database images. As offline databases grow with the number of rendered images, the complexity is bounded by the same $\mathcal{O}(\log n)$ time. However, the time taken for KDtree retrieval (1ms) is insignificant compared to the image descriptor generation (200ms) and local feature matching (100ms), as mentioned in Sec. III, F. Implementation details.

6.2.2 Pose Confidence

In the current system design, we try to reduce false positives and achieve high precision in pose estimations, especially when the sensor is outside the prior map. We use the following criteria to help reduce false positives. Firstly, a retrieved rendered image is rejected if the distance between the global descriptors is too large. Secondly, there has to be a sufficient number of matched local features between the query and database image. Thirdly, SuperGlue gives a confidence score for each pair of matched local features. The matched image is rejected if too many local features have a low confidence score.

In terms of the localization rate in the experiments presented in Section V, we acknowledge that it is harder to objectively define a success criteria for this problem compared to the previous chapter. Following the approach used in other evaluations of visual localization systems, we have chosen a success criteria within 1 or 2 meters and 30 degrees of the ground truth poses.

As a potential avenue for future work, it may be beneficial to incorporate a confidence value or covariance in pose estimation. Several metrics could be considered when constructing an uncertainty value: the descriptor distance between the query and matched rendered images, the number of matched local features and the confidence score for each feature, as well as the optimisation residual from the final step of the PnP solver.

7
SLAM Dataset in Challenging Environments

As SLAM algorithms become more accurate and evolve to support real-life applications, the benchmarks used to evaluate them must also advance. However, many existing datasets do not present sufficiently challenging scenarios to effectively distinguish between top SLAM approaches. There is a recognised need for datasets that utilise the latest sensors to benchmark multi-modal SLAM accuracy and robustness across diverse challenges.

In our efforts to develop and test the odometry algorithm detailed in Chapter 4, we identified a shortage of multi-camera visual inertial datasets. To address this gap, we developed an extension of the Newer College Dataset, incorporating a 128-beam Ouster lidar and four fisheye cameras, with all hardware synchronised.

After its public release and subsequent evaluations, we recognised the need to establish a very accurate ground truth benchmark. In collaboration with Hilti corporation, we introduced a data capture process linked to the high quality TLS scanner. This led us to create the Hilti-Oxford dataset, which features the same multi-camera and IMU setup but employs a more accurate lidar sensor. This dataset includes a variety of challenging environments, from sparse and regular construction sites to a 17th-century neoclassical building with intricate details and

curved surfaces. The multi-modality and diversity of the Hilti-Oxford dataset has attracted significant attention, resulting in 42 submissions from both academic and industrial groups. This interest culminated in the successful second edition of the Hilti SLAM Challenge, which concluded in May 2022. Both datasets have been cited in published papers more than 30 times.

7.1 Newer College Dataset Extension

The following short paper was presented at the *IEEE International Conference on Robotics and Automation (ICRA) Workshop, Robotic Perception and Mapping: Emerging Techniques*. **Note it is not a peer reviewed publication**, but a workshop paper reviewed by the workshop panel. An accompanying video is available online at: https://www.youtube.com/watch?v=-LMq3zU47Pw.

Multi-Camera LiDAR Inertial Extension to the Newer College Dataset

Lintong Zhang, Marco Camurri, David Wisth and Maurice Fallon

***Abstract*— We present a multi-camera LiDAR inertial dataset of 4.5 km walking distance as an expansion of the Newer College Dataset. The global shutter multi-camera device is hardware synchronized with both the IMU and LiDAR, which is more accurate than the original dataset with software synchronization. This dataset also provides six Degrees of Freedom (DoF) ground truth poses at LiDAR frequency (10 Hz). Three data collections are described and an example use case of multi-camera visual-inertial odometry is demonstrated. This expansion dataset contains small and narrow passages, large scale open spaces, as well as vegetated areas, to test localization and mapping systems. Furthermore, some sequences present challenging situations such as abrupt lighting change, texture-less surfaces, and aggressive motion. The dataset is available at: https://ori-drs.github.io/newer-college-dataset/**

I. Introduction

There has been rapid progress in the field of robotic autonomous navigation. High-quality public datasets can propel research and development, allowing consistent evaluation across different algorithms. Our recent Newer College Dataset [1] features a stereo-inertial camera and a dense LiDAR setup, and introduced a novel method of generating accurate high-frequency ground truth poses. In line with the development of cutting-edge sensors, we now present a new handheld device incorporating a more accurately synchronized multi-camera device for challenging visual scenarios, and a wide field of view 128 channel LiDAR that can provide even denser point clouds per single scan.

Recently, many commercial robotic applications rely on state estimation systems containing multi-camera configurations. Despite the fact, the vast majority of datasets SLAM practitioners uses to benchmark their systems are based on either monocular or stereo camera configurations. There exists a gap in considering different types of sensing configurations among industrial applications and academic research, and this dataset would hope to serve such a purpose.

There are several monocular and stereo camera datasets for visual odometry and SLAM purpose. Some of these datasets contain IMU or LiDAR measurements. A notable example is the KITTI dataset [2] that provides LiDAR, IMU, and stereo camera data with GPS/INS ground truth. EuRoC MAV [3] and TUM VI [4] both offer 6 Degree of Freedom (DoF) ground truth poses, with data for IMU and stereo cameras, but not any LiDAR sensors. For a more detailed comparison, we refer the reader to Table I in our original paper [1]. However, there is very few public dataset that offers multi-camera vision for odometry or SLAM related usage.

The authors are with the Oxford Robotics Institute, University of Oxford, UK. {lintong, mcamurri, davidw, mfallon}@robots.ox.ac.uk

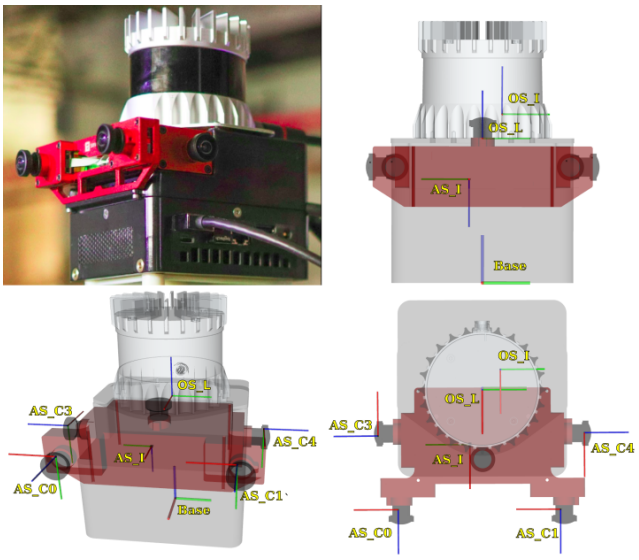


Fig. 1: Our custom built handheld device is on the top left. The other three images are the URDF model with reference frames shown in front, isometric, and top-down views.

Within the field of computer vision, there are a number of public datasets involving multiple cameras. EPFL-RLC [5] dataset used three static HD cameras inside a building to track objects such as pedestrians. WoodScape [6] was the first fisheye autonomous driving dataset with 4 large Fields of View (FoV) cameras. It provided several categories of information including segmentation, depth estimation, and bounding box. These datasets can be used for a variety of computer vision-related research such as object tracking, prediction, and classification, but not for autonomous navigation tasks such as localisation and mapping.

To the best of our knowledge, the only dataset with similar sensors to ours is the Hilti SLAM Challenge dataset [7]. This dataset focused on the construction site environment, providing sparse ground truth poses, with just a few ground truth measurements per sequence. Most sequences are collected in a "stop and go" fashion, where the total station produces a measurement using a reflecting prism during the "stop" periods.

We would like to provide the research community with a comprehensive multi-camera dataset that encompasses vision, LiDAR, inertial measurements, along with high-frequency ground truth poses and accurate prior maps.

Sensor	Type	Rate	Characteristics
LiDAR	Ouster, OS0-128	10 Hz	128 Channels, 50 m Range 90° Vertical FoV 1024 Horizontal Resolution
Cameras	Alphasense	30 Hz	Global shutter (Infrared) 720×540
LiDAR IMU	ICM-20948	100 Hz	3-axis Gyroscope 3-axis Accelerometer
Camera IMU	Bosch BMI085	200 Hz	Sychronized with cameras

TABLE I: Overview of the sensors in our handheld device.

II. The Handheld Device

The handheld multi-camera LiDAR inertial device is shown in Fig. 1. The sensors are rigidly attached to a precisely 3D-printed base. A complete URDF model of the device is available as an open source ROS package[1] (see Fig. 1). The Ouster LiDAR is directly mounted above the cameras for a balanced and compact design, so the top facing camera was removed. Tab. I provides an overview of the various sensors.

The multi-camera sensor in our device is the Alphasense Core Development Kit from Sevensense Robotics AG. An FPGA within the Alphasense synchronizes the IMU and four grayscale fisheye cameras – a frontal stereo pair with an 11 cm baseline and two lateral cameras. Each camera has a FoV of 126° × 92.4° and a resolution of 720 × 540 px. This configuration also produces an overlapping FoV between the front and side cameras of about 36°. The cameras and the embedded cellphone-grade IMU operates at 30 Hz and 200 Hz, respectively. The Ouster OS-0 LiDAR has 128 beams and a 90 degree elevation FoV, which provides a much denser point cloud than the OS1-64 used in the original dataset. Both sensors are cutting-edge devices in mobile robotics research and have recently been used in the DARPA Subterranean Challenge.

Fig. 1 shows the various sensor frames with the following abbreviations:

- Base: The bottom centre of the printed computer case.
- OS_I: The IMU coordinate system in the LiDAR.
- OS_L: The LiDAR coordinate system where the point clouds are read.
- AS_C0, 1, 3, 4: Camera optical frames.
- AS_I: The IMU coordinate system in the Alphasense.

III. Data Collection

We have collected a variety of datasets at different speeds of walking and turning. The three collections of datasets were gathered at different times of the year and they contain some challenging aspects such as fast motions, aggressive shaking, rapid lighting change, and textureless surfaces. The high frequency ground truth trajectories are also provided using the same method in the original paper, summarized in Sec. IV.

The datasets contain different levels of difficulty and are organized according to the aggressiveness of the motion and the type of scenes observed by the cameras.

[1]https://github.com/ori-drs/halo_description

Fig. 2: Trajectories for all collections. **Top:** Maths-Easy (green), Maths-Medium (orange), Maths-Hard (purple); **Mid left:** Quad-Easy (green), Quad-Medium (orange), Quad-Hard (purple); **Mid right:** Stairs; **Bot:** Cloister-Easy (green), Park (orange).

Collection 1 - New College:

- *Quad-Easy* (198 s): Two loops in the quad area with a typical walking speed (247 m).
- *Quad-Medium* (190 s): Two loops of brisk walking with cameras pointing in different directions on different occasions (260 m).
- *Quad-Hard* (187 s): Fast walking with aggressive motion, approaches to the walls, and lighting changes (234 m).
- *Stairs* (118 s): Climbing up and down in a narrow stairway with the cameras subject to doors opening and textureless corridor walls (57 m).

Collection 2 - New College:

- *Cloister* (278 s): Two loops of the cloister corridor and the cloister centre quad (429 m).

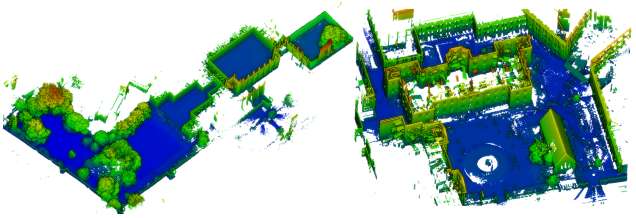


Fig. 3: Ground truth and prior maps. **Left:** New College, Oxford (extended from the original Newer College dataset map); **Right:** Maths Institute, University of Oxford.

- *Park* (1567 s): Long experiment of the entire park and two quads with multiple loops. Route corresponds to the original dataset (2396 m).

Collection 3 - Maths Institute:

- *Maths-Easy* (216 s): Outdoor large scale environment with typical walking speed (264 m).
- *Maths-Medium* (176 s): Brisk walking with cameras occasionally turned to face different directions (304 m).
- *Maths-Hard* (243 s): Fast walking with aggressive motions, textureless surfaces, rapid rotations and shaking of the device up to 5.5 rad/s (321 m).

In Fig. 2, trajectories for each collection are overlaid on the ground truth map using the method explained in the next section. A video example displaying four camera steams and the point cloud of the cloister sequence can be viewed at https://youtu.be/tGXNSlmQOb0.

IV. Ground Truth

The prior map and ground truth pose generation process use the same method described in [1]. We use a survey-grade 3D imaging laser scanner, Leica BLK360, to scan the entire environment. As shown in Fig. 3, we further extend the prior map of New College and provide an additional map for the Maths Institute. Ground truth poses are determined by registering each undistorted LiDAR scan to the prior map using an approach based on the point-to-point Iterative Closest Point method. By processing each scan at a lower speed, we can accurately correct the motion distortion using integrated IMU measurement. The poses are expressed in the "Base" frame shown in Sec. II.

V. Calibration

Similarly to the original stereo camera LiDAR dataset, we use the open source camera and IMU calibration toolbox Kalibr [8] to compute the intrinsic calibration of the Alphasense cameras as well as their extrinsics. We perform spatio-temporal calibration between the cameras and the two IMUs embedded in the Alphasense and the Ouster sensor. The calibration target is 6×6 April grids and each tag size is 8.8 cm. The calibration settings use the pinhole projection model with equidistant distortion. All cameras are calibrated with the IMU, starting with the frontal stereo cameras, then the individual lateral cameras. Since the collections were carried out at different times of the year, we provide a set of calibration files corresponding to each dataset collection.

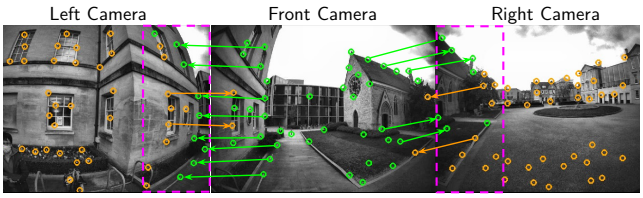


Fig. 4: VILENS-MC takes advantage of any overlapping image regions (purple rectangles) in a multi-camera setup to track features across cameras. This increases feature track length and avoids tracking the same feature independently in different cameras. The arrows indicate features being tracked from one image to another.

A. Synchronization

The recording computer, Intel NUC, is modified to support dual Ethernet ports with hardware timestamp capability. The Ouster LiDAR and Alphasense are synchronized with the NUC using the Precision Time Protocol (PTP), which achieves sub-microsecond accuracy.

Further details about the Alphasense synchronization can be found at its Github page[2]. The detailed synchronization procedure for Ouster LiDAR can be found in the software manual[3] (Section 16 "PTP Quickstart Guide").

VI. Example Usage

To demonstrate one usage of the dataset and the advantage of multi-sensor synchronization, we compared three visual-inertial odometry methods, with reference to LiDAR-generated ground truth.

The first is a multi-camera visual-inertial odometry system (VILENS-MC) [9], developed by the authors. VILENS-MC is based on factor graph optimization which estimates motion by using all cameras simultaneously while retaining a fixed overall feature budget. VILENS-MC introduced cross camera feature tracking as shown in Fig. 4, and focuses on motion tracking in challenging environments by leveraging this multi-camera dataset. Many sequences in this dataset, such as Maths-Hard, Quad-Hard or Stairs, would cause classic stereo inertial odometry approaches to fail. As an example, Fig. 5 shows the trajectories estimated by OpenVINS [10], ORBSLAM-3 [11] and VILENS-MC for the Maths-Hard dataset.

VII. Conclusion

In this paper, we presented an extension of Newer College Vision and LiDAR dataset. By leveraging a highly accurate and detailed prior map, we provided accurate high-frequency 6 DoF ground truth poses, which distinguishes our dataset from existing ones. We combined a cutting edge multi-camera device and a dense 3D LiDAR sensor to provide challenging scenarios, which are considered difficult for current robotic navigation systems. The paper also illustrated an

[2]https://github.com/sevensense-robotics/alphasense_core_manual

[3]https://data.ouster.io/downloads/software-user-manual/software-user-manual-v2p0.pdf

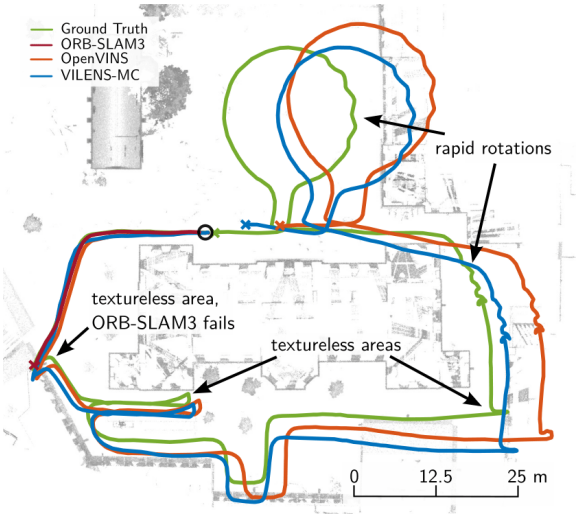


Fig. 5: Top-down view of Maths-Hard dataset comparing the estimated trajectory of ORB-SLAM3, OpenVINS, and VILENS-MC against the ground truth. A black circle marks the start of the trajectory, while colored crosses indicate the last pose of each trajectory. A textureless area, where multiple cameras were facing towards walls was a cause of failure for the stereo odometry methods.

example use case of multi-camera visual inertial odometry. We hope this dataset can propel researchers to further push the boundaries of autonomous navigation by demonstrating how algorithms can become more robust when taking into account challenging scenarios.

Acknowledgment

The authors would like to thank the members of the Oxford Robotics Institute (ORI) who helped with the creation of this dataset release, especially Wayne Tubby. We also thank the personnel of New College for facilitating our data collection.

This research was supported by the Innovate UK-funded ORCA Robotics Hub (EP/R026173/1) and the EU H2020 Project THING. Maurice Fallon is supported by a Royal Society University Research Fellowship.

References

[1] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. 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)*, 2020, pp. 4353–4360.

[2] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[3] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *International Journal of Robotics Research (IJRR)*, 2016.

[4] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1680–1687.

[5] T. Chavdarova and F. Fleuret, "Deep Multi-camera People Detection," in *IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 848–853.

[6] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Pérez, "WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9307–9317.

[7] M. Helmberger, K. Morin, N. Kumar, D. Wang, Y. Yue, G. Cioffi, and D. Scaramuzza, "The Hilti SLAM Challenge Dataset," *arXiv preprint arXiv:2109.11316*, 2021.

[8] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending Kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4304–4311.

[9] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the budget: Feature selection and tracking for multi-camera visual-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1182–1189, 2022.

[10] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4666–4672.

[11] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM," *IEEE Transactions on Robotics (TRO)*, pp. 1–17, 2021.

7.2 Hilti-Oxford Dataset

The following article was published in the *IEEE Robotics and Automation Letters (RA-L)* and it was also presented at the *IEEE International Conference on Robotics and Automation (ICRA)* 2023 [152]. An accompanying video is available online at: https://www.youtube.com/watch?v=-LMq3zU47Pw.

Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping

Lintong Zhang[1], Michael Helmberger[2], Lanke Frank Tarimo Fu[1], David Wisth[1], Marco Camurri[1], Davide Scaramuzza[3], Maurice Fallon[1]

***Abstract*—Simultaneous Localization and Mapping (SLAM) is being deployed in real-world applications, however many state-of-the-art solutions still struggle in many common scenarios. A key necessity in progressing SLAM research is the availability of high-quality datasets and fair and transparent benchmarking. To this end, we have created the Hilti-Oxford Dataset, to push state-of-the-art SLAM systems to their limits. The dataset has a variety of challenges ranging from sparse and regular construction sites to a 17th century neoclassical building with fine details and curved surfaces. To encourage multi-modal SLAM approaches, we designed a data collection platform featuring a lidar, five cameras, and an IMU (Inertial Measurement Unit). With the goal of benchmarking SLAM algorithms for tasks where accuracy and robustness are paramount, we implemented a novel ground truth collection method that enables our dataset to accurately measure SLAM pose errors with millimeter accuracy. To further ensure accuracy, the extrinsics of our platform were verified with a micrometer-accurate scanner, and temporal calibration was managed online using hardware time synchronization. The multi-modality and diversity of our dataset attracted a large field of academic and industrial researchers to enter the second edition of the Hilti SLAM challenge, which concluded in June 2022. The results of the challenge show that while the top three teams could achieve an accuracy of 2 cm or better for some sequences, the performance dropped off in more difficult sequences.**

***Index Terms*—Data Sets for SLAM; SLAM; Mapping;**

I. Introduction

SLAM research has made impressive progress, allowing the transition from lab demonstrations to real-world deployment. Open-source datasets play a key role in this transition, as researchers can progressively improve and compare different SLAM solutions. The TUM [1], EuRoC [2], and KITTI [3] datasets have been a pillar in the robotics community, and their leaderboards are still motivating new and improved algorithms. However, as SLAM algorithms improve to enable real-life applications, so should their benchmarks. We see the need to have more challenging datasets to differentiate top SLAM approaches. We also believe that these datasets should use the latest sensors to benchmark multi-modal SLAM accuracy and robustness under a variety of challenges.

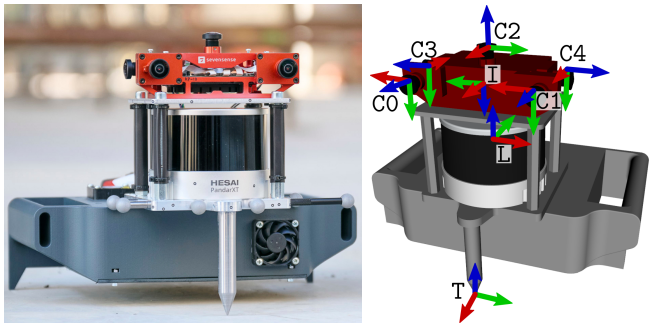


Fig. 1: The handheld device, called Phasma, is composed of five cameras, an IMU sensor, and a 32 beam lidar.

The key motivation for this work is to create a high-quality dataset with a variety of challenging sequences that can propel SLAM-related research. As shown in several works [4], [5], [6], it is beneficial to fuse multiple sensors to improve accuracy and robustness. Hence, we present a SLAM dataset combining vision, lidar, and inertial sensing.

Additionally, accurate ground truth plays a vital role in evaluation, as many lidar-based algorithms are approaching centimeter-level accuracy. Thus, we propose a new approach which can provide millimeter accuracy by using state-of-art surveying equipment. By leveraging a high precision long-range lidar, multiple global shutter cameras, and a synchronized IMU, we provide a comprehensive dataset where SLAM algorithms must perform accurately and robustly in order to be used in real-world applications.

The need for and interest in a high-quality dataset is evidenced by the Hilti SLAM Challenge 2022[1], which received 42 submissions from both industry and academic research groups. In summary, the Hilti-Oxford dataset[2] offers the following contributions:

- Challenging and degenerate scenarios, such as dark corners, narrow stairs, long corridors, and a few dynamic objects, with sequences specifically designed to break existing SLAM algorithms;
- A data collection platform with modern sensors, including an accurate long-range lidar (up to 120 m), five fisheye

Manuscript received Aug 6th, 2022; Revised Oct 14th, 2022; Accepted Nov 16th, 2022.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. M. Fallon thanks the Royal Society for funding his University Research Fellowship.

[1]Oxford Robotics Institute, Department of Engineering Science, University of Oxford, UK. lintong@robots.ox.ac.uk

[2]Hilti AG, Schaan, Liechtenstein

[3]Robotics and Perception Group, Department of Informatics, University of Zurich, Switzerland

Digital Object Identifier (DOI): see top of this page.

[1]Challenge Video: https://www.youtube.com/watch?v=-LMq3zU47Pw

[2]https://hilti-challenge.com/dataset-2022.html

cameras operating at 40 Hz, and inertial sensors. The sensors are mounted on a high-precision machined chassis, with extrinsics verified by a micrometer accurate scanner. All signals have been hardware synchronized;
- A novel sparse ground truth collection method based on a survey-grade scanner and reference targets, which achieves millimeter precision;
- Insights and discussion of the merits of each system and sensor modality based on the high number of submissions.

II. Related Work

Existing benchmark datasets for SLAM can be categorized by their different operation domains. Depending on the domain, different sensory data and varying degrees of ground truth accuracy are provided.

In the visual-inertial odometry domain, EuRoC [2] and TUM VI [1], which provide camera and IMU data, have been extensively used by the research community. EuRoC recorded hardware time-synchronized stereo camera images and IMU measurements from a micro aerial vehicle equipped with a Skybotix stereo VI sensor. While the 11 EuRoC sequences are accompanied by millimeter accurate ground truth poses, their trajectories only covered the indoor bounds of a motion capture system. The TUM VI dataset uses a hand-held data acquisition device which features a global shutter stereo camera and IMU, which were also hardware time-synchronized. Unlike EuRoC, the 28 sequences in TUM VI include segments that extend outdoors, providing a diverse set of scenarios to benchmark visual-inertial SLAM. The segments in TUM VI that extend outdoors start and end in an indoor motion capture environment, providing ground truth poses in these indoor segments.

Datasets such as KITTI [3], WoodScape [7], and UMich [8] are specialized to the autonomous driving domain, and in addition to IMU and camera images, these datasets also provide lidar data. WoodScape and UMich are both equipped with 360° cameras while KITTI used a linear array of stereo cameras, two colored and two grayscale. With a Segway scooter as a platform, the UMich dataset covers both indoor and outdoor scenarios. KITTI and WoodScape, on the other hand, used large vehicles as their platform, so were not able to record indoor environments. All these datasets (KITTI, WoodScape, and UMich) provided ground truth trajectories using GPS/GNSS-INS measurements which are only accurate to several centimeters.

The 2021 Hilti Challenge dataset [9] used similar sensors to our proposed dataset and featured an AlphaSense 5-camera module which offered a 270° continuous field of view. For wide-coverage lidar measurements, [9] used the Ouster OS0-64 lidar with a range accuracy of ± 3 cm, whereas the dataset presented in this paper uses the much more accurate Hesai PandarXT-32 lidar with ± 1 cm range accuracy.

Compared to the aforementioned works, our dataset covers a variety of scenarios ranging from a sparse construction site to a 17th-century theatre with challenging staircases and narrow hallways. Throughout these indoor and outdoor sequences with difficult trajectories, we consistently provide millimeter-accurate ground truth positions at select control points using the method described in Sec. V-B. The challenging sequences and accurate sparse ground truth measurements of our dataset aimed to propel SLAM research into real-world applications such as architectural inspection and construction monitoring, where use cases require sub-centimeter accuracy.

Sensor	Type	Rate	Characteristics
Lidar	Hesai, PandarXT-32	10 Hz	32 Channels, 120 m Range 31° Vertical FoV 1024 Horizontal Resolution
Cameras	Alphasense	40 Hz	5 Global shutter (Infrared) 720×540 pixels
IMU	Bosch BMI085	400 Hz	Sychronized with cameras

TABLE I: Overview of the sensors on the Phasma device.

III. Hardware

Our handheld multi-camera lidar inertial device, called Phasma, is shown in Fig. 1. Phasma is composed of these three sensors, rigidly assembled in a case that has been produced by a precise milling machine. A complete URDF model of the device is also available as an open source ROS package[3]. The Hesai lidar is directly mounted below the cameras for a balanced and compact design so that the upward facing camera is not obstructed. The metal needle tip at the bottom is used to align with target points when determining ground truth (see Sec. V-B). Tab. I provides an overview of the sensor specifications.

Drawing on Hilti's expertise, we manufactured a precise handheld device that had improved accuracy and stability compared the rig used in the previous challenge [9]. Specifically, by using milled components and dowel pins, we ensured an accurate assembly of the sensors. The correct placement of the lidar, metal tip, and cameras were verified using a GOM Atos Q3[4] industrial 3D scanner. The multi-camera sensor is an Alphasense Core Development Kit from Sevensense Robotics AG. An FPGA within the Alphasense hardware synchronizes the IMU and five grayscale fisheye cameras – a frontal stereo pair with an 11 cm baseline, two lateral cameras, and one upward-facing camera. Each camera has a Field of View (FoV) of 126×92.4° and a resolution of 720×540 px. This configuration produces an overlapping FoV between the front and side cameras of about 36°. The cameras and the embedded cellphone-grade IMU operates at 40 Hz and 400 Hz, respectively. The Hesai lidar has 32 beams and a 31° elevation FoV, with a range of 5 cm to 120 m. Notably, the Hesai Pandar has a range accuracy of ± 1 cm and a precision of 0.5 cm (1σ).

A. IMU Calibration

A 90 min sequence of IMU data was collected on a stationary flat surface. We adopted the Allan Variance estimation method[5] to compute the angle random walk, bias instability, and random walk for the gyroscope and the velocity random walk, bias instability, and random walk for the accelerometer. This IMU rosbag is provided with the dataset for the user's convenience.

[3]https://github.com/Hilti-Research/Hilti-SLAM-Challenge-2022
[4]https://www.gom.com/en/products/3d-scanning/atos-q
[5]https://github.com/ori-drs/allan_variance_ros

Fig. 2: Hilti-Oxford dataset environments: *Top:* Construction site. *Middle:* Long office corridor (left), Sheldonian stairs (right). *Bottom:* Sheldonian lower gallery (left) and exterior (right).

B. Camera Calibration

Similarly to the Newer College Dataset Multi-Camera Extension [10], we used the open source camera and IMU calibration toolbox Kalibr [11] to compute the intrinsic and extrinsic calibration of the Alphasense cameras. The calibration used the pinhole projection model with equidistant distortion. We then performed spatio-temporal calibration between the cameras and the IMU embedded in the Alphasense. The large rigid calibration target contained 7×12 April tags, with a tag size of 15 cm. All cameras were calibrated with the IMU, with the front cameras calibrated as stereo cameras and the remaining three calibrated as monocular cameras. In this dataset, we provide the rosbag of the camera and IMU calibration sequence to enable users to conduct their own calibration.

IV. Dataset

This dataset was recorded in two locations. The first is a construction site in Schaan, Liechtenstein, near the Hilti headquarters. It is a live construction site with limited texture and color variation. The site covers an area of $100\,\text{m}\times30\,\text{m}$ with longer range measurements scanning nearby buildings. The site has four floor levels including a basement. The second location is the Sheldonian Theatre, built in 1664 in Oxford, England. The Sheldonian Theatre is used for ceremonial events and graduations and is an architecturally significant *listed building*. As shown in Fig. 2, the building spans 6 floors, from the basement to the octagonal cupola at the top. The cupola is accessible through narrow staircases, only 60 cm across. Both of these locations challenge SLAM systems in different ways.

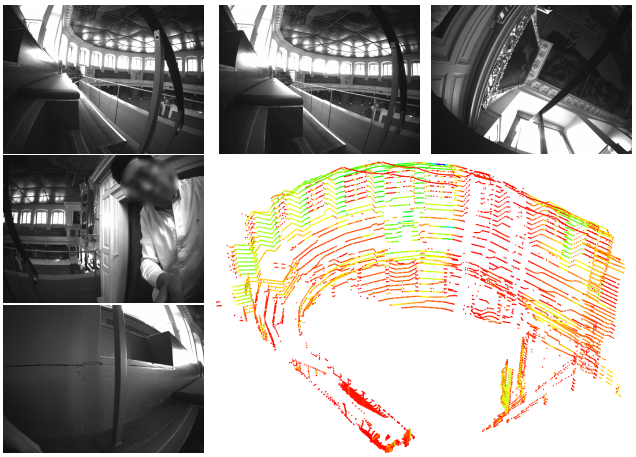


Fig. 3: Dataset example showing images from each camera, and a lidar scan of the upper gallery.

Each dataset sequence is a rosbag that contains five camera image topics, one lidar topic, and one IMU topic. An example of the data is shown in Fig. 3. Below is a summary of each sequence. We qualitatively indicate the difficulty levels based on the environment and motions for each sequence as either easy, medium, or hard. Users can find more information including the top-down trajectories on the dataset website.

A. Challenge Sequences

Hilti Construction Site:

a) **Exp01 Construction Ground Level** (Easy, 227 s):
One loop around the ground level at typical walking speed.
b) **Exp02 Construction Multilevel** (Medium, 430 s):
One loop around the upper level then going down the staircase to the ground level. Some shaking motions with the angular velocity up to 3.5 rad/s
c) **Exp03 Construction Stairs** (Hard, 309 s):
Starting in the staircase, moving into dark corners while going down the stairs, then entering a car park in the basement, and finally returning to the top of the staircase. An operator is walking in front about half of the time.
d) **Exp07 Long Corridor** (Medium, 132 s):
A 100 m long corridor in Hilti's head office of 2 m width and 3 m height. Structurally, both ends of the corridor are higher than the middle section.

Oxford Sheldonian Theatre:

a) **Exp09 Cupola** (Hard, 367 s):
From the ground floor hall going up multiple levels through very narrow staircases to the very top of the theatre, the cupola. Then descending down another set of stairs back to the ground floor hall.
b) **Exp11 Lower Gallery** (Medium, 151 s):
From the ground floor hall to the first floor lower gallery, circling the lower gallery and back to the starting point.
c) **Exp15 Attic to Upper Gallery** (Hard, 260 s):
From the top floor attic space walking down to the upper

gallery, going through some degenerate narrow spaces. Circling around the upper gallery and climbing back up another set of stairs to the attic space.

d) **Exp21 Outside Building** (Easy, 152 s):
Starting outside the theatre, circling the theatre and the main quad, and entering back into the theatre.

These eight sequences formed the 2022 Hilti Challenge.

B. Additional Sequences

We also release some additional sequences, with both sparse and dense ground truth trajectories, which can provide extra challenges for algorithm testing.

a) **Exp04 Construction Upper Level 1** (Easy, 124 s):
One loop in the upper level with a typical walking speed.

b) **Exp05 Construction Upper Level 2** (Easy, 125 s):
A repeat of Exp04 by another operator, offering a different walking pattern.

c) **Exp06 Construction Upper Level 3** (Medium, 150 s):
Similar to 4 and 5 but a faster walking speed with aggressive motions, such as spinning on the spot, and approaching walls and corners.

d) **Exp10 Cupola 2** (Hard, 446 s):
Similar to Exp09 but with faster motions.

e) **Exp14 Basement 2** (Medium, 73 s):
A short sequence that starts from the staircase and goes through the basement with door opening scenarios.

f) **Exp16 Attic to Upper Gallery 2** (Hard, 198 s):
Similar to Exp15 but with faster motions.

g) **Exp18 Corridor Lower Gallery 2** (Hard, 100 s):
A short sequence starts in the corridor and enters the lower gallery from a different set of stairs from Exp11.

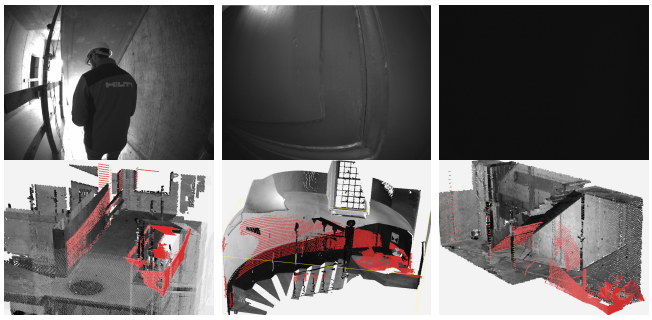
h) **Exp23 The Sheldonian Slam** (Hard, 1049 s):
A large mission around the whole Sheldonian Theatre that visits all spaces and revisits the ground hall several times for loop closure purposes.

In particular, Exp10 and Exp16 are harder than the challenge sequences due to faster motions, and Exp23 is an interesting sequence which is much longer than the others and includes many loop closures, stair climbs, and sensor deprivations.

C. Characteristics of the Sequences

We intentionally introduce challenging and degenerate scenarios into the dataset. These scenarios include aggressive motions, such as shaking and swinging the device, dynamic objects occasionally blocking the field of view, narrow staircases which are geometrically similar, and dark corners where cameras cannot detect features.

The purpose of this is to test the robustness and accuracy of state-of-the-art SLAM systems. For example, Fig. 4 shows: (a) a person walking in front and blocking the field of view; (b) the handheld device being placed down close to one wall with only a few lidar points sensed in the environment (a degenerate mode for lidar-based SLAM); (c) the device entering a very dark corner in the basement, returning few lidar points. This is challenging for both vision- and lidar-based SLAM. In general, lidar-based SLAM suffers in confined space when there are not enough geometric constraints in a scan for registration. For vision-based SLAM, moving around in confined spaces can also result in rapid scene changes and fast image feature flow. We spent an average of 65 % of the time moving inside various confined spaces in Exp03 Construction Stair, Exp09 Cupola, and Exp15 Attic to Upper Gallery sequences.



(a) Exp02 (b) Exp15 (c) Exp02

Fig. 4: Top row shows camera images in challenging scenarios with their corresponding lidar scans in red at the bottom (aligned to our ground truth model in grey).

V. Ground Truth

A. Prior Map

Prior maps of the two facilities were built using the scanner shown in Fig. 6. The Z+F Imager 5016 3D laser scanner is equipped with an integrated HDR camera, internal light, and positioning system. It measures up to 360 m, with a maximum measurement rate of 1 million points/s. It has a field view of 360×320°, an angular accuracy of 14.4 arcsec (both horizontal and vertical) and a linearity error of the laser system of ≤1 mm + 10 ppm/m. The ranging noise is negligible (sub-mm). This scanner allows us to build accurate maps of the environment and establish ground truth points with millimeter accuracy. For the registration of the scans, we used reflective scanner targets as well as plane-to-plane registration followed by block adjustment [12]. Due to sufficient overlap between the laser scans, the reported pairwise registration uncertainty sigma is in the sub-mm range, which is the prior used before the bundle adjustment. The final uncertainty sigma for each scan with respect to the starting scan or master scan is shown in Fig. 5. 91 % and 95 % of scans have position uncertainty within 3mm for the Sheldonian and the construction site respectively. The most significant uncertainty is still just a few mm, as they are the leaf scan with fewer connections in the bundle adjustment graph. More importantly, we have not placed ground truth targets in those leaf scans. Some snapshots of the final registered point clouds are shown in Fig. 7.

B. Sparse Ground Truth For Evaluation

The process to set up a ground truth point is detailed as follows. We first select appropriate locations to set up crosshairs on the floor, by drawing on an adhesive blue marker 8. We then align the metal tip of a checkerboard target (shown in Fig. 8) at the center of the cross hairs. After adjusting the bubble level,

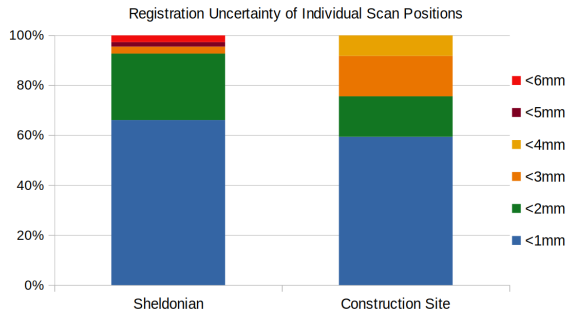


Fig. 5: Prior map individual scan registration uncertainties with respect to the initial scan.

Fig. 6: The Z+F scanner in the Sheldonian Theatre, scanning and taking images.

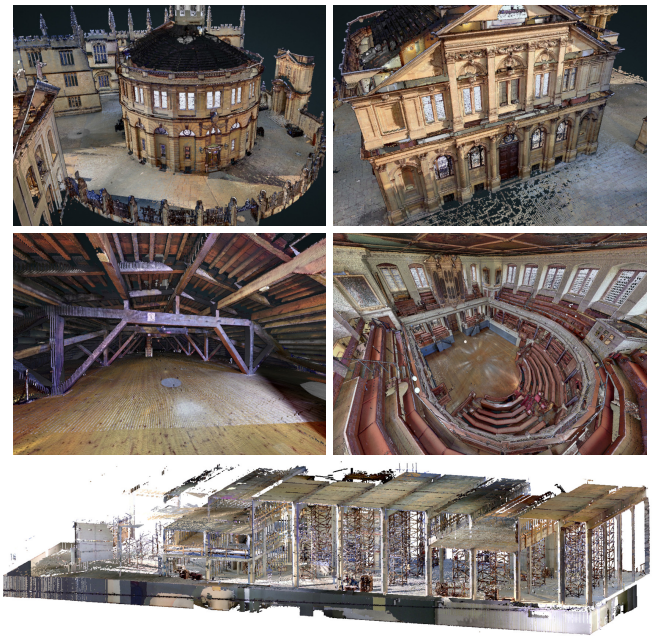


Fig. 7: Final point clouds built with the survey grade scanner. Top: Sheldonian theatre front and back exterior view. Mid: Sheldonian theatre attic and hall view. Bottom: Multi-level construction site side view.

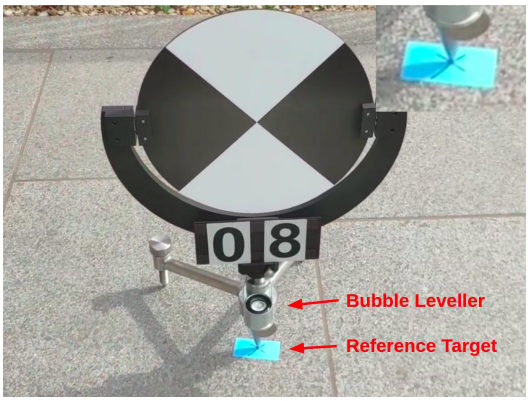


Fig. 8: A reference target placed on the ground and used for point cloud fine registration. Sparse ground truth points were created by placing Phasma at the cross-hairs of the reference target.

each target is labelled numerically. These targets are scanned by the Z+F scanner and used in the fine registration step to create the complete point cloud. We ensure each target can be seen in multiple scans to add additional constraints to the cloud registration. All crosses drawn on the floor can be extracted from the final registered point cloud and used as ground truth evaluation points. During the handheld data gathering stage, we again place the tip of the handheld device at the crosshairs on the floor. To be noted, during this process, the site was closed to visitors and we ensured each blue marker stayed in the same place. Each time we took several seconds to carefully placed the metal tip on the crosshair, to ensure the manual error stays less than 1 mm.

This sparse ground truth method is inspired by site surveying practice where the surveyor takes measurements of construction sites as a continuous monitoring or inspection procedure. We adopted the same equipment to establish ground truth poses for trajectories which is novel for SLAM datasets. It achieves millimeter accuracy but is limited to a small number (between 5 and 10) of instants where the Phasma device has to be laid on the crosshairs. To assist development and evaluation, denser but less accurate ground truth trajectories are provided for the additional sequences.

C. Dense Ground Truth For Development

The dense ground truth poses generation process uses the same method described in [10], [13]. The registered point cloud from the Z+F laser scanner in Sec. V-A was used as a prior map in which to localize. Ground truth poses are determined by registering each undistorted lidar scan to the prior map using an approach based on the point-to-point Iterative Closest Point method. We use our existing VILENS system [6] which is a Lidar-inertial odometry system to process each scan at a lower playback speed. Instead of building a map, we register to the prior ground truth map. We use a factor graph in the pre-integration IMU factor to motion correct the lidar scan. In general, the accuracy of this method is in the 1-2 cm range, so it can be useful to help users develop their SLAM algorithm, such

as identifying precisely where odometry drift has occurred. When a SLAM algorithm approaches <1 cm accuracy, we recommend using the sparse ground truth in Sec. V-B for performance evaluation.

D. Scoring Metric

The aim of the challenge was to understand the state-of-the-art SLAM algorithms for use in the built environment, and in particular for the construction industry. Many real-world applications (*e.g.* autonomous hole drilling) require sub-centimeter accuracy to be useful. This motivated the accuracy-based error metric described below.

First, the control points and the estimated trajectory are aligned using SE(3) Umeyama alignment to account for any differences in coordinate systems [14]. Then the absolute distance error e_i between the ith control point and the estimated trajectory is calculated and given a score s_i, where

$$s_i = \begin{cases} 10 & \text{if } e_i < 0.01\,\text{m} \\ 6 & \text{if } e_i \in [0.01, 0.03\,\text{m}) \\ 3 & \text{if } e_i \in [0.03, 0.06\,\text{m}) \\ 1 & \text{if } e_i \in [0.06, 0.10\,\text{m}) \\ 0 & \text{if } e_i \geq 0.10\,\text{m} \end{cases} \tag{1}$$

The total score for each dataset S_j is the percentage of the maximum possible score (*i.e.* if all points had <1 cm error and scored 10 points),

$$S_j = \left(\frac{1}{10N} \sum_{i=0}^{N} s_i \right) \times 100 \tag{2}$$

where N is the number of ground truth points evaluated in each dataset. This denominator normalizes the score for a particular run to be between 0 and 100, regardless of the number of ground truth points in each dataset. The final score is then a sum of the scores from each experiment.

VI. Challenge Results and Findings

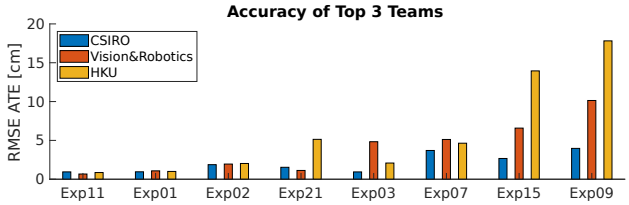
A. Results

A total of 42 academic and industrial groups submitted their results to the *2022 Hilti SLAM Challenge*. The challenge results were announced as part of the Future of Construction workshop[6] at the *IEEE International Conference on Robotics and Automation* in June 2022. To support their submissions, teams were given access to the calibration datasets, as well as three of the additional sequences with dense ground truth poses (see Sec. V-C).

The challenge results are shown in Tab. II. Summary reports of each team's approach are available on the challenge website [7]. The highest scoring team was CSIRO with a score of 563.8. Their Wildcat SLAM [15] algorithm uses a continuous-time trajectory representation for lidar-inertial odometry using sliding-window optimization and online pose graph optimization. This is refined by an offline global optimization module that takes advantage of non-causal information. Of the top 25 teams, all used lidar and IMU data, while only 10 used camera data.

[6]Video: https://www.youtube.com/watch?v=NpfJV_Q_SMk&t=205s

[7]https://hilti-challenge.com/leader-board-2022.html



Team	Exp11	Exp01	Exp02	Exp21	Exp03	Exp07	Exp15	Exp09
CSIRO	**0.9**	**1.0**	1.9	1.5	**0.9**	3.7	2.7	4.0
Vision&Robotics	**0.7**	1.1	2.0	1.1	4.8	5.1	6.6	10.1
HKU	**0.9**	1.0	2.0	5.1	2.1	4.6	13.9	17.8

Fig. 9: Summary of the top three team's RMSE ATE (cm), sorted from smallest to largest error. Results in bold have reached the desired sub-cm accuracy.

The highest scoring vision-only submission was Smart Robotics Lab with a score of 32.5. SRL's OKVIS2.0 [16] produced complete trajectories for all of the sequences, however typical errors were in the 10–20 cm range which resulted in a low score. This highlights the gap in performance between lidar and camera-based SLAM systems, and the susceptibility of vision-based systems to subtle scaling and calibration errors.

B. Discussion

Tab. II describes the details of each algorithm's odometry and SLAM modules. In the odometry column, we classify each algorithm as either filter or optimization based and whether the odometry system is real-time. In the SLAM column, we label each algorithm according to the use of global bundle adjustment (Global BA), loop closure detection (LC), and if the algorithm only uses past measurements (causal). Overall, the scoring metric presented in this paper approximately aligns with the mean RMSE of the Absolute Trajectory Error (ATE). However, some entries, such as KTH&NTU, achieved a better ATE but lower score. This is explained by the fact that most of the poses fell outside the high-scoring sub-3 cm range. Additionally, the *mean* ATE can be deceptive here, as some teams have incomplete trajectories or performed badly in one particular sequence.

Fig. 9 shows a summary of the error on each sequence by the top three teams. Sequences with large open spaces and overlapping areas, where LIDAR scan matching can be highly effective, had the lowest error (Exp01, Exp02, Exp11, Exp21). The sequences with the highest error had challenging geometries for lidar-based algorithms including long narrow corridors (Exp07) and small staircases (Exp03, Exp09, Exp15), as illustrated in Fig. 2. This demonstrates that while the top three teams achieved accuracy close to 1 cm in some of the easier sequences, there is room for improvement in the others.

Another key observation is that the top four solutions were lidar-inertial only solutions, without the use of camera data. It has become common knowledge to fuse IMU measurements to provide a strong prior in SLAM system nowadays. While we still intended to create ill-conditioned situations for lidar-inertial based SLAM to require camera data fusion, the lidar

	Lead Organization	Algorithm	Sensors Used			Odometry		SLAM			Same	Results	
			Lidar	IMU	Cam. (#)	Type	Real-Time	Global BA	Causal	LC	Params	ATE	Score
1	CSIRO	Wildcat SLAM [15]	✓	✓		SW Opt.	✓	✓	✗	✓	✓	2.07	563.8
2	Vision & Robotics	MC2SLAM [17]	✓	✓		SW Opt.	✓	✓	✗	✓	✓	3.94	443.8
3	HKU	FastLIO2[18], BALM [19]	✓	✓		Filter	✓	✓	–	✓	–	5.94	400.4
4	KAIST	Based on [18], [20]	✓	✓		Filter	–	✓	✗	✓	–	19.02	317.5
5	Beihang Uni.	Based on [18], [21]	✓	✓	✓(2)	Filter	✓	✗	✓	✗	✗	22.59	311.6
6	Luxembourg Uni.	Based on [18], [22]	✓	✓	✓(1)	Filter	✓	✗	✓	✗	✓	20.49	303.8
7	MINES ParisTech	CT-ICP [23]	✓	✓		Opt.	✓	✗	✓	✓	–	7.72*	272.8
8	AIST	VITAMIN-E [24], [25]	✓	✓	✓(3)	SW Opt.	✓	✗	✓	✗	✓	16.16	260.5
9	HKUST & Georgia Tech	Based on [18], [26]	✓	✓	✓(5)	Filter	✓	✓	✗	✓	✓	47.50	257.6
10	KTH & NTU	VIRAL SLAM [27]	✓	✓		SW Opt.	✗	✗	✓	✗	✓	6.90	251.9
	Vision-only Results												
1	TUM	OKVIS2.0 [16]		✓	✓(5)	SW Opt.	✓	✓	✗	✓	✓	25.36	32.5
2	Stuttgart Uni. & TUM	Based on [28]		✓	✓(4)	SW Opt.	✗	✗	✓	✗	✗	42.04	22.2

Legend: # = No. of cameras used, – = No information provided, ATE = Mean RMSE ATE (cm), * = Did not submit results for Exp15

TABLE II: HILTI SLAM Challenge 2022 Results

scanner still captured sufficient information to avoid degeneracy in most sequences. For example, in Exp02 (Fig. 4(a)) we had an operator walk in front to block the sensors but the lidar scan was able to capture the slanted staircase ceiling and wooden rails (circled in Fig. 10-Right) which was enough to constrain the estimation. Similarly, when climbing the lower staircases of the Sheldonian, the lidar could scan through small windows onto adjacent buildings and the ground outside the theatre, despite there being insufficient constraints within the small staircase. These seemingly challenging situations were resolved by the accuracy and range of the lidar. We note that the top performing teams relied on dense local lidar submaps to overcome these locally-degenerate scenarios.

The most challenging sequence was Exp09 which entered the narrow upper staircases of the Sheldonian (from 132 s) where there were no windows, providing limited constraints for lidar odometry. Other failure situation example are dark corners under the stair cases (Exp03, 71 s), narrow space (Exp15, 24 s), middle of a long corridor (Exp07, 46 s).

There are some interesting findings on loop closure from the submissions. First, for shorter sequences, e.g. Exp01, teams could simply keep a complete map in their lidar odometry system. Drift was small enough that an explicit loop closure was not actually needed, and the system could implicitly localise to the local map without doing pose graph SLAM. For longer runs, such as Exp03, loop closures were used to but sometimes they distributed the error from one particular section to the whole trajectory. More importantly, teams used loop closures between sequences to create a multi-sequence map, e.g. linking exp09, exp11 and exp15, to further reduce drift.

C. Known Issues

Although we took the utmost care in the creation of this dataset and challenge, there were a few limitations of our approach:

Scoring Metric: The metric used in this evaluation focused on the accuracy of the trajectory and did not consider other performance characteristics of real-time SLAM systems, such as latency and computation. There was a wide range of timing and computational differences for various online and offline processing methods. Some teams used multi-sequence fusion in post-processing to optimize their results - thus their results are likely to be better than when operating on the individual sequences. Also given each team used different hardware, it would be difficult to include this in the scoring. However, we have qualitatively captured these traits in Table II and would refer readers to each team's report on the website.

Lidar-IMU Calibration: While the dataset used highly-accurate, machined components with low tolerances, we did not undertake a separate extrinsic calibration between the lidar and IMU. This may have resulted in a few millimeters of error in the ground truth control points.

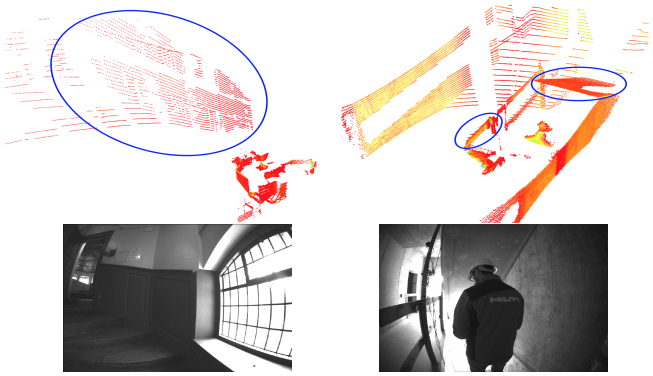


Fig. 10: *Left*: While in a narrow staircase, the lidar sees through a window and scans the adjacent building and the ground. *Right*: An operator walks in front to block the view, but the stair ceiling and rails are scanned and provide enough constraints.

VII. Conclusion

This paper presents the Hilti-Oxford Dataset, comprised of vision, lidar, and inertial sensing collected in two challenging environments. The provision of highly accurate ground truth enables the transparent evaluation of SLAM systems. In this challenge, we found that the top three teams achieved 3 cm accuracy in the construction site sequences but incurred higher errors for the harder sequences from the Sheldonian.

The HILTI SLAM Challenge leaderboard remains live and can accept new submissions for automatic evaluation. We hope this dataset provides researchers with a difficult and diverse challenge to improve their SLAM systems.

VIII. Acknowledgements

We thank the organizers of the Future of Construction workshop at ICRA 2022 for hosting the challenge and Beda Berner from HILTI who produced and calibrated the handheld device used to collect this dataset.

References

[1] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers, "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry," in *IEEE/RSJ Intl Conf. on Intelligent Robots and Systems*, 2018, pp. 1680–1687.

[2] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *Int. Journal of Robotics Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. Journal of Robotics Res.*, vol. 32, no. 11, 2013.

[4] C. Debeunne and D. Vivet, "A Review of Visual-LiDAR Fusion based Simultaneous Localization and Mapping," *Sensors*, vol. 20, no. 7, 2020.

[5] L. Zhang, D. Wisth, M. Camurri, and M. Fallon, "Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1182–1189, 2022.

[6] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots," *IEEE Trans. on Robotics*, 2022.

[7] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Pérez, "WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving," in *IEEE/CVF Intl. Conf. on Computer Vision*, 2019, pp. 9307–9317.

[8] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *Intl. Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.

[9] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza, "The Hilti SLAM Challenge Dataset," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7518–7525, 2022.

[10] L. Zhang, M. Camurri, D. Wisth, and M. Fallon, "Multi-Camera LiDAR Inertial Extension to the Newer College Dataset," *arXiv:2112.08854*, 2021.

[11] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending Kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *IEEE Intl. Conf. on Robotics and Automation*, 2016, pp. 4304–4311.

[12] D. Wujanz, S. Schaller, F. Gielsdorf, and L. Gruendig, "Plane-based registration of several thousand laser scans on standard hardware," *Photogrammetry, Remote Sensing and Spatial Info. Sciences*, vol. XLII-2, pp. 1207–1212, 2018.

[13] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon, "The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2020, pp. 4353–4360.

[14] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM." https://github.com/MichaelGrupp/evo, 2017.

[15] M. Ramezani, K. Khosoussi, G. Catt, P. Moghadam, J. Williams, P. Borges, F. Pauling, and N. Kottege, "Wildcat: Online Continuous-Time 3D Lidar-Inertial SLAM," *arXiv:2205.12595*, 2022.

[16] S. Leutenegger, "OKVIS2: Realtime Scalable Visual-Inertial SLAM with Loop Closure," *arXiv preprint arXiv:2202.09199*, 2022.

[17] F. Neuhaus, T. Koß, R. Kohnen, and D. Paulus, "MC2SLAM: Real-Time Inertial Lidar Odometry Using Two-Scan Motion Compensation," *German Conference on Pattern Recognition*, pp. 60–72, 2019.

[18] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry," *IEEE Trans. on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[19] Z. Liu and F. Zhang, "BALM: Bundle Adjustment for Lidar Mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3184–3191, 2021.

[20] F. A. Maken, F. Ramos, and L. Ott, "Estimating Motion Uncertainty with Bayesian ICP," in *IEEE Intl. Conf. on Robotics and Auto.*, 2020.

[21] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Trans. on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[22] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2020, pp. 4666–4672.

[23] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure," in *IEEE Intl Conf on Robotics and Automation*, 2022, pp. 5580–5586.

[24] M. Yokozuka, S. Oishi, S. Thompson, and A. Banno, "VITAMIN-E: VIsual Tracking and MappINg With Extremely Dense Feature Points," in *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019.

[25] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Globally Consistent and Tightly Coupled 3D LiDAR Inertial Mapping," in *IEEE Intl. Conf. on Robotics and Automation*, 2022, pp. 5622–5628.

[26] F. A. Maken, F. Ramos, and L. Ott, "Estimating motion uncertainty with Bayesian ICP," in *IEEE Intl. Conf. on Robotics and Automation*, 2020, pp. 8602–8608.

[27] T. Nguyen, S. Yuan, M. Cao, T. H. Nguyen, and L. Xie, "VIRAL SLAM: Tightly Coupled Camera-IMU-UWB-Lidar SLAM," *CoRR*, vol. abs/2105.03296, 2021.

[28] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 558–16 569, 2021.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping
Publication Status	x Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and unsubmitted work written in a manuscript style
Publication Details	Lintong Zhang, Michael Helmberger, Lanke Frank Tarimo Fu, David Wisth, Marco Camurri, Davide Scaramuzza, and Maurice Fallon, "Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping," *IEEE Robotics and Automation Letters*, 2023.

Student Confirmation

Student Name:	Lintong Zhang		
Contribution to the Paper	• Collected the dataset and generate ground truth • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	April 16, 2024

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Maurice Fallon			
Supervisor comments Lintong was the lead instigator, implementer and researcher on this project and wrote the bulk of the paper			
Signature		Date	May 03, 2024

This completed form should be included in the thesis, at the end of the relevant chapter.

7.3 Discussion

In this chapter, we described two datasets that were created to support the research in challenging scenarios for odometry and localisation. These datasets incorporate multiple sensors, including lidar, multiple cameras, and an IMU. They are designed to present challenging scenarios for SLAM, such as low-light dark spaces for vision-based SLAM, large empty spaces and areas with degenerate geometry for lidar-based SLAM. The Newer College Dataset (NCD) extension builds upon the original dataset by incorporating multiple cameras and a denser 128-beam lidar, achieving dense ground truth trajectories with centimetre accuracy. Additionally, the Hilti-Oxford dataset aims to provide sparse millimetre-accurate ground truth poses. The latter dataset was the focus of a high profile international challenge in 2022, with particular attention paid to ground truth accuracy.

Since their publication, both datasets have been well-received within the research community and used in numerous studies related to SLAM. One direction for future enhancements would be the incorporation of sequences that use colour imaging sensors. An effort to address this is ongoing within the envelope of the work described in chapter 8, led by my colleague Ethan Tao. This addition would not only broaden the utility of these datasets for tasks involving colour reconstruction but also enhance support for visual place recognition algorithms that depend on colour imagery.

Another critical aspect that warrants discussion regarding the accuracy of dense ground truth poses in the NCD dataset, which currently gives accuracy within a few centimetres. The primary sources of error are identified as follows: 1) the precision of prior map construction, 2) the accuracy of the handheld lidar scanner, and 3) the error from registration of the handheld lidar scan to the prior map. By aligning each lidar scan to the prior map using ICP, we can achieve dense ground truth at the frequency of the lidar, although this process is susceptible to the aforementioned sources of error. Efforts to refine the ground truth accuracy to millimetre-level were undertaken in the Hilti-Oxford dataset by placing survey points on the floor. However, these points are spaced several meters apart, resulting in a spatial density that does not approach that of the NCD ground truth. For

future research, it is beneficial to strive for both dense coverage and millimeter-level accuracy in ground truth generation.

8
Conclusion

This thesis has examined the challenges of odometry and localisation through the integration of vision and lidar sensors. Initially, we provided an overview of the foundational knowledge and algorithms that underpin the discussions in previous chapters. We then reviewed related works, focusing on odometry and localisation techniques that utilise vision and lidar-based methods.

In Chapter 4, we introduced a multi-camera visual-inertial odometry system designed to be robust against common failure scenarios encountered by vision systems. Chapter 5 presented a global lidar localisation system that leverages semantic and instance recognition within prior lidar maps, facilitating its application in indoor environments. Chapter 6 discussed a novel localisation framework that integrates both vision and lidar, enabling cross-modal localisation from cameras to colour lidar 3D maps.

Throughout this thesis, we have explored these themes by incorporating a variety of learning-based techniques for both lidar and vision systems, thereby enhancing their operational capabilities and environmental adaptability. All systems and algorithms developed in this work have been empirically tested using real-world data from robotic platforms or sensor payloads and are capable of operating in real time.

In the following section, we will outline several promising avenues to expand upon the research conducted in this thesis. These proposed areas of further exploration

aim to build on the existing methodologies and insights developed during the thesis.

Multi-Camera and Lidar Inertial Odometry Since the publication of our work outlined in Chapter 4, there have been ongoing advancements in multi-camera odometry and SLAM. Despite this progress, a performance gap remains in terms of accuracy between vision-based SLAM and lidar-based SLAM, where the latter can achieve centimetre-level accuracy. However, lidar-based SLAM systems often encounter limitations in environments with sparse geometric features, such as long corridors or tunnels. Similarly, vision-based SLAM systems can suffer significant performance degradation under varying lighting conditions. Existing research has proposed that a tight integration of camera and lidar sensors could enhance both accuracy and robustness.

In Chapter 6, we have redesigned the hardware setup, as shown in Fig. 6 (left). We reduced four cameras down to three to allow for a more compact assembly. The cameras are recessed into the casing and can maintain calibration over extended periods, with the enclosed casing facilitating outdoor operations.

To address the challenge of vision and lidar integration, we propose several modifications within our existing factor graph framework to better integrate them. First, from a vision perspective, lidar depth information can be overlaid onto camera images to directly acquire image feature depth from the lidar, as illustrated in Chapter 6 Fig. 6 (right). Secondly, we propose utilising joint factor graph optimisation to incorporate vision factors, lidar ICP factors, and pre-integrated IMU factors, following the approach proposed by Wisth *et al.* [141]. An additional enhancement would involve activating vision factors only when lidar factors are likely to fail, ensuring that lidar provides the most accurate pose estimation under normal conditions, with vision activated during potential lidar failures.

This integrated approach, combining multi-camera systems with lidar sensors, promises not only the precision of lidar-based SLAM, but also the added robustness provided by multiple cameras. This makes further exploration into this integrated methodology a worthwhile endeavour.

Object Instance Based Vision Lidar Localisation In Chapter 5, we introduced a method of localising in a prior lidar map with a single lidar. Chapter 6 presented another method of localising in a prior 3D colour map with a single camera. Future work could explore using both modes simultaneously to determine whether this approach is complementary, potentially improving localisation accuracy and providing additional robustness.

Building on the concept of learning through instances introduced in Chapter 5, we suggest that future work could develop enhanced maps that incorporate object instances identified via both vision and lidar data. This approach would ensure that object instances are not only distinctly segmented but also accurately colour-mapped, giving a more detailed and useful representation in the maps. With such a colour instance-based prior map, a tightly coupled vision-lidar localisation system could be developed to jointly estimate poses.

Additionally, we could explore the application of these instance-labelled colour 3D lidar maps in other robotic tasks, such as object retrieval and precise object localisation. This would not only expand the utility of these lidar vision maps but also provide a unified representation of the environments.

Long Term Mapping for Localisation In this thesis, we have demonstrated that the proposed localisation methods can accommodate scene changes to a certain extent. However, due to accumulated changes over months or years, these prior maps will eventually need to be updated. Currently, this update process involves periodically creating a completely new map. Investigating methods for automatic map updates during localisation could significantly enhance the maintenance and utility of these maps, not only for localisation purposes but also for applications such as object and scene understanding.

There are several potential strategies for tackling this issue. One promising approach is to utilise object instances alongside change detection techniques. This would allow dynamic updates to the map. For instance, if a sofa has moved from its recorded position in the prior map, the map could be automatically updated to reflect

the sofa's new location during subsequent localisations. As an additional approach, objects within the environment could be categorised as either stationary or movable, with movable objects specifically excluded from long-term map representations. Alternatively, a more nuanced method could involve a "location change score" for objects that are frequently moved, such as chairs or doors. This score would adjust based on observed movements during each localisation run, providing a scalable and adaptive framework for map updates that reflect the real-time dynamics of the environment and its contents.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. *Ceres Solver*. Version 2.2. 2023. URL: https://github.com/ceres-solver/ceres-solver (page 53).

[2] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. "Fast Radar Motion Estimation with a Learnt Focus of Attention using Weak Supervision". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 1190–1196 (page 1).

[3] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. "What Could Go Wrong? Introspective Radar Odometry in Challenging Environments". In: *IEEE Intelligent Transportation Systems Conf. (ITSC)*. 2019, pp. 2835–2842 (pages 2, 37).

[4] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.6 (2018), pp. 1437–1451 (pages 19, 24, 25, 48).

[5] Ian Baldwin and Paul Newman. "Laser-only road-vehicle localization with dual 2D push-broom LIDARS and 3D priors". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2012, pp. 2490–2497 (page 48).

[6] Tim D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2007 (pages 8, 14).

[7] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. "The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020 (pages 56, 57).

[8] Lukas Bernreiter, Lionel Ott, Juan Nieto, Roland Siegwart, and Cesar Cadena. "Spherical Multi-Modal Place Recognition for Heterogeneous Sensor Systems". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2021, pp. 1743–1750 (page 51).

[9] Gabriele Berton, Gabriele Trivigno, Barbara Caputo, and Carlo Masone. "EigenPlaces: Training Viewpoint Robust Models for Visual Place Recognition". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 11080–11090 (page 48).

[10] Michael Bloesch, Sammy Omari, Marco Hutter, and Roland Siegwart. "Robust Visual Inertial Odometry using a Direct EKF-based Approach". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2015, pp. 298–304 (page 41).

[11] Paulo Borges, Robert Zlot, Michael Bosse, Stephen Nuske, and Ashley Tews. "Vision-based Localization Using an Edge Map Extracted From 3D Laser Range Data". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2010, pp. 4902–4909 (page 51).

[12] Mitch Bryson, Matthew Johnson-Roberson, and Salah Sukkarieh. "Airborne Smoothing and Mapping using Vision and Inertial Sensors". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2009, pp. 2037–2042 (page 40).

[13] Keenan Burnett, David J Yoon, Yuchen Wu, Andrew Z Li, Haowei Zhang, Shichen Lu, Jingxing Qian, Wei-Kang Tseng, Andrew Lambert, Keith YK Leung, Angela P Schoellig, and Timothy D Barfoot. "Boreas: A Multi-Season Autonomous Driving Dataset". In: *Intl. J. of Robot. Res.* 42.1-2 (2023), pp. 33–42 (pages 57, 58).

[14] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. "The EuRoC Micro Aerial Vehicle Datasets". In: *Intl. J. of Robot. Res.* 35.10 (2016), pp. 1157–1163 (page 56).

[15] 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". In: *IEEE Trans. Robotics* (2021), pp. 1–17 (pages 36, 39, 40, 54).

[16] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. "University of Michigan North Campus Long-term Vision and Lidar Dataset". In: *Intl. J. of Robot. Res.* 35.9 (2016), pp. 1023–1035 (page 57).

[17] Sarah Cen and Paul Newman. "Radar-only Ego-motion Estimation in Difficult Settings via Graph Matching". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 298–304 (pages 1, 37).

[18] Yukang Chen, Yanwei Li, X. Zhang, Jian Sun, and Jiaya Jia. "Focal Sparse Convolutional Networks for 3D Object Detection". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5418–5427 (page 47).

[19] Younggun Cho, Giseop Kim, and Ayoung Kim. "Unsupervised Geometry-Aware Deep LiDAR Odometry". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 2145–2152 (page 45).

[20] Christopher Choy, JunYoung Gwak, and Silvio Savarese. "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3075–3084 (page 47).

[21] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction". In: *Eur. Conf. on Computer Vision (ECCV)*. 2016, pp. 628–644 (page 46).

[22] Mark Cummins and Paul Newman. "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance". In: *Intl. J. of Robot. Res.* 27.6 (2008), pp. 647–665 (page 48).

[23] Andrew J. Davison, Ian D. Reid, Nicholas Molton, and Olivier Stasse. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (2007), pp. 1052–1067 (page 54).

[24] Frank Dellaert and Contributors. *GTSAM*. Version 4.2a8. 2022. URL: https://github.com/borglab/gtsam) (page 53).

[25] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Hanover, MA, USA: Now Publishers Inc., 2017 (pages 11, 13).

[26] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. "CT-ICP: Real-time Elastic LiDAR Odometry with Loop Closure". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2022, pp. 5580–5586 (page 43).

[27] Jeffrey Delmerico and Davide Scaramuzza. "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018, pp. 2502–2509 (page 41).

[28] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. "SuperPoint: Self-Supervised Interest Point Detection and Description". In: *IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018, pp. 224–236 (pages 19, 22, 48).

[29] Alexander Dietsche, Giovanni Cioffi, Javier Hidalgo-Carrio, and Davide Scaramuzza. "Powerline Tracking with Event Cameras". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2021 (page 39).

[30] B. Douillard, A. Quadros, P. Morton, J. P. Underwood, M. De Deuge, S. Hugosson, M. Hallström, and T. Bailey. "Scan segments matching for pairwise 3D alignment". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2012, pp. 3033–3040 (page 49).

[31] Renaud Dubé, Andrei Cramariuc, Daniel Dugas, Juan Nieto, Roland Siegwart, and Cesar Cadena. "SegMap: 3D Segment Mapping using Data-Driven Descriptors". In: *Robotics: Science and Systems (RSS)*. Pittsburgh, Pennsylvania, 2018, pp. 20–30 (page 50).

[32] Renaud Dubé, Daniel Dugas, Elena Stumm, Juan Nieto, Roland Siegwart, and Cesar C. Lerma. "SegMatch: Segment Based Place Recognition in 3D Point Clouds". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2017, pp. 5266–5272 (page 50).

[33] Kevin Eckenhoff, Patrick Geneva, and Guoquan Huang. "MIMC-VINS: A Versatile and Resilient Multi-IMU Multi-Camera Visual-Inertial Navigation System". In: *IEEE Trans. Robotics* (2021), pp. 1–21 (page 42).

[34] Jakob Engel, Vladlen Koltun, and Daniel Cremers. "Direct Sparse Odometry". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 40.3 (2018), pp. 611–625 (page 39).

[35] Jakob J. Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *Eur. Conf. on Computer Vision (ECCV)*. 2014 (page 54).

[36] Maurice F. Fallon, John Folkesson, Hunter McClelland, and John J. Leonard. "Relocating Underwater Features Autonomously Using Sonar-Based SLAM". In: *Journal of Oceanic Engineering* 38 (3 2013), pp. 500–513 (page 38).

[37] S. Fazli and L. Kleeman. “Wall Following and Obstacle Avoidance Results from a Multi-DSP Sonar Ring on a Mobile Robot”. In: *IEEE Intl. Conf. Mechatronics and Automation*. Vol. 1. 2005, pp. 432–437 (page 38).

[38] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. “On-Manifold Preintegration for Real-Time Visual–Inertial Odometry”. In: *IEEE Trans. Robotics* 33.1 (2017), pp. 1–21 (page 40).

[39] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. “SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems”. In: *IEEE Trans. Robotics* 33.2 (2017), pp. 249–265 (pages 36, 39, 40).

[40] Matteo Frosi and Matteo Matteucci. “ART-SLAM: Accurate Real-Time 6DoF LiDAR SLAM”. In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 2692–2699 (page 1).

[41] Matteo Frosi and Matteo Matteucci. “ART-SLAM: Accurate Real-Time 6DoF LiDAR SLAM”. In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 2692–2699 (page 55).

[42] Lanke Frank Tarimo Fu, Nived Chebrolu, and Maurice Fallon. “Extrinsic Calibration of Camera to LIDAR Using a Differentiable Checkerboard Model”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2023, pp. 1825–1831 (page 70).

[43] Paul Furgale, Joern Rehder, and Roland Siegwart. “Unified temporal and spatial calibration for multi-sensor systems”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2013, pp. 1280–1286 (page 70).

[44] Paul T Furgale and Timothy D Barfoot. “Visual Teach and Repeat for Long-range Rover Autonomy”. In: *J. Field Robot.* 27.5 (2010), pp. 534–560 (page 1).

[45] Guillermo Gallego, Tobi Delbruck, Garrick Michael Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jorg Conradt, Kostas Daniilidis, and Davide Scaramuzza. “Event-based Vision: A Survey”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* (2020), pp. 1–1 (page 39).

[46] Dorian Galvez-López and Juan D. Tardos. “Bags of Binary Words for Fast Place Recognition in Image Sequences”. In: *IEEE Trans. Robotics* 28.5 (2012), pp. 1188–1197 (pages 24, 48, 54).

[47] Xiang Gao and Zhang Tao. “Unsupervised learning to detect loops using deep neural networks for visual SLAM system”. In: *Autonomous Robots* 41 (2015), pp. 1 –18 (page 54).

[48] A Geiger, P Lenz, C Stiller, and R Urtasun. “Vision Meets Robotics: The KITTI Dataset”. In: *Intl. J. of Robot. Res.* 32.11 (2013) (pages 56, 57).

[49] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. “OpenVINS: A Research Platform for Visual-Inertial Estimation”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 4666–4672 (pages 1, 41).

[50] A. Georgiev and P.K. Allen. “Localization methods for a mobile robot in urban environments”. In: *IEEE Trans. Robotics* 20.5 (2004), pp. 851–864 (page 38).

[51] Riccardo Giubilato, Wolfgang Stürzl, Armin Wedler, and Rudolph Triebel. "Challenges of SLAM in Extremely Unstructured Environments: The DLR Planetary Stereo, Solid-State LiDAR, Inertial Dataset". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.4 (2022), pp. 8721–8728 (page 58).

[52] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. "3D Semantic Segmentation with Submanifold Sparse Convolutional Networks". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 9224–9232 (page 47).

[53] Adam L. Harmat, Michael Trentini, and Inna Sharf. "Multi-Camera Tracking and Mapping for Unmanned Aerial Vehicles in Unstructured Environments". In: *J. of Intelligent & Robotic Systems* 78 (2014), pp. 291 –317 (page 41).

[54] Stephen Hausler, Sourav Garg, Ming Xu, Michael Milford, and Tobias Fischer. "Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 14141–14152 (page 48).

[55] Li He, Xiaolong Wang, and Hong Zhang. "M2DP: A Novel 3D Point Cloud Descriptor and its Application in Loop Closure Detection". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2016, pp. 231–237 (page 49).

[56] Ming He, Chaozheng Zhu, Qian Huang, Baosen Ren, and Jintao Liu. "A Review of Monocular Visual Odometry". In: *The Visual Computer* 36.5 (2020), pp. 1053–1065 (page 41).

[57] Yao He, Huai Yu, Wen Yang, and Sebastian Scherer. "Towards Robust Visual-Inertial Odometry with Multiple Non-Overlapping Monocular Cameras". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2022, pp. 9452–9458 (page 70).

[58] Michael Helmberger, Kristian Morin, Beda Berner, Nitish Kumar, Giovanni Cioffi, and Davide Scaramuzza. "The Hilti SLAM Challenge Dataset". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.3 (2022), pp. 7518–7525 (page 57).

[59] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. "Real-time Loop Closure in 2D LIDAR SLAM". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2016, pp. 1271–1278 (page 55).

[60] F.S. Hover, R.M. Eustice, A. Kim, B.J. Englot, H. Johannsson, M. Kaess, and J.J. Leonard. "Advanced Perception, Navigation and Planning for Autonomous In-Water Ship Hull Inspection". In: *Intl. J. of Robot. Res.* 31.12 (2012), pp. 1445–1464 (page 38).

[61] Ibrahim Hroob, Benedikt Mersch, C. Stachniss, and Marc Hanheide. "Generalizable Stable Points Segmentation for 3D LiDAR Scan-to-Map Long-Term Localization". In: *IEEE Robot. Autom. Lett. (RA-L)* 9 (2024), pp. 3546–3553 (page 47).

[62] Albert S. Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. "Visual Odometry and Mapping for Autonomous Flight using an RGB-D Camera". In: *Intl. J. of Robot. Res.* Springer, 2017, pp. 235–252 (page 39).

[63] *Introducing Deep Learning with MATLAB*. https://uk.mathworks.com/campaigns/offers/next/deep-learning-ebook.html (page 18).

[64] Joshua Jaekel, Joshua G. Mangelson, Sebastian Scherer, and Michael Kaess. "A Robust Multi-Stereo Visual-Inertial Odometry Pipeline". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 4623–4630 (page 41).

[65] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. "iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2011, pp. 3281–3288 (page 53).

[66] J. Kannala and S.S. Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 28.8 (2006), pp. 1335–1340 (page 21).

[67] Christina Kassab, Matias Mattamala, Lintong Zhang, and Maurice Fallon. "Language-EXtended Indoor SLAM (LEXIS): A Versatile System for Real-time Visual Scene Understanding". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2024 (pages 5, 70).

[68] Pushyami Kaveti, Aniket Gupta, Dennis Giaya, Madeline Karp, Colin Keil, Jagatpreet Nir, Zhiyong Zhang, and Hanumant Singh. "Challenges of Indoor SLAM: A Multi-Modal Multi-Floor Dataset for SLAM Evaluation". In: *IEEE Intl. Conf. on Automation Science and Engineering (CASE)*. 2023, pp. 1–8 (page 57).

[69] Pushyami Kaveti, Shankara Narayanan Vaidyanathan, Arvind Thamil Chelvan, and Hanumant Singh. "Design and Evaluation of a Generic Visual SLAM Framework for Multi Camera Systems". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.11 (2023), pp. 7368–7375 (page 70).

[70] Christian Kerl, Jürgen Sturm, and Daniel Cremers. "Robust odometry estimation for RGB-D cameras". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2013, pp. 3748–3754 (page 39).

[71] D.-H Kim, S.-B Han, and Jun Hyun Kim. "Visual Odometry Algorithm Using an RGB-D Sensor and IMU in a Highly Dynamic Environment". In: *Advances in Intelligent Systems and Computing* 345 (2015), pp. 11–26 (page 39).

[72] Giseop Kim and Ayoung Kim. "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 4802–4809 (page 49).

[73] Hanme Kim, Stefan Leutenegger, and Andrew J Davison. "Real-time 3D Reconstruction and 6-DoF Tracking with an Event Camera". In: *Eur. Conf. on Computer Vision (ECCV)*. 2016, pp. 349–364 (page 39).

[74] Georg S. W. Klein and David William Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *IEEE Intl. Sym. on Mixed and Augmented Reality* (2007), pp. 225–234 (page 54).

[75] Kenji Koide, Jun Miura, and Emanuele Menegatti. "A Portable Three-dimensional LIDAR-based System for Long-term and Wide-area People Behavior Measurement". In: *Intl. J. of Advanced Robotic Systems* 16 (2019) (page 55).

[76] Xin Kong, Xuemeng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Feng Wen. "Semantic Graph Based Place Recognition for 3D Point Clouds". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 8216–8223 (page 49).

[77] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. "G2O: A General Framework for Graph Optimization". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2011, pp. 3607–3613 (page 53).

[78] Haowen Lai, Peng Yin, and Sebastian Scherer. "AdaFusion: Visual-LiDAR Fusion With Adaptive Weights for Place Recognition". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.4 (2022), pp. 12038–12045 (page 51).

[79] Alex Junho Lee, Seungwon Song, Hyungtae Lim, Woojoo Lee, and Hyun Myung. "$(LC)^2$: LiDAR-Camera Loop Constraints for Cross-Modal Place Recognition". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.6 (2023), pp. 3589–3596 (page 51).

[80] John J. Leonard and Hugh F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992 (page 37).

[81] Stefan Leutenegger, Paul Timothy Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Y. Siegwart. "Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization". In: *Robotics: Science and Systems (RSS)*. 2013 (page 53).

[82] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. "Keyframe-based Visual–inertial Odometry using Nonlinear Optimization". In: *Intl. J. of Robot. Res.* 34.3 (2015), pp. 314–334 (pages 36, 40).

[83] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun. "Map-based Precision Vehicle Localization in Urban Environments". In: *Robotics: Science and Systems (RSS)*. 2007, pp. 110–118 (page 38).

[84] Dongjiang Li, Xuesong Shi, Qiwei Long, Shenghui Liu, Wei Yang, Fangshi Wang, Qi Wei, and Fei Qiao. "DXSLAM: A Robust and Efficient Visual SLAM System with Deep Features". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020 (page 54).

[85] Qing Li, Shaoyang Chen, Cheng Wang, Xin Li, Chenglu Wen, Ming Cheng, and Jonathan Li. "LO-Net: Deep Real-Time Lidar Odometry". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8465–8474 (page 45).

[86] Wen Li, Shangshu Yu, Cheng Wang, Guosheng Hu, Siqi Shen, and Chenglu Wen. "SGLoc: Scene Geometry Encoding for Outdoor LiDAR Localization". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 9286–9295 (page 47).

[87] Xingchen Li, Yuxuan Xiao, Beibei Wang, Haojie Ren, Yanyong Zhang, and Jianmin Ji. "Automatic Targetless LiDAR-Camera Calibration: A Survey". In: *Artificial Intelligence Review* (2022) (page 22).

[88] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. "PointCNN: Convolution On X-Transformed Points". In: *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*. 2018 (page 46).

[89] Shiwen Liang, Yunzhou Zhang, Rui Tian, Delong Zhu, Linghao Yang, and Zhenzhong Cao. "SemLoc: Accurate and Robust Visual Localization with Semantic and Structural Constraints from Prior Maps". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2022, pp. 4135–4141 (page 52).

[90] Peidong Liu, Marcel Geppert, Lionel Heng, Torsten Sattler, Andreas Geiger, and Marc Pollefeys. "Towards Robust Visual Odometry with a Multi-Camera System". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018 (page 42).

[91] David G Lowe. "Distinctive Image Features from Scale-invariant Keypoints". In: *Intl. J. of Computer Vision* 60 (2004), pp. 91–110 (page 48).

[92] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. "DeepICP: An End-to-End Deep Neural Network for Point Cloud Registration". In: 2019, pp. 12–21 (page 45).

[93] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. "ContextDesc: Local Descriptor Augmentation With Cross-Modality Context". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2522–2531 (page 46).

[94] Zixin Luo, Tianwei Shen, Lei Zhou, Siyu Zhu, Runze Zhang, Yao Yao, Tian Fang, and Long Quan. "GeoDesc: Learning Local Descriptors by Integrating Geometry Constraints". In: *Eur. Conf. on Computer Vision (ECCV)*. 2018 (page 46).

[95] John McCormac, Ronald Clark, Michael Bloesch, Andrew Davison, and Stefan Leutenegger. "Fusion++: Volumetric Object-level SLAM". In: *Intl. Conf. on 3D vision (3DV)*. 2018, pp. 32–41 (page 54).

[96] Colin McManus, Ben Upcroft, and Paul Newman. "Scene Signatures: Localised and Point-less Features for Localisation". In: *Robotics: Science and Systems (RSS)*. Berkeley, CA, USA, 2014, pp. 177 –185 (page 36).

[97] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Eur. Conf. on Computer Vision (ECCV)*. 2020 (page 17).

[98] Michael Milford, Hanme Kim, Stefan Leutenegger, and Andrew Davison. "Towards Visual SLAM with Event-based Cameras". In: *The problem of mobile sensors workshop in conjunction with RSS*. 2015 (page 39).

[99] Anastasios I Mourikis and Stergios I Roumeliotis. "A Multi-state Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2007, pp. 3565–3572 (page 41).

[100] Anastasios I. Mourikis and Stergios I. Roumeliotis. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2007, pp. 3565–3572 (page 54).

[101] M. G. Müller, F. Steidle, M. J. Schuster, P. Lutz, M. Maier, S. Stoneman, T. Tomic, and W. Sturzl. "Robust Visual-Inertial State Estimation with Multiple Odometries and Efficient Mapping on an MAV with Ultra-Wide FOV Stereo Vision". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2018), pp. 3701–3708 (page 42).

[102] Raul Mur-Artal, José M. M. Montiel, and Juan D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Trans. Robotics* 31 (2015), pp. 1147–1163 (page 54).

[103] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew William Fitzgibbon. "KinectFusion: Real-time Dense Surface Mapping and Tracking". In: *IEEE Intl. Sym. on Mixed and Augmented Reality* (2011), pp. 127–136 (page 54).

[104] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense Tracking and Mapping in Real-time". In: *Intl. Conf. on Computer Vision (ICCV)*. 2011, pp. 2320–2327 (page 39).

[105] Eric North, Jacques Georgy, Mohammed Tarbouchi, Umar Iqbal, and Aboelmagd Noureldin. "Enhanced Mobile Robot Outdoor Localization Using INS/GPS Integration". In: *Intl. Conf. on Computer Engineering and Systems (ICCES)*. 2010, pp. 127 –132 (page 38).

[106] Vojtech Panek, Zuzana Kukelova, and Torsten Sattler. "MeshLoc: Mesh-Based Visual Localization". In: *Eur. Conf. on Computer Vision (ECCV)*. 2022 (page 48).

[107] Chanoh Park, Peyman Moghadam, Jason L. Williams, Soohwan Kim, Sridha Sridharan, and Clinton Fookes. "Elasticity Meets Continuous-Time: Map-Centric Dense 3D LiDAR SLAM". In: *IEEE Trans. Robotics* 38.2 (2022), pp. 978–997 (page 55).

[108] Geoffrey Pascoe, William Maddern, and Paul Newman. "Direct Visual Localisation and Calibration for Road Vehicles in Changing City Environments". In: *IEEE Intl. Conf. on Computer Vision Workshop (ICCVW)*. 2015, pp. 98–105 (page 52).

[109] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space". In: *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*. 2017, 5105–5114 (pages 45, 46).

[110] Tong Qin, Peiliang Li, and Shaojie Shen. "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator". In: *IEEE Trans. Robotics* 34.4 (2018), pp. 1004–1020 (pages 1, 40).

[111] Milad Ramezani, Kasra Khosoussi, Gavin Catt, Peyman Moghadam, Jason L. Williams, Paulo Borges, Fred Pauling, and Navinda Kottege. "Wildcat: Online Continuous-Time 3D Lidar-Inertial SLAM". In: *ArXiv* abs/2205.12595 (2022) (page 55).

[112] Sebastian Ratz, Marcin Dymczyk, Roland Y. Siegwart, and Renaud Dubé. "OneShot Global Localization: Instant LiDAR-Visual Pose Estimation". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 5415–5421 (page 51).

[113] Jerome Revaud, Philippe Weinzaepfel, César Roberto de Souza, and Martin Humenberger. "R2D2: Repeatable and Reliable Detector and Descriptor". In: *Intl. Conf. on Neural Information Processing Systems (NeurIPS)*. 2019 (page 48).

[114] Edward Rosten, Reid Porter, and Tom Drummond. "Faster and Better: A Machine Learning Approach to Corner Detection". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.1 (2010), pp. 105–119 (page 22).

[115] Joseph Rowell, Lintong Zhang, and Maurice Fallon. "LiSTA: Geometric Object-Based Change Detection in Cluttered Environments". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2024 (pages 5, 88).

[116] Stefan Saftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. "Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 4358–4364 (page 37).

[117] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. "From Coarse to Fine: Robust Hierarchical Localization at Large Scale". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019 (page 48).

[118] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. "Efficient and Effective Prioritized Matching for Large-Scale Image-Based Localization". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 39.9 (2017), pp. 1744–1756 (page 48).

[119] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2016 (page 48).

[120] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, and D. Cremers. "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 1680–1687 (page 56).

[121] Hochang Seok and Jongwoo Lim. "ROVINS: Robust Omnidirectional Visual Inertial Navigation System". In: *IEEE Robot. Autom. Lett. (RA-L)* 5.4 (2020), pp. 6225–6232 (page 42).

[122] Tixiao Shan and Brendan Englot. "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 4758–4765 (pages 37, 44).

[123] Shaojie Shen, Nathan Michael, and Vijay Kumar. "Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2015, pp. 5303–5310 (page 40).

[124] Kshitij Sirohi, Rohit Mohan, Daniel Buscher, Wolfram Burgard, and Abhinav Valada. "EfficientLPS: Efficient LiDAR Panoptic Segmentation". In: *IEEE Trans. Robotics* 38 (2021), pp. 1894–1914 (page 47).

[125] K. Tateno, F. Tombari, I. Laina, and N. Navab. "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6565–6574 (page 54).

[126] S. Thongchai and K. Kawamura. "Application of Fuzzy Control to a Sonar-based Obstacle Avoidance Mobile Robot". In: *IEEE Intl. Conf. on Control Applications*. 2000, pp. 425–430 (page 38).

[127] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. "Monte Carlo Localization for Mobile Robots". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 1999, pp. 1322–1328 (page 43).

[128] Georgi Tinchev, Simona Nobili, and Maurice Fallon. "Seeing the Wood for the Trees: Reliable Localization in Urban and Natural Environments". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2018, pp. 8239–8246 (page 50).

[129] Georgi Tinchev, Adrian Penate-Sanchez, and Maurice Fallon. "Learning to See the Wood for the Trees: Deep Laser Localization in Urban and Natural Environments on a CPU". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2019, pp. 1327–1334 (page 50).

[130] Marco Tranzatto et al. "Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned". In: *Journal of Field Robotics* (2022) (page 2).

[131] Michael J. Tribou, Adam Harmat, David W.L. Wang, Inna Sharf, and Steven L. Waslander. "Multi-Camera Parallel Tracking and Mapping with Non-Overlapping Fields of View". In: *Intl. J. of Robot. Res.* 34.12 (2015), pp. 1480–1500 (page 41).

[132] Mikaela Angelina Uy and Gim Hee Lee. "PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4470–4479 (page 49).

[133] Kavisha Vidanapathirana, Peyman Moghadam, Ben Harwood, Muming Zhao, Sridha Sridharan, and Clinton Fookes. "Locus: LiDAR-based Place Recognition using Spatiotemporal Higher-Order Pooling". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2020, pp. 5075–5081 (page 50).

[134] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. "LoGG3D-Net: Locally Guided Global Descriptor Learning for 3D Place Recognition". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2022, pp. 2215–2221 (page 49).

[135] Antonio Vitale, Alpha Renner, Celine Nauer, Davide Scaramuzza, and Yulia Sandamirskaya. "Event-driven Vision and Control for UAVs on a Neuromorphic Chip". In: (2021) (page 39).

[136] 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". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.2 (2023), pp. 1029–1036 (page 43).

[137] Thang Vu, Kookhoi Kim, Tung Minh Luu, Xuan Thanh Nguyen, and Chang-Dong Yoo. "SoftGroup for 3D Instance Segmentation on Point Clouds". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 2698–2707 (pages 32, 47).

[138] Jianeng Wang, Matias Mattamala, Christina Kassab, Lintong Zhang, and Maurice Fallon. *Exosense: A Vision-Centric Scene Understanding System For Safe Exoskeleton Navigation*. 2024 (page 5).

[139] Yifu Wang, Yonhon Ng, Inkyu Sa, Alvaro Parra, Cristian Rodriguez, Tao Jun Lin, and Hongdong Li. *MAVIS: Multi-Camera Augmented Visual-Inertial SLAM using SE2(3) Based Exact IMU Pre-integration*. 2023 (page 70).

[140] Thomas Whelan, Stefan Leutenegger, Renato F. Salas-Moreno, B. Glocker, and A. Davison. "ElasticFusion: Dense SLAM Without A Pose Graph". In: *Robotics: Science and Systems (RSS)*. 2015 (page 54).

[141] David Wisth, Marco Camurri, and Maurice Fallon. "VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots". In: *IEEE Trans. Robotics* 39.1 (2023), pp. 309–326 (pages 45, 128).

[142] Ryan W. Wolcott and Ryan M. Eustice. "Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving". In: *Intl. J. of Robot. Res.* 36 (2017), pp. 292 –319 (page 49).

[143] Ryan W. Wolcott and Ryan M. Eustice. "Visual localization within LIDAR maps for automated urban driving". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2014, pp. 176–183 (page 52).

[144] Yuchen Wu, David J. Yoon, Keenan Burnett, Soeren Kammel, Yi Chen, Heethesh Vhavle, and Tim D. Barfoot. "Picking up Speed: Continuous-Time Lidar-Only Odometry Using Doppler Velocity Measurements". In: *IEEE Robot. Autom. Lett. (RA-L)* 8 (2022) (page 37).

[145] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. "FAST-LIO2: Fast Direct LiDAR-Inertial Odometry". In: *IEEE Trans. Robotics* 38.4 (2022), pp. 2053–2073 (pages 1, 37, 44).

[146] Fei Xue, Ignas Budvytis, Daniel Olmeda Reino, and Roberto Cipolla. "Efficient Large-scale Localization by Global Instance Recognition". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 17327–17336 (page 48).

[147] Senthil Yogamani, Ciaran Hughes, Jonathan Horgan, Ganesh Sistu, Sumanth Chennupati, Michal Uricar, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed, Sanjaya Nayak, Saquib Mansoor, Padraig Varley, Xavier Perrotton, Derek Odea, and Patrick Pérez. "WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving". In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 9307–9317 (page 57).

[148] Chao Yu, Zuxin Liu, Xinjun Liu, Fugui Xie, Yi Yang, Qi Wei, and Fei Qiao. "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2018), pp. 1168–1174 (pages 36, 54).

[149] Huai Yu, Weikun Zhen, Wen Yang, Ji Zhang, and Sebastian A. Scherer. "Monocular Camera Localization in Prior LiDAR Maps with 2D-3D Line Correspondences". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)* (2020), pp. 4588–4594 (page 51).

[150] Wenzhen Yuan and Srikumar Ramalingam. "Fast localization and tracking using event sensors". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2016, pp. 4564–4571 (page 39).

[151] Ji Zhang and Sanjiv Singh. "LOAM : Lidar Odometry and Mapping in real-time". In: *Robotics: Science and Systems (RSS)* (2014), pp. 109–111 (pages 1, 37, 44).

[152] Lintong Zhang, Michael Helmberger, Lanke Frank Tarimo Fu, David Wisth, Marco Camurri, Davide Scaramuzza, and Maurice Fallon. "Hilti-Oxford Dataset: A Millimeter-Accurate Benchmark for Simultaneous Localization and Mapping". In: *IEEE Robot. Autom. Lett. (RA-L)* 8.1 (2023), pp. 408–415 (pages 56, 71, 115).

[153] Lintong Zhang, Sundara Tejaswi Digumarti, Georgi Tinchev, and Maurice Fallon. "InstaLoc: One-shot Global Lidar Localisation in Indoor Environments through Instance Learning". In: *Robotics: Science and Systems (RSS)*. 2023 (page 73).

[154] Lintong Zhang, David Wisth, Marco Camurri, and Maurice Fallon. "Balancing the Budget: Feature Selection and Tracking for Multi-Camera Visual-Inertial Odometry". In: *IEEE Robot. Autom. Lett. (RA-L)* 7.2 (2022), pp. 1182–1189 (page 60).

[155] Nan Zhang, Michael Warren, and Timothy D. Barfoot. "Learning Place-and-Time-Dependent Binary Descriptors for Long-Term Visual Localization". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2018, pp. 828–835 (pages 2, 36).

[156] Yi Zhou, Guillermo Gallego, and Shaojie Shen. "Event-Based Stereo Visual Odometry". In: *IEEE Trans. Robotics* (2021), pp. 1–18 (page 39).

[157] Xingxing Zuo, Wenlong Ye, Yulin Yang, Renjie Zheng, Teresa Vidal-Calleja, Guoquan Huang, and Yong Liu. "Multimodal localization: Stereo over LiDAR map". In: *J. Field Robot.* 37.6 (2020), pp. 1003–1026 (page 52).