Uncertainty-Aware Lidar-Visual 3D Reconstruction



Yifu Tao

Hertford College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

2025

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.



Yifu Tao, Hertford College
June 22, 2025

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my DPhil supervisor Prof. Maurice Fallon, for his insightful guidance and tremendous support throughout my DPhil journey. From frequent and valuable feedback, facilitating collaborations to detailed reviewing of my writings, his expertise and efforts have been instrumental in shaping this work.

I am also profoundly grateful to my mentors and collaborators: Nived Chebrolu, Marija Popović, Yiduo Wang, Matias Mattamala, Lintong Zhang, Miguel Ángel Muñoz-Bañón, Yash Bhalgat, Tejaswi Sundara Digumati, Rowan Border, Milad Ramezani, Frank Fu, Jiahao Wang, Marco Cammuri, Johnathan Gammell, Jiarong Lin, Fu Zhang. Your insightful feedback, constructive discussions and support have significantly enriched my work. Your generosity in sharing knowledge and time has been truly inspiring.

A special thanks to my colleagues and friends in DRS and the ORI: Christina Kassab, Tobit Flatscher, Ricardo Cannizzaro, Joseph Jowell, Haedam Oh, Shaohong Zhong, Jianeng Wang, Matthew Graham, Benoit Casseau, Russell Buchanan, Fengyu Zhong, Jin Jin, and many more. Dphil life is much more joyful and exciting with you guys. I am also thankful to the broader engineering team for their help in hardware and software support, and the admin team for coordinating and facilitating travels and trials.

I am thankful to my examiners, Ayoung Kim and Victor Adrian Prisacariu, for taking their time to provide valuable feedback and discussion on my research.

To my partner Yaru Zhang, thank you for being by my side throughout this incredible journey. Your encouragement carried me through the most challenging times. I am deeply grateful for your dedication, even if we were often far apart. The time we spent travelling together in Europe and in China brought light and joy to this Dphil journey.

最后，致我的母亲凌彩飞与父亲陶锡红，感谢你们的养育之恩，一路无条件的支持我，给予我你们不曾获得过的宝贵的教育机会。你们的勤劳和勇敢为我树立了榜样。希望儿子获得的成果能让你们感到自豪。

Abstract

3D reconstruction is a fundamental research problem in robotics and computer vision. Recent advances in learning-based depth estimation and radiance fields offer promising alternatives to classical reconstruction methods, enabling new possibilities for reality capture, robot simulation, and navigation. However, applying these techniques to large-scale outdoor 3D reconstruction remains challenging, particularly due to the inaccuracies in the neural networks' outputs when tested on real-world data collected by mobile robots.

This thesis explores strategies for integrating learning-based depth estimation and radiance fields into scalable dense 3D reconstruction systems. In particular, we developed sensor fusion strategies to combine the complementary strengths of lidar and visual inputs, while also quantifying the uncertainty in the outputs of learning-based approaches which enhances reconstruction accuracy and reliability.

Two entirely different systems for lidar-visual 3D reconstruction are developed. The first method is an online geometric mapping system integrated with probabilistic depth completion. We train a network to be able to upsample, with both interpolation and extrapolation, sparse lidar data using images from an overlapping camera. The network also outputs uncertainty estimates of the completed depths, which are integrated probabilistically into a real-time occupancy mapping system. This online system is particularly useful for path planning and obstacle avoidance.

The second system is an offline, textured mapping method based on neural radiance field (NeRF). This approach builds a dense textured map capable of rendering photorealistic virtual colour images. We incorporate lidar data into the framework by using its range measurements to impose geometric constraints on the depths and surface normals of observed surfaces. This system can scale to large-scale environments ($>15\,000\,\mathrm{m}^2$) with submapping. It is also integrated with a lidar SLAM system to bootstrap the pre-processing of the data into posed images. This method is then further extended with a component to properly quantify the epistemic map uncertainty. Specifically, we estimate the covariance of the posterior distribution of the map parameters in a tractable way using the Laplace approximation and Fisher information matrix. This formulation enables a principled decoupling of the map uncertainty into a visual component, which corresponds to textureless areas, and a lidar component, which corresponds to areas outside the lidar's sensing range.

Finally, we also present an extensive multi-site lidar-visual dataset covering areas over 125 000 m^2, which is suitable for evaluating 3D reconstructions' geometric quality and texture quality. In particular, the dataset includes precise ground truth 3D maps from a terrestrial laser scanner. The multiple colour cameras of the mobile mapping device facilitates the evaluation of radiance field approaches in a SLAM context, and the benchmark we evaluated in the dataset reveals their current limitations of generalisation. We believe that the methods presented in this thesis and this dataset can contribute to the development of the next generation of 3D reconstruction systems.

Contents

List of Figures **vii**

List of Abbreviations **ix**

1 Introduction **1**

1.1 Motivation 1

1.2 Contribution 3

1.3 Publications 4

1.3.1 First-Authored Publications 5

1.3.2 Co-authored Publications 5

1.4 Thesis Outline 6

2 Background **7**

2.1 Preliminaries 7

2.1.1 Notation 7

2.1.2 Bayesian Probability 8

2.1.3 Geometric Transformations 11

2.2 Sensors 13

2.2.1 Cameras 13

2.2.2 Lidar 16

2.3 Representing the Map 18

2.3.1 Formulation 18

2.3.2 Point Representations 19

2.3.3 Surface Representations 19

2.3.4 Volumetric Representations 20

2.4 Building the Map 21

2.4.1 Occupancy Mapping 22

2.4.2 Volumetric Rendering 23

3 Literature Review **25**
3.1 Localisation and Reconstruction from Vision 26
3.1.1 Classical Visual SLAM 26
3.1.2 Learning-aided Visual SLAM 28
3.2 Localisation and Reconstruction from Range Data 29
3.2.1 RGB-D SLAM 30
3.2.2 Lidar SLAM 31
3.3 Lidar-Visual SLAM 32
3.4 Learning-based Depth Estimation 35
3.4.1 Depth Completion using Vision and Depth 35
3.4.2 Depth Estimation from Vision 38
3.5 Radiance Fields 38
3.5.1 NeRF 39
3.5.2 New Formulations and Improving Accuracy 40
3.5.3 Explicit Map Representations for Computation Efficiency . . 41
3.6 Datasets 47
3.6.1 Localisation 47
3.6.2 3D Reconstruction 47
3.6.3 Novel-view Synthesis 48

4 Probabilistic Depth Completion **49**
4.1 3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation 50
4.2 Technical Remarks 60
4.3 Discussion 63

5 Scalable Lidar-Visual NeRF **65**
5.1 SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection 66
5.2 Technical Remarks 75
5.3 Discussion 78

6 Epistemic Uncertainty Quantification in Lidar-Visual NeRF **82**
6.1 SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification 83
6.2 Technical Remarks 101
6.3 Discussion 102

7 Lidar-Visual Dataset for SLAM and Radiance Fields **107**
7.1 The Oxford Spires Dataset: Benchmarking Large-Scale Lidar-Visual Localisation, Reconstruction and Radiance Field Methods 108
7.2 Technical Remarks 126
7.3 Discussion 127

8 Conclusion **129**
8.1 Summary 129
8.2 Future Work 131
8.2.1 Learnt Priors with Radiance Fields 131
8.2.2 Spatial Temporal Scalability for Long-term Mapping 131
8.2.3 Graph-based Scene Representations 132

References **134**

List of Figures

2.1 Illustration of the linear camera projection model. Image from OpenCV [39]. 16
2.2 Generic camera model which is able to model (i) perspective projection (ii) stereographic projection (iii) equidistance projection (iv) equisolid angle projection (v) orthogonal projection, with a unit focal length $f = 1$. r is the distance between the image point and the principal point, and θ is the angle between the optical axis and the incoming ray. Figure from [40]. 17
2.3 Probabilistic graphical model of the 3D reconstruction problem. M is the map, $z_{1...4}$ are the sensor observations, $T_{1...4}$ are the sensor poses. 21

3.1 Probabilistic graphical models of a full SfM (as Markov Random Field), filtering-based SLAM and Keyframe-based SLAM from [49]. A filtering approach marginalises previous poses, but keeps the feature landmarks which can be revisited in future frames. An optimisation approach solves a sparse graph of previous poses (keyframes) and landmarks. 26
3.2 An overview of NeRF's MLP-based radiance field representation and the differential rendering pipeline from [25] 40

4.1 Depth completion results when trained with 64-beam lidar depth as ground truth. The network predicts reasonable results only in specific locations in the image where there is ground truth depth. . 60
4.2 A depth completion network tested with in-distribution input and out-of-distribution input. The network does not generalise to an input with different levels of depth sparsity, even if there is more depth information available which should make the completion task easier. 61

5.1 We used a segmentation network to obtain an image mask for the sky, which is used to regularise the corresponding rays to be empty and therefore removes the floating sky artefacts. 75
5.2 Images that are under-exposed and over-exposed, collected in a forest. 76

5.3 COLMAP results of forest images. Results on the left use the original images which are under-exposed under the canopy, and results on the right use images that are processed by histogram equalisation, which is shown to benefit the image registration. 77
5.4 COLMAP fails to register two image sequences where only monocular images are used. 78
5.5 The complete COLMAP trajectory when using three camera image sequences. 79

6.1 Updated version of Figure 9 in the original paper submission. Here we compare between visual and lidar uncertainty. The middle and right reconstructions on the top row are coloured by uncertainty estimates. Here, the lidar used was a Hesai XT32 which has a narrow field-of-view and cannot scan the upper part of the buildings, and the depth uncertainty estimates can identify such regions (red point clouds). This indicates that only visual information is used when reconstructing these areas. 101
6.2 Different Bayesian deep learning techniques for posterior approximation with different characteristics. Image from Papamarkou *et al.* [193]. 103

7.1 Images of the Bodleian library with different exposure times. The low dynamic range camera is only able to capture either the dark or the bright region in the scene. 126
7.2 Colourised TLS cloud projected onto a 2D image plane. 128

List of Abbreviations

3D . . . . . Three dimensional

3DGS . . . . . 3D Gaussian Splatting

CNN . . . . . Convolutional Neural Networks

FoV . . . . . Field-of-View

GP . . . . . Gaussian Process

GPU . . . . . Graphics Processing Unit

ICP . . . . . Iterative Closest Point

IMU . . . . . Inertial Measurement Unit

MAP . . . . . Maximum A Posteriori

MLE . . . . . Maximum Likelihood Estimation

MLP . . . . . Multi-layer Perceptron

MVS . . . . . Multi-view Stereo

NeRF . . . . . Neural Radiance Field

RGB . . . . . Red-Green-Blue

RGB-D . . . . RGB-Depth

SLAM . . . . . Simultaneous Localisation And Mapping

SfM . . . . . Structure-from-Motion

TLS . . . . . Terrestrial Laser Scanner

1 Introduction

Contents

1.1 Motivation **1**
1.2 Contribution **3**
1.3 Publications **4**
1.3.1 First-Authored Publications 5
1.3.2 Co-authored Publications 5
1.4 Thesis Outline **6**

1.1 Motivation

3D reconstruction is a fundamental problem in robotics and computer vision. A dense 3D map facilitates downstream applications such as robot navigation [1], reality capture, and building inspection [2]. In mobile robotics, 3D reconstruction is often solved concurrently with localisation by Simultaneous Localisation and Mapping (SLAM). 3D reconstruction of outdoor environments primarily relies on two exteroceptive sensor modalities: cameras and lidars. In recent years, significant progress has been made in large-scale dense 3D reconstruction using vision-based [3–5] or lidar-based [6–8] methods.

Fusing multi-modal sensor inputs has proven valuable for both localisation and mapping, as lidar and camera data complement each other. Lidar sensors provide

long-range distance measurements that are accurate but sparse, while cameras capture dense visual information with no direct geometric measurements. Combining these sensor modalities can effectively increase localisation robustness [9–14], particularly in challenging environments such as in the DARPA SubT Challenge [15] where visual localisation is affected by poor lighting and lidar localisation struggles with geometrically similar structures such as tunnels. Sensor fusion has also been adopted for 3D reconstructions, and recent SLAM systems have leveraged both lidar and vision to create textured 3D reconstructions, including colourised lidar point clouds [16, 17] and textured meshes [18].

Parallel to these developments, learning-based depth estimation [19–24] and radiance field [25–29] emerge as promising alternatives for dense 3D reconstruction. Depth estimation neural networks can infer dense depths from monocular camera images [19], and the performance can be further improved by using multi-view inputs [23, 24] or additional sparse depth information [20]. On the other hand, radiance field methods [25, 29, 30] offer a rich and expressive map representation of the 3D scenes, and enable photorealistic novel-view synthesis which is beyond what classical colourised point clouds [5, 16] or meshes [18] can achieve. This rendering capability creates new opportunities in creating realistic simulation environments for robot learning [31] and visual relocalisation.

Despite their promise, applying these emerging methods to large-scale outdoor reconstruction presents several challenges. Learning-based depth estimation remains less accurate than lidar [32], and suffers from domain shifts between training and test data [33]. Furthermore, the depth estimates often lack temporal consistency across frames [34]. Radiance field reconstructions, although visually compelling, lack geometric accuracy in visually ambiguous regions such as textureless surfaces. Moreover, most radiance field methods are only tested with object-centric trajectories [25, 27], whereas mobile robots in outdoor environments often follow linear-motion trajectories with sparse viewpoints, which leads to overfitting and reconstruction artefacts. Finally, large-scale environments pose computational challenges: camera pose estimation using Structure-from-Motion (SfM) becomes

increasingly time-consuming, while the memory demands of radiance field's models grow with the scene size and can exceed hardware capacity. Addressing these limitations is crucial for integrating learning-based depth estimation and radiance field methods into real-world mobile robotics applications.

1.2 Contribution

This thesis explores strategies to incorporate learning-based depth estimation and radiance fields for dense 3D reconstruction in large-scale outdoor environments. Central to our reconstruction systems are the fusion of lidar and visual data, and the quantification of uncertainty in the depth estimates and radiance field reconstructions. Specifically, the lidar data provides strong geometric priors that enhance the accuracy of both depth neural networks and the radiance field's geometry. In terms of the uncertainty estimation, we consider both aleatoric uncertainty and epistemic uncertainty. Aleatoric uncertainty arises from the inherent noise present in e.g. the sensor measurements, while epistemic uncertainty represents the lack of knowledge and information. Quantifying both types of uncertainty enables the identification of unreliable learnt depths estimates and the radiance field reconstructions, which can then be used to improve the reconstruction accuracy and inform downstream robotic tasks such as navigation and next-best-view planning.

We present two reconstruction systems. The first system integrates an aleatoric-uncertainty-aware depth completion network into an occupancy mapping framework. The second system uses neural radiance field (NeRF) to combine both vision and lidar data, and incorporates submapping and bootstrapped bundle adjustment for large-scale reconstruction. This system is further extended with epistemic uncertainty quantification, with decoupled uncertainty based on the distinct characteristics of visual and lidar sensors. Additionally, we introduce a large-scale lidar-visual dataset curated alongside the development of our methods, providing a benchmark for evaluating SLAM systems and emerging radiance field methods. The specific contributions of this thesis are detailed as follows:

- A probabilistic depth completion aided occupancy mapping framework. A neural network is trained to upsample 16-beam lidar data with overlapping camera images. It also predicts aleatoric uncertainty of the completed depth output. The uncertainty-aware depth estimation produced in this step is then fed into an occupancy mapping framework to achieve dense reconstruction comparable to one obtained by using a more expensive 64-beam lidar.

- A large-scale NeRF-based mapping system that is integrated with a lidar-SLAM system. The system combines both camera and lidar data to achieve reconstruction with both photorealistic texture and accurate geometry. It exploits the online SLAM trajectory to bootstrap a Structure-from-Motion procedure to speed up its computation time. It can scale to trajectories of as long as 600 metres with the use of submapping.

- An extension to the previous NeRF mapping system, featuring epistemic uncertainty quantification for each sensor modality's input based on Fisher information. The uncertainty estimation enables the identification of the unreliable portions of a multi-modal reconstructions which are due to situations where the individual sensors inherently struggle. It also proposes to use visibility information to perform submapping with improved scene partitioning.

- A multi-sequence, multi-environment, lidar-visual dataset spanning over $125\,000\,\mathrm{m}^2$ with accurate TLS ground truth suitable for evaluating localisation, 3D reconstruction and novel view synthesis. This sensor setup enables lidar-visual mapping systems and encourages the research community to test their own integration of radiance field methods with SLAM systems with a set of specific accuracy benchmarks.

1.3 Publications

The previously described research efforts have been materialised through four first-author peer-reviewed publications developed during this project (two of which are

under review at present). Additional contributions include two non-first-author peer-reviewed articles (one of which is under review at present).

1.3.1 First-Authored Publications

1. Y. Tao, M. Popović, Y. Wang, S. Digumarti, N. Chebrolu, M. Fallon, *3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation*, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022 [35]

2. Y. Tao, Y. Bhalgat, L. Fu, M. Mattamala, N. Chebrolu, M. Fallon, *SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection* , IEEE International Conference on Robotics and Automation (ICRA), 2024 [36]

3. Y. Tao, M. Fallon, *SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification*, IEEE Transactions on Robotics (T-RO), 2025 (under review)

4. Y. Tao, M. Muñoz-Bañón, L. Zhang, J. Wang, L. Fu, M. Fallon, *The Oxford Spires Dataset: Benchmarking Large-Scale LiDAR-Visual Localisation, Reconstruction and Radiance Field Methods*, International Journal of Robotics Research (IJRR), 2025

1.3.2 Co-authored Publications

1. R. Border, N. Chebrolu, Y. Tao, J. Gammell , M. Fallon, *Osprey: Multi-Session Autonomous Aerial Mapping with LiDAR-based SLAM and Next Best View Planning*, IEEE Transactions on Field Robotics (T-FR), 2024

2. L. Zhang, Y. Tao, J. Lin, F. Zhang, M. Fallon, Visual Localization in 3D Maps: Comparing Point Cloud, Mesh, and NeRF Representations. IEEE Transactions on Robotics (Autonomous Robots) - under review.

1.4 Thesis Outline

The structure of the overall thesis is outlined as follows:

- **Chapter 2 - Background** presents the foundations for the methods developed in the thesis.
- **Chapter 3 - Literature Review** surveys related work in the field of SLAM, 3D reconstruction and view synthesis, and identifies the gaps in the current research and motivates the methods developed in this thesis.
- **Chapter 4 - Probabilistic Depth Completion** presents a learning-based approach to probabilistically densify sparse lidar data with vision, which is integrated with an occupancy mapping framework.
- **Chapter 5 - Scalable Lidar-Visual NeRF** describes a radiance field framework incorporating both lidar and vision in large-scale environments.
- **Chapter 6 - Epistemic Uncertainty Quantification in Lidar-Visual NeRF** develops upon the previous chapter with a novel uncertainty quantification method to identify uncertain reconstruction due to each sensor's own characteristics, which facilitates filtering unreliable reconstruction. It also achieves more extensive evaluation and sophistication.
- **Chapter 7 - Lidar-Visual Dataset for SLAM and Radiance Fields** presents a dataset developed to evaluate the reconstruction methods developed in this thesis. It highlights the generalisation issue in the current radiance field methods, which opens up new research opportunities.
- **Chapter 8 - Conclusion** summarises the key findings of the research, and suggests areas for potential future research directions.

2
Background

Contents

2.1 Preliminaries 7
2.1.1 Notation 7
2.1.2 Bayesian Probability 8
2.1.3 Geometric Transformations 11
2.2 Sensors 13
2.2.1 Cameras 13
2.2.2 Lidar 16
2.3 Representing the Map 18
2.3.1 Formulation 18
2.3.2 Point Representations 19
2.3.3 Surface Representations 19
2.3.4 Volumetric Representations 20
2.4 Building the Map 21
2.4.1 Occupancy Mapping 22
2.4.2 Volumetric Rendering 23

2.1 Preliminaries

2.1.1 Notation

In this thesis, we will use a variety of mathematical objects which are denoted with different typography styles. They are summarised in Tab. 2.1.

Table 2.1: Typefaces for different mathematical objects

Mathematical Object	Typeface	Example
Scalar	Lowercase italic	Occupancy o
Vector	Lowercase bold	RGB Colour $\mathbf{c}$
Matrix	Uppercase bold	Transformation matrix $\mathbf{T}$
Set	Uppercase calligraphic	A set $\mathcal{S}$

For common sets of numbers, we use uppercase blackboard bold to denote them: $\mathbb{N}$ as natural numbers, $\mathbb{Z}$ as integers, $\mathbb{R}$ as real numbers.

We denote a function with domain $\mathcal{X}$ and codomain $\mathcal{Y}$ as $f : \mathcal{X} \to \mathcal{Y}$. We use f for a scalar function, $\mathbf{f}$ for a vector function, $\mathbf{F}$ for a matrix function.

2.1.2 Bayesian Probability

Probability theory provides a mathematical toolbox for *modelling* uncertainty. Bayesian probability is a particular interpretation of probability that represents one's subjective belief state of knowledge. In the field of machine learning and robotics, Bayesian probability has been widely used to develop systems which can reason under uncertainty [37, 38]. In this section, we provide an overview of how we formulate our problem using Bayesian probability theorem.

Many robotic perception and machine learning tasks can be formulated as estimation problems. Specifically, given a set of observations, the goal is typically to estimate the underlying data generation process represented by a set of parameters. This problem is difficult, because the observations are often incomplete and non-deterministic. Formally, we denote the set of observations or data as $\mathcal{D}$. For a supervised machine learning task, each observation can be represented as an input and output pair, and we denote $\mathcal{D} = \{(\boldsymbol{x_n}, \boldsymbol{y_n})\}_{n=1}$ where $(\boldsymbol{x_n}, \boldsymbol{y_n})$ are the n-th input and output pair. The set of parameters which governs the underlying data generation process is denoted as $\boldsymbol{\theta}$.

Many estimation problems can be solved using Bayesian inference. Bayesian inference determines the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ from a prior distribution $p(\boldsymbol{\theta})$ and the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ using Bayes' theorem.

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\mathcal{D}, \boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}} \tag{2.1}$$

In words, Bayes theorem states that the posterior distribution is equivalent to the joint probability distribution $p(\mathcal{D}, \boldsymbol{\theta})$ conditioned by the observations $p(\mathcal{D})$, which can be constructed from the prior distribution $p(\boldsymbol{\theta})$ and the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$.

We can formulate the task of 3D reconstruction (assuming known sensor localisation) within the framework of Bayesian probability theory. The observations $\mathcal{D}$ are "posed sensor measurements", in other words for each camera image or range image there is an accurate corresponding pose of the sensor in 3D. The parameters to be estimated $\boldsymbol{\theta}$ is the underlying 3D map with a certain representation such as voxel grids or a neural networks.

Representing Uncertainty

There are two major types of uncertainties that can modelled: *aleatoric uncertainty* and *epistemic uncertainty*. Aleatoric uncertainty, also called data uncertainty, models stochastic events. This cannot be reduced as new information is acquired, as it is an inherent property of the object. Epistemic uncertainty, also called model uncertainty, represents the lack of knowledge due to inherent limitations in underlying fundamentals. This is also called model uncertainty, and can be reduced as new information is acquired.

In the context of robotic perception, examples of aleatoric uncertainty include the noise characteristics of the lidar range measurements when scanning a static object from a fixed location. Another example is camera image noise that comes from the photodiodes in the digital camera or the electrical circuits digitising the image signal. Epistemic uncertainty in robotic perception essentially represents our confidence in the state of the robot (e.g. the sensor pose) and the map (e.g. the 3D geometry) that we are estimating using our sensor observations.

For a prediction task (e.g. regression), we can represent aleatoric uncertainty and epistemic uncertainty using Bayesian probability theory. When predicting the

output for a given input, the Bayesian approach computes the posterior predictive distribution by integrating over all possibilities of the unknown parameters:

$$p(y \mid x, \mathcal{D}) = \int p(y \mid x, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta} \tag{2.2}$$

The full uncertainty of the output prediction depends on both the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ and the posterior $p(\boldsymbol{\theta} \mid \mathcal{D})$. We can use the variance of the likelihood $p(\mathcal{D} \mid \boldsymbol{\theta})$ to represent the aleatoric uncertainty of the sensor observations, where a small variance corresponds to more confident observations given a certain underlying data generation process defined by $\boldsymbol{\theta}$. Epistemic uncertainty can be represented by the posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$, where a large variance means that there is a larger degree of variability in the estimated parameters $\boldsymbol{\theta}$ which corresponds to higher uncertainty. The likelihood and posterior distributions are often approximated with a Gaussian function (such as in the Laplace approximation), and the variance of the distribution can be interpreted as the uncertainty.

Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) aims to find the best possible set of parameters that explain a set of observed data assuming a certain mapping function $p(\mathcal{D} \mid \boldsymbol{\theta})$ that generates the observation.

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\mathcal{D} \mid \boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log p(\mathcal{D} \mid \boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \left[-\log p(\mathcal{D} \mid \boldsymbol{\theta}) \right] \\
&= \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} p(x_i, y_i \mid \boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{x \sim p(x)} \log p(\mathcal{D} \mid \boldsymbol{\theta})
\end{aligned} \tag{2.3}$$

Maximum A Posteriori Estimation

Compared to MLE, Maximum A Posteriori (MAP) estimation additionally considers the prior distribution $p(\boldsymbol{\theta})$ when estimating the best set of parameters.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \mathcal{D}) \\ &= \arg\max_{\boldsymbol{\theta}} \Big[\log p(\mathcal{D} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \Big] \\ &= \arg\min_{\boldsymbol{\theta}} \Big[-\log p(\mathcal{D} \mid \boldsymbol{\theta}) - \log p(\boldsymbol{\theta}) \Big] \\ &= \arg\min_{\boldsymbol{\theta}} \Big[\sum_{n=1}^{N} \ell(x_n, y_n; \boldsymbol{\theta}) + r(\boldsymbol{\theta}) \Big] \\ &= \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}; \boldsymbol{\theta})\end{aligned} \tag{2.4}$$

where $\ell(x_n, y_n; \boldsymbol{\theta}) = -\log p(y_n | f_\theta(x_n))$ is the loss term (that optimisation methods minimise), and corresponds to the negative log-likelihood for each data sample, and $r(\theta) = -\log p(\boldsymbol{\theta})$ is the weight regularisation that corresponds to the log-prior.

It can be seen from Eq. (2.4) that the total training loss can be interpreted as $\mathcal{L}(\mathcal{D}; \boldsymbol{\theta}) = -\log p(\mathcal{D}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$. The exponential of the negative training loss then corresponds to the unnormalised posterior $p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{1}{Z} p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) = \frac{1}{Z} \exp\Big(-\mathcal{L}(\mathcal{D}; \boldsymbol{\theta}) \Big) \tag{2.5}$$

where the normalising constant Z is the normalising constant, and is defined as:

$$Z = \int p(\mathcal{D} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \tag{2.6}$$

When the prior is a simple uniform distribution, MAP collapses to a MLE problem. The key difference in terms of the computation is that the normalisation step is often very expensive to compute or even intractable.

2.1.3 Geometric Transformations

Geometric transformations are a class of mathematical functions commonly used in fields such as robotics, computer vision and graphics. They can be used to specify the location of a sensor in the 3D space, as well as to adjust the scale of a sensor trajectory.

Formally, a geometric transformation is defined as an invertible function from a set $\mathcal{X}$ to itself $\mathbf{f} : \mathcal{X} \to \mathcal{X}$. We now present some transformations that are used in this thesis.

Rotation

A 3D rotation can be represented as a 3×3 orthonormal matrix $\mathbf{R}$ ($\mathbf{R}^\top = \mathbf{R}^{-1}$) with the constraint that $\det(\mathbf{R}) = 1$. Without the constraint of a positive determinant, the orthonormal matrix can represent a rotation with reflection, which is not considered in this thesis. Every 3D rotation matrix is a member of the *special orthogonal group* denoted SO(3).

Rigid-body Transformation

Rigid-body transformation, also called Rigid motion or Euclidean Transformation, can be used to represent a robot's motion in 3D, including rotation and translation. Formally, it is a function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ defined as

$$\mathbf{f}(\mathbf{x}) = \mathbf{R}\mathbf{x} + \mathbf{t} \tag{2.7}$$

where $\mathbf{p} = [p_x, p_y, p_z]^\top$ is a vector representing a 3D coordinate, $\mathbf{R}$ is the rotation matrix, and $\mathbf{t} \in \mathbb{R}^3$ represent a 3D translation.

A 3D rigid-body transformation can also be represented by a 4×4 matrix $\mathbf{T}$ when the input is a homogeneous coordinate $\tilde{\mathbf{p}} = [p_x, p_y, p_z, 1]^\top$:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \tag{2.8}$$

All 3D rigid-body transformations belong to the *special Euclidean group* denoted SE(3).

Similarity Transformation

Rigid-body transformations preserve angles and distances, and can be used to describe the motion of a rigid body in 3D space. A more general geometric transformation is a *similarity transformation*, which can additionally describe a

change of scale. A similarity transformation that preserves orientation (which excludes reflection) is called *direct similarity transformation*. A 3D direct similarity transformation $\mathbf{f} : \mathbb{R}^3 \to \mathbb{R}^3$ is defined as

$$\mathbf{f}(\mathbf{x}) = s\mathbf{R}\mathbf{x} + \mathbf{t} \tag{2.9}$$

where s is a scalar representing the scale factor.

In this thesis, specifically in Chap. 5 and Chap. 6, the direct similarity transformation is used to rescale the non-metric trajectory from a visual reconstruction system to a fully metrically consistent trajectory using a lidar SLAM system.

Application: Coordinate Transformation

One application of the rigid motion is to express the change of a point's coordinate system. We denote a 3D point whose vector coordinates are expressed in a right-handed Cartesian coordinate system $\mathtt{A}$ as $\mathbf{p}^{\mathtt{A}} = [p_x, p_y, p_z]^\top$. This point can be also expressed in another right-handed Cartesian coordinate system $\mathtt{B}$ as $\mathbf{p}^{\mathtt{B}} = [p'_x, p'_y, p'_z]^\top$. Changing a point's vector coordinate system from $\mathtt{A}$ to $\mathtt{B}$ can be represented by a 3D rigid motion which we denote as $\mathbf{T}_{\mathtt{BA}}$:

$$\mathbf{p}^{\mathtt{B}} = \mathbf{T}_{\mathtt{BA}}\mathbf{p}^{\mathtt{A}} \tag{2.10}$$

2.2 Sensors

2.2.1 Cameras

Cameras are relatively low-cost sensors with abundant visual information, and are ubiquitous in robotic systems. A camera is a passive sensor that captures light information from the environment which can be projected into a 2D image plane in the form of a regular grid.

Raw Image Processing

The basic principle of image capture is to direct incoming light through a lens and convert it into electrical signals using an array of photosensitive sensors. The intensity of the light hitting each charge-coupled device (CCD) sensor determines the brightness and colour of the corresponding pixel. In a modern digital camera, each pixel sensor is a wavelength-specific filter, which results in it being able to capture only a specific colour, from red, green and blue. This leads to what we called *raw bayer pattern image*. The raw image can then be processed through a debayering process which interpolates the neighbouring pixels to form a 3-channel RGB value for each pixel. The debayered image can also be passed through an Image Signal Processor (ISP) to make the image more visually appealing. A typical ISP pipeline includes demosaicing, white balance, gamma correction and tone mapping.

In mobile robotics, the camera collects light while in motion. This means that the information received by the CCD sensor is averaged across a certain duration of time. While the sensor is not static, this can lead to *motion blur*. While using low exposure time can reduce this effect, it also leads to lower light intensity and an increase prominence of thermal noise comes when the environment is dark. In this thesis, our test data contains both indoor and outdoor scenes where the lighting conditions are different. Therefore, during data acquisition, our cameras use an auto-exposure feature to achieve good-quality image exposure in different environments.

Camera Projection Model

The camera projects rays of light onto a certain place on the image plane. In this section, we describe the basic camera projection process, which transforms a 3D ray onto a 2D image.

Consider a point in 3D, which can define a ray given the camera centre. The first step is to transform the point from an arbitrary frame (let's say the World frame W) to the camera's frame using a transformation matrix $T_{\mathtt{CW}}$. Then, a point in camera frame $\mathbf{p}^C$ can be transformed onto the image plane by the projection function $\mathbf{f} : \mathbb{R}^3 \to \mathbb{Z}^2$. The points that fall within the image boundary defined

by the image height and width can be associated to image pixels. This process can be described by Eq. (2.11).

$$\mathbf{u} = \mathbf{f}(\mathbf{T}_{\mathtt{BA}} \cdot \mathbf{p}^{\mathtt{W}}) \tag{2.11}$$

Linear Camera Model

The simplest projection model is the perspective projection, which is simply a linear mapping

$$\begin{aligned}
\mathbf{f}(\mathbf{p}^{\mathtt{C}}) &= \mathbf{M}\mathbf{p}^{\mathtt{C}} \\
&= \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} f_x p_x + c_x p_z \\ f_y p_y + c_y p_z \\ p_z \end{bmatrix} \\
&= p_z \begin{bmatrix} \frac{f_x p_x}{p_z} + c_x \\ \frac{f_y p_y}{p_z} + c_y \\ 1 \end{bmatrix} \\
&= p_z \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}
\end{aligned} \tag{2.12}$$

Where f_x, f_y are the focal lengths, and c_x, c_y are the camera centre (also called the principle point), u, v are the image coordinates of the projected points. This projection is illustrated in Fig. 2.1.

Non-linear Camera Model

Different lenses have different Field-of-Views (FoV) and light acquisition behaviours. For wide FoV lenses, such as Fisheye lenses, the perspective projection is no longer suitable. This is because light rays at large angles relative to the camera's optical axis are mapped to excessively large pixel locations (which are outside the image boundary). Non-linear camera projection models address this issue by instead mapping with a non-linear function, which allows the rays from large angles to be projected within the image boundary.

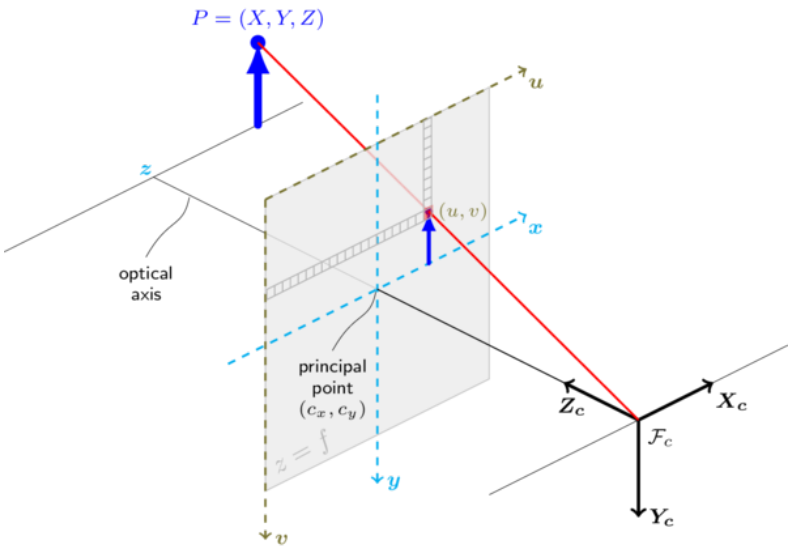


Figure 2.1: Illustration of the linear camera projection model. Image from OpenCV [39].

There are several non-linear projection functions typically used by wide FoV fisheye lenses, including stereographic projection, equidistance projection, equisolid angle projection, and orthogonal projection. These projections, as well as perspective projection, can be approximated with the generic camera model [40], which models the projection function as $r(\theta)$ with a Taylor expansion:

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^5 + k_5\theta^9 + \dots, \tag{2.13}$$

where θ represents the angle between the optical axis and the ray, and $(k_1, \dots, k_5)$ are the parameters of the projection function. Here, the even terms of the Taylor expansion are excluded, which ensures the function is an odd function with radial symmetry: the rays with negative θ will be mapped in the same way as the ones with positive θ. Fig. 2.2 summarises common projection functions $r(\theta)$.

2.2.2 Lidar

Range sensor is a frequently used sensor modality in SLAM applications. Example range sensors include lidar, time-of-flight (TOF) cameras, RGB-D cameras, and stereo cameras. For outdoor environments, lidar is a commonly used sensor. It measures the distance to a surface by emitting a laser beam, and the reflected

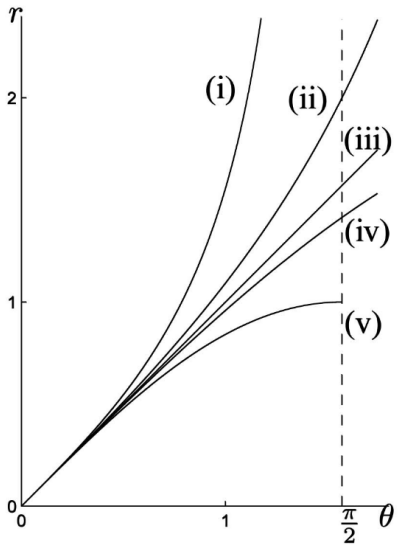


Figure 2.2: Generic camera model which is able to model (i) perspective projection (ii) stereographic projection (iii) equidistance projection (iv) equisolid angle projection (v) orthogonal projection, with a unit focal length $f = 1$. r is the distance between the image point and the principal point, and θ is the angle between the optical axis and the incoming ray. Figure from [40].

laser is received by a detector. The time difference between emission and detection can then be used to calculate the distance travelled by the light. A lidar has a certain scanning pattern, with the most common one involving a continuous 360 scanning pattern, which can be represented by a projection model. Certain types of lidar have projection models that allow the lidar point clouds to be represented in the form of a regular grid. This is refered to as *range image* $\mathbf{R} \in \mathbb{R}^{B \times W}$ with the number of beam denoted as B and horizontal resolution as W.

A mobile lidar's range data often contain noise with a measurement variance in the order of centimetres. In addition, many rotating lidars collect their measurements in a continuous fashion. This means that the lidar measurements in each beam within one rotation all have different acquisition time. When the lidar and the environment are stationary, this time difference has a smaller effect. However, when the lidar sensor is in motion, this effect has to be considered when converting the raw range measurements to 3D point clouds. This problem is often referred to as *motion distortion*. One common approach is to use inertial data, often from an inertial measurement unit (IMU), to compute the sensor trajectory within the

lidar data collection cycle, and transform all lidar measurements to a single sensor frame at a certain time. This motion correction technique is also referred to as *undistortion* of the lidar measurements.

2.3 Representing the Map

2.3.1 Formulation

A 3D map that represents the entities of the 3D environment can be modelled as a function $o\colon \mathbb{R}^3 \to \mathcal{Y}$. The codomain $\mathcal{Y}$ of the function depends on the quantity to be modelled. In lidar-based mapping, the most commonly used map representations are occupancy [41] and distance (to the closest surface) [1]. Occupancy represents whether the space is occupied. This binary state is often represented probabilistically, which allows incremental Bayesian updates of the map as later reviewed in Sec. 2.4.1). The Signed Distance Function (SDF) is an implicit representation of the underlying geometry. The surface is deemed to be the distance field's zero-level set. The sign of the distance function represents whether the location is inside an object or outside an object. The distance entity is useful when carrying out nearest neighbour search operations which are often performed in SLAM methods and used for motion planning [1]. When the distance is truncated, it is called the Truncated Signed Distance Function (TSDF).

3D representations for view-synthesis such as NeRF [25] uses a more sophisticated 3D map representation. The function takes in not only a 3D point, but also a viewing angle (θ, ϕ) to model view-dependent colours. The function output of a NeRF is the emitted colour $\mathbf{c} = (r, g, b)$ and a volume density $\sigma \in \mathbb{R}$. The most general framework for view synthesis is the *light field*, also called the *plenoptic function*. It describes the amount of light received by the sensor from any direction at any point in a 3D space. This allows one to form a visual image of a static scene from an arbitrary pose.

After presenting different map formulations, we now move on to the representations of these maps. Representing the complete function space using finite discrete parameters requires some form of approximation. The representations can be

broadly divided into point representations, surface representations and volumetric representations. In the next sections, we will review each of these representations.

2.3.2 Point Representations

Point cloud is an unstructured discrete map representation of a scene. It can be defined as a set of 3D points $\mathcal{P} = \{\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p_n}\}$. Point representations are flexible due to their unstructured nature, and are widely used. For example, point clouds can be used to represent range measurements from a lidar, and can be used to visualise a depth image obtained from stereo cameras. Point representations however can become inefficient when there are accumulated range observations in the same space, which can lead to excessively large file sizes.

2.3.3 Surface Representations

Surface representations model the boundary of objects. Surface representations are commonly used in computer graphics, as they are more lightweight (than volumetric representations) and also enable efficient rendering via rasterisation. The major types of surface representations are surfels and meshes.

Surfels extend points clouds to represent a collection of surface patches by adding a measure of the surface orientation. It can be defined as $\mathcal{S} = \{\mathbf{s_1}, \mathbf{s_2}, \ldots, \mathbf{s_n}\}$ where $\mathbf{s_i} = (\mathbf{p_i}, \mathbf{n_i})$ where $\mathbf{n_i}$ is the surface normal vector. Often, a surfels also contains a radius quantity which represents the size of the surfel. A surfel that includes colour information can be efficiently rendered by the splatting algorithm [42].

A Polygon mesh is a collection of vertices, edges and faces which represent a polyhedral object. A common type of polygon mesh is a *triangle mesh* where each face is represented by a triangle which can be used to model different polygons. Compared to surfels, meshes directly represent the connectivity information between two adjacent surfaces. This makes meshes more suitable to represent watertight objects, and are therefore commonly used to represent digital assets in computer graphics and simulation engines.

2.3.4 Volumetric Representations

Compared to surface representations, volumetric representations model the 3D volume. Volumetric representations are necessary for applications such as medical imaging and engineering simulation (e.g. fluids). In the field of computer graphics, volumetric representations are needed to model fog and smoke. In the field of robotic perception, volumetric representations can be used to explicitly model free space, which facilitates safe motion planning for a robot.

Volumetric representations can be further divided into discrete volumetric representations and continuous volumetric representations, with the former using discrete geometric primitives and the latter using continuous function approximators.

Voxel grids are popular discrete volumetric representations. A voxel grid is obtained by partitioning the space into cubes, which can be represented by a 3D array. It can be seen as the 3D version of the 2D grid which is used for representing images. A grid's physical scale can be defined by the user (as the physical size of the bounding box), and the physical resolution of the voxel grid can then be determined.

While regular grids are used ubiquitously for 2D data (image), the space complexity of a 3D voxel grid makes it not manageable for large-scale scenes without needing to fall back to a compensation of coarse resolution. However, the 3D scenes we encounter are often sparse: there exists a large amount of space that is empty, and many objects can be reconstructed with coarse resolutions (planar surface). One can take advantage of the sparsity to build compact representations which can efficiently approximate the full function space of 3D maps. Surface representations leverage sparsity by not modelling the free space and also the inside of an object. For voxel grids, Octree is a common representation to efficiently store voxel data by using multiple levels of resolutions. For regions with higher geometry complexity (e.g. thin objects), a voxel can be subdivided into eight octants. This leads to a local high-resolution reconstruction, while low-resolution voxels are used for regions with low geometry complexity. Voxel hashing [43] is another technique for improving memory efficiency by storing surface voxels in a hash table, which then actually resembles a surface representation since it does not store free space in the memory.

Voxel grids are a structured discrete volumetric representation, which requires the pre-definition of resolution and size. In comparison, unstructured representation such as 3D Gaussian is more flexible for inserting, deleting and carrying out local modifications. The 3D Gaussians have been particularly popular for 3D scene representation mainly following the seminal publication on 3D Gaussian Splatting [29]. In this work, each 3D Gaussian is defined by its spatial location, 3D covariance, opacity, and spherical harmonics representing the view-dependent colour.

All of these representations are discrete representations of 3D volume. Alternatively, the function that represents a 3D map could instead be modelled by a multi-layer perception (MLP), a Gaussian process (GP) or other function approximators. MLPs are widely used in NeRF methods [25], while GPs are often used in occupancy mapping [44].

2.4 Building the Map

After discussing each map representation, we now discuss how the map can be built using the sensor observations. We used occupancy grid maps in Chap. 4 and used NeRFs in Chap. 5 and Chap. 6. In this section, we describe the basic principles of building these maps using camera and/or range data. In both cases, the mapping process can be formulated as a MAP estimation given a certain sensor model and the prior of the map, with the probabilistic graphical model shown in Fig. 2.3.

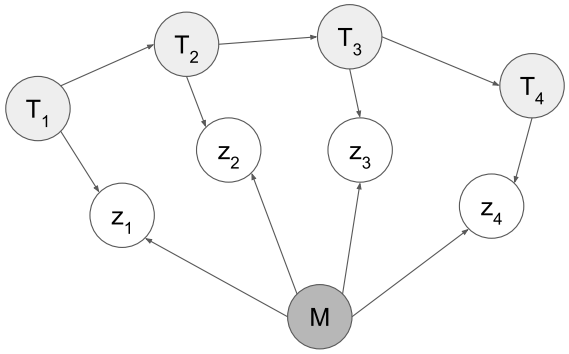


Figure 2.3: Probabilistic graphical model of the 3D reconstruction problem. M is the map, $z_{1...4}$ are the sensor observations, $T_{1...4}$ are the sensor poses.

2.4.1 Occupancy Mapping

We have a set of observations from a ranging sensor (lidar) $z_{1:t}$ together with sensor poses $T_{1:t}$, and we are trying to estimate the map defined by a set of voxel grids $\mathcal{M} = \{m_1, \ldots, m_n\}$ where m_i presents the probability of the i^{th} cell being occupied.

We now derive the incremental occupancy mapping following [38]. Assuming the cells' occupancy are independent of each other, the posterior distribution can be factorised. The MAP inference can then be formulated as

$$\begin{aligned} &\arg\max p(\mathcal{M} \mid z_{1:t}, T_{1:t}) \\ =&\arg\max \prod_{m_i \in \mathcal{M}} p(m_i \mid z_{1:t}, T_{1:t}) \end{aligned} \tag{2.14}$$

The state of the map is either occupied or not, i.e. a binary state. This estimation problem can be solved by a binary Bayes filter.

For a single cell m_i, applying Bayes rule, and assuming that the measurement z_t only depends on the map state, i.e. $p(z_t \mid m_i, z_{1:t-1}) = p(z_t \mid m_i)$:

$$p(m_i \mid z_{1:t}) = \frac{p(z_t \mid m_i) p(m_i \mid z_{1:t-1})}{p(z_t \mid z_{1:t-1})} \tag{2.15}$$

By leveraging opposite map state $\neg m_i$ which can be defined in a similar way as Eq. (2.15), one can obtain the following equation:

$$\frac{p(m_i \mid z_{1:t})}{1 - p(m_i \mid z_{1:t})} = \frac{p(m_i \mid z_{1:t-1})}{1 - p(m_i \mid z_{1:t-1})} \cdot \frac{p(m_i \mid z_t)}{1 - p(m_i \mid z_t)} \cdot \frac{1 - p(m_i)}{p(m_i)} \tag{2.16}$$

Taking the logarithm of both sides, and define $L(m_i) = \log\left(\frac{p(m_i)}{1-p(m_i)}\right)$, which is also called the *log-odds* of voxel m_i, then we obtain the following expression:

$$L(m_i|z_{1:t}) = L(m_i|z_{1:t-1}) + L(m_i|z_t) - L(m_i) \tag{2.17}$$

where $L(m_i \mid z_t)$ is the inverse sensor model in log odds form, representing the probability that cell m_i is occupied given the current measurement z_t.

Having completed these steps, the occupancy mapping can be performed incrementally: we can simply add the new log-odds to the previous cell value stored in the voxel. In Eq. (2.17), $L(m_i)$ is the prior probability of a cell being occupied, and is assumed to be a uniform distribution. Therefore, $p(m_i) = 0.5$, and $L(m_i) = 0$.

2.4.2 Volumetric Rendering

Volumetric rendering techniques have been used in NeRF[25], 3DGS [29], and many follow-up works. It is an image formation process from a volumetric scene representation. In this section, we review the image formation model common for both techniques.

In the radiance field formulation, a point in space has a colour $\mathbf{c}$ and volume density σ. To render from a particular viewpoint, we cast rays $\mathbf{r}(t) = \mathbf{o}+t\mathbf{d}$ from the camera origin $\mathbf{o}$ along the viewing direction $\mathbf{d}$ of each pixel $\mathbf{u}$ in the image plane, and render the pixel-colour by integrating over the set of points samples along the ray.

For each point sample, the probability of it being hit by a ray in a small interval dt depends on the volume density of this point and the points that the ray travels through before it. The probability of the ray not hitting the previously-visited points at the distance t is called transmittance, denoted as $T(t)$. The transmittance can be solved from the following differential equation:

$$T(t) + T'(t)dt = T(t) - T(t)\sigma(t)dt \tag{2.18}$$

Solving the differential equation, we obtain

$$T(t) = \exp(-\int_{t_0}^{t} \sigma(s)ds) \tag{2.19}$$

The expected colour $\hat{\mathbf{C}}(\mathbf{r})$ can be computed as

$$\hat{\mathbf{C}}(\mathbf{r}) = \int T(t)\sigma(t)\mathbf{c}(t)dt, \tag{2.20}$$

where $\mathbf{c}(t)$ is the colour of point sample at distance t.

Since applying Eq. (2.19) to Eq. (2.20) leads to a nested integral, we use the quadrature approximation [45, 46] to simplify the integration, assuming the density and colour are constant within each interval defined by the samples along the ray. With this, the final pixel colour can be expressed as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=0}^{N} w_i \mathbf{c}_i, \tag{2.21}$$

where the weighting coefficient w_i is defined by

$$w_i = \exp\left(-\sum_{j=1}^{i-1} \delta_j \sigma_j\right)(1 - \exp\left(-\delta_i \sigma_i\right)). \tag{2.22}$$

where δ_i is the distance between adjacent samples.

The radiance field can be trained with a squared photometric loss given the ground truth colour $\mathbf{C}(\mathbf{r})$ from the input image:

$$\mathcal{L}_{\text{Colour}} = \sum_{\mathbf{r}\in\mathcal{D}} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||^2 \tag{2.23}$$

where $\mathcal{D}$ is the whole training dataset used to generate the rays $\mathbf{r}$ and ground truth colour $\mathbf{C}(\mathbf{r})$.

While both using this image formation model, 3DGS [29] differs from NeRF as it does not sample points along the ray, but instead directly rasterise the 3D Gaussians. During training, 3D Gaussians' position, covariance, colour and opacity are optimised, while for NeRF the MLP is optimised.

3
Literature Review

Contents

3.1 Localisation and Reconstruction from Vision 26
3.1.1 Classical Visual SLAM 26
3.1.2 Learning-aided Visual SLAM 28
3.2 Localisation and Reconstruction from Range Data . . 29
3.2.1 RGB-D SLAM 30
3.2.2 Lidar SLAM 31
3.3 Lidar-Visual SLAM 32
3.4 Learning-based Depth Estimation 35
3.4.1 Depth Completion using Vision and Depth 35
3.4.2 Depth Estimation from Vision 38
3.5 Radiance Fields 38
3.5.1 NeRF 39
3.5.2 New Formulations and Improving Accuracy 40
3.5.3 Explicit Map Representations for Computation Efficiency 41
3.6 Datasets 47
3.6.1 Localisation 47
3.6.2 3D Reconstruction 47
3.6.3 Novel-view Synthesis 48

This chapter provides an overview on the state-of-the-art methods related to 3D reconstruction. We first discuss localisation and reconstruction methods using visual and range data. Next, we discuss SLAM systems that combine both lidar and visual information. We then discuss learning-based depth estimation methods which combine lidar and visual data, and emerging radiance field representations

which enable volumetric reconstruction using vision with photorealistic rendering capability. Finally, we discuss datasets that are used for localisation, reconstruction and novel-view synthesis.

3.1 Localisation and Reconstruction from Vision

3.1.1 Classical Visual SLAM

MonoSLAM [47] was the first Visual SLAM system developed that can operate in real-time with a monocular camera. It is based on an extended Kalman filter (EKF), and could estimate both the camera pose and the 3D visual features as multivariate Gaussian distributions. PTAM [48] introduces the concept of keyframe which enables performing more expensive optimisation (Bundle Adjustment) at a lower rate (the keyframe rate). Compared to filter-based methods which marginalise the whole trajectory history, keyframe-based methods retain a smaller set of previous frames and optimise them in the back-end. For the same computational cost, keyframe-based methods are shown to be more accurate than filtering [49], and have been widely adopted as a result.

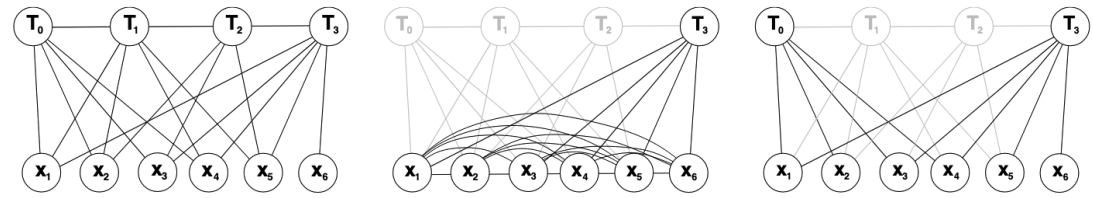


Figure 3.1: Probabilistic graphical models of a full SfM (as Markov Random Field), filtering-based SLAM and Keyframe-based SLAM from [49]. A filtering approach marginalises previous poses, but keeps the feature landmarks which can be revisited in future frames. An optimisation approach solves a sparse graph of previous poses (keyframes) and landmarks.

A number of works tackled PTAM's limitation of scalability to allow the mapping of larger environments. One approach is to replace the full Bundle Adjustment with efficient alternatives including sliding-window Bundle Adjustment [50] and double-window optimisation [51]. Covisibility information [51] can be used to perform

tracking and mapping in a local covisible area regardless of the size of the global map, and this also allows one to perform pose graph optimisation with loop closure detection. Additionally, monocular visual SLAM inevitably contains scale drift [50] which has to be considered by the pose graph optimisation techniques upon loop closures. These techniques, together with place recognition methods like DBoW2 [52], have been integrated in ORB-SLAM [53, 54]. Further extensions of ORB-SLAM were developed to include support for stereo and RGB-D cameras [54] and a multi-map system which allows reusing and merging maps from different sessions [55].

The aforementioned visual SLAM systems all rely on detecting and matching visual features for tracking and map building, and are referred to as *indirect* methods. Another class of methods, called *direct* methods, do not use explicit visual features, but use the pixel intensities directly. The optimisation goal is not to minimise the geometric reprojection error, but instead photometric error. This approach was adopted first by DTAM [3]. To improve computational efficiency, LSD-SLAM [4] selects the pixels with only sufficient contrast. While direct methods can also output dense geometry, they impose a geometry prior and encourage smoothness. In DSO [56], such a geometry prior was not incorporated with the belief that it can introduce a bias and reduce long-term large-scale accuracy. Hybrid systems such as SVO [57] combines the benefits of direct and indirect methods. SVO uses a direct method for tracking at high frame rate, and extracts features for keyframes which together with the trajectory are optimised in the back-end.

Direct and indirect methods have their own advantages and limitations. As discussed in [53], direct methods do not require feature extraction, and are more robust to blur and low-texture environments. However, photometric consistency requires a narrower baseline of the frame matching and is prone to rolling-shutter and auto-exposure effects (if the photometric camera model is not modelled well). In comparison, indirect methods can match features with a wide baseline, and the features are more robust to viewpoint and lighting changes.

Inertial sensors are commonly used within visual SLAM/Odometry methods, and leads to visual-inertial navigation systems (VINS). An inertial measurement unit

(IMU) contains accelerometres and gyroscopes, and can measure linear acceleration and angular velocity. This allows one to recover metric scale and gravity direction which are complementary to a monocular visual SLAM system. Popular visual-inertial systems include VINS-Mono [58], ORB-SLAM3 [55, 59], OpenVINS [60], Kimera [61] and OKVIS2 [62].

3.1.2 Learning-aided Visual SLAM

Learning-based methods using convolutional neural networks (CNN) for estimating geometric prior from a monocular image, such as depth [63] and surface normals [64], have been used within visual SLAM systems. Using learning-based depth estimation also makes it possible to recover scale information from a monocular image sequence [63] which is not otherwise possible using classical methods [50]. We discuss learning-based depth estimation methods in more detail in Sec. 3.4.

While the learnt depth prior is useful, one challenge is the inconsistency of the depth estimation across frames in the 3D space. To tackle this, CNN-SLAM [63] treats the learning-based depth estimation as an initial guess and refines it with small baseline stereo matching similar to LSD-SLAM [4]. The stereo matching is particularly helpful in refining edge reconstruction where learning-based depth tends to be blurry. CodeSLAM [34] presents a novel idea of achieving cross-frame consistency by optimising in the latent space of an auto-encoder-based depth estimation network. The estimated dense depth is conditioned on the input image, and can be updated by changing the latent variables in the auto-encoder. Overlapping frames are used to optimise the latent variable by minimising the photometric loss. The loss function's Jacobian with respect to the relative transformation and the latent variable can then be computed. This allows the learnt depth estimation to be consistent across frames. While CodeSLAM [34] focuses on visual odometry, its follow-up work DeepFactors [65] extends it to a full SLAM system with loop closure, relocalisation and full-scale map optimisation. TANDEM [66] combines a visual odometry module DSO [56] with deep Multi-view

Stereo Network to estimate dense depth from multiple frames, which are fused in a volumetric mapping system.

Deep neural networks have also been used for pose estimation. DeepVO [67] presents an end-to-end visual odometry system based on deep recurrent CNNs. DROID-SLAM [68] is a competitive SLAM system which uses a differentiable deep bundle adjustment layer to match the camera poses and depth estimation with the optical flow estimation. Purely learning-based SLAM system often requires more computation power from the hardware (with GPUs) because of the use of neural networks.

More recent methods adopt the DUST3R [24] framework which predicts pointmaps from two images and can be used for 3D vision tasks including pose estimation and depth estimation. For example, MAST3R-SLAM [69] presents a SLAM system which is developed upon MAST3R [70] and includes efficient techniques for matching, tracking, factor graph construction and optimisation, and loop closure. This system achieves competitive relocalisation and dense reconstruction performance compared to alternative methods such as ORB-SLAM3 [55] and DROID-SLAM [68].

This section mainly discusses learning-aided visual SLAM methods, where a trained model is tested in unseen environments. Later in Sec. 3.5.3 we also discuss SLAM methods based on radiance fields. Although NeRF uses a neural network and techniques in deep learning, it is essentially a reconstruction method for a particular scene rather than a method that generalises to different scenes. Hence, we do not include them in this section.

3.2 Localisation and Reconstruction from Range Data

Common depth sensors include depth camera (for indoor environments) and lidar (for outdoor environments). SLAM methods with depth data can also be categorised into indirect (edge and plane features) and direct methods (ICP). When processing offline, Bundle adjustment techniques have also been explored in BALM [71, 72] and HBA [73]. In this section, we focus on range data based SLAM systems.

3.2.1 RGB-D SLAM

The availability of consumer-grade depth cameras such as Kinect leads to the emergence of RGB-D SLAM systems. The seminal work KinectFusion [74] uses a volumetric fusion method [75] to incrementally reconstruct the TSDF map running on a GPU. Its visual odometry module registers each frame to the map itself using ICP. The final map achieves textured sub-centimetre-resolution reconstruction.

A number of subsequent work focus on improving KinectFusion's scalability. Kintinuous [76] keeps a local map, and upon entering a new environment it stores previously mapped area to a mesh representation. The map can be deformed upon loop closure. This extends the scalability of the original KinectFusion and map environments of multiple indoor rooms and outdoor roads.

Improving the efficiency of the data structure is another direction to explore to increase the scalability. A fixed-resolution voxel grid has cubic space complexity, which makes high-resolution maps on large-scale scenes difficult. Octrees have been used as a more efficient representation in SLAM systems [7, 41, 77]. Another efficient implementation is voxel hashing [43], which stores voxel values in a hash table. This technique has been incorporated in large-scale depth SLAM systems such as InfiniTAM [78]. Handling loop closures is a non-trivial task for the structured volumetric representation. One approach is dividing the scene into submaps [79], and upon loop closure adjusting the submaps according to the new pose graph. Another approach proposed in BundleFusion [80] is to de-integrate the frame from the map, and integrate again with a newly optimised pose.

In addition to voxel grids, surfels [81] have also been adopted by depth fusion SLAM systems. Keller *et al.* introduced a surfel-based reconstruction pipeline [82] which performs pose estimation and depth fusion using a unified data representation. ElasticFusion [83] adopts the surfel-based representation and uses a deformation graph to achieve map consistency upon loop closure. The elastic map representation overcomes the limitation of Kintinuous [76] which cannot reuse previous maps and does not work well on dense local loop closures. In addition to rendering a splatted depth image as in Keller *et al.* [82], ElasticFusion also renders a splatted colour

image, and performs pose estimation by jointly minimising both geometric loss from the depth image and photometric loss from the colour image. BAD SLAM [84] also adopts the surfel representation which is optimised in a back-end direct bundle adjustment minimising photometric error.

3.2.2 Lidar SLAM

LOAM [6] is a seminal work on real-time lidar odometry estimation. It first extracts lidar features (edges and planes) which are used to obtain high frame rate odometry. A mapping module running at a lower frame rate (1-2 Hz) registers lidar point clouds using Iterative Closest Point (ICP). The point clouds are then downsampled to reduce computation. This framework was later adopted in many subsequent lidar odometry systems. LeGO-LOAM [85] targets ground robots and introduces a ground segmentation method which then reduces computation. CT-ICP [86] incorporates pose continuity to estimate motion undistortion, which improves registration accuracy. KISS-ICP [87] combines point-to-point ICP with an adaptive threshold for correspondence matching, and generalises in a variety of environments without fine-tuning.

Loop closure detection is crucial for lidar SLAM systems to obtain a globally consistent map. Place recognition modules allow the drift in the estimated odometry to be reduced with pose-graph optimisation. Scan Context [88, 89] is a popular real-time lidar descriptor which is suitable for place recognition in many outdoor scenes, in particular urban environments. It captures the highest structural points, and is shown to be robust to lateral and rotational changes, and also dynamic objects. Learning-based methods (e.g. LoGG3D-Net [90]) have also been proposed, and can output a learnt global descriptor for lidar-based place recognition.

Similar to visual-inertial systems, IMU has been incorporated in many lidar odometry/SLAM systems. In particular, the IMU measurement can be used to compensate for the motion distortion in a rotating lidar. The IMU preintegration can be directly used in odometry estimation, as in e.g. LIOM [91]. LIO-SAM [92] uses the lidar odometry estimation to estimate the IMU bias. The old scans are

marginalised to ensure real-time performance. As a filter-based method, FAST-LIO uses an extended Kalman filter with a new Kalman gain formula which is more computationally efficient. FAST-LIO2 [93] further improves the map data structure efficiency, and allows registering the lidar scan directly to the map.

Similar to RGB-D SLAM, Voxel grids have also been used for dense lidar mapping which facilitates robot motion planning. Voxblox [1] proposes a real-time mapping system that can grow dynamically without the specification of the map size a priori. Adaptive-resolution voxel grids based on octree have been used by methods including Supereight [7, 94] to efficiently represent both coarse and fine geometry of the scene. For large-scale scenes, submaps approaches are popular [8, 95] since they can incorporate global deformations upon loop closure detection.

Finally, some recent methods use neural implicit representation for lidar mapping. SHINE-Mapping [96] uses octree-based sparse voxel grids to store implicit map features, which are interpolated and decoded into SDF by a neural network. NeRF-LOAM [97] proposed an odometry and mapping framework with a dynamic voxel growing method. To incorporate loop closures, PIN-SLAM [98] proposed to use a point-based neural representation which allows elastic deformation of the local maps.

3.3 Lidar-Visual SLAM

SLAM systems using both vision and lidar can be broadly categorised as *loosely coupled* or *tightly coupled* based on the degree of sensor integration. A loosely-coupled system typically has two separate state estimators for vision and lidar, while tightly-coupled approaches try to have a unified representation and solution. The fusion can be done at the raw measurements level (*early-fusion*) or at the estimated state level (*late-fusion*). For all lidar-visual SLAM systems, the lidar-camera calibration is crucial [10].

Zhang *et al.* proposed VLOAM [9], a loosely-coupled odometry system that has a high-frequency visual odometry module (DEMO [99]) that handles rapid motion and provides initialisation of pose which is refined by a lower frequency lidar odometry module (LOAM [6]) which is robust to challenging lighting conditions.

A common approach for lidar-aided visual odometry/SLAM system is to use the 3D lidar points to provide depth measurement for the visual tracking module [10, 99]. This idea can also be applied when using a RGB-D camera [99]. When the visual tracking module is indirect, the sparse lidar depth might not correspond to the visual feature and thus might require interpolation which introduces potential errors. Systems such as DVL-SLAM [11] use direct methods for visual tracking and do not need to extract visual features. There are also lidar-visual systems that use vision to augment lidar SLAM. For example, using visual place recognition for loop closure detection [100]. Another approach is to use the running visual trajectory estimation to initialise ICP [9, 101].

More recent lidar-visual SLAM systems fuse the two sensor modalities in a tightly-coupled manner. We first discuss filtering-based systems, and then discuss optimisation-based systems.

One early example of filter-based lidar-visual odometry systems is LIC-Fusion [102]. It estimates odometry by fusing IMU measurements, sparse visual features and lidar features using the multi-state constrain Kalman filter (MSCKF) [103].

Optimisation-based approaches often use factor graph formulation for combining the heterogeneous sensors. LVI-SAM [12] maintains a joint factor graph with IMU preintegration constraints, visual-inertial odometry constraints, lidar odometry constraints and loop closure constraints. It has a visual-inertial subsystem (based on VINS-Mono [58]) and a lidar-inertial subsystem (based on LIO-SAM [92]). The two subsystems can function independently and can also exchange information: the lidar subsystem provides depth for visual feature tracking, and loop closure is first detected by the visual subsystem and refined by the lidar subsystem.

VILENS [104, 105] uses a single factor graph to fuse lidar, vision, IMU and also kinematics from a legged robot. The factor graph includes constraints from ICP-based lidar odometry, leg odometry from quadruped, and visual and lidar features. To handle sensor failure mode, it uses robust loss functions to reduce the effect of outliers.

R2LIVE [14] has both a (higher frequency) filter-based odometry and a (lower frequency) factor graph optimisation. The real-time odometry system combines lidar, IMU and visual using an iterated Kalman filter. The factor graph optimisation refines a windowed local map with visual landmarks and keyframe poses.

R3LIVE [16] maintains a lidar subsystem (based on FAST-LIO2 [93]) and a VIO subsystem. Compared to R2LIVE [14], it uses direct methods for both lidar and visual subsystems. Since direct methods use photometric loss which is prone to changing lighting conditions and exposure time, R3LIVE++ [106] further improves the system by performing offline camera photometric calibration (camera nonlinear response function and lens vignetting effect) and online camera exposure time estimation to recover the true radiance information.

More recently, Zheng *et al.* proposed FAST-LIVO2 [17]. This system sequentially updates the state from the lidar and image. Specifically, it first registers the lidar scan to the map by minimising point-to-plane error, and then updates this lidar-inertial odometry estimation by minimising (sparse) photometric error between the visible visual map points and the current image. Compared to R3LIVE [16], FAST-LIVO2 does not operate on a pixel level, but on an image patch level. This enables it to use a one-step frame-to-map registration (by sparse image alignment), while in R3LIVE the frame-to-map registration requires initialisation from frame-to-frame registration using optical flow. Another key difference is that FAST-LIVO2 directly uses raw image patches, while R3LIVE uses colourised point clouds which have lower resolution.

ImMesh [18] is able to reconstruct triangle meshes online in large-scale scenes. In this work, the scene is represented by a hierarchical voxel structure which enables fast voxel query. The meshing module first projects 3D vertices of a voxel to a 2D local plane within the voxel, and then performs the meshing operation. ImMesh also demonstrates that it can generate texture meshes when combined with R3LIVE [16].

Many lidar-visual SLAM fuse heterogeneous sensor measurements for state estimation. The map representation is typically (sparse) 3D points— visual features for camera images and point cloud (or line and plane features) for lidar. R2LIVE [14]

is indirect and maintains a local map of visual landmarks and lidar features. R3LIVE [16] uses direct methods and maintains a lidar point cloud map colourised by the camera image. The colourised map is used to perform visual frame-to-map registration by photometric error, and is used additionally to estimate camera image exposure time online in R3LVIE++ [106]. FAST-LIVO2 [17] replaces the colourised point cloud in R3LIVE with point cloud with image patches, which have higher resolution. ImMesh's mesh model is associated with a voxel grid, and the mesh within a voxel is updated incrementally upon new point cloud observations and efficiently with updates in a 2D plane.

In summary, dense map representations used in these systems are mainly based on the lidar point cloud, which can be colourised by the image at pixel-level or patch-level. Our mapping system presented in Chap. 5 and Chap. 6 instead uses a radiance field which is a more sophisticated volumetric textured map representation and more suitable for novel-view synthesis in post-processing.

3.4 Learning-based Depth Estimation

Depth estimation is the task of inferring dense depth information from a camera viewpoint. When the input is an incomplete depth image or a combination of sparse depth and camera image, the task is often referred to as *Depth Completion*. Here, we focus on the problem of depth completion, and briefly discuss vision-based monocular and multi-view depth estimation (without lidar input).

3.4.1 Depth Completion using Vision and Depth

When sparse depth is available, depth completion can be performed by classical image processing techniques such as dilation and hole filling [107]. These techniques utilise the fact that there is smoothness in the final depth image, and the depth completion is mainly based on interpolation. This approach works less well when there is sparser depth. Modern depth completion methods mostly rely on training neural networks to capture depth cues from large datasets of images. Ma *et al.* [20] proposed to add sparse depth in addition to the RGB camera image as input

to a neural network based on ResNet [108], and demonstrated that it improves performance compared to the vision-only depth estimation baseline. This work is further improved in [109] to include a U-Net [110] encoder-decoder structure with a self-supervised training framework. The self-supervised training framework leverages the sequential monocular camera stream and odometry estimates computed by Perspective-n-Point (PNP). Given a sequence of images, the current camera image can be warped to the next image given dense depth and the relative pose, which allows one to optimise the depth estimates using a photometric loss from the warped image and the actual image.

There are several techniques that are shown to improve the accuracy of the completed depth. One is to utilise additional auxiliary data modalities such as surface normals and semantics. DeepLiDAR [111] includes an intermediate trained network for surface normal prediction which is then used together with the sparse depth as the input to a network to obtain completed dense depths. This is then combined with a vision-only depth estimation network with a learnt weighting to obtain the final dense depths. Empirically, the completed depth using surface normal is shown to be more accurate in short-range regions, while the depth estimated from vision is more accurate in far distances. In addition to surface normals, semantic segmentation masks are also shown to be beneficial for training depth completion networks [112].

One important aspect of the CNN-based depth completion networks is the sparsity of the depth input. A CNN takes in the input in the form of a regular grid (dense image). For a sparse lidar point cloud projected onto an image plane and overlaid on top of a camera image, there are only a fraction of the pixels in the camera image that have a lidar depth value. The missing values in the sparse lidar depth are often encoded with a stock value of 0 metres. However, CNN is not designed specifically for this kind of data. Uhrig *et al.* [113] demonstrated that a depth completion network trained on a given sparsity cannot generalise to other sparsities. It is probably surprising to see that when the network is given more depth cues (by increasing the depth sparsity) in test time it performs worse

than given less depth cues (by using the same sparsity as the training stage). This phenomenon is a result of the convolutional kernel treating the missing depth also as an input. Knutsson and Westin [114] proposed normalised convolution which takes the input sparsity into account when performing the convolution operation, and can generalise to different degrees of missing data in the input. The key idea is to normalise the convolution results by the number of valid input pixels (which are non-zero). This technique has been applied in CNNs to handle sparse depth images. Uhrig *et al.* [113] and Hua *et al.* [115] both tackle depth completion from sparse depth inputs and normalise the convolution results given the input sparsity, and the main difference is whether to include the convolution kernel values when computing the normalising constant. Eldesoky *et al.* [116] applied normalised convolution to depth completion with both sparse depth and RGB camera image, and also propagated the confidence from the convolution to the deeper layers of the network. A subsequent work [117] went on to model the uncertainty of both the input depth and the completed dense depth.

Another common issue of learning-based depth estimation is that the depth estimated is often not sharp and tends to be blurred. One of the reasons for this is that the L1 or L2 loss used to train the network does not encourage sharp local structures. One line of work tackles this problem with linear propagation models to learn an affinity matrix. An affinity matrix quantifies the similarity between points in a given space. For a depth image, the depths of pixels belonging to the same object are often correlated/similar, and therefore depth completion could benefit from this approach. Liu *et al.* [118] propose to learn the affinity matrix with spatial propagation networks which can be used to refine coarse network predictions. Cheng *et al.* [119] adopts this for depth completion, and updates depth values within a local convolutional context. This is further improved by learning adaptive convolutional kernel sizes and the number of iterations for the propagation [120]. Such spatial propagation technique has been adopted to refine the estimated depth [121].

3.4.2 Depth Estimation from Vision

Monocular depth estimation has been enabled by advanced deep learning architectures and large-scale datasets. An early learning-based method from Eigen [19] uses CNNs to regress depth with long skip connections similar to U-Net [110]. Recent methods leverage large-scale datasets to improve the results. One key challenge is the variance in the ground truth depth annotations and ranges, and scene characteristics. Ranftl *et al.* [122] proposes a robust training objective which is invariant to the changes in depth range and scale, which enables them to train a single model from multiple datasets. This is adopted in more recent methods such as DepthAnything [22] which further increases the scale of the training data by additionally adding unlabelled monocular images and using pre-trained models.

Deep networks have also been used for multi-view inputs. Methods such as MVSNet [23] often use convolutional layers to generate a cost volume from the input images, which is then further processed with operations such as 3D convolutions to become the final depth estimation. A recent method named DUSt3R [24] proposes a new framework which outputs not depth but instead a set of points. This method does not require unknown camera intrinsics and extrinsics while the point output can be used to recover depths, pixel correspondences and camera parameters. This transformer-based method can take as input monocular or multi-view images, and has achieved impressive results in 3D reconstruction. This publication has led to several follow-up works. For example, MASt3R [70] predicts additional per-pixel features which improve pixel matching and pose estimation quality. This method is adopted in recent SLAM system MASt3R-SLAM [69].

3.5 Radiance Fields

In this section, we review radiance field methods which are mainly designed for the task of novel-view synthesis. We primarily focus on neural radiance field (NeRF) [25], and follow-up works which improve its accuracy and computation efficiency using alternative representations such as voxels [28] and 3D Gaussians [29].

3.5.1 NeRF

The seminal work by Mildenhall *et al.* [25] introduced Neural Radiance Field (NeRF) which created a major step forward in novel view synthesis. This method uses a Multi-layer Perceptron (MLP) to represent a continuous radiance field, and proposes a differential rendering framework which allows the optimisation of the field using calibrated and posed images using a photometric loss function. NeRF uses volumetric rendering to generate colours of a scene from a particular viewing angle. It uses the volume density of the points sampled along the viewing ray to estimate the transmittance (how much of the light will be attenuated) of discretised intervals using the quadrature rule as discussed by Max [45]. One key issue in using a MLP as the scene representation is that it tends to fit a low-frequency function, while the 3D objects to be reconstructed typically exhibit high-frequency geometry and colour patterns. This was addressed by using a positional encoding (a mapping with sinusoidal functions) to transform the input 5D coordinate to a higher dimensional space. Fourier mapping is not only applicable for representing the radiance field, but also in a range of tasks including image regression, 3D shape regression and MRI reconstruction [123]. Another contribution is a coarse-to-fine sampling strategy, which aims to sample more points near object boundaries instead of using uniform sampling. Specifically, NeRF includes two MLPs, one as a "coarse" scene representation and another as a "fine" scene representation. The "coarse" network is used to create a proposal sampling for the "fine" network which then provides the actual scene information for rendering.

Since the publication of the original NeRF paper [25], there has been a very large number of follow-up works that follow its general pipeline for novel-view synthesis and 3D reconstruction of static scenes and dynamic scenes. Here, we review a few influential works which have improved the rendering quality and computational efficiency of the NeRF method.

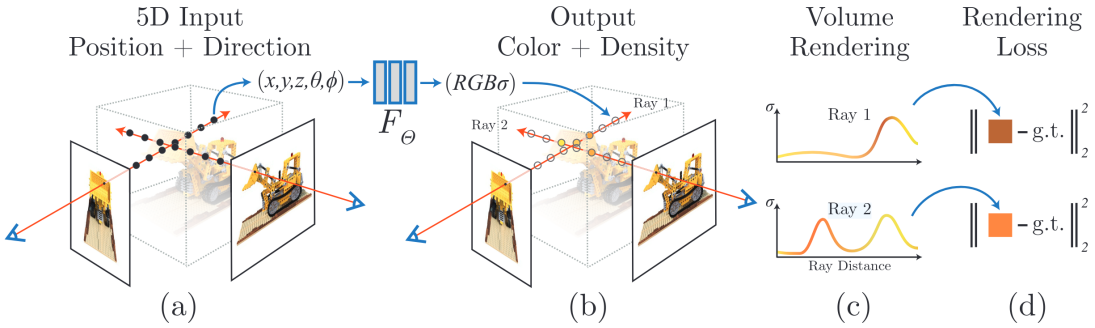


Figure 3.2: An overview of NeRF's MLP-based radiance field representation and the differential rendering pipeline from [25]

3.5.2 New Formulations and Improving Accuracy

The first line of work sought to improve NeRF's rendering quality. One key limitation of NeRF's formulation is the point sampling mechanism which can suffer from aliasing. When the training images observe the scene from different distances and therefore different camera resolutions, the rendering can exhibit blurs or aliasing artefacts. Mip-NeRF [26] proposed to replace ray casting with a model based on cones which models the volume of a frustum. The 3D position and the surrounding volume (modelled by a 3D Gaussian) is encoded by the integrated positional encoding. This generalises the positional encoding in NeRF [25], and compactly represents the regions instead of just a point in 3D space. Mip-NeRF improves the accuracy of NeRF rendering quality—especially in scenarios where the scene is observed at different resolutions. Another direction of research has been to support the modelling of unbounded scenes. NeRF++ [124] introduces an inverted sphere parametrisation. It divides the space into a foreground and the unbounded background separately. This approach then used another network to model the background. Mip-NeRF 360 [27] models the background with a parametrisation that is suitable for volumes [125] instead of points, which results in very competitive rendering quality. To model crowd-sourced images which have dynamic objects and varying lighting conditions and exposures, NeRF-W [126] proposes a per-frame appearance encoding and uncertainty estimation.

The rendered images are evaluated using metrics including Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM) and Learned Perceptual Image Patch Similarity (LPIPS). PSNR is essentially a mean square error expressed in a logarithmic scale. Since it just measures the photometric differences, it is affected by appearance and lighting differences in the image, and is not good at indicating the loss of image structure (e.g. blurred image and sharp image might have similar PSNR). SSIM evaluates the perceptual similarity between two images by considering structural differences, luminance and contrast. LPIPS is computed based on the differences in the image features obtained using a pre-trained deep neural network.

3.5.3 Explicit Map Representations for Computation Efficiency

Voxel grids

A second line of work aims to improve the computation efficiency of NeRF. When the radiance field is modelled by a MLP, as in the original NeRF framework [25], rendering a pixel requires sampling multiple points in the network by a forward pass. This is computationally expensive. Training a MLP-based NeRF takes 1-2 days for a single small-scale scene. A number of works propose to use alternative 3D data structures to replace the MLP. One popular choice is to use a voxel grid. PlenOctree [127] first showed that the test-time rendering of a NeRF can be significantly accelerated by distilling the trained NeRF into an Octree. Each voxel stores the colour and density values extracted from the trained NeRF. In particular, the view-dependent colour (useful for non-Lambertian materials) is modelled by a spherical function and represented by spherical harmonics. This representation increased the speed of test-time rendering by 5 orders of magnitude with the same image quality (100+ FPS on a NVIDIA V100 GPU). Building on top of the voxel representation, Plenoxel [30] demonstrates that the radiance field can be trained directly by voxels without a neural network and achieves 100× faster training time (reducing from 1 day to 11 minutes). To model a continuous radiance field with a discrete data structure, the volume density and spherical harmonics are interpolated.

Instant-NGP [28] demonstrated that the dense voxel grid can be further optimised to reduce memory usage by using multi-resolution voxel hashing. This method uses a hashing function to transform 3D coordinates to voxel indices at different resolutions. Voxel features from each resolution are concatenated and decoded by a lightweight MLP to determine the final colour and density of the 3D point. While voxel hashing techniques typically address hash collision [43], Instant-NGP suggested that hash collision can be ignored to avoid control flow and increase computation efficiency. This method also includes an impressive CUDA MLP implementation which was key to allowing the training time to be reduced to around just 10 minutes, and the hash encoding scheme reduces the time further to 1 minute. While the voxel representation is less storage-efficient than MLP-based NeRF, the multi-resolution voxel hashing is shown to be more storage-efficient than using multi-resolution dense voxel grids. Explicit representations such as Plenoxels [30] have the advantage of *local update* compared to purely MLP-based methods such as NeRF [25]. It can update a single voxel without affecting voxels that are not connected to it, while updating the MLP will change the entire scene representation. This is one of the reasons that explicit representations are faster to train [28].

More recently, Zip-NeRF [128] addresses the aliasing issue presented in voxel-based representations such as Instant-NGP [28]. For an interval along the ray, it constructs multiple samples to approximate the shape of the conical frustum.

Point-based Representations

In addition to voxel-based representations, there are other explicit representations such as points and surfels that have been used to represent radiance fields. Point-NeRF [129] uses neural 3D point clouds to represent the scene. The neural points can be initialised from a 3D point cloud of the scenes obtained from e.g. Structure-from-Motion or Multi-view Stereo. In addition, the neural points can be pruned if the volume density is low, and added to the boundary of the clouds to fill in missed area. SurfelNeRF [130] uses surfels as the scene representation, and each surfel contains colour, surface normal, radius, confidence and neural features. Nearby

surfels are progressively merged. Additionally, SurfelNeRF also used differentiable rasterisation for rendering the surfels.

3D Gaussians

3D Gaussians is an unstructured scene representation similar to points and surfels, but it represents a 3D volume. 3D Gaussian Splatting (3DGS) [29] is a seminal work which uses an explicit representation to achieve test-time rendering an order of magnitude faster than InstantNGP [28] and Plenoxels [30]. It also outperforms slower but more accurate methods such as Mip-NeRF 360 [27] in terms of rendering quality while using much less training time. While 3DGS uses the same image formation model (α-blending) as NeRF, it avoids the expensive computation of per-pixel ray marching by using differentiable rasterisation of a set of 3D Gaussians as the main engine for optimising the radiance field. The 3D Gaussians can be added or removed during optimisation to achieve a compact yet expressive scene representation. The adaptive density control shares similarities with Point-NeRF [129].

Since its publication, a number of follow-up works have been published to further improve the method. Several methods aim to improve the geometry and rendering quality by applying regularisation on the 3D Gaussians. Mip-Splatting [131] applies a 3D smoothing filter to constrain the size of the 3D Gaussians based on the maximal sampling frequency, which is shown to remove high-frequency artefacts when zooming in. SuGaR [132] introduces a regularisation loss to encourage the Gaussians to be distributed over the surface, and then runs Poisson reconstruction to extract a triangle mesh from the 3D Gaussians. 2DGS [133] replaces the 3D Gaussians volumes with 2D Gaussian disks. This surface-oriented representation is shown to produce detailed and smoothed 3D reconstruction.

3DGS's impressive performance in terms of accuracy, training and rendering speed has made it a popular framework. Because of the explicit representations, it also uses much bigger storage compared to MLP-based radiance field representations such as NeRF [25]. There is a trade-off between computational speed and memory footprint for these radiance field representations.

NeRF for 3D Reconstruction

When the reconstruction targets are objects which can be represented as surfaces, distance fields are a natural choice of representation as it impose a geometric prior of surface compared to a 3D volumetric field. There are several works that replace the volume density field with a signed distance function (SDF). VolSDF [134] represents the volume density as a function of the SDF, which combines the volumetric rendering formulation from NeRF [25] while enforcing the surface in the 3D space. The SDF is regularised with the Eikonal loss [135] which encourages the (2-norm of) the gradients of the distance field to be constant, and therefore encourages the distance value to be consistent. NeuS [136] introduces a new volume rendering scheme to specifically address the bias introduced by a standard volume rendering method. Specifically, it proposed a weight function that is unbiased in the first-order approximation of SDF. Another approach for surface reconstruction is proposed in UNISURF [137] which progressively decreases the sampling region and also poses a surface normal loss to encourage the normals of nearby points to be similar. The aforementioned methods all use MLP and, as a result, are slow to train and render. NeuS2 [138] adopts the multi-resolution hash encoding in Instant-NGP [28] to accelerate training, and also extends the method to reconstruct dynamic scenes with a global transformation prediction component. Neuralangelo [139] also combines hash encoding [28] with NeuS [136]. It additionally proposes to use numerical gradients to compute the surface normals (using the Eikonal loss) which is more suitable for the hash encoding. It also adopts a progressive optimisation scheme to reconstruct the scene coarse-to-fine.

Many of the methods for improving NeRF's geometry reconstruction incorporate a geometric prior of surfaces, often via the use of SDF formulation. The resultant reconstruction often has more smooth surface compared to volume density based NeRF approaches.

NeRF for SLAM

The rich representation capability of a radiance field makes it an attractive alternative map representation in SLAM systems. iMAP [140] first demonstrated that MLP-based NeRF can be used as the only scene representation for a real-time RGB-D SLAM system. iMAP adopts PTAM [48]'s pipeline of parallel tracking and mapping modules. The tracking is performed by aligning the RGB-D data with rendered depth and colour images from the NeRF. Since rendering from a NeRF is computationally expensive for real-time operation, iMAP only selects pixels that are the most informative. Specifically, it uniformly samples the image first, and then uses the loss statistics of these samples to resample, which leads to more samples being in regions with higher loss. The improvements in NeRF described in Sec. 3.5 have been adopted to the later works on real-time NeRF-SLAM systems. NICE-SLAM [141] adopts the use of voxel-grid data structure as the data structure for the radiance field representation, and demonstrates better computation efficiency and improved reconstruction results compared to iMAP [140]. SDF has also been adopted by surface reconstruction systems taking RGB-D data such as Neural RGB-D [142].

In addition to the camera image input, many of the NeRF-SLAM systems use geometric information such as depth and surface normal, obtained from a sensor or a trained neural network. iMAP [140] uses depth information from a RGB-D camera to regularise the rendered depth using a normalised L1 loss. DS-NeRF [143] proposed a new depth loss formulation which encourages the weight distribution to be close to a Dirac delta function, and used COLMAP to obtain sparse 3D visual features as depth supervision. In addition to COLMAP sparse depth, other depth estimation techniques such as MVS [144], learning-based depth completion [145] have also been used as the depth supervision for training a NeRF. Another geometric cue useful for 3D reconstruction is surface normal. While surface normals are typically estimated by a local plane fitting to a point cloud, MonoSDF [146] leveraged pretrained neural networks to estimate a surface normal image from an RGB camera image. In general, geometric cues are useful priors to regularise the 3D reconstruction especially when the input image is sparse.

More recently, 3DGS [29] has also been used in real-time SLAM systems. One example is MonoGS [147]. For camera pose estimation, it derives the Jacobian on Lie group of the camera pose with respect to the 3D Gaussian map within the rasterisation pipeline. It also regulates the Gaussian to be isotropic to avoid the typical over-fitted elongated Gaussians along the viewing direction.

Large-scale Radiance Fields

Submapping has been applied to NeRF representations for large- scale reconstruction. Various partitioning strategies have been explored, such as grid-based partitioning [148] and using road intersections [149]. Merging NeRF submaps presents challenges due to the ambiguity of their boundaries and the potential differences in appearance encoding between submaps [126]. Block-NeRF [149] addressed this by selecting submap candidates based on distance and visibility, then merging them in the 2D image space through interpolation and appearance matching during test time. Submapping has also been used for 3D Gaussian Splatting methods, and the explicit representation of 3D Gaussians makes it easier to define submap boundaries. One key challenge is the artefacts in the submap boundaries. To address this, VastGaussian [150] proposes a visibility-aware progressive data selection strategy. Each submap is expanded such that reconstruction on submap boundaries can be improved. The submaps are later merged into a complete scene. CityGaussian [151] leverages a coarsely trained map with fewer Gaussians to obtain a prior map with lower quality. This prior map is used to guide pose selection and the training of each submap block. After training, each submap is filtered by a bounding box and directly concatenated.

In addition to submapping approaches, there are also recent methods that leverage distributed computation to train a single large map. NeRF-XL [152] proposed a novel distributed training and rendering formulation which is shown to be equivalent to the classic single-GPU case, with minimised communication between GPUs.

Summary

In summary, radiance fields emerge as a popular representation for novel-view synthesis. The initial work from Mildenhall *et al.* [25] introduces a framework to optimise the radiance field representations using differentiable rendering. Different representations, have been proposed with trade-offs in rendering quality, geometry quality, memory consumption and computation efficiency. Many of these advances have been incorporated in open-sourced implementations such as Nerfacto and Splatfacto [153], and have been used in downstream applications such as SLAM.

3.6 Datasets

High-quality and properly curated datasets play a crucial role in developing and evaluating robotic perception algorithms. In this section, we review datasets that can be used for tasks including localisation, 3D reconstruction and novel-view synthesis.

3.6.1 Localisation

TUM RGB-D [154] is one of the first datasets developed to evaluate SLAM algorithms. Due to the difficulty of obtaining ground truth 3D point clouds, it focuses on evaluating localisation performance with ground truth trajectories acquired with a motion capture system. For visual-inertial systems, EuROC [155] and TUM VI [156] have been used widely. For outdoor environments, lidar is commonly used due to its accurate long-range measurements. In the context of autonomous driving, lidar is included in datasets such as KITTI [157] and Complex Urban [158]. Accurate trajectories used as ground truth can be obtained from GNSS data [157], motion capture system [159], registering against an accurate 3D map [160] or using reference targets [161].

3.6.2 3D Reconstruction

Evaluating 3D reconstruction with real-world data is difficult, primarily due to the costly acquisition of ground truth 3D maps. Some SLAM datasets such as ICL-NUIM [162] use simulation data for 3D ground truth. RGB-D SLAM systems have

also been used for ground truth, although at a lower accuracy [163, 164]. Precise real-world ground truth are often collected using terrestrial laser scanner (TLS). EuROC and ScanNet++ are two indoor datasets that provide TLS ground truth maps. For outdoor SLAM datasets, Newer College [160] and Hilti-Oxford-2022 [161] have accurate ground truth coupled with greyscale cameras. Recent datasets [165, 166] also include colour cameras along with lidar and TLS ground truth.

In the field of computer vision, datasets with 3D ground truth reconstruction exist for MVS research, often in small-scale scenes. Middlebury is one of the early datasets where ground truth was collected by a structured light scanner. DTU [167], ETH3D [168], Tanks and Temples [169] are all popular benchmarks with TLS ground truth.

3.6.3 Novel-view Synthesis

For radiance field methods, synthetic datasets [25, 170] with perfect trajectories and controlled lighting are popular. Real-world datasets [27, 171] with object-centric image captures are also used. Recently, ScanNet++ [172] used a more challenging approach to provide test images captured independently from the training sequence with a higher-quality DSLR camera.

4

Probabilistic Depth Completion

Contents

4.1 3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation 50
4.2 Technical Remarks 60
4.3 Discussion 63

The first contribution of this thesis is a mapping system based on occupancy mapping. Compared to camera image data, lidar measurements are much sparser. For example, the reprojection of a 64-beam lidar into an overlapping camera only covers around 6% of the pixels in the KITTI dataset [157]. Sparse lidar measurements will not only result in incomplete reconstruction, but also limit free space detection which is useful for robot motion planning.

Research in vision-guided depth completion has demonstrated promising results [20]. However, the completed depth is subject to outliers especially in object boundaries and in the sky region. This hinders its use in robotic mapping and navigation applications where reliable reconstruction and free space detection are crucial for safe operations.

This chapter presents a probabilistic depth completion method integrated with occupancy mapping. The depth completion neural network can identify the regions

prone to errors with an aleatoric uncertainty estimate, which can be directly integrated into the sensor measurement model within an occupancy mapping framework. The system uses as input a 16-beam lidar and an overlapping monocular camera, and achieves comparable results as using a 64-beam lidar. It is tested with a legged robot scanning a building quad, and we demonstrated that the maps produced with the completed depth facilitate path planning by identifying more free space in the environment than the map reconstructed using the 16-beam lidar data.

4.1 3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation

The article in this section was presented at the *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022* [35]. An accompanying video is available online at `https://youtu.be/db3HetMq5h4`.

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation

Yifu Tao[1], Marija Popović[2], Yiduo Wang[1], Sundara Tejaswi Digumarti[1], Nived Chebrolu[1], and Maurice Fallon[1]

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981531

***Abstract*— Safe motion planning in robotics requires planning into space which has been verified to be free of obstacles. However, obtaining such environment representations using lidars is challenging by virtue of the sparsity of their depth measurements. We present a learning-aided 3D lidar reconstruction framework that upsamples sparse lidar depth measurements with the aid of overlapping camera images so as to generate denser reconstructions with more definitively free space than can be achieved with the raw lidar measurements alone. We use a neural network with an encoder-decoder structure to predict dense depth images along with depth uncertainty estimates which are fused using a volumetric mapping system. We conduct experiments on real-world outdoor datasets captured using a handheld sensing device and a legged robot. Using input data from a 16-beam lidar mapping a building network, our experiments showed that the amount of estimated free space was increased by more than 40% with our approach. We also show that our approach trained on a synthetic dataset generalises well to real-world outdoor scenes without additional fine-tuning. Finally, we demonstrate how motion planning tasks can benefit from these denser reconstructions.**

I. Introduction

Dense 3D reconstruction is a key task in a range of robotic applications including exploration, industrial inspection and obstacle avoidance. Lidar is the dominant sensor used for mapping outdoor environments as it provides accurate long-range depth measurements. However, lidar readings are sparse, especially when compared to depth images obtained from RGB-D cameras. For instance, the reprojection of a Velodyne 64-beam lidar only covers around 6% of the pixels of an overlapping camera image in the KITTI dataset [1]. This drops to about 1.6% for a 16-beam sensor.

Mapping with sparse lidar sensors results in incomplete reconstructions which in turn causes path planning algorithms to either fail or be inefficient due to the limited free space detected. This becomes a critical issue, particularly in robotic applications where path planning is performed online and a reconstruction which minimises unobserved space is necessary for safe operation. While high-end lidars (e.g. 64 or 128 beams) provide denser point clouds of the environment, their prices are typically restrictive for low-cost mobile robot platforms such as delivery robots.

To address the sparsity of lidar sensors, many works use texture information from an overlapping (monocular) camera to infer a complete depth image for every pixel of the camera,

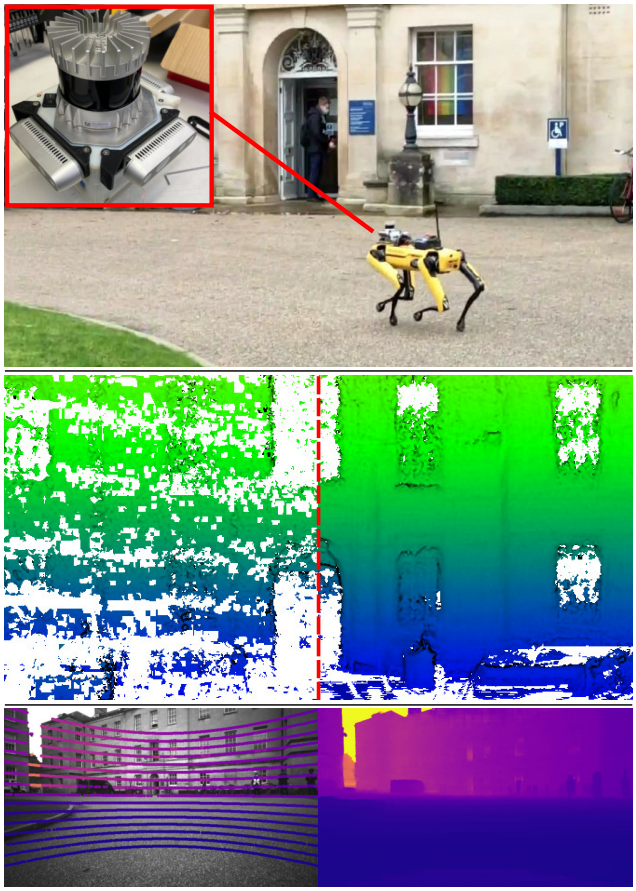


Fig. 1: Depth completion and mapping results from our Maths Institute dataset. Top: A view of a building scanned by the legged robot. Inset shows the scanning device with a lidar and 3 cameras. Middle: 3D reconstruction of the same building using the sparse 16-beam lidar (left) & completed depth images estimated by our approach (right). Bottom: Forward camera image with sparse lidar overlay, our input (left) and a completed depth image output by our approach (right). Video: https://tinyurl.com/depth-completion

This research is supported by the EPSRC ORCA Robotics Hub (EP/R026173/1), the EU H2020 Project MEMMO (780684), and partially supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2070 – 390732324. M. Fallon is supported by a Royal Society University Research Fellowship. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

[1] These authors are with the Oxford Robotics Institute, University of Oxford, UK. {yifu, ywang, tejaswid,nived, mfallon}@robots.ox.ac.uk [2] This author is with the Cluster of Excellence PhenoRob, Institute of Geodesy and Geoinformation, University of Bonn. mpopovic@uni-bonn.de

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

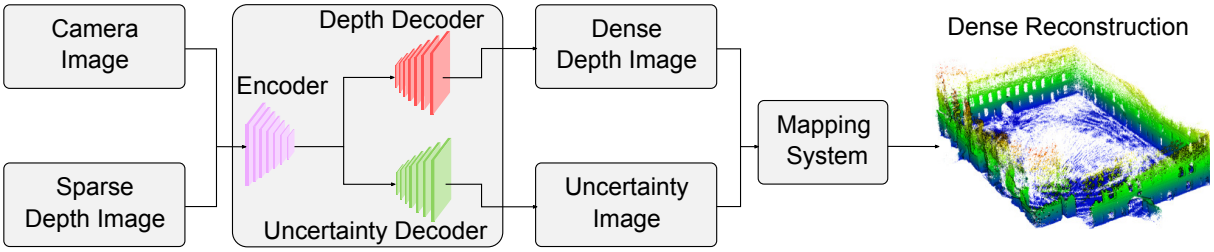


Fig. 2: System pipeline: A camera image and a sparse lidar depth image are the inputs to the network. The network predicts dense depth images along with corresponding uncertainty values. These outputs are fed into a volumetric mapping system to obtain a 3D reconstruction. Our experiments used networks whose architectures are based on [2–4].

which is known as *depth completion*. Recent learning-based approaches [5, 6] have made great progress in improving accuracy. However, the completed depth is still subject to outliers, which would then generate distorted reconstruction and false positive free space volumes in regions that are truly unknown or occupied.

In this work, we propose a pipeline for large-scale mapping with probabilistic lidar depth completion which uses sparse lidar sensing and one or more overlapping camera images. Specifically, the neural network takes a pair of camera and sparse lidar depth images as input, and predicts a dense depth image along with an associated depth uncertainty image. These two predictions are then fused in our volumetric mapping system. This approach can reconstruct a more complete structure and detect more correct free space thanks to this high density completed depth image. The accuracy of both the surface reconstruction and the free space is improved by rejecting predicted depth that has high uncertainty predicted by the network. We deploy the proposed system on a legged robot platform. Unlike most lidar depth completion works which typically use a single monocular camera image having only a small overlap with the lidar, we use a three-camera setup (left, forward and right facing) to give about 270 °overlap with the lidar.

In summary, the contributions of this work are:

1) Extension of a depth completion framework to incorporate uncertainty predictions.
2) Integration of the uncertainty-aware depth completion network within a probabilistic volumetric mapping framework.
3) Evaluation of the framework on real-world outdoor datasets demonstrating deployment on a robot.

II. Related Work

A. Depth Completion

Depth completion is the task of inferring a dense depth image from a sparse one, with or without the guidance of a corresponding camera image. Classical image processing techniques [7] have been applied for unguided depth completion. While these approaches require minimal computational resources, they do not exploit camera image texture, and typically under-perform image-guided methods.

Most recent works on guided depth completion are learning-based. Ma *et al*. [2] developed a depth completion network using ResNet [8] as a backbone, and later improved it by adopting a UNet [9] encoder-decoder paradigm within a self-supervised training framework [3]. Additional information such as surface normals [10] and semantics [11] have been shown to be useful for depth completion. Optimizations based on learning affinity matrix [5, 12] have also been proposed to further refine the predicted depth and improve accuracy. In this work, we leverage the accuracy of such image-guided networks while keeping the network compact. This allows the system to be deployed on mobile GPUs.

Multimodal fusion of visual and depth sensing using deep neural networks remains challenging. Strategies include concatenating the two modalities to form a combined input (early fusion) as well as fusing features extracted from the two modalities at a later stage (late fusion) have been explored [10, 11]. A further challenge arises when the networks are required to generalize to lidars with different sparsity patterns and resolutions [1, 13]. In this paper, we keep the same sparsity level in input depth data during training and at test time.

B. Depth Completion for 3D Reconstruction and Navigation

While many works aim to address the problem of depth completion on a per-image basis, only a few investigate the use of the dense depth output for applications such as reconstruction and navigation. Fehr *et al*. [14] proposed a planning system that predicts dense depth images using a CNN [2] and uses them as the input for the Voxblox mapping system [15]. Depth estimation has been used in visual SLAM systems to recover metric scale [16, 17], and to complete sparse visual features [18, 19]. In this work, we make depth completion more applicable for navigation by connecting it with a volumetric mapping system to achieve both denser reconstruction and to detect more free space.

An issue that often arises with depth completion networks trained on depth images generated by projecting 3D lidar points (e.g. KITTI [1]) is the invalid prediction for regions of the sky due to the lack of ground truth depth there. As a consequence, directly using these predictions results in noisy reconstructions and incorrect identification of free space. In this work, we utilise synthetic datasets with sky annotations so that the network could learn to predict a very large depth value for the sky.

Uncertainty estimates, derived from heuristics [17] or learning [20], have been used to reject wrong predictions during reconstruction. Our work is closely related to [20],

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

which proposed a framework for depth completion and mapping in indoor scenarios using RGB-D cameras. The authors designed a CNN to predict both dense depth and depth uncertainty, which are then fed into an occupancy-based volumetric mapping system [21]. The resulting occupancy maps uncover more *obstacle-free* space compared to using the raw depth images and facilitate improved path planning. In our work, we extend these concepts to real-world outdoor environments where the lidar depth data is much sparser than depth cameras and the scenes have a much larger scale. In comparison to approaches that use RGB-D images as input, our method learns to complete missing information between the lidar points, rather than filling holes in depth images which is characteristic of RGB-D images.

III. Approach

An overview of our proposed system for volumetric mapping with probabilistic depth completion is shown in Fig. 2. The inputs to our system are sets of grey-scale camera images and sparse lidar depth images. This input is processed by a neural network with a shared encoder and two separate decoders to generate a completed dense depth image and the corresponding depth uncertainty prediction. We filter unreliable depth predictions considering their range and predicted uncertainty. The remaining predicted depth and uncertainty are then fed into an efficient probabilistic volumetric mapping system [21] to create a dense reconstruction.

A. Network Architecture

Our focus in this paper is *not* redesigning depth completion network architecture, as the performance on the KITTI Depth Completion Benchmark [1] is saturated, and accuracy comes with more computation burden. Rather, our aim is to obtain probabilistic depth completions applicable to robotic tasks including 3D reconstruction and path planning. Therefore, we choose three baseline depth completion networks from the KITTI Depth Completion Benchmark [1]: 1. ***S2D*** [2], a ResNet-based [8] vanilla Encoder-Decoder network. 2. ***U-S2D*** [3], a ResNet-based Encoder-Decoder network with long skip-connections with concatenation. 3. ***FastDepth****[1] [4], an efficient network with additive skip connection. Note that the network used in our framework is replaceable depending on the application.

In this work, we extend the depth completion network to incorporate depth uncertainty prediction by adding an additional identical decoder to the chosen networks. For the uncertainty decoder in FastDepth*, we only keep the last three layers and one additive skip connection for efficiency.

B. Loss Function and Training Strategy

Our depth uncertainty network aims to predict aleatoric uncertainty which comes from the noise inherent in the observations [22, 23]. In the context of depth completion, depth at object boundaries usually has higher aleatoric uncertainty. Assuming the depth measurement follows a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, the depth network is trained to estimate the mean μ, while the uncertainty network estimates the standard deviation σ, the aleatoric uncertainty.

The loss function [22] for jointly optimising the depth completion network and the depth uncertainty network is defined as:

$$L_{\text{unc}} = \frac{1}{N}\sum_{p=1}^{N} \frac{|y_p - f(\mathbf{x}_p)|^2}{\sigma(\mathbf{x}_p)^2} + \log\left(\sigma(\mathbf{x}_p)^2\right) \quad (1)$$

where $\mathbf{x}_p$ is the input vector of features for the depth completion network (sparse depth and camera image), y_p is the ground truth depth measurement, $f(.)$ and $\sigma(.)$ are the completed depth and associated depth uncertainty output by the network.

The predicted uncertainty has an inverse component in the first term and a proportional part in Eq. (1). This means that the uncertainty prediction should be optimised to be high at erroneous predictions, and low elsewhere. This uncertainty prediction is then used by the volumetric mapping system to update free space and reject incorrect depth predictions which can improve the quality of both the reconstruction and the free space estimation.

C. Probabilistic Volumetric Mapping

The predicted depth measurements are then integrated into a volumetric reconstruction for use in robotic applications such as path planning. We use the state-of-the-art multi-resolution volumetric mapping pipeline [21] to leverage the explicitly represented free space. This pipeline demonstrates greater efficiency and lower memory usage when integrating long-range lidar scans in large-scale environments [24], compared to conventional occupancy mapping methods such as OctoMap [25].

For each voxel in the 3D reconstruction, we store its log-odds occupancy probability. Free, unknown and occupied space have negative, zero and positive occupancy log-odds probabilities, respectively. Computation of the log-odds occupancy probability $L(d)$ at distance d along a ray $\mathbf{r}$ uses a piecewise linear function [21]:

$$L(d) = \begin{cases} l_{\min} & d \leq 3\sigma(d_r) \\ \frac{-l_{\min}}{3\sigma(d_r)}d & 3\sigma(d_r) < d \leq \frac{k_\tau d_r}{2} \\ \frac{-l_{\min}}{3\sigma(d_r)}\frac{k_\tau d_r}{2} & \frac{k_\tau d_r}{2} < d \leq k_\tau d_r \\ \text{no update} & \text{otherwise} \end{cases} \quad (2)$$

where $l_{\min}$ denotes the minimum occupancy probability in log-odds, d_r is the measured depth along a ray $\mathbf{r}$, k_τ is a scaling factor controlling how much occupied space is created behind a measured surface (surface thickness), and $\sigma(d_r)$ is the uncertainty of a measurement d_r.

In the method in [20], a quadratic model for $\sigma(d_r)$ is used, which suited reconstruction with an RGB-D camera. In our work, we estimate $\sigma(d_r)$ using a linear model as:

$$\sigma(d_r) = \begin{cases} \sigma_{\min} & d_r \leq \frac{\sigma_{\min}}{k_\sigma} \\ k_\sigma d_r & \frac{\sigma_{\min}}{k_\sigma} < d_r < \frac{\sigma_{\max}}{k_\sigma} \\ \sigma_{\max} & \frac{\sigma_{\max}}{k_\sigma} \leq d_r \end{cases} \quad (3)$$

[1]We add one convolution layer for the depth image input, and the feature is concatenated with the first layer of the image feature as in [3].

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

to better represent the characteristics of a lidar. $\sigma_{\min}$, $\sigma_{\max}$ and k_σ are constants that depend on the sensor specification.

Our mapping system utilises the depth-dependent uncertainty $\sigma(d_i)$ for raw depth d_i from the input lidar measurement, and the network-predicted uncertainty for the predicted depth d_p, similar to [20]. Since our predicted uncertainty is trained to coincide with depth error, we reject predicted depth when the predicted uncertainty is higher than expected. Specifically, we invalidate a completed depth measurement if the predicted uncertainty is more than 2 times greater than the sensor uncertainty. Analysis of this rejection is presented in Sec. IV-E. Unlike [20], we did not update free space for these uncertain depths. This is shown to remove more incorrect free space which is detrimental to safe navigation.

IV. Experimental Results

A. Datasets

Depth completion is heavily studied for autonomous driving [1], but relatively under-explored in aerial, legged or UGV platforms where the camera images have different depth distribution and objects than images captured on the street. Thus, we evaluated our approach on two separate datasets that target mobile robotics applications where explicit free space is required.

1) Handheld Dataset: The first dataset is a handheld-device dataset: the Newer College Dataset (NCD) [26]. The dataset was collected using a 64-beam 3D lidar and stereoscopic-inertial camera at typical walking speeds. Specifically, we used the left grey-scale camera images from an Intel Realsense D435i and lidar scans from a 64-beam Ouster lidar scanner, synchronised by VILENS [27, 28], as the input to our completion network. Sparse depth images were generated by projecting 3D lidar points onto the camera image plane. The dataset [26] also provided a highly-accurate ground truth point cloud which is useful for evaluating the reconstruction. The ground truth trajectory of the handheld-device was provided by the dataset. Poses of the handheld-device were computed by localizing individual scans against this ground truth map using VILENS.

Training the depth completion network requires ground truth depth. We followed the approach of the KITTI Depth Completion Benchmark [1] and accumulated 11 consecutive laser scans to increase the density of the depth images and use this as the ground truth depth. We simulate a sparser and lower-cost 16-beam lidar providing input to our network by downsampling the original 64-beam lidar readings by removing scan lines.

We generated 11087 samples from the long experiment in NCD and 6651 samples from the short experiment. The long experiment samples are used for training, while 1000 samples from the short experiment are randomly selected for testing and the remaining 5651 samples for validation. The camera image, input and ground truth depth images all have the same resolution of 848 $\times$ 480 pixels.

2) Robot Dataset with three-camera setup: Most lidar depth completion works use only a forward-facing monocular camera image. Thus, around three-fourths of the lidar points are not considered during depth completion. Instead, we built a device with three Intel Realsense D435i cameras and a 64-beam Ouster lidar scanner. The three cameras were arranged as shown in Fig. 3 facing front, left and right respectively. This device was mounted on a legged robot, Boston Dynamics Spot. Data was collected while the robot was manually operated around an outdoor campus environment. Similar pre-processing as in the above Handheld Dataset was applied to generate the input and ground truth depth images, and the poses of the scanning device. We refer to this dataset as the Mathematics Institute (Maths Inst.) dataset and use it solely for testing our proposed method.

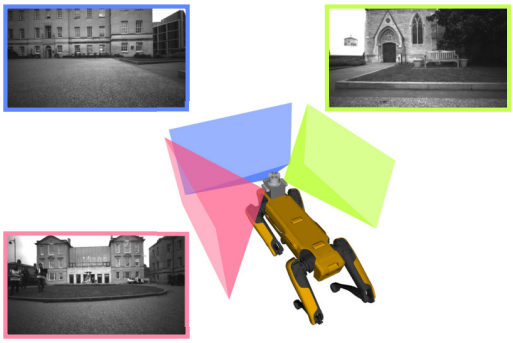


Fig. 3: Our mapping device has cameras facing left, forward and right beneath a LIDAR and is mounted Spot robot. The multi-camera setup has an overlap of about 270°with the lidar.

3) Synthetic Pre-training Dataset: In addition to NCD, we also use the Virtual KITTI 2 dataset (vKITTI) [29] for training the depth network. This dataset contains photo-realistic image sequences captured from a simulated vehicle with ground truth depth images. The depth for regions of the sky is encoded as 256m, in contrast to real-world datasets like KITTI or NCD where the depth there is unannotated (not covered by lidar). Thus, the depth network can predict a pre-defined large depth in such regions.

To generate the sparse input depth image, we applied a 16-beam lidar mask from NCD to the dense ground truth depth to keep input depth sparsity consistent with the above datasets, similar to [30]. This is crucial for generalisation as discussed in IV-C.3, since conventional convolution kernels are known to be sensitive to changes in the density of valid pixels in the input [1, 11]. We prepared a training set with 40520 images and a validation set with 1000 images. Colour images were converted to grey-scale and cropped to have the same width as the images of the above datasets.

B. Implementation Details

1) Network Training Details: Our implementation is built upon [31]. The depth completion network follows the open-source implementation from [2–4]. We used the Adam optimiser with an initial learning rate of 10e-4 and decrease it by a factor of 10 every 5 epochs. The training was performed on an NVIDIA Tesla V100 GPU.

Minimising the uncertainty loss function L_{unc} in Eq. (1) corresponds to optimising the depth completion network and

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

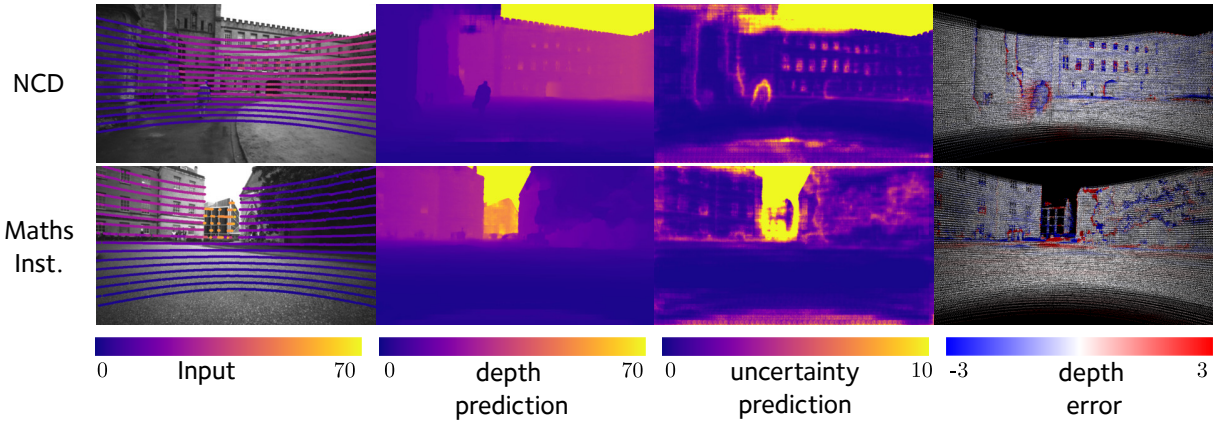


Fig. 4: Example results from vKITTI rel + NCD unc, U-S2D in Tab. I. Left to right: camera image overlaid with sparse lidar depth, dense completed depth, uncertainty prediction, error in depth prediction. All measurements are in metres. Regions of high uncertainty correspond to regions of higher error in depth prediction, allowing us to reject these points for cleaner reconstructions.

the depth uncertainty network simultaneously. We observed that this joint training does not always lead to optimal depth accuracy. Hence, we experimented with a two-stage training procedure: pre-training the shared encoder and the depth decoder with a selected loss function first, and then training the uncertainty decoder using L_{unc}, while freezing the weights of the shared encoder and the depth decoder. For each stage, we trained the network for 30 epochs and selected the best-performing model based on the validation set. Even though this leads to sub-optimal performance on the uncertainty objective L_{unc}, the depth prediction is more important and is hence prioritised over uncertainty.

2) Volumetric Mapping System: We used the volumetric mapping system from Funk *et al.* [21] to fuse the 3D reconstruction using the lidar data and poses as described in Sec. III-C with a voxel resolution of 6.5 cm. The linear inverse sensor model is set such that surface thickness $k_\tau = 0.026$, linear uncertainty model coefficient $k_\sigma = 0.052$, $\sigma_{\min} = 0.06\,\text{m}$, $\sigma_{\max} = 0.20\,\text{m}$.

C. Evaluation of the Depth Completion Network

We perform a quantitative evaluation of the network with multiple training strategies and loss functions on two real-world datasets, as summarised in Tab. I. Qualitative results are shown in Fig. 4. We experimented with training on real-world (NCD) and/or synthetic datasets (vKITTI).

1) Evaluation Metrics: As an established dataset for depth completion evaluation, KITTI uses RMSE as the primary metric for evaluation. However, RMSE and MAE penalise inaccurate depth predictions for distant objects heavily. In robotics applications, depth prediction for nearby objects is as important as those which are further away. In order to balance accuracy across different depth ranges, we use mean absolute relative error (REL) as our primary loss function for the depth completion network, which is defined as

$$REL = \frac{1}{N}\sum_{p=1}^{N}\left|\frac{y_p - f\left(\mathbf{x}_p\right)}{y_p}\right|. \tag{4}$$

We also report the mean absolute error of the inverse depth (iMAE), and the percentage of predicted pixels whose relative error is less than a threshold $\delta \in \{1.05, 1.10, 1.25\}$.

For uncertainty metrics, in addition to our uncertainty loss function L_{unc}, we also evaluated uncertainty using the Area Under the Sparsification Error curve (AUSE) [32]. This metric captures the correlation between the estimated uncertainty and prediction error.

2) Two-stage Training: Training the depth and uncertainty networks together using the uncertainty loss L_{unc} in NCD (NCD unc in Tab. I) leads to good performance on L_{unc} itself. However, there is no explicit objective for the depth network, and some metrics such as REL are not always optimal. We then experiment with a two-stage training strategy to separate the optimisation of depth and uncertainty estimation: we train the depth network first with our chosen loss function REL, and then train the uncertainty decoder using L_{unc} (NCD rel + NCD unc). Such two-stage training enables us to utilise both synthetic and real-world datasets: we train the depth network on vKITTI, and the uncertainty decoder on NCD (vKITTI rel + NCD unc in Tab. I), to get the best of both worlds: the depth network benefits from large training samples and precise depth (including the sky) in vKITTI and the uncertainty decoder captures real-world depth characteristics in NCD. We use U-S2D trained with this strategy for the reconstruction experiments, for its generally good performance on depth and uncertainty metrics.

3) Generalisability: We train S2D on the KITTI Depth Completion dataset with raw 64-beam depth images and downsampled 16-beam depth images. However, the depth estimation is significantly worse when tested on the NCD/Maths Inst. datasets. We hypothesise that the lack of generalisability is due to inconsistency in the input depth sparsity pattern between KITTI and our test datasets. When we apply the same input depth mask from NCD to vKITTI as the training dataset, the depth network only trained on vKITTI generalises to both NCD and Maths Inst without any fine-tuning, validating our hypothesis and approach.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Depth prediction network performance under multiple training strategies, IV-C.

Training Scheme	Network	RMSE ↓ (m)	MAE ↓ (m)	REL ↓ (%)	iMAE ↓ (km^{-1})	$\delta_{1.05}$ ↑ (%)	$\delta_{1.10}$ ↑ (%)	$\delta_{1.25}$ ↑ (%)	L_{unc}↓	AUSE ↓
NCD										
Linear Interpolation	-	3.96	1.51	16.4	-	68.4	77.2	85.8	-	-
NCD unc	S2D	**3.84**	1.69	17.1	22.0	59.6	72.5	84.8	**0.88**	0.12
NCD rel + NCD unc	S2D	4.58	1.76	10.8	24.5	52.4	76.0	88.1	1.52	0.16
NCD rel + NCD unc	U-S2D	4.41	1.68	**9.9**	79.5	61.0	80.7	89.7	1.83	0.16
vKITTI rel + NCD unc	FastDepth*	3.98	**1.37**	11.2	53.5	68.2	80.4	89.9	1.71	0.20
vKITTI rel + NCD unc	S2D	6.53	1.85	16.1	18.8	56.0	73.0	88.5	1.28	**0.11**
vKITTI rel + NCD unc	U-S2D	5.21	1.42	11.5	**17.4**	**81.9**	**89.8**	**94.3**	1.29	0.14
Maths Inst.										
Linear Interpolation	-	**2.69**	0.89	18.6	-	60.5	72.7	85.7	-	-
KITTI 16 rel	S2D	3.21	1.28	27.7	84.4	37.0	53.4	73.6	-	-
KITTI 64 rel	S2D	3.64	1.77	50.1	99.1	27.5	44.7	67.1	-	-
NCD unc	S2D	2.81	1.22	26.6	60.9	35.0	53.4	77.6	3.72	0.14
vKITTI rel + NCD unc	FastDepth*	2.79	**0.83**	**14.5**	53.0	58.8	73.7	88.4	**3.43**	0.22
vKITTI rel + NCD unc	S2D	4.22	1.08	19.6	45.7	50.2	67.2	86.4	8.8	**0.13**
vKITTI rel + NCD unc	U-S2D	3.72	0.89	15.3	**41.5**	**62.3**	**75.9**	**88.8**	5.02	0.15
KITTI Validation Set										
KITTI 16-beam	S2D	1.34	0.45	2.4	1.9	90.5	96.7	99.3	-	-
KITTI 64-beam	S2D	**1.05**	**0.34**	**1.8**	**1.5**	**93.9**	**98.0**	**99.5**	-	-

4) Runtime Analysis: We test the three probabilistic depth completion networks on a mobile GPU Nvidia Quadro M2200, which resembles the resources of a typical mobile robot. The inference speed for FastDepth* is 9.7Hz, S2D 8.3Hz, and U-S2D 1.4 Hz, allowing for near real-time operation on a robot. Note that for a fair comparison, FastDepth* is not pruned and not hardware-optimised (TVM). This, along with the additional uncertainty decoder and larger image size, contributes to slower inference than reported in [4].

D. Evaluation of the Reconstruction and Free Space

We perform a quantitative evaluation of reconstruction quality and the free space predicted by our proposed method (U-S2D, vKITTI rel + NCD unc from Tab. I). We compare reconstructions generated with the following configurations:

1) 16-beam lidar depth image, linear sensor uncertainty
2) 64-beam lidar depth image, linear sensor uncertainty
3) completed depth image, predicted uncertainty

To evaluate the accuracy of free and occupied space in the reconstruction, we create an occupancy map using the centimetre-accurate 3D point cloud captured by a survey-grade Leica laser scanner cloud [26]. We generate the ground truth free and occupied space by performing ray casting from the sensor location to every point in the ground truth cloud. The ground truth map uses the same resolution as our generated reconstructions.

The quantitative results are summarised in Tab. II. We analyse the following two aspects:

1) Reconstruction: The reconstruction is evaluated in terms of accuracy and completeness. For accuracy, we create meshes using marching cubes (Fig. 5) from the volumetric map and sample points from them. Then, we compute the average distance from the sampled points to the ground truth point cloud. For completeness, we calculate the volume of occupied space in the ground truth occupancy map that overlaps with our reconstruction. Our reconstruction using completed depth recovers significantly more ground truth occupied space than the reconstruction with raw 16 beams, while retaining an average point-to-point error below 0.2 m.

TABLE II: Evaluation using different depth images for mapping.

Section	Type	Reconstruction Error (m)	Reconstruction Vol. (m^3)	Free Space Correct (m^3)	Free Space Incorrect
NCD	raw 16 (5 Hz)	0.08	25.5	6025.7	1.55%
	raw 64	0.09	52.0	12916.1	1.50%
	completed mono	0.17	100.7	10479.2	0.85%
Maths Inst.	raw 16	0.12	28.4	3675.1	2.3%
	raw 64	0.13	46.7	6798.6	2.9%
	completed mono	0.15	23.9	2791.8	2.2%
	completed 3-cam	0.14	37.8	5416.5	2.8%

Rejection ratio $\rho = 2$ is used for completed depth based reconstruction in NCD. We used lidar scan at 5 Hz. Only depth within 50m is integrated.

2) Free Space: We compare the free space detected by our reconstruction to free space reconstructions derived from occupancy maps produced from the ground truth point clouds of both NCD and Maths Inst. datasets. A free voxel is identified as being incorrect free space if it is free in the reconstruction but either occupied or unknown in the ground truth. We observe that the reconstruction using our approach with completed depth reveals much more correct free space than when using raw 16-beam lidar depth. The performance is similar to using the full dense 64-beam lidar. We also demonstrate the advantage of depth completion using the

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

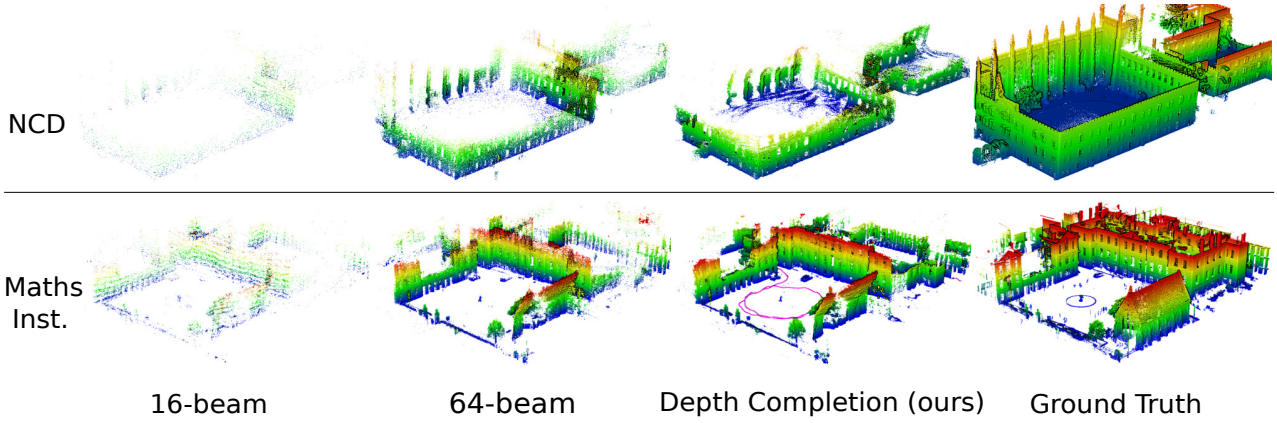


Fig. 5: Mesh reconstruction using different inputs for NCD with a monocular camera (top row) and Maths Inst. dataset with multi-cameras (bottom row). The trajectory followed by the robot in the Maths Inst. dataset is shown in the third image on the bottom row.

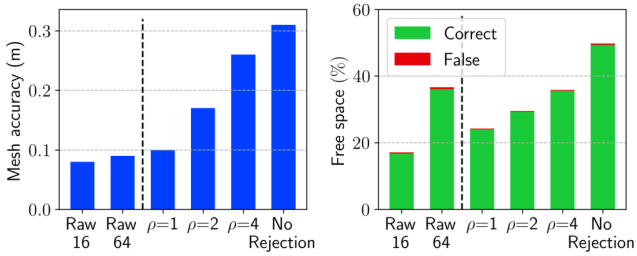


Fig. 6: Performance for different uncertainty rejection ratios.

three-camera setup compared to only upsampling with a single camera. The three-camera configuration recovers around twice the amount ($\sim$5416m^3) of free space as compared to the single camera configuration ($\sim$2791m^3) The incorrect free space from our method is within 2.5% of the total discovered free space, which is important for safe navigation.

E. Effect of Uncertainty Rejection

The uncertainty prediction from our network estimates the standard deviation of the predicted depth. Here, we present an ablation study on the effect of uncertainty rejection. We only update free space and surface information when the predicted uncertainty is less than ρ times the expected linear uncertainty given by Eq. (3). For NCD, we tested different choices of ρ and calculate the mesh accuracy, percentage of correct free space and wrong free space. We used the sparse 16-beam lidar depth as a mapping baseline. As shown in Fig. 6, both the reconstruction error and correct space increase with ρ, as expected. The detected free space is comparable to the raw 64 beam sensor for $\rho = 4$. In our experiments, we used $\rho = 2$ for a balance between reconstruction accuracy and free space detection. We illustrate the effect of uncertainty rejection in Fig. 7.

F. Path Planning

Our final experiments demonstrate the advantage of using an occupancy map, with probabilistic depth completion, for path planning. We illustrate planning results using 64-beam, 16-beam lidar and completed depth as inputs respectively. We use the RRT* planner [33] to generate a path in the

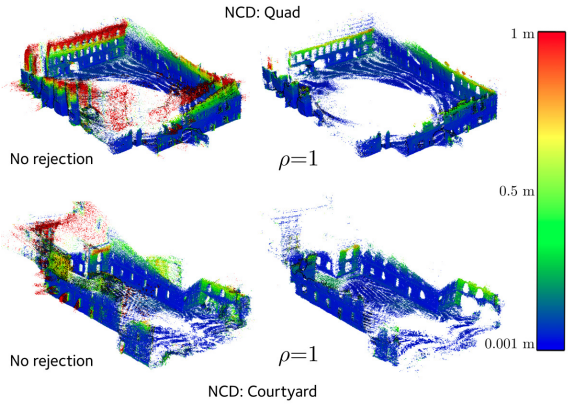


Fig. 7: Example mesh reconstructions using NCD with varying uncertainty rejection thresholds. Point-to-point errors are visualised with a colour map. Without any uncertainty rejection, we have a noisier reconstruction coloured in red.

map that passes through narrow garden passages between two buildings. As demonstrated in Fig. 8, RRT* fails to find a path around a tight corner in the map created using 16-beam lidar, as its sparse measurement cannot detect sufficient free space. In contrast, the much denser reconstruction from our approach allows RRT* to find a path in the same scenario.

V. Conclusions and Future Work

In summary, we propose a 3D lidar reconstruction framework with probabilistic depth completion, which achieves denser reconstructions as well as more free space detection compared to using sparse lidar depth images. Such an approach enables the use of low-cost sparse lidar scanners to achieve performances similar to expensive scanners with higher beam density. We demonstrate that our approach is suitable for deployment on real-world robots for applications such as 3D reconstruction and path planning.

The proposed approach facilitates map compression while preserving dense reconstructions. We would like to analyse this capability for tasks such as localization and navigation in the future. Extending the depth completion approach with wide field-of-view cameras, such as fisheye cameras is also of interest.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

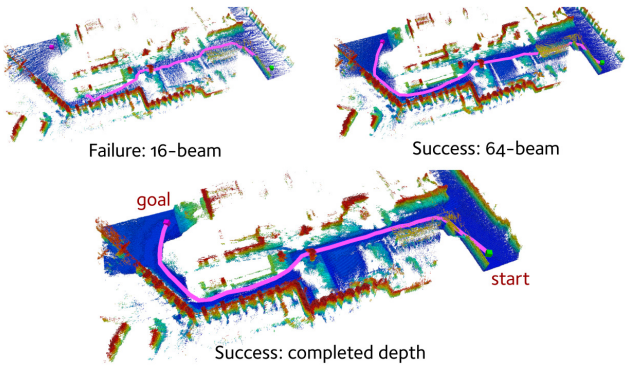


Fig. 8: Planning using a map built with 16 beam data is sparse and fails to find a path, whereas our approach using probabilistic completed depth makes it feasible.

ACKNOWLEDGMENT

The authors would like to thank David Wisth, Milad Ramezani and Matias Mattamala for their help on generating the training datasets for depth completion, Michal Staniaszek for his assistance with operating the Spot robot.

REFERENCES

[1] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant CNNs," *Intl. Conf. on 3D Vision (3DV)*, p. 11–20, 2017.

[2] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2018, pp. 4796–4803.

[3] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from LiDAR and monocular camera," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 3288–3295.

[4] D. Wofk, F. Ma, T.-J. Yang, S. Karaman, and V. Sze, "Fastdepth: Fast monocular depth estimation on embedded systems," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6101–6108.

[5] X. Cheng, P. Wang, C. Guan, and R. Yang, "CSPN++: Learning context and resource aware convolutional spatial propagation networks for depth completion," *AAAI Conf. on Artificial Intelligence*, vol. 34, no. 07, pp. 10 615–10 622, Apr. 2020.

[6] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "PENet: Towards precise and efficient image guided depth completion," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021.

[7] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the CPU," *Conf. on Computer and Robot Vision (CRV)*, p. 16–22, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[10] J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, "DeepLiDAR: Deep surface normal guided depth prediction for outdoor scene from sparse LiDAR data and single color image," in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2019, pp. 3313–3322.

[11] M. Jaritz, R. d. Charette, E. Wirbel, X. Perrotton, and F. Nashashibi, "Sparse and dense data with CNNs: Depth completion and semantic segmentation," *Intl. Conf. on 3D Vision (3DV)*, p. 52–60, 2018.

[12] X. Cheng, P. Wang, and R. Yang, "Depth estimation via affinity learned with convolutional spatial propagation network," in *Eur. Conf. on Computer Vision (ECCV)*, 2018, pp. 103–119.

[13] A. Eldesokey, M. Felsberg, and F. S. Khan, "Confidence propagation through CNNs for guided sparse depth regression," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 42, no. 10, pp. 2423–2436, 2020.

[14] M. Fehr, T. Taubner, Y. Liu, R. Siegwart, and C. Cadena, "Predicting Unobserved Space for Planning via Depth Map Augmentation," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 30–36.

[15] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, p. 1366–1373, 2017.

[16] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction," *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, p. 6565–6574, 2017.

[17] X. Ye, X. Ji, B. Sun, S. Chen, Z. Wang, and H. Li, "DRM-SLAM: Towards dense reconstruction of monocular SLAM with scene depth fusion," *Neurocomputing*, vol. 396, p. 76–91, 2020.

[18] X. Zuo, N. Merrill, W. Li, Y. Liu, M. Pollefeys, and G. Huang, "Codevio: Visual-inertial odometry with learned optimizable dense depth," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 382–14 388.

[19] H. Matsuki, R. Scona, J. Czarnowski, and A. J. Davison, "Codemapping: Real-time dense mapping for sparse slam using compact scene representations," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7105–7112, 2021.

[20] M. Popović, F. Thomas, S. Papatheodorou, N. Funk, T. Vidal-Calleja, and S. Leutenegger, "Volumetric occupancy mapping with probabilistic depth completion for robotic navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5072–5079, 2021.

[21] N. Funk, J. Tarrio, S. Papatheodorou, M. Popović, P. F. Alcantarilla, and S. Leutenegger, "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3553–3560, 2021.

[22] D. Nix and A. Weigend, "Estimating the mean and variance of the target probability distribution," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 1, 1994, pp. 55–60 vol.1.

[23] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Conf. on Neural Information Processing Systems (NIPS)*, 2017, pp. 5575–5585.

[24] Y. Wang, N. Funk, M. Ramezani, S. Papatheodorou, M. Popović, M. Camurri, S. Leutenegger, and M. Fallon, "Elastic and Efficient LiDAR Reconstruction for Large-Scale Exploration Tasks," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2021.

[25] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.

[26] 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 (IROS)*, 2020.

[27] 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.

[28] 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.

[29] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," 2020.

[30] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Trans. on Image Processing*, vol. 30, p. 1116–1129, 2021.

[31] L. Teixeira, M. R. Oswald, M. Pollefeys, and M. Chli, "Aerial single-view depth completion with image-guided uncertainty estimation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, p. 1055–1062, 2019.

[32] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Eur. Conf. on Computer Vision (ECCV)*, 2018, pp. 652–667.

[33] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Intl. J. of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on September 26,2023 at 15:13:18 UTC from IEEE Xplore. Restrictions apply.

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).**

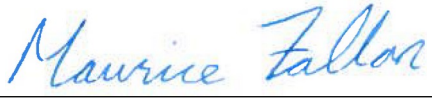
Title of Paper	3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation
Publication Status	✕Published ☐ Accepted for Publication ☐Submitted for Publication ☐Unpublished and unsubmitted work written in a manuscript style
Publication Details	Y. Tao, M. Popovi ´c, Y. Wang, S. T. Digumarti, N. Chebrolu, and M. Fallon, "3D lidar reconstruction with probabilistic depth completion for robotic navigation," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022, pp. 5339–5346.

Student Confirmation

Student Name:	Yifu Tao		
Contribution to the Paper	• Contributed to development of the idea • Developed the depth completion network integrated with a mapping pipeline • Conducted experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	22nd January 2025

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: Maurice Fallon			
Supervisor comments Yifu was the lead investigator, implementer and researcher on this project and wrote the bulk of the paper			
Signature		Date	22nd January 2025

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

4.2 Technical Remarks

Generalisation

The ability to generalise to unseen data is a critical attribute of a machine learning model. For the task of vision-guided depth completion, we encountered several different types of domain gaps.

In the early stage of this work, we trained a depth completion network directly on real data, with 16-beam lidar input and 64-beam lidar ground truth. We found two interesting results when testing the network.

The first finding is that the network only works in locations where there is depth ground truth. In Fig. 4.1, the network is trained with 64-beam lidar in the form of a depth image as ground truth. The dense completed depth results are reasonable within the 64-beam-lidar-mask, but are otherwise random. This is due to the fact that the ground truth depth has a stable sparsity pattern which is determined by the lidar projection model and camera projection model. The network is therefore only trained to predict depth at these specific locations (the ground truth mask), and the U-Net [110] structure does not allow it to generalise in regions elsewhere.

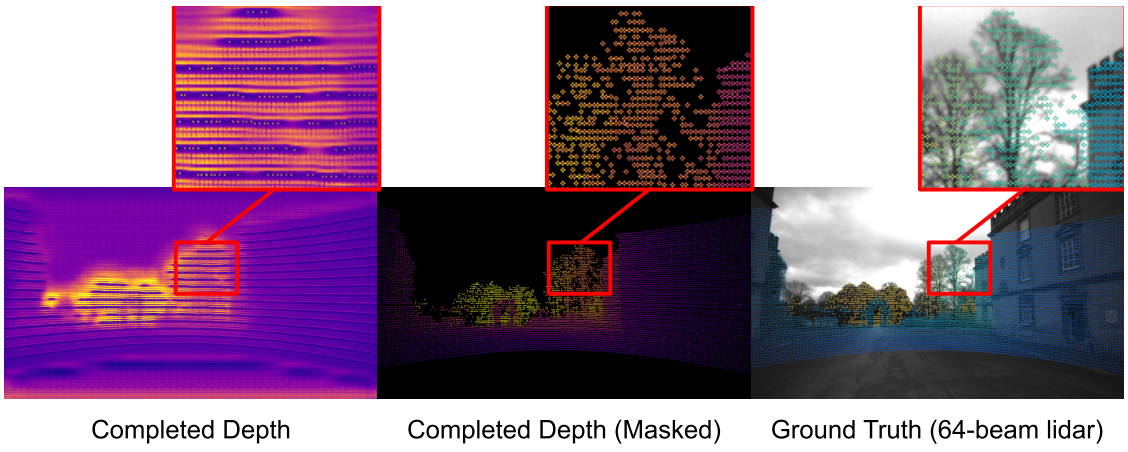


Figure 4.1: Depth completion results when trained with 64-beam lidar depth as ground truth. The network predicts reasonable results only in specific locations in the image where there is ground truth depth.

This motivated us to increase the density of the depth image. This can be achieved by accumulating lidar scans around the current pose using the estimated trajectory from the SLAM system. One limitation is that the trajectory inevitably

includes error which is propagated to the ground truth lidar depth. This then further motivated us to use synthetic datasets, which provide accurate ground truth depth, in the final system presented in this chapter.

The second finding is the network output does not generalise when the input lidar type is different from one used in training. One might expect that the task of completing sparse depth would be made simpler when the input depth is denser: when the input depth is sufficiently dense, a simple linear interpolation could work well assuming that there are planar structures in the scene whose depth is being completed; when the input depth is very sparse, the depth completion problem is more ambiguous and extrapolation is required.

However, providing denser input depth does not help the depth completion network. In Fig. 4.2, the network is trained with 16-beam lidar as input. However, when provided with a denser depth input of 64-beam lidar, the results are much worse.

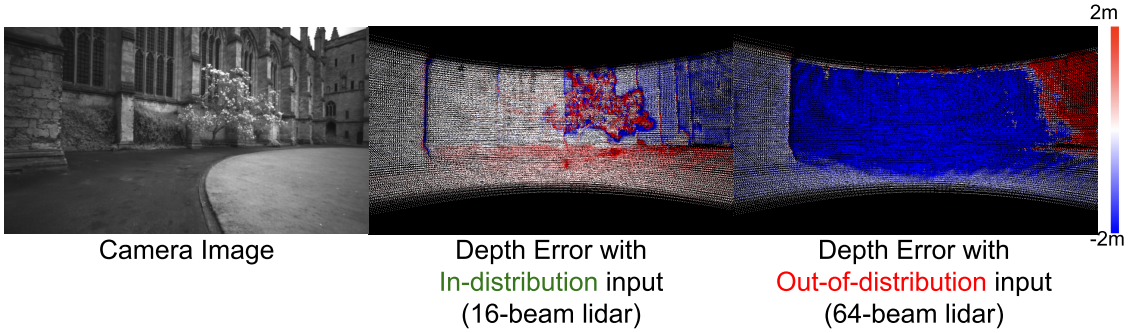


Figure 4.2: A depth completion network tested with in-distribution input and out-of-distribution input. The network does not generalise to an input with different levels of depth sparsity, even if there is more depth information available which should make the completion task easier.

This is a fundamental characteristic of the network's convolution operation when provided with incomplete data. As discussed in Sec. 3.4, the missing data in the depth image still has a value (of zero), and the convolution result is affected when the sparsity (number of zeros in the receptive field) changes. Based on this observation, we then used a fixed lidar sparsity pattern in the input—we trained the network with 16-beam lidar input and tested it with 16-beam lidar input. In fact, we found that if the input lidar sparsity pattern is kept similar, our depth network

trained on synthetic data can generalise well to real-world scenes. Additionally, there are alternative approaches in the literature to handle the generalisation issue of convolutional neural networks when taking inputs of different levels of sparsity. For example, normalised convolution layers [114] are shown to generalise to different levels of sparsity in the input, and have been applied to depth estimation networks [116]. These are interesting approaches that can be also used in our method for it to generalise to different types of lidar.

Synthetic Data: Calibration and the Sky

Generating training data for depth completion from real-world data is challenging. The input camera image and lidar data have to be synchronised (via lidar motion undistortion) and precisely calibrated. The ground truth depths obtained from accumulated lidar data such as those from KITTI [157] contain errors from the lidar sensor noise and trajectory error. In addition, lidar-based depth ground truth contains no information in the sky. Network trained with such data output random depth values in the sky, which lead to undesirable artefacts in mapping.

To mitigate theses issues, we leveraged synthetic data from the virtual KITTI dataset [173] which provides perfect complete depth images (including the sky) synchronised with the camera images. Since it does not provide input lidar data, we simulated the lidar depth input by sampling the ground truth depth image with a pattern we obtained from our real-world lidar depth image. We found this strategy allows the network to generalise when using real-world camera images with 16-beam lidar data as input. Additionally, we encode the depth of the sky to be a large value (256 metres). This allows the network to output a pre-defined large depth for the sky in test time, which can be filtered during the mapping process.

We used real-world data for test time depth completion. The lidar-camera synchronisation was handled by motion-unditorting the lidar to the camera timestamp. The lidar-camera extrinsics calibration was obtained by manually fine-tuning the rotation angles from a prior extrinsics from the 3D CAD model of the sensor setup. The resultant is less precise compared to a proper calibration such as DiffCal [174]

which is used in subsequent works in Chap. 5,Chap. 6 and Chap. 7. Imperfect input data can result in less accurate depth completion. This is particularly significant to object boundaries where a rotation error leads to a large change of depth. Our uncertainty estimation partially mitigates this, since the uncertainty is typically high in the boundary areas. This also implies that this approach is less suitable for reconstructing detailed geometries, which then motivates our NeRF-based reconstruction system presented in Chap. 5 and Chap. 6.

4.3 Discussion

Epistemic Uncertainty

We used the probabilistic depth completion network to predict the *aleatoric* uncertainty which is the inherent noise in the data. This often appears in object boundaries where there is a sharp depth discontinuity in the ground truth depth. However, the output will also have *epistemic* uncertainty which is due to the limited training data. When the test data is very different from the training data, the model's output will be inherently unpredictable and uncertain. Techniques such as dropout [175] and ensemble learning [176] can be used to allow the estimation of epistemic uncertainty, which can result in a more calibrated and accurate full uncertainty prediction. One limitation of these techniques is that they require multiple forward passes for the uncertainty estimation, and are hence computationally expensive. Recently, evidential learning has been adopted by methods on semantic segmentation [177] as an efficient alternative for epistemic uncertainty prediction.

Geometric Consistency

The monocular depth estimation task that we are solving is inherently an ill-posed problem. The completed depth, when reprojected into 3D, is not guaranteed to be geometrically consistent across a sequence of observations. In our work, we address this inconsistency issue by using the occupancy mapping framework which probabilistically fuses the measurements. In the literature, there are other

approaches to handle the geometric consistency problem in depth estimation networks. One approach proposed in CodeSLAM [34] is to learn a latent space for the output depth, and to optimise the latent space such that photometric consistency is achieved when one image is warped to another overlapping image based on the estimated depth. Another approach is to not perform monocular depth estimation but instead deep Multi-View Stereo. This has been adopted by SLAM systems such as TANDEM [66]. More recently, DUSt3R [24] introduces a method that takes in multi-view images and directly outputs a single point cloud. Compared to the aforementioned methods, our approach of integrating with occupancy mapping (which is effectively a Bayes filter) is a loosely-coupled approach.

The work presented in Chap. 5 and Chap. 6 addresses the consistency problem by adopting a map-centric approach, rather than the view-centric approach which the depth completion method has taken. The map-centric approach using NeRF maintains a single global map and optimises multiple frames to achieve photometric consistency and geometry consistency.

Finally, while it is important to achieve geometric consistency across frames for 3D reconstruction, there are also applications where monocular depth estimation is sufficient. For example, when there are dynamic obstacles which the robot must manoeuvre to avoid, monocular depth completion is very useful to identify and track such objects, which can be otherwise challenging if only using a sparse lidar. Our network runs up to 10 Hz, and is hence suitable for this application.

5
Scalable Lidar-Visual NeRF

Contents

5.1 SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection 66
5.2 Technical Remarks 75
5.3 Discussion 78

This chapter presents a map-centric reconstruction framework based on the Neural Radiance Field (NeRF) representation. This is motivated by NeRF's capability to render photorealistic images, which is not matched by classical visual reconstruction methods such as MVS. However, NeRF methods are often tested with small-scale scenes with object-centric trajectories. When mapping a large-scale outdoor scene, the camera images collected by a mobile robot are relatively sparse and do not always form good view constraints which are needed to determine the 3D structure. In these kinds of conditions, the NeRF mapping may look photorealistic, but the underlying geometry would not be precise, especially for visually ambiguous or textureless areas such as blank walls or tarmacadam roads. For such areas, lidar data can be useful since it provides range measurements regardless of texture. In addition, mapping large-scale scenes with a single NeRF is challenging due to increased demand on computer and GPU memory, and the model's representation

capability. Large-scale environments result in more image observations, which in turn makes the pose computation, often done by a Structure-from-Motion (SfM) method, more time-consuming. These challenges need to be addressed for NeRFs to be applied in a robotics context.

We introduce a system named SiLVR that integrates the NeRF framework with a lidar SLAM system to map large-scale urban environments. We use lidar to provide geometric constraints of depth and surface normal to the reconstruction to overcome the issue of visual ambiguity. Our system also leverages the online lidar SLAM trajectory to bootstrap SfM to accelerate the pose estimation as well as to obtain the correct (metric) scale. The full trajectory is then partitioned into submaps which allows the reconstruction to scale without raising hardware requirements.

This system also benefits from the lidar-visual hardware sensor setup. We used three synchronised cameras whose wide Field-of-View are instrumental for the visual pose estimation and visual reconstruction. The lidar is synchronised with the images via motion undistortion, and is precisely calibrated to the camera using DiffCal [174]. This effort is further developed into the lidar-visual dataset named Oxford Spires which is presented in Chap. 7.

5.1 SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection

The article in this section was presented in *IEEE International Conference on Robotics and Automation (ICRA) 2024* [36]. An accompanying video is available online at `https://youtu.be/kA11bdMbhMo`.

2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection

Yifu Tao[1], Yash Bhalgat[2], Lanke Frank Tarimo Fu[1], Matias Mattamala[1], Nived Chebrolu[1], and Maurice Fallon[1]

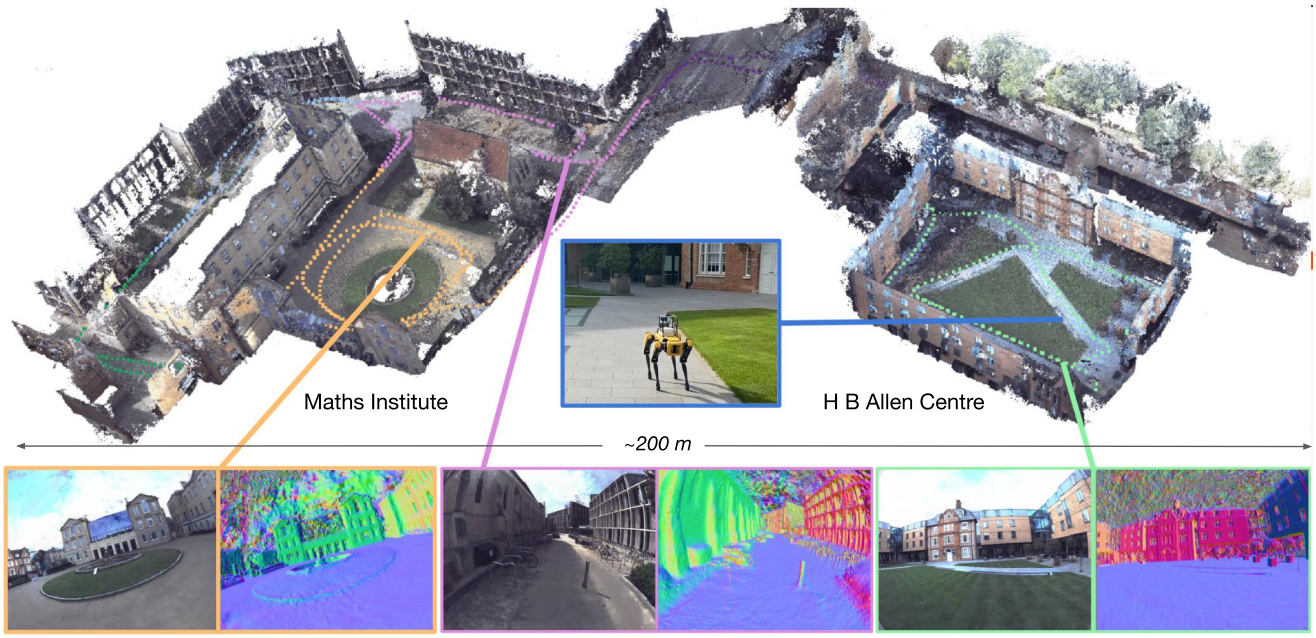


Fig. 1: Large-scale reconstruction consisting of 8 submaps of Maths Institute and H B Allen Centre in Oxford. The bottom row shows the novel views synthesised from the model and surface normals at three different locations. The trajectory of each submap is visualised in a different colour.

***Abstract*— We present a neural-field-based large-scale reconstruction system that fuses lidar and vision data to generate high-quality reconstructions that are geometrically accurate and capture photo-realistic textures. This system adapts the state-of-the-art neural radiance field (NeRF) representation to also incorporate lidar data which adds strong geometric constraints on the depth and surface normals. We exploit the trajectory from a real-time lidar SLAM system to bootstrap a Structure-from-Motion (SfM) procedure to both significantly reduce the computation time and to provide metric scale which is crucial for lidar depth loss. We use submapping to scale the system to large-scale environments captured over long trajectories. We demonstrate the reconstruction system with data from a multi-camera, lidar sensor suite onboard a legged robot, hand-held while scanning building scenes for 600 metres, and onboard an aerial robot surveying a multi-storey mock disaster site-building. Website: `https://ori-drs.github.io/projects/silvr/`**

[1]Oxford Robotics Inst., Dept. of Eng. Science, Univ. of Oxford, UK. {yifu, fu, matias, nived, mfallon}@robots.ox.ac.uk
[2]Visual Geometry Group, Dept. of Eng. Science, Univ. of Oxford, UK. yashsb }@robots.ox.ac.uk.
This project has been partly funded by the Horizon Europe project Digiforest (101070405). Maurice Fallon is supported by a Royal Society University Research Fellowship. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising.

I. INTRODUCTION

Dense 3D reconstruction is a task that underpins a range of robotic applications such as industrial inspection and autonomous navigation. Common sensors used for reconstruction include cameras and lidar. Camera-based reconstruction systems use techniques including Structure-from-Motion (SfM) and Multi-View Stereo (MVS) to produce dense textured reconstructions [1]. However, these systems rely on good lighting conditions as well as having observed the many view constraints. They also struggle with textureless areas. Lidar provides accurate geometric information at long range — as it directly measures distances to surfaces. This makes it desirable for large-scale outdoor environments, but the sensor measurement is usually much sparser than a camera image. It also does not capture colour which is important for applications such as virtual reality and 3D asset generation.

Classical reconstruction systems have used point clouds, occupancy maps, and sign-distance fields (SDF) as their internal 3D representation. Recently, neural radiance field (NeRF) [2] has gained popularity for visual reconstruction. With differentiable rendering, this approach optimises a continuous 3D representation by minimising the difference between the rendered image and reference camera images. It

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611278

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

can achieve state-of-the-art novel view synthesis quality.

As with traditional vision-based reconstruction methods, NeRF struggles to estimate accurate geometry in locations where there is limited multi-view input and little texture. Autonomous systems commonly encounter these situations - for example, reconstructing a wall with uniform colour when traveling directly towards it. This problem could be addressed by using lidar sensing to give accurate geometric measurements in these textureless objects. In addition, the use of lidar data can mitigate the need to capture multiple camera views. It is impractical for an inspection robot (e.g. The Spot quadruped on an industrial facility) to execute object-centric trajectories just to improve the visual reconstruction. This motivates the development of a reconstruction system that fuses these sensors.

In this work, we present a NeRF-based reconstruction system that integrates both lidar and visual information to generate accurate, textured 3D reconstruction which also provides photo-realistic novel view synthesis. Our method builds upon NeRF implementation [3] utilising hash encoding [4] that takes minutes to achieve photo-realistic rendering. This is extended with geometric constraints from lidar to improve reconstruction quality. The use of lidar enables depth measurements even from featureless areas, which cannot be obtained from SfM [5]. Surface normal can also be computed from a lidar scan, which is more robust than learning-based priors [6] which can suffer from input data distribution shift in real-world deployment.

This system is demonstrated using data from a perception sensor suite which contains three wide field-of-view cameras and a 3D lidar to enable robots to reconstruct in all directions. We take advantage of a lidar-inertial odometry and SLAM mapping system [7] as part of the pipeline. Experiments are presented using a drone, a legged robot, and a handheld device in industrial and urban environments.

In summary, our main contributions are:

- A dense textured 3D reconstruction system that achieves accurate geometry that's on-par with lidar, and photorealistic novel view synthesis
- Integration with a lidar SLAM system, so that the NeRF is trained with both depth and surface normal obtained from lidar data, and metric-scale trajectory with a reduction in computation time by 50% compared with up-to-scale offline Structure-from-Motion method commonly used in literature.
- A sub-mapping system that scales to large outdoor environments — with trajectories over 600 metres
- Evaluation of the system on real-world large-scale outdoor datasets captured from multiple robotic platforms.

II. RELATED WORKS

A. Large-scale 3D Reconstruction

Lidar is the dominant sensor for accurate 3D reconstruction of large-scale environments [8], [9], thanks to its accurate, long-range measurements. Volumetric lidar mapping relies on high-frequency odometry estimates from scan-matching and IMU measurements [7], [10]–[12]. Yet, lidar-based systems may yield partial reconstructions, especially when robots explore scenes and have limited field-of-view sensors. Visual cues can be used to densify lidar mapping [13].

Alternatively, SfM systems such as COLMAP [1] can generate large-scale textured reconstructions from images by first estimating camera poses using sparse feature points in a joint bundle adjustment process, followed by refinement with multi-view stereo algorithms [14]. However, they face challenges in low-texture areas, repetitive patterns, and feature matching across views can be problematic with changing lighting or non-Lambertian materials. Urban Radiance Fields [15] proposes using lidar sweeps along with RGB images to optimise a neural radiance field model that can be used for 3D surface extraction. Our work shares this approach, fusing both sensor modalities to generate precise geometry, overcoming limitations of vision-only approaches in low-texture areas, and being much denser than lidar-only reconstructions.

A common strategy to extend dense reconstructions to large-scale areas is through submapping [16]–[20]. These approaches partition the scene into individual local submaps which can incorporate the effect of loop closure corrections while still producing a consistent global map while achieving significantly lower runtime. Our work also adopts the submapping approach and partitions large-scale scenes into local maps (approximately 50x50m) using a globally-consistent lidar SLAM trajectory. This increases the representation capability and improves reconstruction, especially for thin objects.

B. Neural Field Representation

Neural Radiance Fields (NeRF) [2] proposed using a multilayer perceptron (MLP) to represent a continuous radiance field with differentiable volume rendering to reconstruct novel views. It implicitly induces multi-view consistency with geometric priors in the learned encodings. NeRF and its many variants used frequency encoding [21] to encode spatial coordinates, but these suffer from long training times, typically a few hours per scene. Alternative explicit representations of radiance fields, including dense voxel-grids with trainable per-vertex features [4], [22] and more recently 3D Gaussians [23] are shown to accelerate the training, at the cost of being more memory intensive. Octree or sparse-grid structures [4], [24] can reduce memory usage by pruning grid-features in empty space. Our work is built upon Nerfacto [3] which incorporates the main features from other NeRF works [4], [25], [26] that have been found to work well with real data.

While NeRFs excel at high-quality view synthesis, obtaining a 3D surface from these representations remains challenging, mainly due to the flexible volumetric representation being under-constrained by the limited multi-view inputs. One approach to improve the reconstruction is to impose depth regularisation [5], [15] or surface normal regularisation [6]. Another approach is to impose surface priors on the volumetric field and use representations such as Signed Distance Field (SDF) [27], [28] to enforce a surface reconstruction output, although the novel view synthesis

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

quality might be compromised [29] with this approach. Our method is built upon a volume density representation which is extended with depth [5] and surface normal [6] regularisations from lidar measurements instead of using SfM [5] or learnt priors [6]. In particular, it can significantly improve the reconstruction quality in texture-less areas with smooth surface.

Neural field representations have been used for lidar-base mapping [30], [31], showing promise in generating more complete and compact reconstructions than traditional methods. While these works also build upon implicit map representation, they do not use visual data for building the map. Our system uses visual information and multi-view geometry constraints, therefore can reconstruct regions outside lidar field-of view.

III. METHOD

We present a system for large-scale 3D reconstruction based on a NeRF representation tailored for robotic inspection tasks. We use a custom-designed sensor payload with a 3D lidar sensor, three fisheye cameras, and an IMU, which is suitable for use on various robot platforms. In Sec. III-B, we present our approach to fusing both lidar and vision information during the optimization phase to ensure high-quality reconstruction. Furthermore, we employ a submapping strategy to scale the approach to large areas described in Sec. III-D. An overview of the system is presented in Fig. 2.

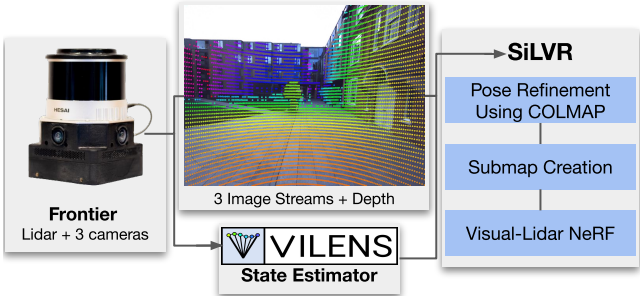


Fig. 2: System Overview: Frontier, our custom perception unit, has three fisheye colour cameras with an IMU and a 3D lidar. Our online state estimator's trajectory is refined with COLMAP and partitioned into submaps. The camera image, lidar depth, and normal image are used to train a NeRF to get the final 3D reconstruction.

A. NeRF-based Scene Representation

Our work builds on the differentiable volume rendering framework used for novel view synthesis [2], [32]–[34]. To render a novel view from a NeRF given a viewpoint, we cast rays from the camera origin along the viewing direction for each pixel u in the image plane, and render the pixel-colour by integrating over points sampled along the ray. This volume rendering integral is approximated using quadrature rule [35], [36] as $\hat{c}_u = \sum_{i=0}^{N} w_i c_i$, where

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

Here, σ_i and c_i are the predicted density and color for the sampled 3D points and $\hat{c}_u$ is the rendered pixel color.

Our implementation is built on top of the Nerfacto method from Nerfstudio [3]. Nerfacto's rendering quality is comparable to state-of-the-art methods such as MipNeRF-360 [25] while achieving a substantial acceleration in reconstruction speed since it also incorporates efficient hash encoding from Instant-NGP [4]. We also use scene contraction proposed in [25] to improve memory efficiency and represent scenes with high-resolution content near the input camera locations. The contraction function non-linearly maps any point in space into a cube of side length 2, and represents the scene within this contracted space. Since there is large variation in exposure and lighting conditions, we use a per-frame appearance encoding for each image, similar to [15], [26].

B. Geometric Constraints from Lidar Measurements

3D reconstruction with NeRF becomes challenging when the surface has uniform texture and limited multi-view constraints. Lidar measurements are complementary as it can provide accurate measurements in such scenarios. In our work, we incorporate the lidar measurements in the NeRF optimization. Specifically, we impose a lidar-based depth regularisation by adding a depth loss [5] defined as the KL-Divergence between a normal distribution around the lidar depth-measurement $\mathbf{D}$ and the rendered ray distribution $h(t)$ from the NeRF model:

$$\mathcal{L}_{\text{Depth}} = \sum_{\mathbf{r}\in\mathcal{R}} \text{KL}[\mathcal{N}(\mathbf{D}, \hat{\sigma}) \| h(t)] \quad (2)$$

We also run a semantic segmentation network [37] to obtain a sky mask, and minimise the weights of these rays similar to [15].

While the depth loss improves 3D reconstruction, we found that the surface contains wavy artifacts in regions where it is expected to be smooth (see Fig. 5). To mitigate this, we compute the surface normal as the negative gradient of the NeRF's density field, and impose a further surface normal regularisation loss, inspired by [6]:

$$\mathcal{L}_{\text{normal}} = \sum_{\mathbf{r}\in\mathcal{R}} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \left\|1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r})\right\|_1 \quad (3)$$

C. Bootstrapping Camera Poses from SLAM with scale

Obtaining accurate camera poses is crucial as the pose accuracy directly impacts the fidelity of the reconstructed model. A popular approach used in most NeRF works is to obtain camera poses using offline Structure-from-Motion methods such as COLMAP [1]. However, COLMAP has the following limitations: (1) long computation time, especially for large image collections collected over a long trajectory, and (2) inability to register all frames into one global map when there is limited visual overlap between the images. These issues limit its application in building a large-scale globally consistent map for robotic applications.

In our work, we use our lidar-inertial odometry and SLAM system VILENS [7]. While VILENS achieves state-of-the-art

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

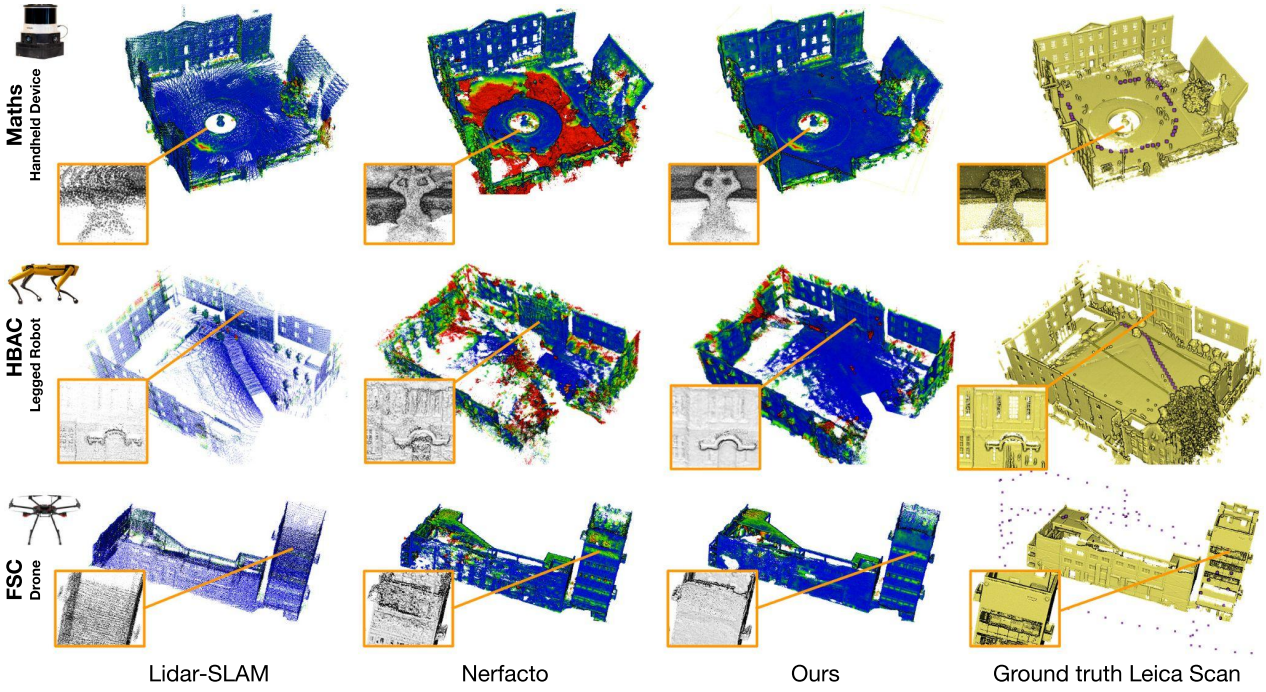


Fig. 3: Comparison of reconstruction quality of Lidar-SLAM, Nerfacto (vision-only) and our approach. Reconstructions are coloured with point-to-point distance to the ground truth with increasing error from blue (0m) to red (1m). The trajectory is shown in purple and overlaid on the ground truth scan captured using a Leica BLK360. The zoomed-in views show the difference in reconstruction quality. Overall, our approach is more complete w.r.t lidar-only reconstruction, and geometrically more consistent w.r.t vision-only reconstruction.

results for online motion tracking, we found that the camera poses obtained are less accurate than those of COLMAP. This results in *blurring* artifacts in the renderings of the NeRF model. Several works [20], [38] use noisy pose inputs and then jointly refine the poses in parallel with the NeRF optimization to generate better results. In our experiments with a collection of large number of images, we found that while this pose-refinement approach sometimes leads to slightly better PSNR (peak signal-to-noise ratio), the resulting rendering is still less sharp compared to using the COLMAP estimated poses, and the training process is usually unstable.

To overcome the above limitations, we use the SLAM poses as a pose prior and refine the trajectory using COLMAP. Specifically, we replace the COLMAP *mapper* with *point triangulator* which reads prior poses. This method has the advantage of being faster, as it converts the incremental Structure-from-Motion into a global bundle adjustment problem, and more importantly, results in COLMAP being able to merge all available images in a single map. For a mission spanning over 20 minutes, our COLMAP-with-prior pipeline achieves similar rendering quality, while only taking half the computation time compared to a fully offline COLMAP run. The computation time is similar to the robot's mission duration, making it more applicable for robotic applications.

After COLMAP computation, we rescale the trajectory using Sim(3) Umeyama alignment with lidar-slam trajectory, so that the final trajectory is metric. This is essential to use lidar measurements in III-B, since lidar depth is also metric.

D. Scaling NeRF with Submapping

Training a NeRF for large-scale scenes is challenging as a single NeRF model has limited representation power and hardware constraints such as RAM usage when loading thousands of images. We adopt a sub-mapping approach, and partition the COLMAP-refined SLAM trajectory into clusters using Spectral Clustering [39]. Different from [20], our representation is based on Instant-NGP [4] which is orders of magnitude faster than a Multilayer Perception. The submaps are trained in their local coordinate frames, and the final reconstruction are *Sim(3)* transformed to the world coordinate frame using their metric pose from III-C.

To generate 3D reconstructions from NeRF, we sample the rays used to train each model and render the colour and depth to generate a 3D point. When evaluating the 3D reconstruction, we found that submaps contain artifacts, especially around their boundary. One cause of the artifacts is that the boundary regions are observed sparsely with limited view constraints. To tackle this, we identify regions with low surface density and remove them when merging submap clouds to get the final reconstruction.

IV. EXPERIMENTAL RESULTS

A. Hardware and Datasets

We evaluate our methods on a custom perception unit called Frontier, a multi-camera lidar inertial device. It includes three 1.6 MP colour[1] fisheye Alphasense cameras on 3 sides of

[1]Raw Bayered images are processed using `https://github.com/leggedrobotics/raw_image_pipeline`

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

the device, with an IMU from Sevensense Robotics AG and a Hesai Pandar QT64 lidar. We used the Frontier to collect datasets on multiple platforms: a legged robot (Boston Dynamics Spot), a drone (DJI M600), and a hand-held device. Spot and handheld Frontier datasets were taken at the H B Allen Centre (HBAC) and Mathematical Institute (Maths Inst.) in Oxford. The DJI M600 drone was operated at the Fire Service College (FSC). In the FSC dataset, we only use the rectified front camera image as the drone propellers were visible in the left and right camera images.

We use VILENS [7], a lidar-inertial SLAM system online to provide a globally consistent trajectory and motion-corrected lidar measurements. The online SLAM trajectory is further optimised by COLMAP offline, which improves the visual reconstruction quality, as shown later in Tab. II. The lidar point clouds are projected as a depth image. The surface normal is computed from the lidar range image, and also projected as a normal image. We calibrate our cameras-to-IMU extrinsics with Kalibr [40] and cameras-to-lidar with [41]. When running COLMAP, we further optimise the intrinsics from Kalibr. For training the NeRF model, we used an Nvidia RTX 3080 Ti and one iterations takes 4096 rays.

B. Evaluation Metrics

To evaluate the geometry of the reconstruction, we report *Accuracy* and *Completeness* following the conventions of the DTU dataset [42]. Accuracy is measured as the distance from the reconstruction to the reference 3D model (ground truth) and encapsulates the reconstruction quality. Completeness is the distance from the point-wise reference to the reconstruction and shows how much of the surface is captured by the reconstruction. For the ground truth, we use the centimetre-accurate point cloud captured by a survey-grade Leica BLK360 laser scanner and register the lidar point clouds using transformation obtained from ICP, which is also used to register NeRF reconstructions.

We also evaluate the visual quality by reporting the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [43]. Note that our images have various exposure times, which lower the test PSNR even if the reconstructed image is photorealistic.

C. Evaluation of the 3D Reconstruction

We perform a quantitative evaluation of our method on real-world datasets captured by different robotic platforms. We compare the point cloud reconstructions generated with the following configurations:

1) Lidar-SLAM: lidar point cloud registered with SLAM poses refined by COLMAP
2) Nerfacto [3]: baseline method using only the images
3) SiLVR: Our method using photometric loss, depth loss, and surface normal loss

We sample the training rays and generate 10 million points from each submap for Nerfacto and SiLVR. For Nerfacto, we excluded the points belonging to the sky by computing a sky segmentation mask with further manual cropping. For lidar-slam, we only include points within the camera field-of-view. This omits points pointing backwards that may be occluded by the operator.

TABLE I: Evaluation of 3D Reconstruction Quality

Method	Accuracy↓ (m)	Completeness↓ (m)	PSNR↑ train	PSNR↑ test	SSIM↑ test
Maths Quad					
lidar-SLAM	**0.06**	0.15	/	/	/
Nerfacto mono	1.38	0.33	**31.3**	17.7	0.65
Nerfacto 3-cam	0.76	0.31	28.0	**21.1**	**0.72**
Ours mono	0.08	0.12	30.1	17.5	0.61
Ours 3-cam	0.08	**0.11**	27.3	20.5	0.71
Oxford HBAC					
lidar-SLAM	**0.05**	0.25	/	/	/
Nerfacto mono	0.49	5.40	**32.6**	19.5	0.65
Nerfacto 3-cam	0.28	0.40	29.8	20.6	**0.74**
Ours mono	0.30	4.60	31.0	**21.2**	**0.74**
Ours 3-cam	0.09	**0.18**	28.8	19.7	**0.74**
FSC					
lidar-SLAM	**0.08**	**0.08**	/	/	/
Nerfacto mono	0.14	0.11	**28.8**	**19.1**	**0.76**
Ours mono	0.11	0.09	27.7	**19.1**	0.75

NeRF models are trained for 10000 iterations for 5 minutes; When computing accuracy and completeness, we crop regions where there is a change or no overlap between the ground truth and lidar scans.

We summarise the quantitative results in Tab. I, and show 3D reconstructions in Fig. 3. Compared to Nerfacto, our method incorporates lidar measurements and has significantly better geometry which is shown in terms of accuracy and completeness. Nerfacto struggles when reconstructing the uniformly-coloured ground in Maths Inst., and the quad area in HBAC where the robot only walked forwards. Compared to lidar-SLAM, our method generally achieves more complete reconstruction since it uses dense visual information, while the accuracy (8-11cm) is nearly on-par with lidar-slam (6-8cm) and much better than Nerfacto (14-76cm).



Fig. 4: Comparison of reconstruction of HBAC building using the front camera only vs. using all the three cameras. The three-camera setup generates more complete and accurate reconstructions compared to using only a single front-facing camera. The multi-camera setting is important in robotic applications where it would be infeasible to actively scan the entire scene to obtain strong viewpoint constraints.

The advantage of our multi-camera sensor setup is demonstrated qualitatively in Fig. 4. Compared to the three-camera

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

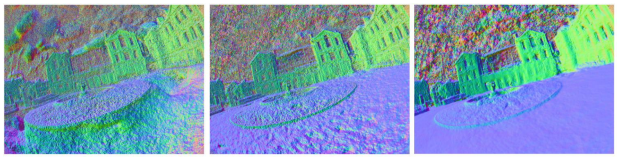


Fig. 5: Comparison of surface normal renderings of the Maths Institute. Incorporating normal constraints in addition to depth from lidar improves the smoothness of the reconstruction. Right: The smooth reconstruction of the ground portion highlights this improvement.

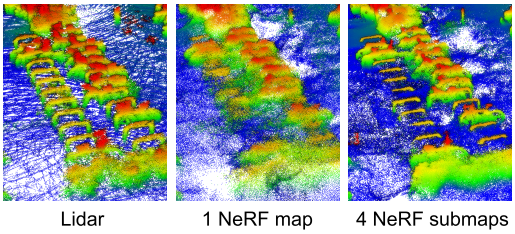


Fig. 6: Effect of submap size on reconstruction quality. A larger number of submaps for a given area results in better reconstruction on thin objects such as the bike racks on the right.

setup, using only the front-facing camera leads not only to an incomplete reconstruction but also worse geometry. Visual reconstruction with photometric loss is biased towards generating a good rendering only at the input viewing angle. The reconstruction using the front-only camera in Fig. 4 is trained with images viewing a shallow angle of the scene, and results in a poor geometric reconstruction when rendered from an unseen angle.

D. Effect of Lidar Surface Normal Loss

While the depth loss Eq. (2) provides geometric constraints, we observe that the resulting 3D reconstruction is not necessarily smooth for flat surfaces. The results in Fig. 5 demonstrate how the surface normal loss Eq. (3) further constrains the reconstruction geometry and improves the reconstruction quality. Nerfacto fails to estimate the depth of the ground due to its uniform texture. Using the lidar depth loss ensures ground reconstruction is at the right height, however, the surface is still not smooth. The surface normal loss furthers smoothens the surface and results in a higher-quality reconstruction.

TABLE II: Ablation: Effect of Bootstrapping SLAM Poses

Method	Features	Prior	Traj. Registered (%)	PSNR↑ Train	PSNR↑ Test	SSIM↑ Test	Time (s)
VILENS	/	/	100.0	23.0	17.4	0.64	Online
NeRF refined	/	/	100.0	23.2	17.9	0.65	Online
COLMAP Sequential	1024		57.6	25.9	19.1	0.71	3299.2
	1024	✓	100.0	26.2	**20.6**	**0.74**	1729.9
	8192		94.0	26.1	19.8	0.72	7850.0
	8192	✓	100.0	26.2	20.4	0.73	4448.4
COLMAP VocabTree	1024		54.7	26.2	19.0	0.71	4444.8
	1024	✓	100.0	26.3	20.4	0.73	1052.5
	8192		94.8	**26.6**	19.9	0.72	37067.5
	8192	✓	100.0	26.3	20.4	**0.74**	11015.3

Results evaluated on HBAC-Maths dataset with 3254 images and duration of 1270s. Models trained for 4000 iterations. PSNR and SSIM are evaluated after masking out the sky.

E. Effect of Bootstrapping SLAM Poses

We compare the performance of different strategies for computing poses: online SLAM poses, SLAM poses with NeRF pose refinement [3], SLAM poses with COLMAP [1] in different configurations, and COLMAP without any prior poses. For COLMAP, we tested different numbers of features extracted per image, and two different COLMAP feature matching algorithms: sequential matching with loop closures and Vocabulary Tree Matcher.

The results are summarised in Tab. II. For all COLMAP configurations, providing the SLAM prior poses not only accelerates pose computation, but also leads to better test rendering, compared to the offline COLMAP. Our SLAM prior poses also register all images in the trajectory, while when not provided, COLMAP only registers 55%-95% images. Extracting more visual features per image (from 1024 to 8192) leads to higher percentage of image registration and better visual reconstruction (PSNR and SSIM). This comes at the expense of a higher computation time, especially with the VocabTree matcher. Using the COLMAP Sequential Matcher is generally faster than Vocabulary Tree Matcher.

F. Submapping for Large-Scale Environments

We show a large-scale reconstruction of HBAC-Maths with the handheld Frontier using submapping in Fig. 1, as well as the trajectory for each submap. To demonstrate the advantage of submapping, we compare the 3D reconstruction and rendering quality using one NeRF model for the entire sequence versus NeRF models built with multiple submaps. We present the qualitative results in Fig. 6. The reconstruction of the bike racks in Maths Inst. is blurred when only using a single NeRF map due to its limited representation capability for storing all objects over a large area. While using only a dedicated submap for that local area, the reconstruction quality improves significantly as seen in Fig. 6 (right).

V. CONCLUSIONS

In summary, we proposed a large-scale 3D reconstruction system fusing both lidar and vision in a neural field via differentiable rendering. The proposed sensor fusion overcomes the limitations of individual sensors, namely the sparsity of the lidar and the fragility of vision in the presence of texture-less surface, and limited multi-view constraints. We demonstrate large-scale reconstruction results from real-world datasets collected from multiple robot platforms in conditions suited for inspection tasks.

ACKNOWLEDGMENT

The authors would like to thank Ren Komatsu for his help on software development, Tobit Flatscher for deploying spot robot, Rowan Border for drone data collection, and Sundara Tejaswi Digumarti for his helpful discussions.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2016.

[2] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[3] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *SIGGRAPH*, 2023.

[4] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022.

[5] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised nerf: Fewer views and faster training for free," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2022, pp. 12 882–12 891.

[6] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," *Conf. on Neural Information Processing Systems (NeurIPS)*, 2022.

[7] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Trans. Robotics*, vol. 39, no. 1, pp. 309–326, 2023.

[8] J. Behley and C. Stachniss, "Efficient surfel-based slam using 3d laser range data in urban environments," in *Robotics: Science and Systems (RSS)*, 2018.

[9] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.

[10] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems (RSS)*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[11] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.

[12] 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.

[13] Y. Tao, M. Popović, Y. Wang, S. T. Digumarti, N. Chebrolu, and M. Fallon, "3d lidar reconstruction with probabilistic depth completion for robotic navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 5339–5346.

[14] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[15] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2022, pp. 12 932–12 942.

[16] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An atlas framework for scalable mapping," in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1899–1906 vol.2.

[17] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph slam and planning in 3d environments," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 2175–2182.

[18] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, 2020.

[19] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon, "Strategies for large scale elastic and semantic lidar reconstruction," *Journal of Robotics and Autonomous Systems*, vol. 155, p. 104185, 2022.

[20] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-nerf: Scalable large scene neural view synthesis," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2022, pp. 8248–8258.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Conf. on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[22] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[23] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023.

[24] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *Intl. Conf. on Computer Vision (ICCV)*, 2021.

[25] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2022.

[26] 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 Int. Conf. Computer Vision and Pattern Recognition*, 2021.

[27] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2023.

[28] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *Conf. on Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 4805–4815, 2021.

[29] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *Conf. on Neural Information Processing Systems (NeurIPS)*, 2021.

[30] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[31] J. Deng, Q. Wu, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, and L. Pei, "Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[32] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, "Neural volumes: learning dynamic renderable volumes from images," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[33] V. Sitzmann, J. Thies, F. Heide, M. Nießner, G. Wetzstein, and M. Zollhofer, "Deepvoxels: Learning persistent 3d feature embeddings," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2019, pp. 2437–2446.

[34] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," *Conf. on Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[35] N. Max, "Optical models for direct volume rendering," *IEEE Trans. on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[36] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *SIGGRAPH*, vol. 18, no. 3, pp. 165–174, 1984.

[37] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[38] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural rgb-d surface reconstruction," in *IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2022, pp. 6290–6301.

[39] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[40] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 1280–1286.

[41] 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.

[42] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *International Journal of Computer Vision*, pp. 1–16, 2016.

[43] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

Authorized licensed use limited to: Bodleian Libraries of the University of Oxford. Downloaded on August 20,2024 at 21:44:43 UTC from IEEE Xplore. Restrictions apply.

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	SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection
Publication Status	✕Published ☐ Accepted for Publication ☐Submitted for Publication ☐Unpublished and unsubmitted work written in a manuscript style
Publication Details	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 International Conference on Robotics and Automation (ICRA), 2024.

Student Confirmation

Student Name:	Yifu Tao		
Contribution to the Paper	• Contributed to development of the idea • Developed the nerf data pipeline and SiLVR mapping system • Conducted experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature	陶逸夫	Date	22nd January 2025

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: Maurice Fallon			
Supervisor comments Yifu was the lead investigator, implementer and researcher on this project and wrote the bulk of the paper			
Signature	Maurice Fallon	Date	22nd January 2025

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

5.2 Technical Remarks

Sky Segmentation and Training

The sky region is often reconstructed by the NeRF as floating point clouds near the tops of the buildings. These artefacts are always observed in the rendering, but when exported to point clouds, they appear as floating structures in the map. Such artefacts hinder the extraction of the building tops, and distort the reconstruction evaluation metrics if not manually filtered.

To address this, we filtered the sky artefacts automatically using a sky segmentation network, which is inspired by URF [178]. Specifically, we first apply Detectron 2 [1] to the camera image to obtain a sky mask. We then enforce zero density along rays corresponding to the sky by minimising the weights of the point samples. This process is illustrated in Fig. 5.1. This process is illustrated in Fig. 5.1. With this sky loss, the rays directed at the sky acquire a large expected depth and can be effectively filtered.

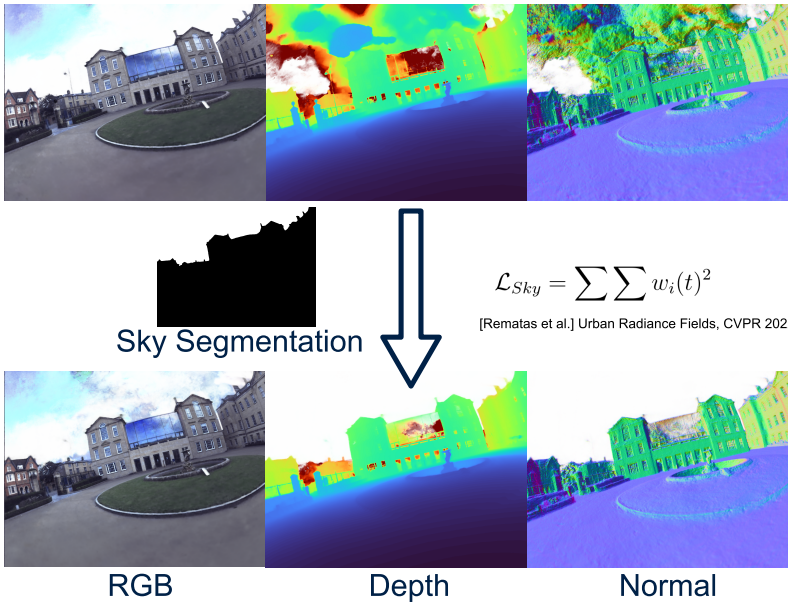


Figure 5.1: We used a segmentation network to obtain an image mask for the sky, which is used to regularise the corresponding rays to be empty and therefore removes the floating sky artefacts.

[1] https://github.com/facebookresearch/detectron2

Image Exposure

The camera image quality plays a crucial role when estimating the poses of the images, which directly affects the final NeRF quality. One crucial aspect is the exposure of the camera image. In Fig. 5.2, we show two challenging cases of images collected in the forest. When the image is under-exposed, the image is dark and can contain thermal noise. When the image is over-exposed, there can be glares in the image which results in loss of object textures.



Figure 5.2: Images that are under-exposed and over-exposed, collected in a forest.

While proactive control of the exposure time has been demonstrated to be able to handle diverse lighting changes [179], the cameras from SevenSense do not support customised control of the exposure time. To capture both dark and bright environments, we used the camera's auto-exposure function when collecting data, which is controlled by its driver to achieve a targetted median intensity [2].

COLMAP is a SfM method which is commonly used by radiance field techniques when working with real-world image collections. The accuracy of the pose estimates produced by COLMAP is affected by poor lighting conditions. In the case of under-exposed images, the SIFT feature extraction used in COLMAP underperforms. In Fig. 5.3, COLMAP is only able to register the raw images on the axis road next to the forest area which has relatively good lighting. The original camera image is under-exposed due to the canopy, and COLMAP failed to register the whole

[2] https://github.com/sevensense-robotics/core_research_manual/blob/master/pages/sensor_settings.md

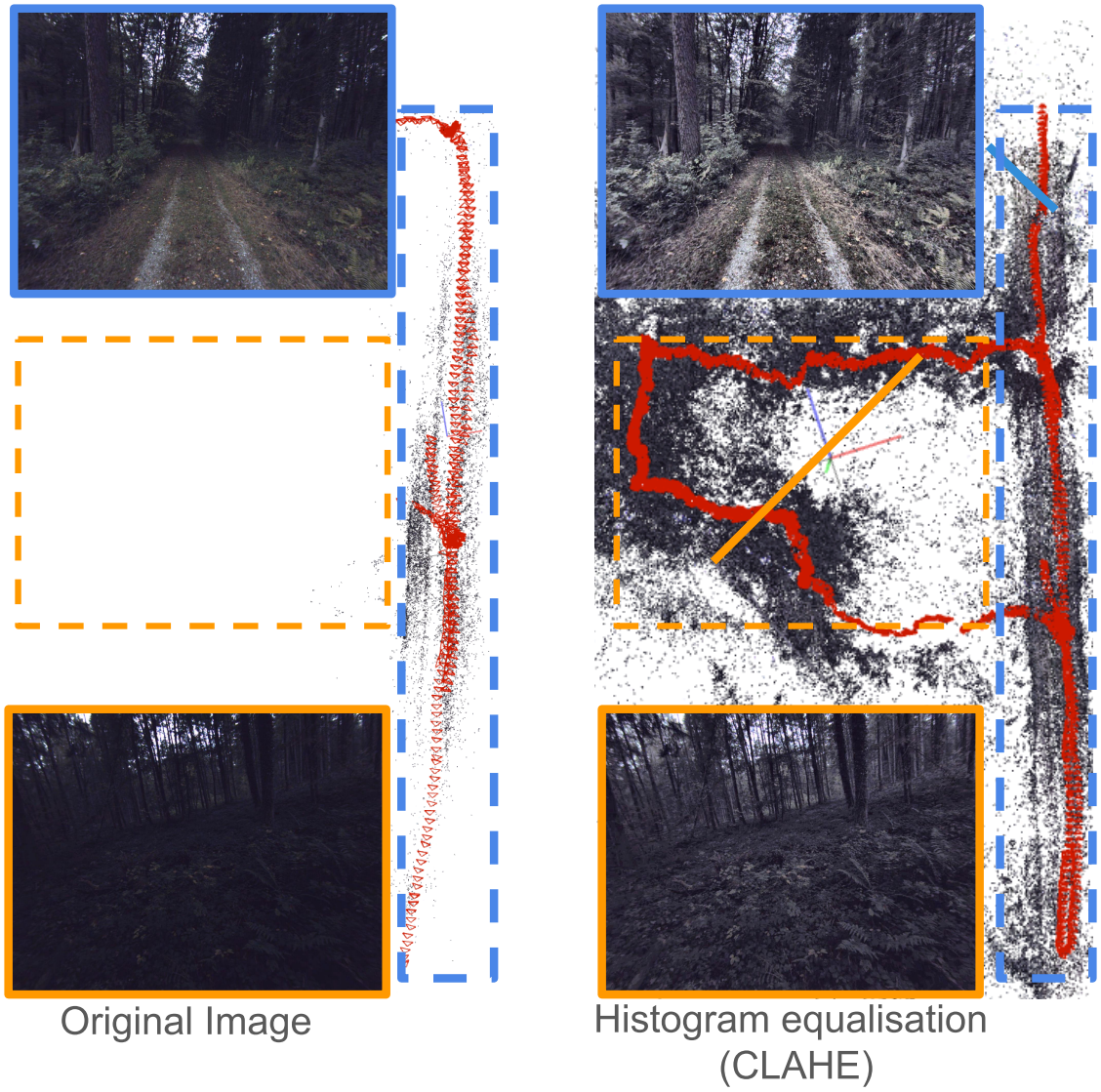


Figure 5.3: COLMAP results of forest images. Results on the left use the original images which are under-exposed under the canopy, and results on the right use images that are processed by histogram equalisation, which is shown to benefit the image registration.

section inside the forest. We found that histogram equalisation can improve the performance of COLMAP in these conditions.

In addition, If the motion is fast or the exposure time is too long (e.g. in a dark room), the image will also contain motion blur. To reduce the effect of motion blur, our data collection procedure requires a slow walking speed.

Importance of the Multi-camera for COLMAP

In addition to issues with lighting and camera exposure, COLMAP is also affected by the degree of overlap present in the views. In Fig. 5.4, we used COLMAP to register two monocular (front) image sequences of a floor inside a building. The final map is incorrectly merged due to a falsely matched image.

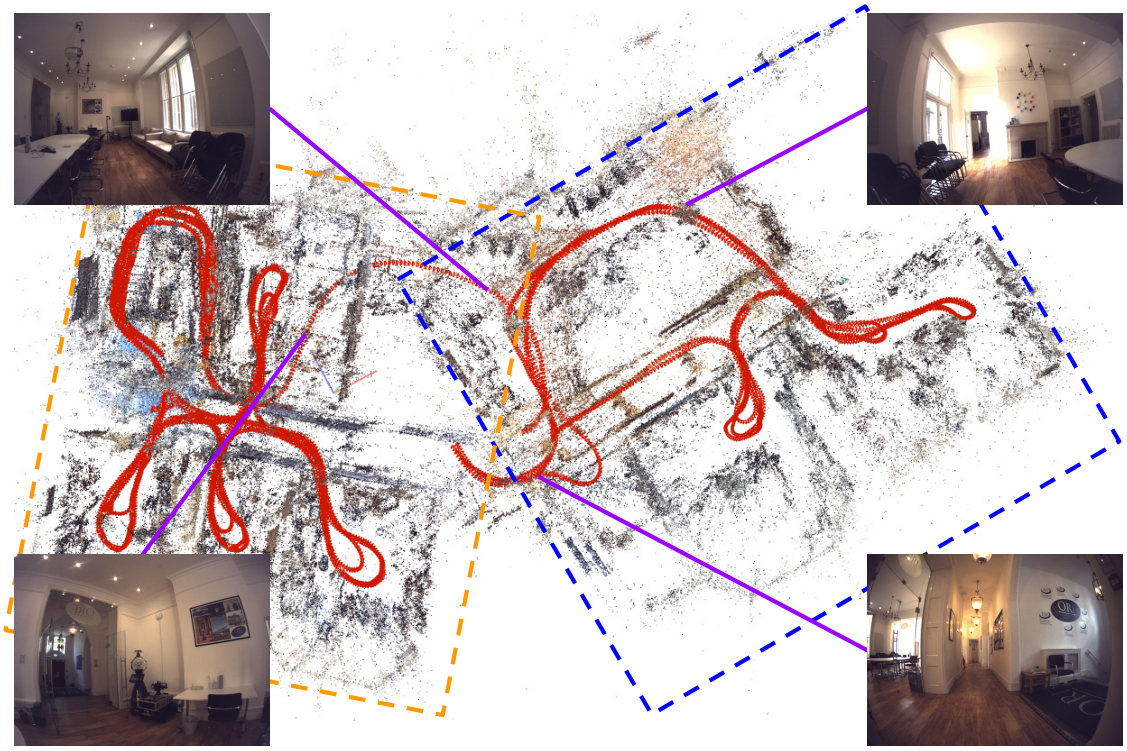


Figure 5.4: COLMAP fails to register two image sequences where only monocular images are used.

Conversely, when we use three cameras, the image collections are successfully registered into one map as shown in Fig. 5.5. We did not include a fourth backwared-facing camera because the device is often handheld, and the operator would obstruct its view.

5.3 Discussion

This chapter presented a method for large-scale lidar-visual NeRF mapping. The method is developed further to incorporate uncertainty quantification in Chap. 6, and tested with a dataset presented in Chap. 7. In this section, we focus on the use of lidar

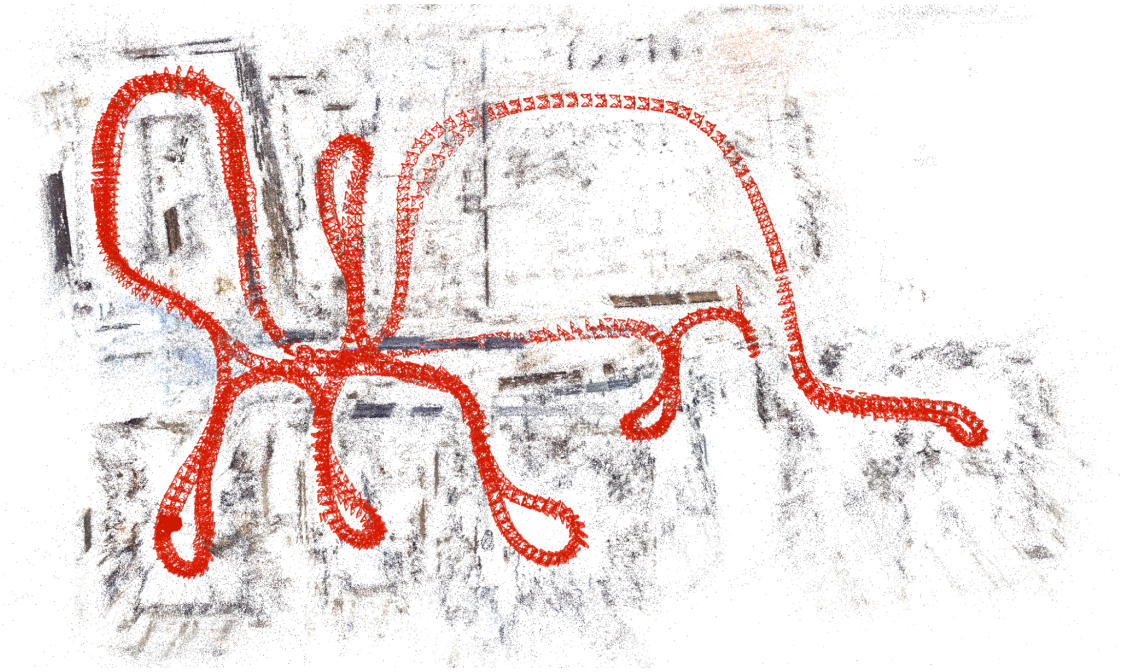


Figure 5.5: The complete COLMAP trajectory when using three camera image sequences.

surface normal in NeRF training, as well as some recent works that can potentially be used to improve the pose estimation pre-processing step for NeRF training.

Surface Normal and Thin Object

The use of surface normal essentially introduces a geometric prior which effectively enforces smoothness constraints to the reconstruction. Specifically, the smoothness prior comes from the fact that we used local plane fitting to estimate the surface normals from the lidar point cloud.

One potential issue with this surface normal regularisation is that it can excessively smooth out geometric detail. For a scene that contains complex geometry such as thin objects, the plane-fitting-based surface normal estimation from a local lidar point cloud is typically less accurate, and this additional prior can be detrimental to the reconstruction. Nevertheless, In our work, we found that thin objects such as bike racks can still be reconstructed. This relies on the photometric loss from the visual data. When training the lidar-visual NeRF, we used relatively small weights for the surface normal loss. This prevents the geometric prior from outweighing visual cues in the camera images for 3D reconstruction,

and only allows the regularisation to affect visually ambiguous areas such as a uniformly textured wall.

The use of a geometric prior is ubiquitous in dense visual SLAM systems. As discussed by the authors of DSO [56], dense or semi-dense approaches exploit the connectedness of the image region to formulate a geometry prior. Without this, estimating dense 3D information is ill-posed from passive vision alone. One unique characteristic of DSO is that it avoids imposing the geometric prior by instead using a sparse map representation, as they argued that such prior can also introduce a bias which reduces long-term accuracy. Indeed, where abundant image observations with good view constraints (sufficient baselines) are available, visual reconstruction can be very accurate. In this case, geometric priors which use heuristics might limit the accuracy of vision-only reconstruction. This is potentially true for the lidar-based geometric prior in the form of the surface normal loss, as the plane-fitting algorithm we used is inaccurate for thin objects like bike racks or lamp posts. It might be interesting to systematically study when purely visual reconstruction can surpass lidar reconstruction whose accuracy can be limited by the inherent sensor noise.

Pose Estimation

In our earlier experiments, we tried to use a SLAM system's output directly for training the NeRF. We found that monocular visual SLAM [55] was typically not robust enough to estimate a trajectory for many of our test sequences; meanwhile, we found that pose estimated by lidar SLAM [105], while robust, was marginally less accurate than COLMAP SfM which resulted in inferior NeRF reconstructions. We preferred using visual Bundle Adjustment to refine an online lidar SLAM trajectory, which is more efficient than a full incremental SfM procedure, but is still computed in post-processing.

Recently, several promising methods for time-efficient and accurate visual pose estimation have been published. One approach is GLOMAP [180], which performs global rather than incremental SfM. This system is shown to achieve pose accuracy comparable to COLMAP [181] when tested in large-scale environments.

Another approach, FAST-LIVO2 [17], combines lidar and visual data to jointly estimate sensor odometry online, and is particularly suitable for robotic applications. FAST-LIVO2 demonstrated that the output trajectory is sufficiently accurate for NeRF reconstruction. Some methods estimate camera poses directly with NeRF training [182], leveraging prior poses [183] or monocular depth priors [184]. In addition, scene coordinate regression [185] offers a more compact map and faster computation than feature-based methods, and has been extended to an SfM setting [186]. These efficient pose estimation methods have the potential to enable more streamlined radiance field reconstruction than using incremental SfM.

6

Epistemic Uncertainty Quantification in Lidar-Visual NeRF

Contents

6.1 SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification 83
6.2 Technical Remarks 101
6.3 Discussion 102

This chapter presents an extension of Chap. 5 as an evolved paper where parts of the text and experiments are reused from the publication [36] in Chap. 5. Building on top of the NeRF system [36], we addressed the problem of epistemic uncertainty quantification in the NeRF reconstruction given multi-modal sensor input. As a dense 3D representation, not all parts of the NeRF are uniformly accurate. Inaccurate 3D reconstructions can be caused by multiple issues, including insufficient texture in the camera images, limited camera views that have sufficient baselines offset, or being outside the Field-of-View or range of the lidar. The NeRF reconstruction (i.e. the radiance field function) is deterministic, and has no explicit notion of uncertainty. Quantifying the uncertainty contribution of each sensor can help to improve the accuracy of the geometry by using uncertainty-based filtering. This can enable downstream tasks such as active mapping and navigation.

We propose a method to estimate the epistemic uncertainty of the NeRF reconstruction. In general, we estimate how much the NeRF can change without affecting the rendering quality in the training data. We approximate the model's posterior distribution as a Gaussian by using the Laplace approximation (LA). The covariance of the Gaussian distribution, which we treat as the model uncertainty, is then estimated by the Fisher information matrix (FIM) of the NeRF map parameters. The log-likelihood in the FIM's Jacobian matrix corresponds to the loss function, which contains a photometric part (from vision) and a geometric part (from lidar). This allows us to disentangle the Jacobian and hence the approximated uncertainty into a visual component and a lidar component. We demonstrated that the part of the NeRF reconstruction with low visual uncertainty estimation is in fact similar to the visual features in SfM that are reliable and used as landmarks. In comparison, the lidar uncertainty can be used to identify regions with few lidar measurements, which are typically outside the lidar's range and Field-of-View. This provides a principled way to combine the heterogeneous sensor modalities for reconstruction with uncertainty estimates that correspond to each sensor's inherent characteristics.

6.1 SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification

The manuscript in this section was submitted to the *IEEE Transactions on Robotics* journal in January 2025. An accompanying video is available online at `https://youtu.be/AU6gvRFfCMQ`.

SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification

Anonymous Submission

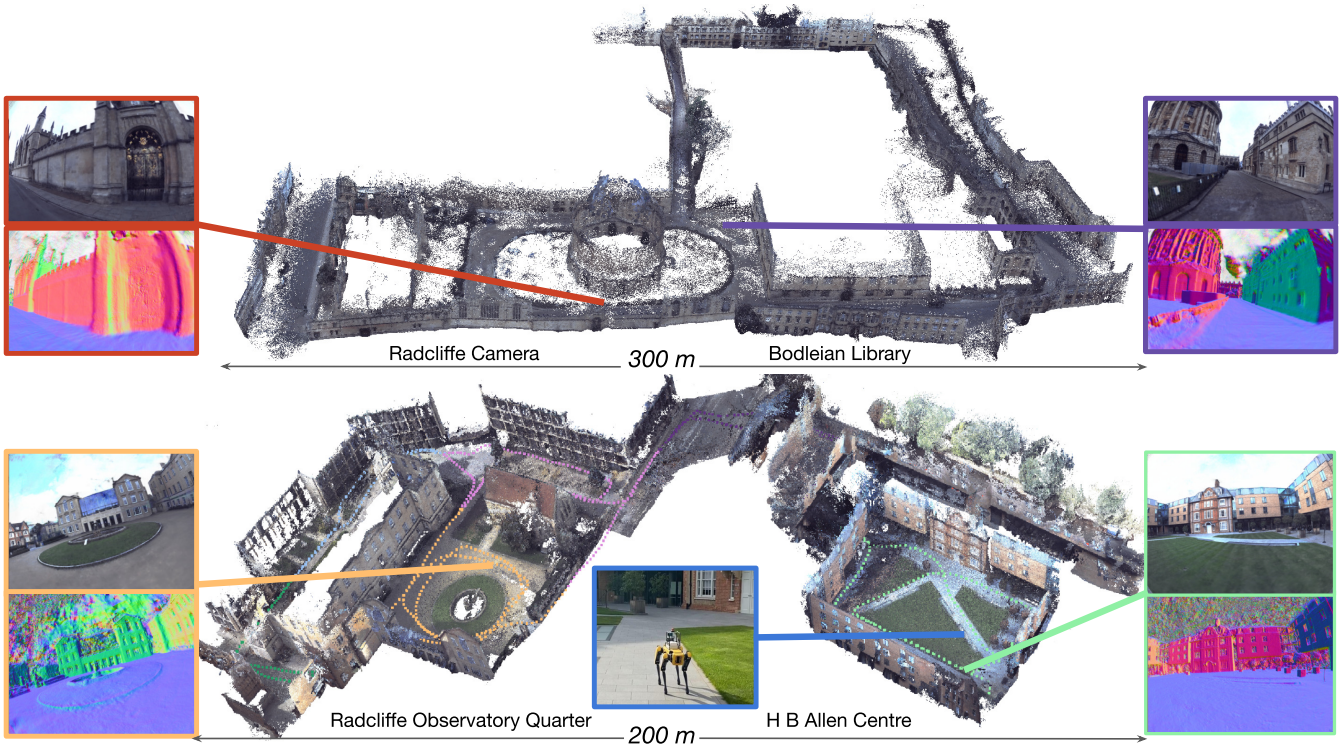


Fig. 1: Two large-scale reconstructions generated by SiLVR. Rendered RGB and surface normal images from the reconstructions are shown on each side. SiLVR combines visual and lidar information to create geometrically accurate maps with photorealistic textures, while considering sensor uncertainty. SiLVR uses submaps to scale to large-scale building complexes.

***Abstract*—We present a neural radiance field (NeRF) based large-scale reconstruction system that fuses lidar and vision data to generate high-quality reconstructions that are geometrically accurate and capture photorealistic texture. Our system adopts the state-of-the-art NeRF representation to additionally incorporate lidar. Adding lidar data adds strong geometric constraints on the depth and surface normals, which is particularly useful when modelling uniform texture surfaces which contain ambiguous visual reconstruction cues. Furthermore, we estimate the epistemic uncertainty of the reconstruction as the spatial variance of each point location in the radiance field given the sensor observations from camera and lidar. This enables the identification of areas that are reliably reconstructed by each sensor modality, allowing the map to be filtered according to the estimated uncertainty. Our system can also exploit the trajectory produced by a real-time pose-graph lidar SLAM system during online mapping to bootstrap a (post-processed) Structure-from-Motion (SfM) reconstruction procedure reducing SfM training time by up to 70%. It also helps to properly constrain the overall metric scale which is essential for the lidar depth loss. The globally-consistent trajectory can then be divided into submaps using Spectral Clustering to group sets of co-visible images together. This submapping approach is more suitable for visual reconstruction than distance-based partitioning. Each submap is filtered according to point-wise uncertainty estimates and merged to obtain the final large-scale 3D reconstruction. We demonstrate the reconstruction system using a multi-camera, lidar sensor suite in experiments involving both robot-mounted and handheld scanning. Our test datasets cover a total area of more than 20,000 m^2, including multiple university buildings and an aerial survey of a multi-storey. Quantitative evaluation is provided by comparing to maps produced by a commercial tripod scanner. The code and dataset will be made open-source.**

***Index Terms*—Mapping, Sensor Fusion, RGB-D Perception, Neural Radiance Field (NeRF), Uncertainty Estimation**

I. Introduction

Dense 3D reconstruction is a core component of a range of robotics applications such as industrial inspection and autonomous navigation. Common sensors used for reconstruction include cameras and lidars. Camera-based reconstruction systems use techniques including Structure-from-Motion (SfM [1]) and Multi-View Stereo (MVS) [2] to produce dense textured reconstructions. However, these approaches rely on good lighting conditions and can require exhaustive data collection to capture data from diverse viewpoints. They also struggle with textureless areas such as bare walls, ceilings and floors. A lidar sensor provides accurate geometric information

at long range—as it actively measures distances to surfaces. This makes lidar desirable for large-scale outdoor environments. However, lidar scans are much sparser than camera images. They also do not capture colour which is important for applications such as virtual reality and 3D asset generation.

Classical reconstruction systems have used point clouds, occupancy maps, and sign-distance fields (SDF) as their internal 3D representation. Recently, radiance field representations, namely neural radiance fields (NeRF) [3] and 3D Gaussian Splatting (3DGS) [4] have gained popularity for visual reconstruction. Taking advantage of differentiable rendering, these techniques optimise a 3D representation by minimising the difference between a rendered image and a reference camera image. These methods can synthesise novel views with near photorealistic quality, which can be useful for robotic inspection and visual localisation.

As with traditional vision-based reconstruction methods, NeRF struggles to estimate accurate geometry in locations where there is limited multi-view input (i.e. when images are only taken from a single direction) or sparse texture. These challenges affect not only volume-density-based fields, but also SDF-based fields [5] and 3D Gaussian representations [4], as can be seen later in Fig. 7. Autonomous systems commonly encounter these situations—for example, a robot reconstructing a wall with uniform colour when travelling directly towards it. In this paper, we show how this problem can be addressed by using lidar sensing to provide accurate geometric measurements of these types of textureless objects. In addition, the use of lidar data can reduce the need to capture varied camera views. It is impractical for an inspection robot (e.g., Spot quadruped on an industrial facility) to execute object-centric trajectories just to improve visual reconstruction. This motivates the development of a reconstruction system that fuses these sensors.

The reconstruction task becomes even more challenging when the scene is large-scale (e.g., urban districts). Running an incremental SfM algorithm such as COLMAP [1] for a large-scale scene is time-consuming, and is not guaranteed to succeed in registering all input images (especially if parts of the scene have poor lighting), which then could lead to an unreconstructed region in the map. As the scene and the dataset size grow, the model capacity of a NeRF and memory constraints become a bottleneck. Simply increasing the size of the model parameters (e.g., hash table size in Instant-NGP [6] or the number of 3D Gaussians [4]) can exceed available computer memory when working on larger scenes. This motivates the development of a scene partitioning strategy. Some existing methods require manually partitioning using heuristics [7] or evenly partitioning the scene using a grid [8]. The limitation of a simple grid-based partition is that the view orientation and visibility are not considered, while it is important to consider these factors while carrying out clustering for visual reconstruction. For example, an image taken from a corridor outside of a room but looking into it ought to be considered part of a submap of that room.

The reconstructed 3D map from a NeRF pipeline can contain artefacts due to unconstrained views, and removing them can be challenging. The implicit representation used in NeRF, while providing tremendous model size compression compared to explicit representations such as 3DGS [4], can generate reconstruction artefacts outside of the observed area. The standard NeRF formulation has no notion of uncertainty for parts of the reconstruction that are unreliable. The regions which can be confidently reconstructed by a camera or a lidar are usually complementary: regions that have uniform texture or lack multi-view constraints often have high *visual uncertainty*, and regions that are outside of the lidar Field-of-View or beyond the lidar's sensing range often have high *lidar uncertainty*. These reconstruction artefacts make it challenging when merging submaps, as each submap will contain noisier reconstruction on its boundary as the measurement density falls off. This can lead to artefacts in the overall merged map. These issues motivate the need for a principled approach to quantifying the reconstruction uncertainty of NeRF maps.

In this work, we present SiLVR, a submap-based NeRF reconstruction system that integrates both lidar and visual information to produce accurate, textured, and uncertainty-aware 3D reconstructions which can also synthesise photorealistic novel views. SiLVR builds upon existing NeRF research and the Nerfacto implementation [9] which utilises hash encoding [6] that is significantly faster than MLP-based NeRF [3] (it takes less than 5 minutes to train a NeRF for one submap in our experiments). We extend this work by adding geometric constraints from lidar to improve reconstruction quality. Our use of lidar data is particularly important for modelling featureless areas where geometry cannot be accurately reconstructed using 3D SfM features [10]. In addition, we also estimate surface normals from the lidar scans to encourage smooth surface reconstruction. This approach does not suffer from input data distribution shift, a characteristic of learning-based normal estimation approaches [11].

Compared to prior NeRF-based reconstruction systems that incorporate lidar [12] or depth cameras [13], [11], our key innovation is a rigorous study of how to quantify uncertainty in the resultant reconstruction, which enables improved reconstruction accuracy as well as facilitating downstream tasks such as active mapping and navigation. After training, SiLVR computes the epistemic uncertainty of the NeRF with the Laplace approximation (LA) [14] and Fisher-Information-approximated Hessian matrix. As an efficient alternative to ensemble learning (which requires multiple iterations of training of the NeRF model and is time-consuming), we build upon the formulation of the perturbation field proposed in BayesRays [15] and use the spatial variance of the field as the measure of epistemic uncertainty. This is used to filter out reconstruction artefacts which improves the final reconstruction accuracy. We also adapt a previously developed lidar SLAM algorithm [16], [17] to bootstrap the SfM component accelerating its operation by up to 70%) and to enforce an accurate metric scale. For mapping large-scale building complexes or a city block, we adopt a submapping approach and apply graph partitioning algorithm [18] with visibility information to divide the complete trajectory into submaps. We study how the use of visibility information allows the submaps to be more self-contained and to have fewer artefacts at their boundaries compared to methods that only consider spatial

information [12].

In summary, our contributions are as follows:

- Epistemic uncertainty quantification of the multi-modal NeRF pipeline which considers lidar and visual uncertainties differently. This allows us to filter parts of the reconstruction that are unreliable by directly using sensor-specific characteristics. To do this, we extended BayesRays [15]'s vision-only uncertainty framework to also support lidar depth. This allows us to identify areas with reliable reconstruction (e.g., where there are visual features or abundant lidar measurements) and unreliable areas (e.g., uniformly-textured surfaces which have also not been scanned by lidar). Filtering based on the uncertainty estimates improves the reconstruction accuracy. This is a more principled approach compared to the surface-density-based filtering [12].
- A submapping strategy that leverages per-image visibility information. Compared to the distance-based clustering method [12], we develop a visibility-based clustering method which reduces visual overlap between submaps and in turn creates fewer artefacts at submap boundaries.
- Large-scale evaluation using two large-scale datasets from the Oxford Spires dataset [19] with quantitative results from millimetre-accurate 3D ground truth. Further comparison is made to baseline radiance field methods that use SDF [5] and 3D Gaussians [4] representations.

II. Related Work

A. Classical 3D Reconstruction

Lidars and cameras are the two main modalities used in robotic perception and specifically for 3D reconstruction. In this section, we review classical lidar-based and vision-based reconstruction methods. For each sensor modality, trajectory estimation is a typical first step in a reconstruction pipeline. Then, we review strategies for extending these methods to large-scale scenes.

Lidar is the dominant sensor used for accurate 3D reconstruction of large-scale environments [20], [21]. It actively transmits laser pulses to measure ranges and as a result is accurate up to ranges of as much 100m. With these distance measurements, Lidar odometry typically uses a variant of Iterative Closest Point (ICP), and often integrates high-frequency IMU measurements [22], [23], [24], [16]. Small errors in odometry can accumulate over time resulting in trajectory drift. This drift can be mitigated when a sensor revisits a previous place and detects loop closure with pose graph optimisation [25] which allows a consistent map to be maintained. After registering all the lidar scans, the (surface) reconstruction problem is then to fuse discrete observations into a map. Example map representations include surfels [26], [27], voxels [28], [29], [30], [31] and wavelets [32]. Despite its advantages, lidar has its own limitations. Lidar is much more expensive than cameras, while the measurements are typically much sparser than camera images. The measurements have inherent noise with ranging errors in the order of centimetres, which makes it difficult to reconstruct small objects accurately in indoor scenes. Finally, lidar data contains no texture or colour, so the final reconstruction is only geometric and cannot be used for applications such as view synthesis, which requires texture.

Alternatively, large-scale textured reconstructions can be recovered from camera images alone via SfM. Given the correspondences between images, a SfM system [1] can optimise a set of camera poses, camera intrinsics, and 3D sparse feature points. This can then be used by a MVS system [33] to compute dense depth for each frame and in turn to create a dense 3D point cloud. Compared to lidar, cameras are much more affordable, and also provide texture and colour. However, the performance of visual reconstruction method depends on environmental conditions, and the quality of feature matching. This makes the resultant reconstruction less reliable in scenes that contain repetitive patterns, low-texture surfaces, dynamic objects, poor lighting conditions and non-Lambertian materials.

When the scene to be reconstructed is large-scale (e.g., urban districts or multi-room indoor environments), computer memory becomes a limiting factor. Attempting to map a large scene while constraining output map size will result in a lower resolution reconstruction, which results in a map that lacks detail. A common strategy to extend dense reconstructions to large-scale areas is to divide the scene into submaps [34]. For visual reconstruction with many thousands of images [35], a submapping approach can significantly reduce computation time and memory requirements. One approach used in large-scale MVS is the submap partitioning [36] which groups images into clusters while not degrading the final resultant reconstruction. After partitioning, each submap should be filtered and merged into one unified model. For online lidar mapping systems, the motivation for using submapping techniques is to accommodate loop closure corrections into the already-built map (Occupancy grid or TSDF). These systems construct submaps that are attached to a pose graph [37], [38], [39], and can deform each submap by reoptimising the pose graph upon loop closure.

B. Radiance Field Representation

Neural Radiance Fields (NeRF) was first proposed in the seminal paper from Mildenhall et al. [3]. The technique uses a multilayer perceptron (MLP) to represent a continuous radiance field, and uses differentiable volume rendering to reconstruct novel views. It minimises the photometric error between the rendered image and the input image, which implicitly achieves multi-view consistency. NeRF and its many variants use frequency encoding [40] to encode spatial coordinates, but these suffer from long training times, typically a few hours per scene. Alternative explicit representations of radiance fields, including dense voxel-grids with trainable per-vertex features [41], [6] and more recently 3D Gaussians [4] are shown to accelerate the training, at the cost of being more memory intensive. Octree or sparse-grid structures [42], [6] can reduce memory usage by pruning grid-features in empty space. Our work is built upon Nerfacto from the open-sourced Nerfstudio project [9]. It incorporates the main features from other NeRF works [6], [43], [44] that have been found to work well with real data.

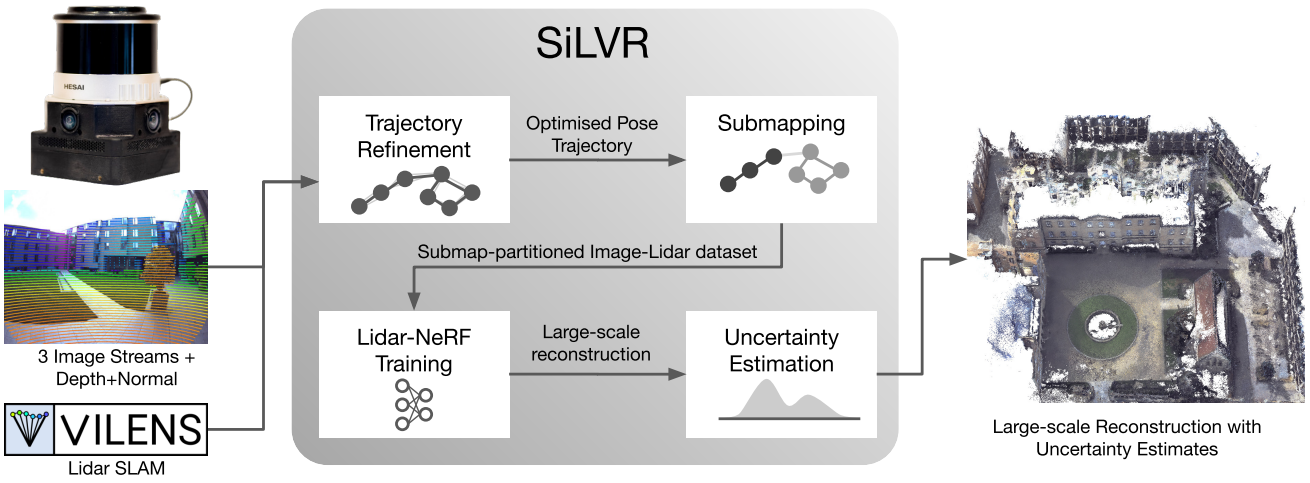


Fig. 2: System overview: SiLVR builds large-scale reconstructions using images and lidar data, and a pose trajectory estimated by a separate odometry system. The sensor streams are provided by the *Frontier*, our custom perception payload carrying three fisheye colour cameras, IMU measurements, and a 3D lidar. When collecting the data, we used VILENS [16] to estimate the trajectory of the sensor, which is refined in post-processing using COLMAP [1] and partitioned into submaps. The camera image, lidar depth, and a derivative normal image are used to train a NeRF to achieve a final 3D reconstruction. After training the NeRF, SiLVR estimates the epistemic uncertainty of the radiance field. Finally, the point cloud reconstruction is extracted from the NeRF by rendering a depth for each of the training rays. The point cloud is then filtered using per-point uncertainty estimates to remove unreliable reconstructions.

While NeRFs excel at high-quality view synthesis, obtaining a 3D surface reconstruction of similar quality remains challenging, mainly due to the flexible volumetric representation being under-constrained by the limited multi-view inputs. One approach to improve the reconstruction is to impose depth regularisation [10], [45] or surface normal regularisation [11] which can be obtained from depth sensors or be estimated by a neural network. Another approach is to impose surface priors on the volumetric field and use representations such as Signed Distance Field (SDF) [46], [47] and 2D Gaussians [48] to enforce a surface reconstruction output, although the novel view synthesis quality might be compromised [5] with this approach. Our method uses a volume density representation which is extended with depth [10] and surface normal [11] regularisations from lidar measurements instead of using SfM [10] or learnt priors [11]. This can significantly improve the reconstruction quality in texture-less areas with smooth surfaces.

Neural field representations have been used for lidar-based mapping [49], [50], [51], showing promise in generating more complete and compact reconstructions than traditional methods. While these works also build upon implicit map representations, they do not use visual data to build the map. Our system uses visual information and multi-view geometry constraints. Because of this, it can reconstruct regions outside of the lidar's field-of view.

C. Uncertainty in Neural Radiance Fields

The standard formulation of NeRF has no notion of uncertainty. The lack of uncertainty makes it difficult to apply them in robotics applications because a NeRF reconstruction could contain artefacts. From a Bayesian machine learning perspective, one could model the data uncertainty (aleatoric uncertainty) and model uncertainty (epistemic uncertainty) [52] in the NeRF model. The data uncertainty models how the image observation differs from the trained NeRF, and the source of errors includes dynamic objects, lighting conditions and non-Lambertian surfaces. Dynamic object masking [53] and appearance encoding [44] have been used to model or mitigate data uncertainty.

The model uncertainty aims to capture the variance of the radiance field given the training data. For example, for a uniformly-textured area (e.g., sky) with parallel viewing angles, there are infinite possible NeRF solutions that can lead to exactly the same image pixel observation. In comparison, the NeRF of a textured object with a clear boundary and observations from multiple viewpoints would have low model uncertainty, and this is similar to the well-conditioned scenario for photogrammetry. The most straightforward and reliable way to quantify model uncertainty is to train an ensemble of models with different initialisations [54]. BayesRays [15] proposes to model the uncertainty of a perturbation field instead, and estimates the uncertainty with the Laplace approximation. We extend BayesRays's perturbation field formulation to also incorporate lidar data, which allows us to obtain uncertainty estimates for both sensor modalities, and filter the results reconstruction considering each sensor's own characteristics.

D. Large-scale Neural Radiance Fields

Submapping has been used in NeRF representations for city-scale reconstruction. There are several partitioning strategies that are based on grid partitioning [8] or using road intersections [7]. Merging NeRF submaps is difficult, since each NeRF

submap's boundary can be ambiguous, and the appearance encoding of each submap can be different [44]. Block-NeRF [7] merges submaps by first selecting submap candidates based on distance and visibility, and combines submaps in the 2D image space with interpolation and test-time appearance matching. These methods either require manual submap partitioning [7], or partition the scene into 2D grids [8]. Our work adopts the submapping approach, and develops partitioning strategies based on visibility, which is advantageous compared to 2D grid partitioning of image data that are close in Euclidean distance but in fact belong to isolated regions (e.g., rooms). We also develop novel strategies for submap merging which uses epistemic uncertainty estimation.

III. Preliminaries

A. Radiance Field Formulation

We start by adopting the radiance field representation and the differentiable volume rendering framework originally proposed by Mildenhall et. al [3]. The radiance field models the scene as a function $R : (\mathbf{p}, \mathbf{d}) \mapsto (\mathbf{c}, \sigma)$ where the input includes a 3D location $\mathbf{p} = (p_x, p_y, p_z)$ and 2D viewing direction $\mathbf{d} = (\phi, \psi)$, the output is an emitted colour $\mathbf{c} = (r, g, b)$ and a volume density σ. To render a novel view using a NeRF from a particular viewpoint, we cast rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ from the camera origin $\mathbf{o}$ along the viewing direction $\mathbf{d}$ of each pixel $\mathbf{u}$ in the image plane, and render the pixel-colour by integrating over the set of points sampled along the ray. The pixel colour $\hat{\mathbf{C}}(\mathbf{r})$ is computed by the volume rendering integral which is approximated using the quadrature rule [55], [56] as

$$\hat{\mathbf{C}}(\mathbf{r}) = \sum_{i=0}^{N} w_i \mathbf{c}_i, \tag{1}$$

where $\mathbf{c}_i$ is the colour of the i-th point sample along the ray, and its weighting coefficient w_i is defined by

$$w_i = \exp\left(-\sum_{j=1}^{i-1} \delta_j \sigma_j\right)(1 - \exp\left(-\delta_i \sigma_i\right)). \tag{2}$$

where δ_i is the distance between adjacent samples.

The radiance field can be trained with a squared photometric loss given the ground truth colour $\mathbf{C}(\mathbf{r})$ from the input image:

$$\mathcal{L}_{\text{Colour}} = \sum_{\mathbf{r} \in \mathcal{D}} ||\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})||^2 \tag{3}$$

where $\mathcal{D}$ is the whole training dataset used to generate the rays $\mathbf{r}$ and ground truth colour $\mathbf{C}(\mathbf{r})$.

Our system extends upon Nerfacto, which is a specific pipeline implemented within the Nerfstudio framework [9]. Nerfacto's rendering quality is comparable to state-of-the-art methods such as MipNeRF-360 [43] while achieving a substantial acceleration in reconstruction speed as it also incorporates efficient hash encoding which was proposed by the authors of Instant-NGP [6]. The scene contraction, proposed in [43], is also used to improve memory efficiency and to represent scenes with high-resolution content near the input camera locations. The contraction function non-linearly maps any point in space into a cube of side length 2, and represents the scene within this contracted space. In real-world outdoor environments, there is often large variation in exposure and lighting conditions. Because of this we use a per-frame appearance encoding for each image, similar to [45], [44].

B. Bayesian Interpretation of the NeRF training

The NeRF reconstruction i.e. the radiance field function f is deterministic, and has no explicit notion of uncertainty. In practice, different parts of the NeRF reconstruction are inherently more uncertain or less reliable. For example, ill-conditioned visual constraints from non-textured areas or insufficient visual parallax can cause the visual reconstruction accuracy to deteriorate. Another example specifically relevant to our work is that a surface is more uncertain if it has only been sparsely scanned by the lidar with limited range and Field-of-View compared to a surface that is densely scanned. Quantifying these uncertainties associated with the NeRF reconstruction allows one to identify the unreliable reconstruction and filter them accordingly, hence improving reconstruction accuracy. It can also enable downstream tasks such as active mapping and navigation [57].

The Bayesian probability theory provides useful tools for quantifying the uncertainties in neural models [52], which can benefit the NeRF reconstruction. For a regression task with the dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}$ (where x_n, y_n are the n-th pair of the input and output) and model parameters θ, the uncertainty of the predictive distribution $p(y|x, \mathcal{D})$ can be approximated by considering the data (aleatoric) uncertainty in the likelihood $p(y|\theta, x)$ and model (epistemic) uncertainty in the posterior $p(\theta|\mathcal{D})$.

We first describe the NeRF training from a Bayesian perspective. When training the NeRF, we seek to minimise the total training loss $\mathcal{L}(\mathcal{D}; \theta)$ with respect to the NeRF parameters θ (e.g., using the photometric loss from Eq. (3) if only vision is provided). This is equivalent to computing the maximum a-posteriori (MAP) estimate $\hat{\theta}$

$$\begin{aligned}
\hat{\theta} &= \arg\max_{\theta} p(\theta \mid \mathcal{D}) \\
&= \arg\max_{\theta} \big[\log p(\mathcal{D} \mid \theta) + \log p(\theta)\big] \\
&= \arg\min_{\theta} \big[-\log p(\mathcal{D} \mid \theta) - \log p(\theta)\big] \\
&= \arg\min_{\theta} \big[\sum_{n=1}^{N} \ell(x_n, y_n; \theta) + r(\theta)\big] \\
&= \arg\min_{\theta} \mathcal{L}(\mathcal{D}; \theta)
\end{aligned} \tag{4}$$

where $\ell(x_n, y_n; \theta) = -\log p(y_n | f_\theta(x_n))$ is the loss term that corresponds to the negative log-likelihood for each data sample, and $r(\theta) = -\log p(\theta)$ is the weight regularisation that corresponds to the log-prior.

It can be seen from Eq. (4) that the total training loss can be interpreted as $\mathcal{L}(\mathcal{D}; \theta) = -\log p(\mathcal{D}|\theta) - \log p(\theta)$. The exponential of the negative training loss then corresponds to the unnormalised posterior $p(\mathcal{D}|\theta)p(\theta)$:

$$p(\theta \mid \mathcal{D}) = \frac{1}{Z} p(\mathcal{D} \mid \theta) p(\theta) = \frac{1}{Z} \exp\big(-\mathcal{L}(\mathcal{D}; \theta)\big) \tag{5}$$

where the normalising constant Z is the normalising constant, and is defined as:

$$Z = \int p(\mathcal{D} \mid \theta) p(\theta)\, d\theta \tag{6}$$

Here, the posterior $p(\theta|\mathcal{D})$ is used for the uncertainty estimation described later in Sec. IV-C

IV. Method

A. Geometric Constraints from Lidar Measurements

Image-based 3D reconstruction with NeRF becomes challenging when a surface has uniform texture and limited multi-view constraints. Lidar measurements are complementary as they can provide accurate depth measurements in such scenarios. In our work, we incorporate the lidar measurements directly into the NeRF optimization. Specifically, we project the lidar point cloud from VILENS-SLAM's pose graph onto the image plane using the camera intrinsics and lidar-camera extrinsics (described in Sec. V-B) to form a depth image. We denote the lidar depth images as $\mathcal{D}_d$. Each frame of the lidar point cloud is motion-undistorted to the nearest image's timestamp, and hence synchronised with the image data. Example overlays can be found in Fig. 4.

1) Lidar Depth Constraints: We follow the depth regularisation approach proposed by DS-NERF [10] for RGB-D cameras, which minimises the Kullback-Leibler (KL) divergence between a (narrow) normal distribution around the lidar depth-measurement $\mathbf{D}$ and the rendered ray distribution $h(t)$ from the NeRF model:

$$\mathcal{L}_{\text{Depth-KL}} = \sum_{\mathbf{r} \in \mathcal{D}_d} \text{KL}[\mathcal{N}(\mathbf{D}, \hat{\sigma}) \| h(t)] \tag{7}$$

During training, we adopt a coarse-to-fine approach by gradually reducing the covariance of the target Normal distribution. Then, the effect of this loss function is that the surface is encouraged to be close to a delta function.

2) Surface Normal Constraints from Lidar Measurements: While the depth loss improves 3D reconstruction, we found that the surface it produces will still contain wavy artefacts in regions where it is expected to be smooth (see Fig. 3). To mitigate this, we regularise the surface normal of the NeRF with lidar data. For each point $\mathbf{p}$ in the radiance field R, its surface normal can be computed as the negative gradient of the volume density σ. To obtain the training labels for the surface normal, we use the lidar range image and estimate the surface normals by local plane fitting. The surface normals are projected onto the image plane to generate the surface normal images, denoted as $\mathcal{D}_n$ similar to the lidar depth images. Then, we introduce an additional surface normal regularisation loss in the NeRF training, inspired by [11]:

$$\mathcal{L}_{\text{Normal}} = \sum_{\mathbf{r} \in \mathcal{D}_n} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \left\| 1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r}) \right\|_1 \tag{8}$$

Compared to learning-based surface normal estimation from the camera image used in [11], our surface normal is estimated from the 3D lidar point cloud and does not suffer generalisation issues. The effect of the surface normal regularisation can be seen in Fig. 3.

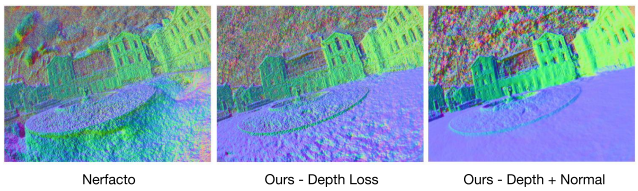


Fig. 3: Comparison of surface normal renderings of the Maths Institute. Incorporating normal constraints in addition to depth from lidar improves the smoothness of the reconstruction. Right: The smooth reconstruction of the ground portion highlights this improvement.

B. Sky Segmentation

Since we focus on large-scale reconstructions of outdoor spaces, the sky and clouds are often present in our training image data. Vision-only NeRF reconstruction typically tries to model it as unconstrained floating white or blue points, which become artefacts in the final reconstruction. To remove these "sky points" from the training procedure, we used a semantic segmentation network [1] to obtain a sky mask that is used to regularise the corresponding camera rays to be empty. Specifically, for the rays $\mathbf{r}$ that correspond to the sky mask (denoted as $\mathcal{D}_s$), we minimise the weights of all samples on these rays, similar to [45]:

$$\mathcal{L}_{\text{Sky}} = \sum_{\mathbf{r} \in \mathcal{D}_s} \sum_i w_i^2 \tag{9}$$

C. Epistemic Uncertainty of the NeRF reconstruction

We aim to obtain a explicit metric of uncertainty of our NeRF reconstruction. Particularly, following Sec. III-B, we aim to quantify the epistemic uncertainty that is directly related to the covariance of the posterior distribution $p(\theta|\mathcal{D})$ using the Laplace approximation. We first describe the reformulation of the parameters θ, and then we derive the epistemic uncertainty estimate using the approximation.

1) Perturbation Field Reformulation: While a NeRF representation presents convenient advantages for scene compression, its parameters θ do not directly correspond to the 3D scenes. A change of one parameter in the MLP could change the whole radiance field, and it is difficult to obtain uncertainty for a specific 3D location. This is in contrast to other approaches such as 3D Gaussian Splatting [4], where the parameters have a direct representation in the world. This introduces additional challenges when estimating the uncertainty of the parameters.

To obtain the spatial uncertainty of a NeRF, we adopt the perturbation field formulation introduced in BayesRays [15]. Specifically, we construct the perturbation field $\mathcal{P}$ of every 3D coordinate $\mathbf{x}$. We define the perturbation field as $\mathcal{P}_{\theta_N}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^3$, where θ_P denotes the parameters in the perturbation field. A 3D coordinate's perturbation can be obtained using trilinear interpolation:

$$\mathcal{P}_{\theta_N}(\mathbf{x}) = \text{Trilinear}(\mathbf{x}, \theta_N) \tag{10}$$

[1]We used Detectron2 from https://github.com/facebookresearch/detectron2

The introduction of the perturbation field enables us to obtain uncertainties of a specific location in the 3D space, which is crucial for our application. From the Bayesian formulation of our problem, this means that the parameter θ in the posterior $p(\theta|\mathcal{D})$ (whose covariance we estimate as our model uncertainty) is not the NeRF parameters θ_N (MLP weights and the hash grids), but the perturbation field θ_P (perturbation value stored in the grid vertices.

2) Epistemic Uncertainty Estimation using Laplace Approximation: We estimate the epistemic uncertainty of the NeRF reconstruction by estimating the covariance of the posterior $p(\theta|\mathcal{D})$. Using Laplace approximation, we estimate the otherwise intractable posterior distribution as a Gaussian function, and then use its covariance as the estimated uncertainty of our reconstruction. Our derivation follows the presentation by Daxberger et al. [14].

First, we take a second-order Taylor Series expansion of the loss function at the MAP estimate $\hat{\theta}$ as follows:

$$\mathcal{L}(\mathcal{D};\theta) \approx \mathcal{L}(\mathcal{D};\hat{\theta}) + \frac{1}{2}(\theta-\hat{\theta})^\top \mathbf{H}(\theta-\hat{\theta}), \tag{11}$$

where $\mathbf{H} = \nabla^2_\theta \mathcal{L}(\mathcal{D};\theta)|_{\hat{\theta}}$ is the Hessian matrix at the MAP estimate $\hat{\theta}$. Here, the first-order term $\nabla_\theta \mathcal{L}(\mathcal{D};\theta)|^\top_{\hat{\theta}}(\theta-\hat{\theta})$ does not appear as it is zero at the MAP estimate.

Substituting the approximation in Eq. (11) into Eq. (6), we obtain:

$$\begin{aligned} Z &= \int p(\mathcal{D} \mid \theta) p(\theta)\, d\theta \\ &= \int \exp(-\mathcal{L}(\mathcal{D};\theta)) d\theta \\ &\approx \exp(-\mathcal{L}(\mathcal{D};\hat{\theta})) \int \exp\left(-\frac{1}{2}(\theta-\hat{\theta})^\top \mathbf{H}(\theta-\hat{\theta})\right) d\theta \\ &= \exp(-\mathcal{L}(\mathcal{D};\hat{\theta})) \frac{(2\pi)^{\frac{D}{2}}}{(\det \mathbf{H})^{\frac{1}{2}}} \end{aligned} \tag{12}$$

where D denotes the dimensionality of the parameters θ.

Then, we substitute the Taylor expansion Eq. (11) and expression of the normalization constant Eq. (12) into the posterior Eq. (5):

$$p(\theta|\mathcal{D}) \approx \frac{(\det \mathbf{H})^{\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \exp\left(-\frac{1}{2}(\theta-\hat{\theta})^\top \mathbf{H}(\theta-\hat{\theta})\right) \tag{13}$$

which turns out to be a Gaussian distribution $\mathcal{N}(\theta;\hat{\theta},\mathbf{\Sigma})$ with mean $\hat{\theta}$ and covariance $\mathbf{\Sigma} = \mathbf{H}^{-1}$.

By now, we have shown that the posterior can be approximated as a Gaussian distribution. The mean of the Gaussian $\hat{\theta}$ is the MAP estimate of the parameters θ. Here, the parameters θ that we are estimating are the perturbation field θ_P introduced in Sec. IV-C1. Assuming that the NeRF reconstruction has converged to local minima after training, small perturbation should not make the reconstruction, and hence the MAP estimate of the perturbation field is $\mathbf{0}$ (no perturbation). Then, the major computation needed is to determine the covariance, or the inverse of the Hessian. If we assume the prior is a zero-mean Gaussian $p(\theta) = \mathcal{N}(\theta;\mathbf{0},\gamma^2\mathbf{I})$, then the full Hessian at the location of the MAP estimate is

$$\begin{aligned} \mathbf{H} &= \nabla^2_\theta \mathcal{L}(\mathcal{D};\theta)|_{\hat{\theta}} \\ &= -\gamma^{-2}\mathbf{I} - \sum_{n=1}^{N} \nabla^2_\theta \log p(y_n|f_\theta(x_n))|_{\hat{\theta}} \end{aligned} \tag{14}$$

While the second term in the Hessian from Eq. (14) is usually intractable, it can be approximated by the Fisher information matrix [58]:

$$\mathbf{H} \approx -\gamma^{-2}\mathbf{I} - \sum_{n=1}^{N} \mathbf{J}\mathbf{J}^\top \tag{15}$$

where $\mathbf{J} = \nabla_\theta \log p(y_n|f_\theta(x_n))|_{\hat{\theta}} = -\nabla_\theta \ell(x_n, y_n;\theta)$ is the Jacobian matrix of the NeRF model.

Since the Fisher information matrix is still expensive to compute, we can make a further assumption that each parameter (perturbation field) is independent of each other, and use a diagonal approximation to the Hessian:

$$\mathbf{H} \approx -\gamma^{-2}\mathbf{I} - \text{diag}(\mathbf{J}\mathbf{J}^\top) \tag{16}$$

This means that for the Hessian matrix, its diagonal elements H_{ii} can be computed as

$$H_{ii} \approx \sum_{j=1}^{n} \left(\frac{\partial \ell_j}{\partial \theta_i}\right)^2 + \gamma^{-2} \tag{17}$$

The Hessian matrix is initialised as all zeroes, i.e. infinite covariance. As a result, the parameters that are not involved in the outputs (the rendered pixels or depth) will have very high epistemic uncertainty because changing them will not change the outputs and the training loss. Since the outputs are rendered according to the rays from the training images, the unobserved regions can be identified as having very high uncertainty and can be filtered out. Identifying unobserved areas (similar to the unknown space in occupancy mapping) is particularly important for NeRF, as the underlying MLP can output arbitrary colour and density in some locations, leading to "hallucinated" reconstruction points.

3) Visual and Depth Uncertainty: The NeRF model is trained with a total training loss function $\mathcal{L}(\mathcal{D};\hat{\theta})$ that contains the photometric loss $\mathcal{L}_{Colour}$ and the depth loss $\mathcal{L}_{Depth}$. This means that the Jacobian $\mathbf{J}$ can be decoupled into a colour component $\mathbf{J}_{Colour}$ and depth component $\mathbf{J}_{Depth}$, from which we can then approximate the Hessian that corresponds only to the visual information $\mathbf{H}_{Colour}$, and the Hessian that corresponds to only to the lidar depth information $\mathbf{H}_{Depth}$. Therefore, we can compute the epistemic uncertainty for *each observation modality* by changing the loss function. Note that the total training loss function contains other terms including the surface normal loss $\mathcal{L}_{Normal}$. In our study, we focus on just the photometric loss $\mathcal{L}_{Colour}$ and depth loss $\mathcal{L}_{Depth}$. This is because these two losses are the dominant components of the total loss $\mathcal{L}(\mathcal{D};\hat{\theta})$ (with our weighting coefficients for each loss chosen).

The nature of the different sensor modalities leads to different uncertainty characteristics. The visual uncertainty captures areas that can be geometrically perturbed while not changing

the colour. As later shown in Fig. 8, "low" visual uncertainty corresponds to distinct visual features and high-frequency areas. "Higher" visual uncertainty typically corresponds to areas with uniform texture, since perturbing a point in an area with similar colours can lead to small changes in the rendered colour. Here, the characteristics of the visual uncertainties are similar to those in SfM where visual features are used as landmarks for Bundle Adjustment whereas uniform texture areas are often not mapped.

In comparison, low lidar depth uncertainty often appears in regions with abundant lidar observation—whether there are visual features or not. This means a road surface with uniform texture can have *lower* lidar depth uncertainty but *higher* visual uncertainty. Low lidar depth uncertainty is also observed at object boundaries, since perturbing that point can lead to a drastic change in the depth (from foreground depth to background depth). Regions with high lidar depth uncertainty are often areas where there are fewer lidar observations, such as the sky, distant regions that are beyond the lidar's sensing range, and regions outside the lidar's Field-of-View.

The decoupling of the uncertainties enables a principled interpretation of the lidar-visual reconstruction. We can identify parts of the reconstructions that are reliable (surface with visual features, and/or abundant lidar observations) and unreliable (no lidar observation and uniform-textured surfaces), given each sensor modality's own characteristics.

D. Large-scale Pose Trajectory Estimation

1) Refining Lidar SLAM trajectory with Bundle Adjustment: Providing the NeRF reconstruction method with accurate camera poses is crucial as their accuracy directly impacts the fidelity of the reconstructed model. A popular approach used in most NeRF works is to estimate camera poses using (offline) Structure-from-Motion methods such as COLMAP [1]. However, we observed the following limitations when testing COLMAP: (1) long computation times, especially for large image collections collected spanning a long trajectory (e.g., 3000 images can take more than 1 hour (as shown in Tab. IV), and (2) inability to register all frames into one global map when there is limited visual overlap between the images. These issues undermine the goal of building complete, large-scale, globally consistent maps.

In our work, we use our lidar-inertial odometry and SLAM system VILENS [16]. While VILENS achieves state-of-the-art results for lidar-based online motion tracking, we found that the camera poses obtained are less precise than those of COLMAP. This results in *blurring* artefacts in the images rendered by the NeRF model. Several works [7], [13] use noisy pose inputs and then jointly refine the poses within the NeRF optimization to generate better results. However, as shown later in Tab. IV, our experiments showed that this pose-refinement approach can produce results which are inferior to using poses estimated by COLMAP.

To overcome these limitations, we propose to use the pose trajectory from VILENS SLAM as a prior and refine it by running Bundle Adjustment using COLMAP [1]. Specifically, we first run the feature extraction and matching on the dataset, and then use the Lidar SLAM poses to triangulate visual feature points, and run a few iterations of BA. This method is faster than a typical incremental SfM, as it reduces the incremental Structure-from-Motion to a Bundle Adjustment problem. More importantly, having an accurate prior for all the image poses means that COLMAP will be able to produce a full solution and does not fail to register some of the images—as would be the case for SfM without initialisation. For a mission spanning over 20 minutes, our COLMAP-with-prior pipeline achieves similar rendering quality, while typically taking half the computation time of a standard COLMAP run. The computation time is similar to the time required by a robot to collect the data, making it more suitable for robotic applications.

After COLMAP computation, we rescale the trajectory using $Sim(3)$ Umeyama alignment to the Lidar-SLAM trajectory, so that the final trajectory is metrically scaled. This step is crucial because the lidar measurements used in Sec. IV-A1 are also metric. The depth regularisation cannot be used if the scale of the scene and the scale of the depth are not consistent.

2) Submapping of Pose Trajectory: To divide the whole map into smaller manageable areas, we partition the entire trajectory into shorter trajectories, which we define as submaps. The submaps are clustered considering image visibility rather than using space partitioning or distance-based clustering [12]. The goal is to exclude an image from a submap if it does not contribute to the submap reconstruction—for example, if the scene observed is not visible from the other images in the submap.

We formulate the clustering problem as a graph partitioning problem, where each node is an image, and the edges between the nodes are weighted by a similarity score. We measure the similarity between two images using a co-visibility metric. Two images are co-visible if some feature points on one image can be viewed from the other image. Specifically, the co-visibility metric for an image pair is computed as the number of visual feature points computed by COLMAP that appear in both images. After constructing the graph, we use the Normalised Cuts algorithm [18] to obtain a partitioning which minimally breaks edges, i.e. to remove the link between unrelated nodes. In practice, this means that images that are co-visible are grouped together, and image pairs that have less visual overlap are identified as the submap boundary.

Once we divide the full map into submaps as a set of clustered images, we independently train each NeRF submap. After training, we compute the epistemic uncertainty of the reconstruction. We can export a point cloud by rendering colour and depth using the training data rays, and we filter points that have high uncertainty.

Site Name	Robotic Platform	GT
HB Allen Centre	BD Spot	Leica BLK360
Fire Service College	DJI M600 Drone	Leica BLK360
Radcliffe Observatory Quarter	Handheld Frontier	Leica RTC360
Bodleian Library	Handheld Frontier	Leica RTC360

TABLE I: Details of real-world datasets that are used for evaluation.

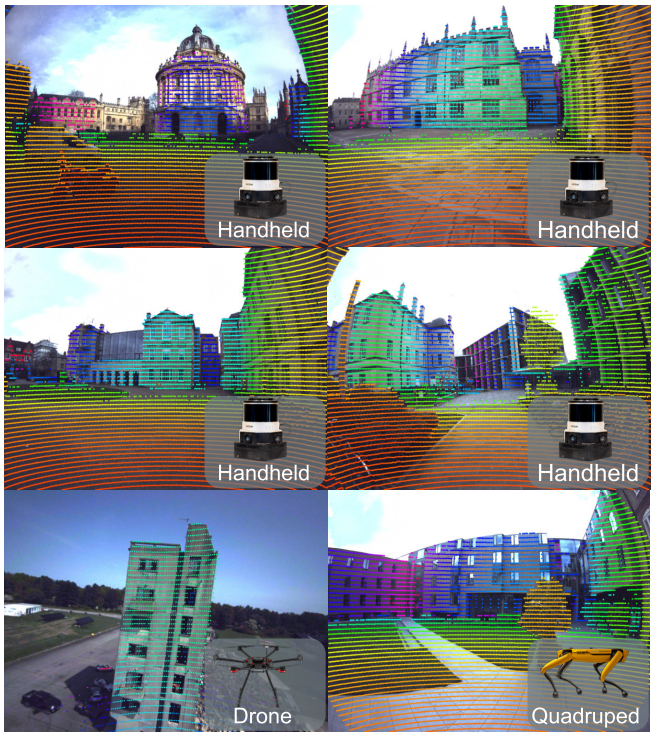


Fig. 4: Sample Data from our diverse robotic datasets. Here each image is overlaid with projected lidar point cloud to demonstrate the accuracy of the sensor calibration.

V. Experimental Setup

A. Hardware and Datasets

We evaluate our methods using a custom perception unit called Frontier shown in Fig. 2. It includes three 1.6 MP fish-eye colour[2] Alphasense cameras (from Sevensense Robotics AG) on 3 sides of the device, as well as a synchronised IMU. The 3-camera setup enables omnidirectional NeRF mapping from a single walking pass through a test site. We installed a Hesai QT64 lidar (104° Field-of-View, 60 metres maximum range) on top of the device. We deployed the Frontier in different modes—onboard a legged robot (Boston Dynamics Spot), a drone (DJI M600, with the system described in [59]), or simply handheld (using the Oxford Spires dataset [19]. Some data is collected with the Frontier mounted on a human operator's backpack).

In addition, we used a survey-grade terrestrial lidar scanner (TLS) to obtain a millimetre-accurate point cloud which we later used to create a reference ground truth model. We use either the entry-level Leica BLK360 or the professional-grade Leica RTC360 to obtain ground truth maps of the sites.

We tested our method using data collected in the following sites: H B Allen Centre (HBAC), Fire Service College (FSC), Radcliffe Observatory Quarter (ROQ), and the Bodleian Library, all in Oxford, UK. The large-scale sites, ROQ and the Bodleian Library, cover areas of 5,000 m^2 and 15,000 m^2, respectively. The hardware details are in Tab. I, and some sample lidar-camera overlays are shown in Fig. 4.

[2]To produce RGB images, we debayer and white-balance the raw bayered images using https://github.com/leggedrobotics/raw_image_pipeline

TABLE II: Evaluation of 3D Reconstruction Quality of Small Scenes

Method	Accuracy↓ (m)	Completeness↓ (m)	PSNR↑ train	PSNR↑ test	SSIM↑ test
Oxford HBAC					
VILENS-SLAM	**0.05**	0.25	/	/	/
Nerfacto mono	0.49	5.40	**32.6**	19.5	0.65
Nerfacto 3-cam	0.28	0.40	29.8	20.6	**0.74**
Ours mono	0.30	4.60	31.0	**21.2**	**0.74**
Ours 3-cam	0.09	**0.18**	28.8	19.7	**0.74**
FSC					
VILENS-SLAM	**0.08**	**0.08**	/	/	/
Nerfacto mono	0.14	0.11	**28.8**	**19.1**	**0.76**
Ours mono	0.11	0.09	27.7	**19.1**	0.75

B. Implementation Details

When collecting the data, we use VILENS [16], a lidar-inertial SLAM system running online to estimate a globally consistent trajectory and to motion correct the lidar measurements. The SLAM trajectory estimated online can also be further optimised using Bundle Adjustment as described in Sec. IV-D1. This improves the visual reconstruction quality, as shown later in Tab. IV. Individual lidar scans are projected to form a sparse depth image coinciding with the camera image (i.e. the same camera pose and intrinsic parameters). Lidar point normals are also estimated using the lidar range image, and projected as a sparse normal image. We use the calibrations provided by the Oxford Spires dataset [19]. In this dataset, the intrinsics and extrinsics of the set of cameras are estimated using Kalibr [60], and a single extrinsic transformation between the three cameras and the lidar is estimated using DiffCal [61]. When running COLMAP, we further optimise the camera intrinsics produced by Kalibr. To train the NeRF model, we used an Nvidia RTX 3080 Ti. One training iteration takes 4096 rays.

C. Evaluation Details

1) 3D Reconstruction Metrics: To evaluate the geometry of the reconstruction, we report *Accuracy* and *Completeness* following the conventions of the DTU dataset [62]. Accuracy is measured as the point-to-point distance from the reconstruction to the (ground truth) reference 3D model and indicates the reconstruction quality. Completeness is the distance from the point-wise reference to the reconstruction and shows how much of the surface has been captured by the reconstruction.

In addition, we also compute *Precision* and *Recall* with a pre-defined error threshold. A point in the reconstruction which is below this threshold can be considered to be a true positive. We use both 5cm and 10cm for the threshold following [19].

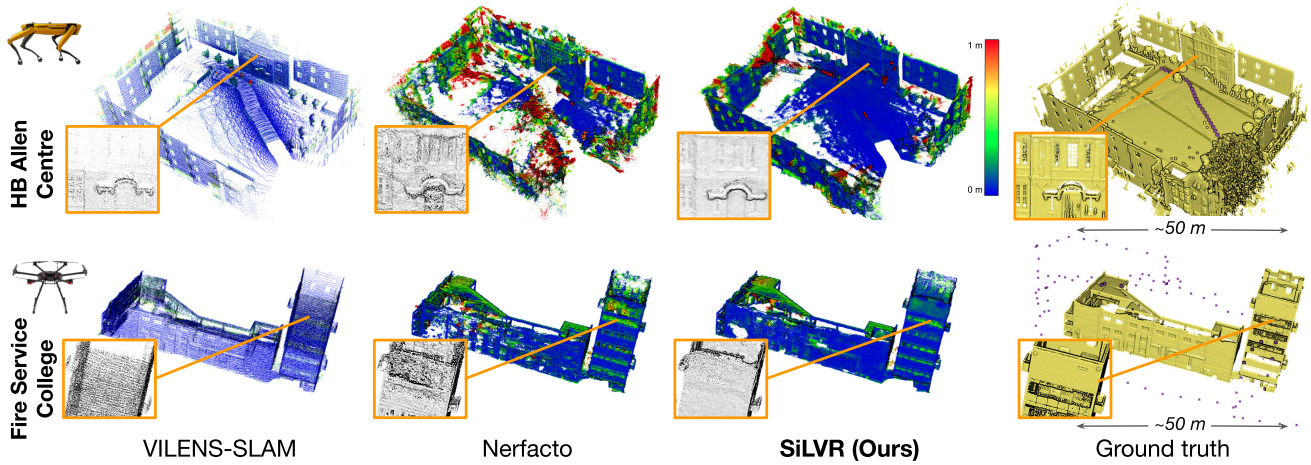


Fig. 5: Comparison of reconstruction quality of VILENS-SLAM, Nerfacto (vision-only) and our approach in small-scale scenes. Reconstructions are coloured using the point-to-point distance between the respective reconstructions and the ground truth scan with increasing error from blue (0m) to red (1m). The trajectory is shown in purple and overlaid on the ground truth scan captured using a Leica BLK360. The zoomed-in views show the difference in reconstruction quality. Overall, our approach is more complete w.r.t lidar-only reconstruction, and geometrically more consistent w.r.t vision-only reconstruction.

Reconstruction \ Reference	Occupied	Free	Unknown
Occupied	True Positive	False Positive	Filter
Free	False Negative	True Negative	Filter
Unknown	Filter	Filter	

Fig. 6: Classification of different occupancy categories for the reconstruction and reference models.

2) Map Filtering for Fair Evaluation: Perfect accuracy and completeness scores (of zero in both cases) would be achieved if the two point clouds are identical, and any deviation is penalised by a higher value. In practice, the ground truth model and the reconstruction do not perfectly overlap, as they scan slightly different parts of the scene from different viewpoints. Two situations can occur which do not correspond to mapping error:

1) Missing regions in the ground truth map: the ground truth map can have undetected areas of the scene that were captured in the Frontier data sequence. This would lead to an artificially higher accuracy score for the Frontier data in such regions. These are in effect *false positives*.
2) Extra regions in the ground truth map: the ground truth map can contain areas that the Frontier device did not scan. In this case, the NeRF reconstruction of these extra regions will be missing. These are undesirable *false negatives*, and the completeness score would again be artificially higher than it should be.

These overestimated error measures are typically much higher than the errors which occur in well-defined regions (both for the TLS ground truth and the Frontier data), and can then skew the results metrics. This makes comparison between different reconstruction methods difficult.

To address this issue, we filter the non-overlapping regions that we consider should not be included in the evaluation — for both the reconstruction and the ground truth. Specifically, our evaluation system first builds an occupancy map of the ground truth reconstruction using Octomap [28]. We then remove points in the reconstruction that are not in the octree (i.e. in the *unknown* space). Similarly, we build an occupancy map of the lidar point clouds, and remove ground truth points within the unknown space. Manual filtering is also applied for regions that are inside the buildings.

3) Rendering Metrics: We evaluate the visual quality of the reconstructions by reporting the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [63], which are standard metrics in the radiance field literature. Note that our images have variable exposure times which lowers the test PSNR even if the reconstructed images have a very high degree of photorealism.

VI. Experimental Results

A. Evaluation of the 3D Reconstruction

We perform a quantitative evaluation of our method using real-world datasets captured by different robotic platforms. Of the evaluation datasets used, HBAC and FSC are small-scale scenes (just one building or a single enclosed space), while ROQ and Bodleian Library are large-scale scenes (connected building complexes). We compare the output point cloud reconstructions from the following algorithms:

1) VILENS-SLAM: lidar point clouds are registered with poses computed by an odometry system VILENS [16] and pose graph optimisation [17]

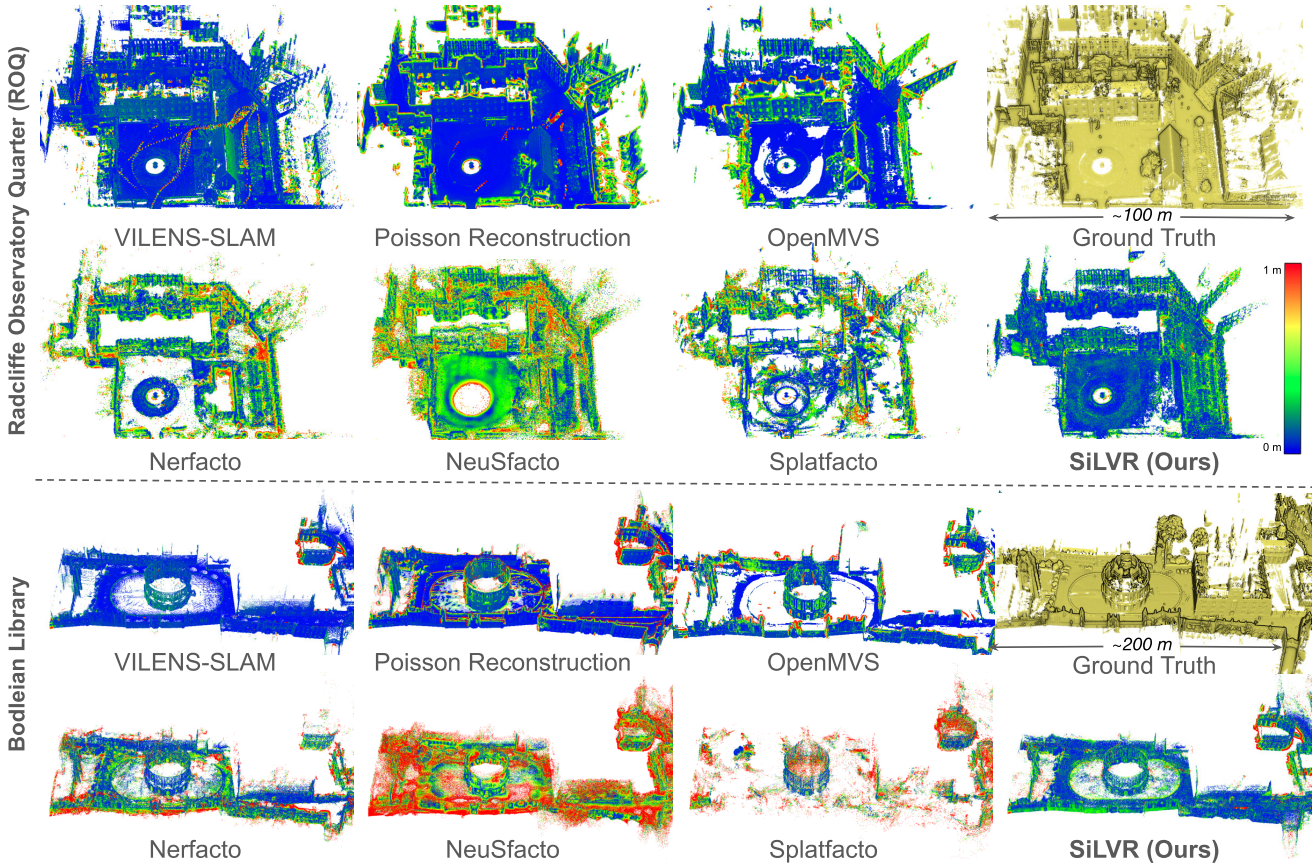


Fig. 7: Comparison of reconstruction quality of our method and other baseline methods in two real-world large-scale scenes. Among the radiance field baseline methods, SiLVR's reconstruction is the most accurate and complete, especially on the ground where there is insufficient visual constraints.

2) Poisson Reconstruction [64]: surface reconstruction using point clouds from VILENS-SLAM
3) OpenMVS [3]: multi-view stereo reconstruction
4) Nerfacto [9]: vision-only radiance field reconstruction using volume density
5) NeuSfacto [5]: vision-only radiance field reconstruction using SDF
6) Splatafacto [65]: vision-only radiance field reconstruction using 3D Gaussians which are initialised using SfM visual features from COLMAP
7) SiLVR: Our proposed method using photometric loss, depth loss, and surface normal loss

Note that all the methods (except VILENS-SLAM and Poisson reconstruction) use the same set of poses and input images. For the large-scale datasets ROQ and Bodleian Library, we use the same submap partitioning for all the radiance field approaches (Nerfacto, NeuSfacto, Splatfacto, and SiLVR).

We summarise the quantitative results in Tab. II and Tab. III, and show the 3D reconstructions in Fig. 5 and Fig. 7. Among all methods, lidar-only method VILENS-SLAM is the most accurate and complete. OpenMVS produces much more accurate and complete reconstructions compared to the radiance field reconstruction, but produces a poor reconstruction of the ground compared to lidar-based methods. This is expected as there is little texture on the ground. All the radiance field reconstructions are less accurate and less complete compared to VILENS-SLAM and OpenMVS. Nerfacto fails to estimate most of the ground geometry accurately in ROQ (the reconstruction is below the ground and is filtered by the occupancy map described in Sec. V-C2). Compared to Nerfacto, NeuSfacto reconstructs the ground surface better, but still at an incorrect height compared to the ground truth. This shows that while the SDF formulation poses a geometric prior on the scene (enforcing that there should be a surface rather than an arbitrary volumetric field), it still cannot estimate the surface accurately if there are insufficient visual constraints. The 3D Gaussians exported by Splatfacto also cannot reconstruct the ground accurately. These 3D Gaussians are located mostly on the visual features of the sites—since they are initialised by the COLMAP feature points.

Compared to the vision-only methods, SiLVR incorporates lidar measurements and has significantly better reconstruction fidelity especially on the ground. Compared to VILENS-SLAM, SiLVR achieves more complete reconstruction for the shorter sequences (e.g., HBAC in Fig. 5) since it uses dense visual information. When there are many accumulated lidar points (which is the case for the large-scale datasets in Fig. 7), this advantage is less prominent.

[3]Available at https://github.com/cdcseacave/openMVS

TABLE III: Evaluation of 3D reconstruction quality. classical methods (Lidar SLAM, Poisson Reconstruction and MVS) and radiance file methods are grouped separately. The best results in each group are indicated in bold.

Method	Accuracy↓ (m)	Completeness↓ (m)	5cm Precision	5cm Recall	5cm F-Score	10cm Precision	10cm Recall	10cm F-Score	PSNR↑ train	PSNR↑ test	SSIM↑ test	LPIPS↓
Radcliffe Observatory Quarter (ROQ)												
VILENS-SLAM	**0.077**	**1.214**	**0.552**	**0.367**	**0.441**	**0.832**	**0.625**	**0.714**	/	/	/	/
Poisson Reconstruction	0.146	1.768	0.406	0.274	0.327	0.658	0.558	0.604	/	/	/	/
OpenMVS	0.123	1.570	0.460	0.353	0.399	0.688	0.495	0.576	/	/	/	/
Nerfacto 3-cam	0.916	2.272	0.220	0.072	0.109	0.392	0.189	0.256	**25.71**	**20.92**	**0.714**	**0.490**
Splatfacto 3-cam	0.478	2.415	0.240	0.044	0.074	0.395	0.151	0.218	21.96	19.95	0.712	0.514
NeuSfacto 3-cam	0.699	2.763	0.051	0.021	0.030	0.115	0.098	0.106	21.17	16.97	0.521	0.549
SiLVR (Ours)	**0.095**	**1.803**	**0.416**	**0.150**	**0.221**	**0.699**	**0.344**	**0.461**	24.73	20.90	0.653	0.551
Bodleian Library												
VILENS-SLAM	1.017	**0.736**	**0.324**	0.098	0.150	**0.518**	0.290	**0.372**	/	/	/	/
Poisson Reconstruction	1.256	1.230	0.239	0.104	0.145	0.400	**0.334**	0.364	/	/	/	/
OpenMVS	**0.955**	2.257	0.223	**0.129**	**0.163**	0.429	0.280	0.339	/	/	/	/
Nerfacto 3-cam	2.841	1.124	0.092	0.030	0.045	0.190	0.132	0.156	**28.92**	**23.03**	0.827	**0.715**
Splatfacto 3-cam	13.532	1.275	0.020	0.004	0.007	0.044	0.027	0.033	23.92	22.19	**0.850**	0.748
NeuSfacto 3-cam	2.656	**1.074**	0.015	0.007	0.010	0.035	0.042	0.038	24.00	20.61	0.619	0.731
SiLVR (Ours)	**1.292**	1.532	**0.129**	**0.041**	**0.063**	**0.276**	**0.170**	**0.211**	28.00	22.94	0.754	0.779

Regarding the rendering quality, Nerfacto achieves the best results among the radiance field reconstructions. We found that NeuSfacto's training takes longer than all the other methods, and the rendering quality is worse than the other methods. SiLVR achieves a balance between the rendering quality and the 3D reconstruction quality.

B. Evaluation of Epistemic Uncertainty Estimation

We evaluate the epistemic uncertainty estimates using both synthetic data and real-world data. The synthetic data is generated using the CARLA simulator [66], which provides perfect pose trajectories and ground truth maps. We simulated a vehicle with a lidar and three-cameras in a configuration similar to the Frontier. Meanwhile, the real-world dataset used in this section is the Radcliffe Observatory Quarter (ROQ).

1) Evaluation Metrics: We evaluate the epistemic uncertainty estimates using the Sparsification Plot which has been used for evaluating confidence estimates in the literature [67], [68], [15]. The Sparsification Plot is used to evaluate how the uncertainty estimates coincide with the actual errors (in our case, point-to-point distance to the ground truth). In these plots, reconstruction points with the highest uncertainty are gradually removed, and the average errors of the remaining reconstruction points are calculated to form a graph. If the uncertainty estimates align perfectly with the actual errors, then the reconstruction points with the the highest errors are always removed first, which leads to the steepest decrease of the remaining error as the most uncertain points are being removed. This ideal sparsification curve is referred to as *Oracle Sparsification*. In practice, the uncertainty estimates do not align with the actual errors perfectly. Specifically, when a reconstruction point has a higher uncertainty but lower error compared to another point, the uncertainty estimates are considered not perfect. The area between the sparsification and its oracle indicates how different the uncertainty estimates are from the ideal ones, and can then be used to compute the Area Under Sparsification Error (AUSE). A smaller difference between the sparsification and its oracle results in a lower AUSE, and indicates that the uncertainty estimates are better because they align better with the actual errors.

In addition to the Sparsification Plot which focuses on the error of the *remaining* reconstructions, we also analyse the error of the reconstruction that is *being removed* (according to the uncertainty). This is achieved by plotting the errors of the reconstruction that have different levels of uncertainties, which we refer to as the Error Plot. While the errors in the Sparsification Plots are normalised (since AUSE is scale-invariant), we use metric errors in the Error Plot to keep the scale information.

2) Results: In Fig. 8, we evaluate the decoupled epistemic uncertainty estimates quantitatively using the Sparsification Plot and the Error Plot, and qualitatively by showing the reconstructions with low and high uncertainty estimates. As shown in the Error Plots, both visual and lidar depth uncertainty can indicate the degree of the reconstruction error. In particular, we can observe how the visual uncertainty and lidar depth uncertainty capture different parts of the scene according to the sensor characteristics. From the reconstruction error figure of both CARLA and ROQ, it can be seen that reconstructions with low visual uncertainty generally are places where visual features can be detected. In fact, this corresponds to the visual features that can be reliably estimated by classical SfM and visual SLAM methods. In ROQ, much of the ground in the quad has relatively higher visual uncertainty but lower lidar depth uncertainty. Essentially, even if there are few visual features on the ground which are not ideal for visual reconstruction, the lidar measurements provide sufficient information to produce an accurate ground reconstruction. When uncertainty estimates are high, we found that lidar depth uncertainty is a better indicator of the reconstruction error than visual uncertainty. From the Error Plot, the average error of points with high lidar depth uncertainty estimates (blue curve) is generally higher than the error of points with high visual uncertainty estimates

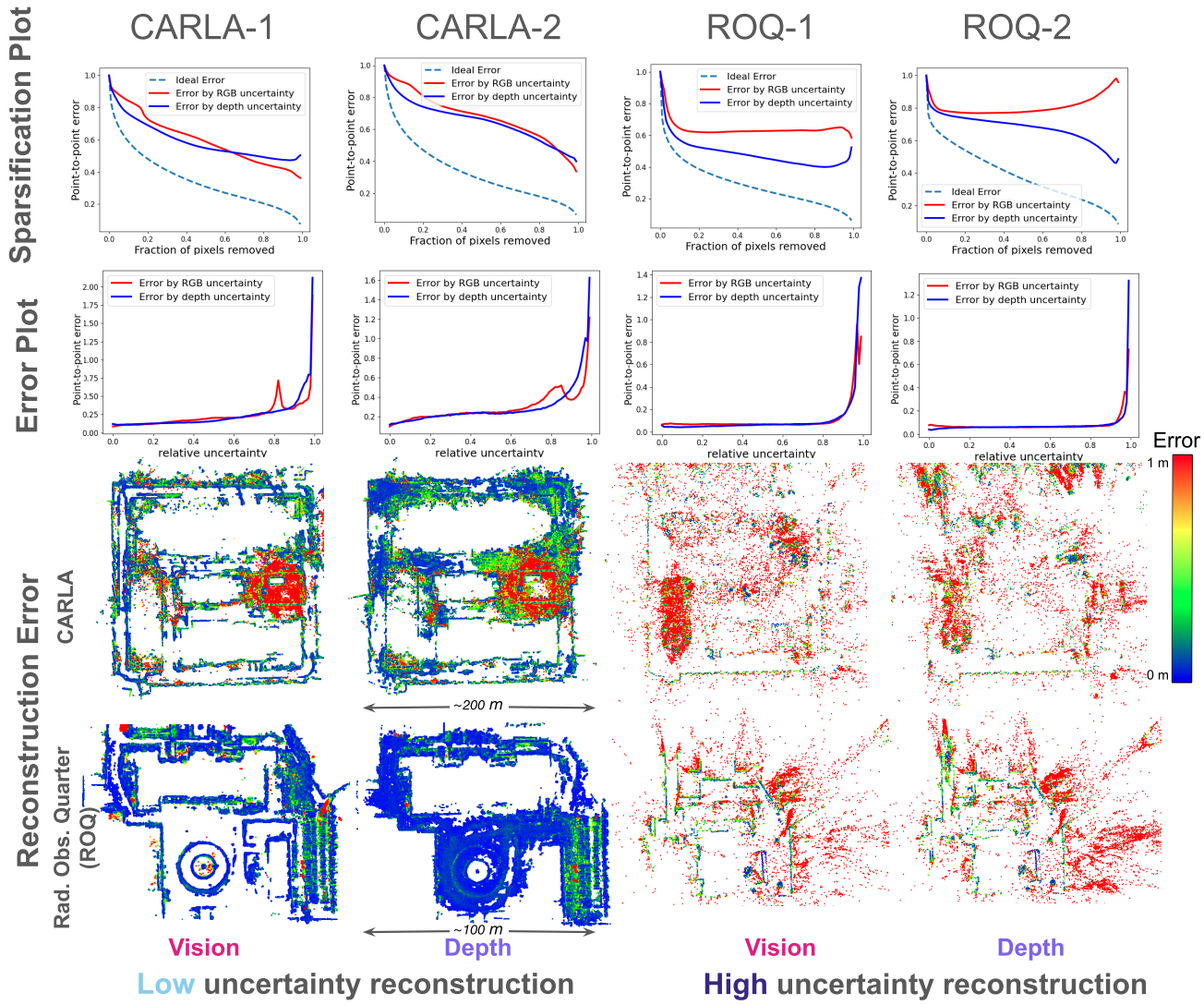


Fig. 8: Qualitative and quantitative evaluation of epistemic uncertainty estimates using both synthetic (CARLA) and real-world (ROQ) datasets. The Sparsification Plot shows how the (normalised) reconstruction error decreases as more uncertain points are removed, and how close it is to the oracle error reduction curve. The Error Plot additionally shows the distribution of error at different uncertainty estimates. Low visual uncertainty corresponds to visual features which are well constrained by the image view constraints, similar to the SfM and Visual SLAM systems. Low lidar depth uncertainty indicates that there is abundant lidar observation, and hence the geometry is also constrained.

(red curve). This can also be shown in the Sparsification Plot: as the first 20% of the reconstruction of higher uncertainty are being removed, the sparsification curve by depth uncertainty (blue curve) is closer to the ideal oracle sparsification (dashed blue curve) compared to the sparsification curve by visual uncertainty (red curve). This is because the depth uncertainty estimates remove points with higher errors than the visual uncertainty estimates, and hence the errors of the remaining points are lower.

The advantage of the disentangled uncertainty estimates is also demonstrated in Fig. 9 where we use a narrow vertical field-of-view lidar with wider vertical field-of-view cameras. Here, the lidar is not able to scan the upper part of the two buildings highlighted in 9, but the cameras can. Because of this, when we compute the epistemic uncertainty, we can see high lidar depth uncertainty for the upper part of both buildings. This indicates that the reconstruction is derived primarily from the visual data. In summary, from a sensor fusion point of view, our epistemic uncertainty estimation framework provides a systematic analysis of each sensor's contribution to the final reconstruction.

C. View Selection Strategy

In this section, we compare our visibility-based submapping strategy with an alternative distance-based submapping strategy proposed in [12]. As shown in Fig. 10, the building highlighted is divided into two submaps when using the distance-based submapping method. This is not ideal as it reduces the number of view constraints, which makes each

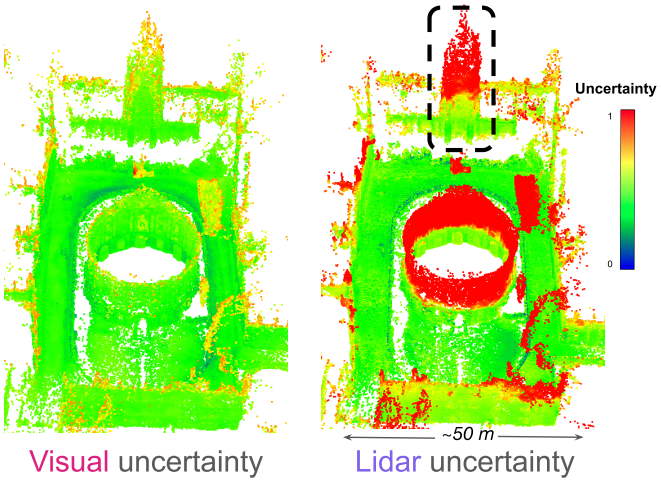


Fig. 9: Comparison between visual and lidar uncertainty. The reconstructions at the top are coloured by uncertainty estimates (red is high uncertainty, green is low uncertainty). Here, the lidar used was a Hesai XT32 which has a narrow field-of-view and cannot scan the upper part of the buildings, and the depth uncertainty estimates can identify such regions (red point clouds). This indicates that only visual information is used when reconstructing these areas.

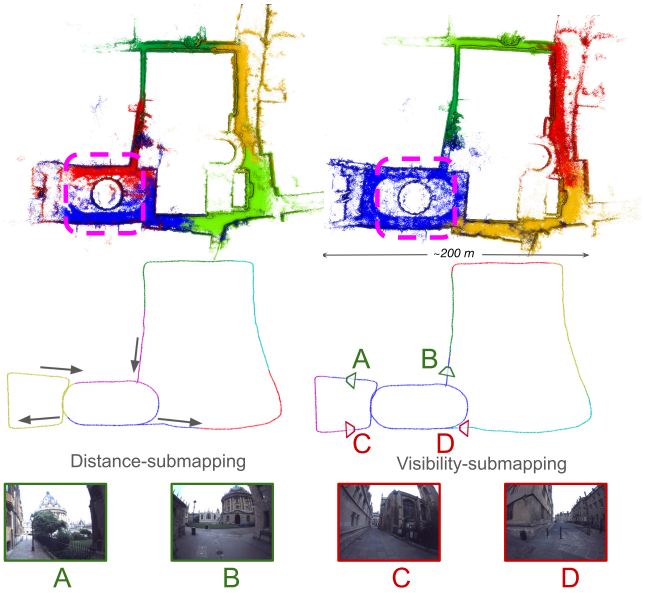


Fig. 10: Submapping strategy comparison. Visibility information guides the clustering algorithm to group images looking at the same object together, which then leads to more accurate and complete reconstruction of that object than algorithms that only consider distances.

submap's partial reconstruction of that building have a lower quality. In addition, distance-based submapping put the poses in A and C into the same submap, which is in fact not ideal. While pose A and pose C are spatially close, they have opposite viewing directions: pose A is looking at the highlighted building, while pose C is looking away from it. In comparison, visibility-based submapping moves pose A into the submap that contains the highlighted building, and pose C into another submap.

D. Multi-Camera Setup Ablation Study

The advantage of our multi-camera sensor setup is demonstrated qualitatively in Fig. 11. Compared to the three-camera setup, using only the front-facing camera leads to a reconstruction that is not only incomplete, but also with poorer geometry. Visual reconstruction with the photometric loss is limited to generating a good quality rendering only at the input viewing angle. The reconstruction using the front-only camera in Fig. 11 is trained with images looking in a single direction in the scene. This results in a poor geometric reconstruction when rendered from an unseen angle. In comparison, reconstruction with three cameras generates a more complete and more accurate reconstruction.

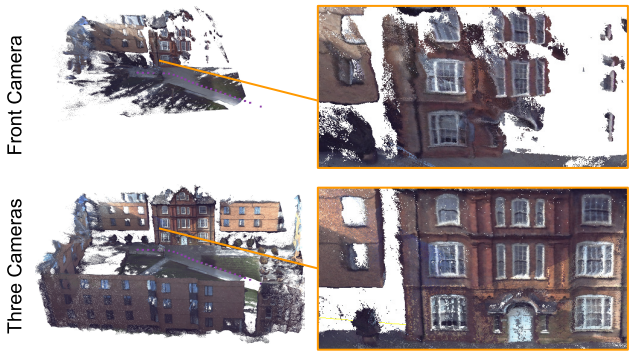


Fig. 11: Comparison of reconstruction of HBAC building using the front camera only vs. using all the three cameras. The three-camera setup generates more complete and accurate reconstructions compared to using only a single front-facing camera. The multi-camera setting is important in robotic applications where it would be infeasible to actively scan the entire scene to obtain strong viewpoint constraints.

E. Effect of Bootstrapping SLAM Poses

We compare the performance of different strategies for computing poses: SLAM poses produced online, SLAM poses later refined using NeRF [9], SLAM poses refined using COLMAP [1]'s Bundle Adjustment in different configurations, and COLMAP without any prior poses. For COLMAP, we tested different numbers of features extracted per image, as well as two different COLMAP feature matching algorithms: sequential matching with loop closures and Vocabulary Tree Matcher [69].

The results are summarised in Tab. IV. For all COLMAP configurations, providing the SLAM prior poses not only accelerates pose computation, but also leads to better test rendering, compared to COLMAP without any initialisation. Our SLAM prior poses can also register all the images in the trajectory; meanwhile COLMAP on its own only registers only 55%-95% images. Extracting more visual features per image (from 1024 to 8192) leads to a higher percentage of image

registration and better visual reconstruction (PSNR and SSIM). This comes at the expense of a higher computation time, especially with the VocabTree matcher. Using the COLMAP Sequential Matcher is generally faster than Vocabulary Tree Matcher.

TABLE IV: Ablation: Effect of Bootstrapping w/ SLAM Poses

Method	Features	Prior	Traj. Registered (%)	PSNR↑ Train	PSNR↑ Test	SSIM↑ Test	Time (s)
VILENS	/	/	100.0	23.0	17.4	0.64	Online
NeRF refined	/	/	100.0	23.2	17.9	0.65	Online
COLMAP Sequential	1024		57.6	25.9	19.1	0.71	3299.2
	1024	✓	100.0	26.2	**20.6**	**0.74**	1729.9
	8192		94.0	26.1	19.8	0.72	7850.0
	8192	✓	100.0	26.2	20.4	0.73	4448.4
COLMAP VocabTree	1024		54.7	26.2	19.0	0.71	4444.8
	1024	✓	100.0	26.3	20.4	0.73	1052.5
	8192		94.8	**26.6**	19.9	0.72	37067.5
	8192	✓	100.0	26.3	20.4	**0.74**	11015.3

Results evaluated on HBAC-Maths dataset with 3254 images and duration of 1270s. Models trained for 4000 iterations. PSNR and SSIM were evaluated after masking out the sky.

VII. Conclusions

In summary, we proposed a large-scale 3D reconstruction system fusing both lidar and vision in a neural radiance field. The proposed approach combines the advantages of the two sensor modalities and generates reconstructions with both photo-realistic textures as well as accurate geometry. We proposed a principled approach to quantification reconstruction uncertainty considering each sensor's characteristics, which enables us to identify unreliable reconstruction artefacts and filter them out to improve reconstruction accuracy. With our proposed submapping approach, we demonstrate large-scale reconstruction results from real-world datasets collected in different robot platforms in conditions suited to industrial inspection tasks.

References

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

[2] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *European Conference on Computer Vision (ECCV)*, 2016.

[3] 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," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[4] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for real-time radiance field rendering," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 4, July 2023.

[5] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "NeuS: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *International Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[6] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022.

[7] M. Tancik, V. Casser, X. Yan, S. Pradhan, B. Mildenhall, P. P. Srinivasan, J. T. Barron, and H. Kretzschmar, "Block-NeRF: Scalable large scene neural view synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8248–8258.

[8] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-NeRF: Scalable construction of large-scale NeRF for virtual fly-throughs," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 922–12 931.

[9] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, J. Kerr, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja, D. McAllister, and A. Kanazawa, "Nerfstudio: A modular framework for neural radiance field development," in *SIGGRAPH*, 2023.

[10] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 882–12 891.

[11] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," *International Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[12] 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 International Conference on Robotics and Automation (ICRA)*, 2024.

[13] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 6290–6301.

[14] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux-effortless bayesian deep learning," *International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 20 089–20 103, 2021.

[15] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi, "Bayes' Rays: Uncertainty quantification in neural radiance fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[16] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2023.

[17] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, "Online LiDAR-SLAM for legged robots with robust registration and deep-learned loop closure," in *IEEE Robotics and Automation Letters*, May 2020, pp. 4158–4164.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[19] Y. Tao, M. Á. Muñoz-Bañón, L. Zhang, J. Wang, L. F. T. Fu, and M. Fallon, "The Oxford Spires dataset: Benchmarking large-scale LiDAR-visual localisation, reconstruction and radiance field methods," *arXiv preprint arXiv:2411.10546*, 2024.

[20] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Robotics: Science and Systems (RSS)*, 2018.

[21] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 672–10 678.

[22] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems (RSS)*, vol. 2, no. 9, 2014.

[23] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super Odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.

[24] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[25] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 403–429, 2006.

[26] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[27] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic LiDAR Fusion: Dense map-centric continuous-time SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1206–1213.

[28] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013.

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[30] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D euclidean signed distance fields for on-board MAV planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1366–1373.

[31] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. J. Kelly, and S. Leutenegger, "Efficient octree-based volumetric SLAM supporting signed-distance and occupancy mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, Apr. 2018.

[32] V. Reijgwart, C. Cadena, R. Siegwart, and L. Ott, "Efficient volumetric mapping of multi-scale environments using wavelet-based compression," 2023.

[33] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[34] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An *Atlas* framework for scalable mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1899–1906 vol.2.

[35] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building Rome in a day," *Communications of the ACM*, vol. 54, no. 10, p. 105–112, 2011.

[36] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1434–1441.

[37] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2175–2182.

[38] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, 2020.

[39] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon, "Strategies for large scale elastic and semantic LiDAR reconstruction," *Journal of Robotics and Autonomous Systems*, vol. 155, p. 104185, 2022.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[41] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5501–5510.

[42] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *International Conference on Computer Vision (ICCV)*, 2021.

[43] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[44] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[45] K. Rematas, A. Liu, P. P. Srinivasan, J. T. Barron, A. Tagliasacchi, T. Funkhouser, and V. Ferrari, "Urban radiance fields," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 12 932–12 942.

[46] Z. Li, T. Müller, A. Evans, R. H. Taylor, M. Unberath, M.-Y. Liu, and C.-H. Lin, "Neuralangelo: High-fidelity neural surface reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[47] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," *International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 4805–4815, 2021.

[48] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D Gaussian Splatting for geometrically accurate radiance fields," in *SIGGRAPH*, 2024.

[49] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "SHINE-Mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[50] J. Deng, Q. Wu, X. Chen, S. Xia, Z. Sun, G. Liu, W. Yu, and L. Pei, "NeRF-LOAM: Neural implicit representation for large-scale incremental lidar odometry and mapping," in *International Conference on Computer Vision (ICCV)*, 2023.

[51] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss, "PIN-SLAM: LiDAR SLAM Using a Point-Based Implicit Neural Representation for Achieving Global Map Consistency," *IEEE Transactions on Robotics*, vol. 40, pp. 4045–4064, 2024.

[52] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[53] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "RobustNeRF: Ignoring distractors with robust losses," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 20 626–20 636.

[54] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *International Conference on Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[55] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, no. 2, pp. 99–108, 1995.

[56] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *SIGGRAPH*, vol. 18, no. 3, pp. 165–174, 1984.

[57] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019.

[58] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[59] R. Border, N. Chebrolu, Y. Tao, J. D. Gammell, and M. Fallon, "Osprey: Multisession autonomous aerial mapping with LiDAR-Based SLAM and next best view planning," *IEEE Transactions on Field Robotics*, vol. 1, pp. 113–130, 2024.

[60] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 1280–1286.

[61] L. F. T. Fu, N. Chebrolu, and M. Fallon, "Extrinsic calibration of camera to lidar using a differentiable checkerboard model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[62] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *International Journal of Computer Vision*, pp. 1–16, 2016.

[63] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[64] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Eurographics Symposium on Geometry Processing*, vol. 7, no. 4, 2006.

[65] V. Ye and A. Kanazawa, "Mathematical supplement for the `gsplat` library," 2023.

[66] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.

[67] E. Ilg, O. Cicek, S. Galesso, A. Klein, O. Makansi, F. Hutter, and T. Brox, "Uncertainty estimates and multi-hypotheses networks for optical flow," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 652–667.

[68] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware CNNs for depth completion: Uncertainty from beginning to end," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[69] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Asian Conference on Computer Vision (ACCV)*, 2017, pp. 321–337.

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	SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification
Publication Status	□Published □ Accepted for Publication ✕Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Y. Tao and M. Fallon, "SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction with Uncertainty Quantification," under review by IEEE Transactions on Robotics, 2025

Student Confirmation

Student Name:	Yifu Tao		
Contribution to the Paper	• Contributed to development of the idea • Developed the uncertainty quatification and submapping modules • Conducted experiments • Performed data analysis and processing • Wrote the majority of the paper		
Signature		Date	22nd January 2025

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: Maurice Fallon			
Supervisor comments Yifu was the lead investigator, implementer and researcher on this project and wrote the bulk of the paper			
Signature		Date	22nd January 2025

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

6.2 Technical Remarks

Additional Information on Figure 9

In Figure 9 in the original paper submission, we did not provide enough context about the scene and the input data. In Fig. 6.1, we provide an updated version of Figure 9 which now includes a colourised point cloud exported from the NeRF reconstruction, and the input lidar-camera overlay. The input shows how the lidar did not scan the top of the building highlighted, which leads to high lidar uncertainty in the final reconstruction.

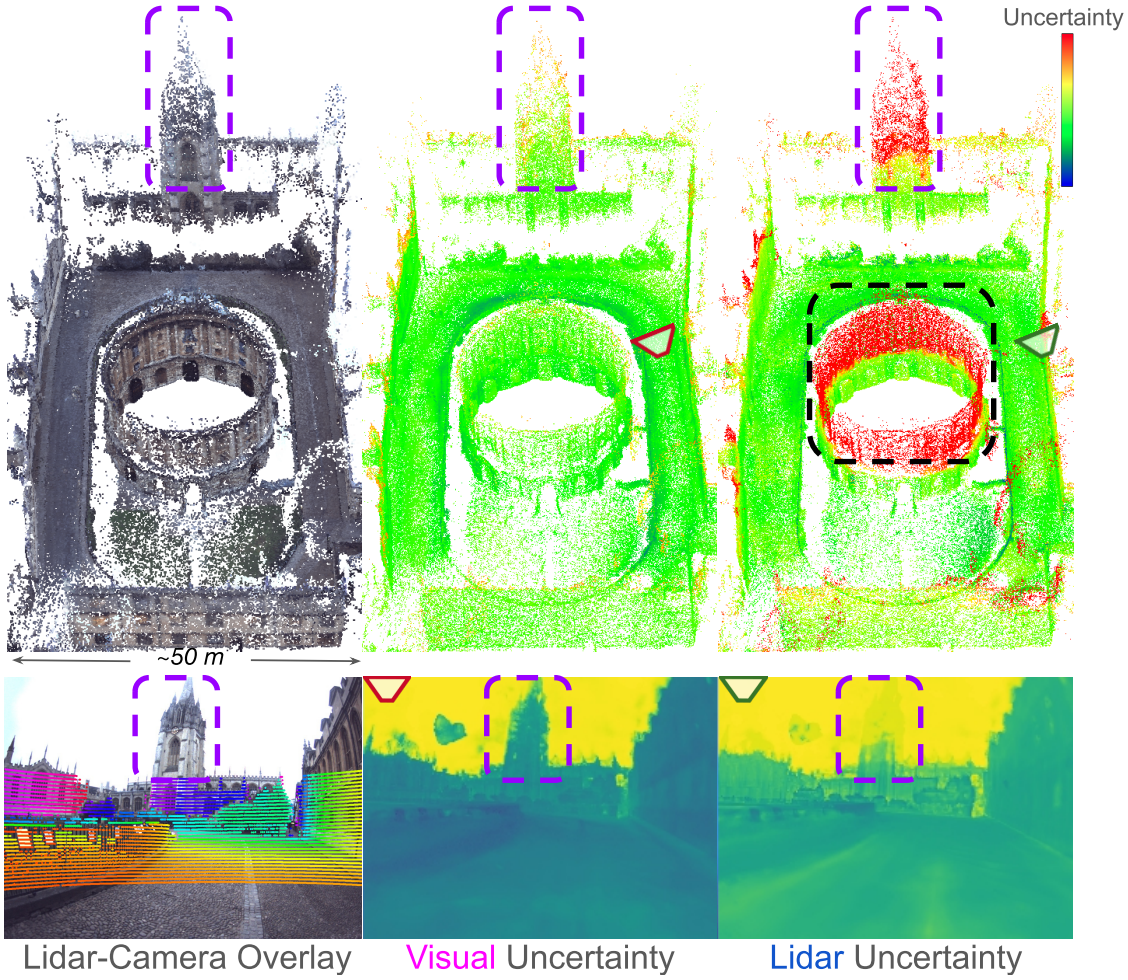


Figure 6.1: Updated version of Figure 9 in the original paper submission. Here we compare between visual and lidar uncertainty. The middle and right reconstructions on the top row are coloured by uncertainty estimates. Here, the lidar used was a Hesai XT32 which has a narrow field-of-view and cannot scan the upper part of the buildings, and the depth uncertainty estimates can identify such regions (red point clouds). This indicates that only visual information is used when reconstructing these areas.

Choice of Parameters for Uncertainty Estimation

Estimating epistemic uncertainty using the Laplace approximation (LA) is a popular approach in the Bayesian Deep Learning literature [187]. LA has been applied to radiance field methods with implicit representation (NeRF, as in BayesRays [188]) and explicit representation (Plenoxel and 3DGS, as in FisherRF [189]). The major difference between BayesRays [188] and FisherRF [189] is the choice of the parameter whose posterior is used to estimate the epistemic uncertainty. In FisherRF [189], the map parameters are the density and the radiance/colour of the voxel or the 3D Gaussian. BayesRays [188] instead introduces a deformation field whose parameters are used for uncertainty estimation. This is motivated by the fact that estimating the covariance of the MLP weights of a NeRF does not directly translate to the uncertainty in the 3D space, which is more meaningful for robotic tasks.

Different choices of parameters result in different final uncertainty estimations. While the 3D representations used in BayesRays and FisherRF are different, the choices of parameters are not necessarily incompatible. For example, the choice of density as the parameters (which FisherRF applies to 3DGS and Plenoxel) can also be applied to a NeRF. In this case, we construct a density-changing field, and compute the Jacobian of the output colour with respect to the density of the samples along the ray. For a uniform-textured surface, changing the density of a point on the surface might not affect the final colour, since the colour of the neighbouring points are all similar. It is plausible to expect that this shares similarity with the deformation field formulation used by BayesRays. One interesting avenue for future investigation is to compare the behaviour of these different formulations. This approach could potentially provide a unifying framework.

6.3 Discussion

Comparison with other Uncertainty Estimation Techniques

Uncertainty estimation is a crucial concept in SLAM and machine learning. In MonoSLAM [47], the visual landmarks are represented as Gaussians, with their

covariance encoding uncertainty. In occupancy mapping, each voxel's occupancy is modelled probabilistically, which also captures the confidence of the map estimation. These modelling techniques are used for explicit discrete map representations. NeRF, on the other hand, represents the 3D space as a neural network, which is an implicit representation of a continuous map. This motivated the use of uncertainty quantification techniques in deep learning, particularly Bayesian neural networks (BNN).

BNN can be implemented using techniques such as Markov chain Monte Carlo (MCMC) [175, 190], deep ensembles [191], variational inference [192] and Laplace approximation [187]. These techniques have their own advantages and limitations. Deep ensembles require training multiple networks with different initialisations, while MCMC requires sampling the posterior distribution. Both methods are computationally expensive. In contrast, Laplace approximation requires no additional cost during training, and can be efficiently estimated after the training. However, it is a local approximation, and therefore cannot effectively model multi-modal posterior distributions. Variational methods approximate the posterior distribution with a pre-determined distribution family (often Gaussian), and it also struggles to capture multi-modal posterior just as Laplace approximation. Fig. 6.2 summarises different Bayesian deep learning techniques' limitations.

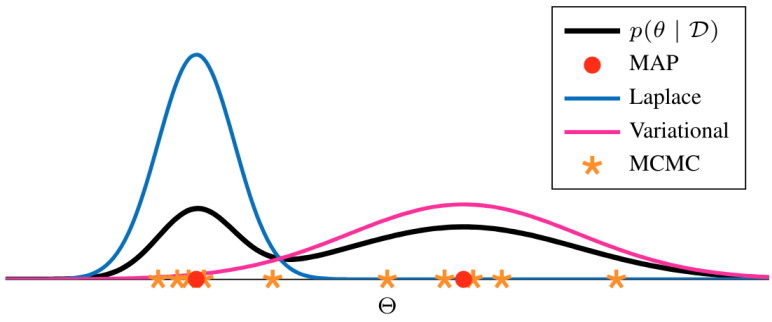


Figure 6.2: Different Bayesian deep learning techniques for posterior approximation with different characteristics. Image from Papamarkou *et al.* [193].

Finally, it is worth noting that Gaussian Process (GP), which also models continuous functions as neural networks, provides a principled framework to

represent uncertainties in the model. It is worthwhile to explore the possibility of representing the radiance field as a Gaussian Process.

Aleatoric Uncertainty

The work focuses on epistemic uncertainty, which identifies regions that can be reliably reconstructed based on the conditioning of visual data and lidar data. A natural extension is to also incorporate aleatoric uncertainty, which measures the inherent variance in the sensor measurements that are irreducible given more observations. In this section, we discuss the aleatoric uncertainties in camera images and in lidar measurements.

For the vision-based NeRF formulation, aleatoric uncertainty arises from the inherent variances in camera images, including appearance changes due ot lighting conditions, weather and seasonal variations, and dynamic objects. There exist methods that model such variances in the camera images. For example, NeRF-W [126] introduced a method to model dynamic objects in the training images using the Bayesian aleatoric uncertainty framework [194] in a similar manner to our work in Chap. 4. In addition, thermal noise might be present in the image, especially in low-light environments such as forests or underground areas (see Fig. 5.3). Recent work, such as RawNeRF [21], shows promise in denoising images captured in the dark.

While Bayesian aleatoric uncertainty framework [194] can model measurement variance, some variations can be explicitly modelled without assuming Gaussian-distributed measurements. One notable source of variation in image appearance is exposure time, which could be obtained from the camera (if the hardware allows) or estimated within a camera sequence. For example, R3LIVE++ [106] uses photometric calibration of the camera to both maintain a radiance map (true colour) and to estimate the exposure time for each image frame. Modelling the variation in the exposure time is beneficial, since it has a significant effect on the per-frame appearance variance in a sequence. This approach is more interpretable than the per-frame latent appearance encoding [126, 195] used in our work.

In common with cameras, lidar measurements also contain aleatoric noise. The aleatoric uncertainty in a lidar measurement is often documented as the standard deviation of the lidar's range reading, which then can be modelled in the same way as the Bayesian framework used in Chap. 4. In addition, lidar measurements' accuracy is also affected by the incident angle and the material property. Consideration of surface normals and lidar intensity might be useful for the noise modelling of the lidar data. Incorporating the aleatoric uncertainty of the visual and lidar data into the reconstruction framework could potentially allow a more complete probabilistic formulation.

Active Mapping

The uncertainty estimation can also be used to enable downstream robotics tasks. One interesting area is active mapping. Given that parts of the reconstruction are not well constrained by vision or not scanned by lidar, a robot moving in the environment could plan its path so as to actively improve the reconstruction quality. Furthermore, extending active mapping systems based purely on lidar measurements (e.g. Osprey [196]) with the consideration of visual data conditioning is quite an interesting area to explore. However, the active mapping system might not necessarily need a sophisticated representation such as NeRF which requires heavy computation. A traditional active visual or lidar SLAM system coupled with a NeRF mapping back-end is a plausible combination and is worth investigation.

Submapping

While this chapter focuses on uncertainty, it also introduces a strategy for submapping which improved the spatial scalability. The proposed visibility-based method draws inspiration from classical large-scale MVS methods such as CMVS [197]. Visual reconstruction and lidar reconstruction requires different submapping strategies. Lidar-based systems can take advantage of direct depth measurements and often spawn a new submap by a simple travel distance threshold [95]. In comparison, visual reconstruction depends on a cluster of co-visible images with sufficient baselines. In this case, visibility is more important than spatial proximity.

One limitation of our visibility-based submapping method is the need for a pre-defined number of submaps. Determining this number is non-trivial, as it is challenging to assess whether a NeRF model is too small for a given image collection. This challenge stems from the fact that the information content within a radiance field differs from the physical size of the scene. For example, a physically large area with simple geometry and uniform colour patterns may be relatively easy for a NeRF to model. In comparison, a smaller-sized tree with leaves and branches which exhibit high-frequency geometric details may require a larger model size with greater representational capability. An overly conservative submap size would lead to an excessive number of submaps, which are more memory-expensive and can be unnecessary; In contrast, too few submaps can degrade the final map quality. One promising research direction is to develop techniques for quantifying the information content (consider both colour and geometry) in a scene, and leveraging it to obtain a more optimal submapping process.

7

Lidar-Visual Dataset for SLAM and Radiance Fields

Contents

7.1 The Oxford Spires Dataset: Benchmarking Large-Scale Lidar-Visual Localisation, Reconstruction and Radiance Field Methods 108
7.2 Technical Remarks 126
7.3 Discussion 127

To develop the lidar-visual mapping system presented in Chap. 5 and Chap. 6, a well characterised and properly curated lidar-visual dataset is necessary. Constructing such a dataset is challenging, as it requires precise time synchronisation of the lidar and image data, along with accurate extrinsics calibration between the lidar and camera. We found that the existing SLAM datasets are not suitable for evaluating large-scale outdoor 3D reconstruction and novel-view synthesis. The majority of SLAM datasets lack high-quality ground truth 3D maps, and are therefore not suitable for evaluating 3D reconstruction. In addition, colour cameras are crucial for radiance field methods, but many SLAM datasets only include greyscale global shutter images as they are sufficient for visual odometry.

This motivated us to create the Oxford Spires dataset, which provides high-

resolution colour image streams from three cameras, a 3D wide Field-of-View lidar data and inertial data from a mobile handheld device, The millimetre-accurate ground truth scans enable us to evaluate the 3D reconstructions using vision (e.g. MVS, NeRF) and lidar data. In particular, our novel-view synthesis benchmark demonstrates the limited generalisation capability of state-of-the-art radiance field methods when the tested poses are distant from the training sequences, an under-explored issue in the literature. We believe this dataset can encourage the integration of SLAM systems with radiance field approaches for large-scale outdoor textured mapping.

7.1 The Oxford Spires Dataset: Benchmarking Large-Scale Lidar-Visual Localisation, Reconstruction and Radiance Field Methods

The manuscript in this section was accepted to the *International Journal of Robotics Research* journal in 2025. An accompanying video is available online at `https://youtu.be/AKZ-YrOob_4`.

Journal Title
XX(X):1–16
©The Author(s) 2025
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

The Oxford Spires Dataset: Benchmarking Large-Scale LiDAR-Visual Localisation, Reconstruction and Radiance Field Methods

Yifu Tao*[1], Miguel Ángel Muñoz-Bañón*[1,2], Lintong Zhang[1], Jiahao Wang[1], Lanke Frank Tarimo Fu[1], and Maurice Fallon[1]

Abstract

This paper introduces a large-scale multi-modal dataset captured in and around well-known landmarks in Oxford using a custom-built multi-sensor perception unit as well as a millimetre-accurate map from a Terrestrial LiDAR Scanner (TLS). The perception unit includes three synchronised global shutter colour cameras, an automotive 3D LiDAR scanner, and an inertial sensor — all precisely calibrated. We also establish benchmarks for tasks involving localisation, reconstruction, and novel-view synthesis, which enable the evaluation of Simultaneous Localisation and Mapping (SLAM) methods, Structure-from-Motion (SfM) and Multi-view Stereo (MVS) methods as well as radiance field methods such as Neural Radiance Fields (NeRF) and 3D Gaussian Splatting. To evaluate 3D reconstruction the TLS 3D models are used as ground truth. Localisation ground truth is computed by registering the mobile LiDAR scans to the TLS 3D models. Radiance field methods are evaluated not only with poses sampled from the input trajectory, but also from viewpoints that are from trajectories which are distant from the training poses. Our evaluation demonstrates a key limitation of state-of-the-art radiance field methods: we show that they tend to overfit to the training poses/images and do not generalise well to out-of-sequence poses. They also underperform in 3D reconstruction compared to MVS systems using the same visual inputs. Our dataset and benchmarks are intended to facilitate better integration of radiance field methods and SLAM systems. The raw and processed data, along with software for parsing and evaluation, can be accessed at https://dynamic.robots.ox.ac.uk/datasets/oxford-spires/.

Keywords

Dataset, Localisation, 3D Reconstruction, Novel-View Synthesis, SLAM, NeRF, Radiance Field, LiDAR Camera Sensor Fusion, Colour Reconstruction, Calibration

1 Introduction

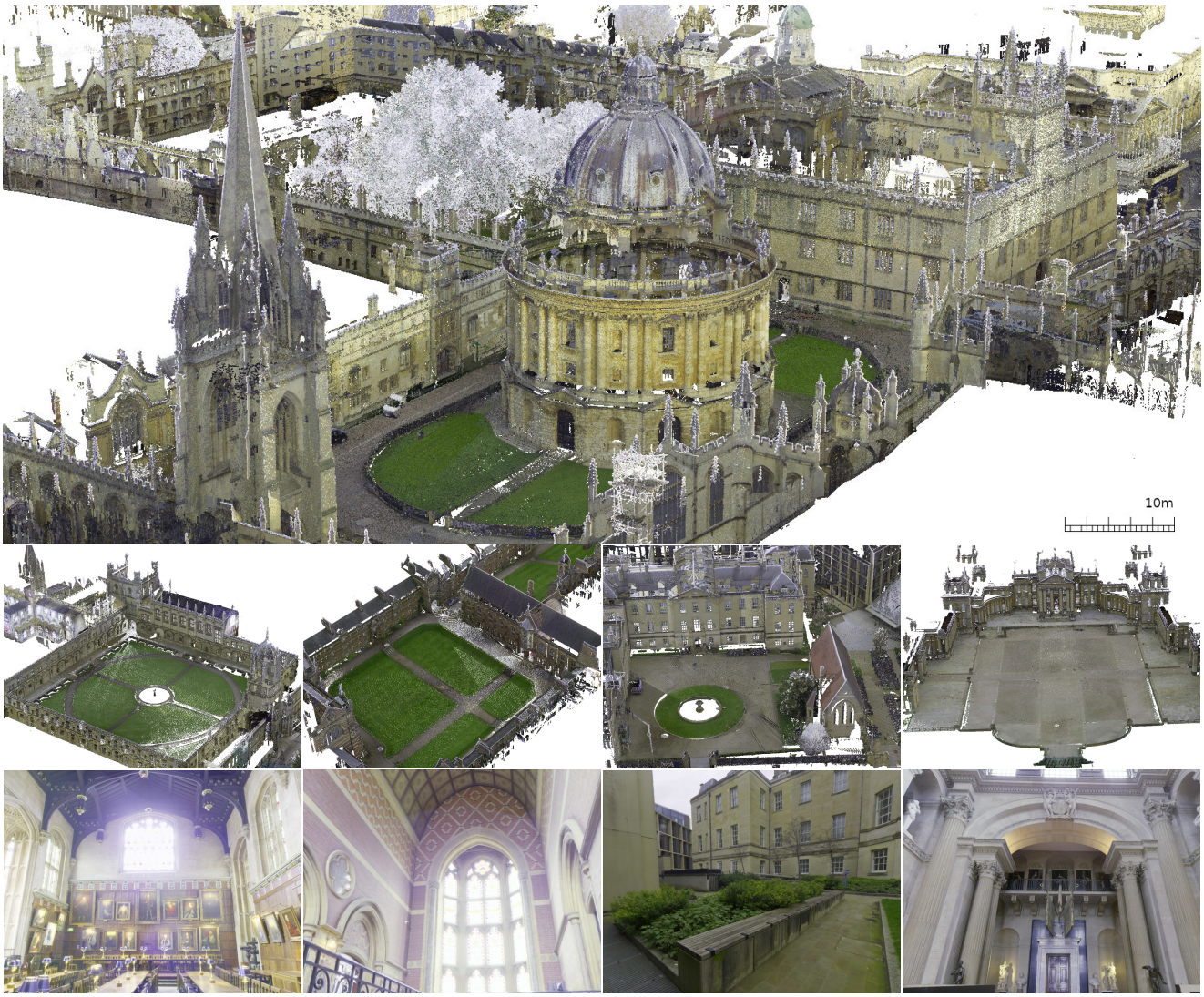
Localisation and 3D reconstruction are fundamental problems in both robotics and computer vision, with applications spanning autonomous driving, building inspection and augmented reality. There are methods that focus on localisation (e.g. visual or lidar odometry, Structure-from-Motion (SfM), place recognition and relocalisation) and others that focus on 3D reconstruction/mapping (e.g. Multi-view Stereo (MVS) and occupancy mapping). In mobile robotics, both problems can be solved concurrently by Simultaneous Localisation and Mapping (SLAM) methods, which are our primary focus in this work. For large-scale outdoor environments, cameras and LiDARs are the most commonly used exteroceptive sensor modalities for environment perception tasks. Additionally, the Inertial Measurement Unit (IMU) is a common interoceptive sensor. The two mentioned exteroceptive sensor technologies have complementary characteristics: LiDAR captures long-range depth measurements that are accurate but sparse, while camera images capture texture with higher resolution.

The evaluation of the outdoor SLAM systems has primarily focused on localisation accuracy, conversely quantitative evaluation of the 3D reconstruction quality is often lacking. An important reason for this is the limited availability of high-quality ground truth. For 3D reconstruction, ground truth is typically collected using survey-grade Terrestrial LiDAR Scanners (TLS) which are expensive (Zhang et al. 2022). Compared to indoor scenes were MVS and RGB-D SLAM systems are often evaluated, the large-scale of outdoor environments make TLS data collection laborious. As a result, many outdoor SLAM datasets do not include precise ground truth reconstruction from TLS and rely on ground truth trajectories from other sensors such GNSS-RTK (Geiger et al. 2013).

In addition to geometric reconstruction, colour reconstruction is becoming more important with the advances of

[1]Oxford Robotics Inst., Dept. of Eng. Science, Univ. of Oxford, UK
[2]Group of Automation, Robotics and Computer Vision (AUROVA), University of Alicante, Spain

Corresponding author:
Yifu Tao, Oxford Robotics Inst., Dept. of Eng. Science, Univ. of Oxford OX1 3PJ, UK.
Email: oxfordspiresdataset@robots.ox.ac.uk



(a) Christ Church College **(b)** Keble College **(c)** Radcliffe Obs. Quarter **(d)** Blenheim Palace

Figure 1. *Top*: Point cloud of the Radcliffe Camera and the Bodleian Library captured by the TLS. Note the scale bar in the bottom-right corner. *Bottom*: Views of the TLS maps from other sites. Each column is a different site. The upper images show views of colour point clouds while the lower images were taken by the scanner's cameras.

radiance field methods including Neural Radiance Fields (NeRF) (Mildenhall et al. 2021) and 3D Gaussian Splatting (Kerbl et al. 2023). These methods take as input calibrated camera images and their precise 3D poses (typically estimated using SfM), and output a dense 3D field with volume density (similar to differential opacity) and view-dependent colour. The output radiance field can be used to synthesise photorealistic images using volume rendering techniques. Since radiance field methods are capable of representing complex geometry and appearance, some SLAM systems have adopted it as their underlying 3D map representation (Sucar et al. 2021; Zhu et al. 2022).

Despite the rapid development of radiance field methods, their use in outdoor mobile robot perception has been less well explored. Radiance field methods are often evaluated by the quality of images rendered using datasets where the image set point at a single object observed in controlled lighting conditions, and often indoors. For a mobile robot operating in an outdoor environment, the trajectory is usually not object-centric, and viewpoints are relatively sparse compared to the size of the scene. Inferring 3D structure from monocular images alone is more challenging if provided with fewer viewpoint constraints, and this can lead to artefacts (e.g. the elongated Gaussians along the viewing direction mentioned in Matsuki et al. (2024)) that are not noticeable if only evaluated from nearby poses. In addition, radiance field methods can also generate "floater" artefacts to overfit per-frame lighting conditions (as discussed in Tancik et al. (2023)) and the texture of the sky. Both are common challenges in outdoor environments. Such artefacts can lead to inferior 3D reconstruction and poorer photo-realistic rendering from a pose that is far from the training sequence. To develop radiance field methods that can be integrated with outdoor SLAM systems, it is crucial to have a dataset with colour images, LiDAR and accurate ground truth trajectory and reconstruction.

In this work, we introduce the Oxford Spires Dataset, a large-scale dataset collected across six historical landmarks in Oxford, UK, covering more than $20\,000\,\mathrm{m}^2$ per site. The total area recorded in this dataset is more than $125\,000\,\mathrm{m}^2$,

about the size of a small town. It provides high-resolution RGB image streams from three cameras, 3D wide Field-of-View LiDAR data, and inertial data from a mobile handheld device. It is accompanied by millimetre-accurate reference scans which serve as the reference ground truth 3D model for reconstruction systems. We also use it to determine the ground truth trajectories of the handheld device. The setup of three colour cameras facing forward, left and to the right is a particularly novel characteristic as similar datasets contain TLS-based ground truth, LiDAR scans but only forward-facing colour camera(s) (Wei et al. 2024; Nguyen et al. 2024; Liu et al. 2024). The side-facing cameras provide increased Field-of-View that not only increases the texture mapping coverage but also provides view constraints which are crucial for vision-only systems to infer the 3D structure. The three-camera setup also makes textured mapping more tractable from a simple linear pass through an environment, rather than requiring exhaustive scanning. This makes our dataset suitable for evaluating radiance field methods in outdoor mobile robotics and reality capture contexts. Leveraging this rich combination of sensor data, we also introduce three benchmarks for localisation, 3D reconstruction, and novel-view synthesis. We use the benchmark to evaluate state-of-the-art SLAM systems, SfM-MVS systems and radiance field systems. In particular, the novel-view synthesis benchmark features test data not only sampled from a single reference trajectory, but also from other sequences where the device travelled in an opposing direction and along a trajectory far from the reference. The evaluation results of state-of-the-art radiance field methods highlight the problem of overfitting to the training data and an inability to generalise to distant viewpoints. Our dataset opens up new research avenues in this space. We release the raw sensor data as well as processed data including example outputs from a LiDAR SLAM system (such as motion-undistorted point clouds) and a SfM system (which can be used for MVS, NeRF and 3D Gaussian Splatting), as well as ground truth trajectories and reconstruction. Software for parsing the data and evaluating the systems presented in the three benchmarks is also made available.

In summary, our main contributions are as follows:

- A large-scale outdoor dataset collected at six historical sites, covering an average area of about $10\,000\,\mathrm{m}^2$ each. In total, 24 sequences were recorded, with the average distance travelled in each sequence exceeding 400 metres.
- The dataset is collected with a sensor suite comprising three 1.6 megapixel global shutter fisheye RGB cameras, a wide Field-of-View 64 beam 3D LiDAR, and inertial data, paired with millimetre-accurate reference 3D models captured using a TLS.
- We provide precise calibrations for the synchronised sensors — including 3 fisheye cameras, IMU and the 104° FoV LiDAR.
- Three benchmarks for localisation, reconstruction and novel-view synthesis with ground truth generated using the 3D models from the TLS. In this paper, we evaluated state-of-the-art SLAM, SfM, MVS, NeRF and 3D Gaussian Splatting methods for each.
- Evaluation software is released for using the dataset and benchmarking methods.

2 Related Work

In this section, we overview related datasets that are available for evaluating localisation, 3D reconstruction and novel-view synthesis. A summary of these datasets is presented in Tab. 1.

2.1 Datasets for Evaluating Localisation

Localisation is a key task in robotics and computer vision, and is performed by methods including odometry, SLAM, SfM, place recognition or relocalisation in a prior map. In indoor environments, cameras and RGB-D sensors are commonly used. TUM RGB-D (Sturm et al. 2012) is one of the first benchmarks which sought to evaluate localisation performance using ground truth trajectories. While for visual-inertial SLAM systems, EuROC (Burri et al. 2016) and TUM VI (Schubert et al. 2018) are popular datasets used in the research community. For outdoor environments, LiDAR is a common sensor modality and has been used in robotics datasets such as New College (Smith et al. 2009) and NCLT (Carlevaris-Bianco et al. 2016). Other datasets focus on evaluating odometry and SLAM trajectories in the context of autonomous driving, including KITTI (Geiger et al. 2013), Complex Urban (Jeong et al. 2019), and WoodScape (Yogamani et al. 2019). Several datasets including RELLIS-3D (Jiang et al. 2021), Botanic Garden (Liu et al. 2024), and WHU-Helmet (Li et al. 2023) focus on unstructured natural environments (forest and rural areas). Robot platforms were used to collect datasets including quadrupeds (Brizi et al. 2024; Wei et al. 2024) and aerial robots (Li et al. 2024).

To evaluate the accuracy of localisation systems, a precise ground truth trajectory is essential. For self-driving datasets, ground truth trajectories are often obtained by fusing GNSS data with inertial and LiDAR data (Geiger et al. 2013). One limitation of GNSS-based ground truth is that it is not reliable in areas such as urban canyons. Motion capture systems can also be used to obtain ground truth trajectories (Helmberger et al. 2022), although they are often limited to indoor environments. In outdoor environments, Newer College (Ramezani et al. 2020b) generate a centimetre-accurate ground truth by registering LiDAR scans against an accurate prior map obtained using TLS. The approach of registering mobile lidar scans to an accurate prior map was also adopted by the authors of BotanicGarden (Liu et al. 2024) and MCD (Nguyen et al. 2024). Hilti-Oxford (Zhang et al. 2022) is notable in achieving millimetre accuracy ground truth for a sample set of stationary poses by using reference targets. Our work follows the approach used in Newer College (Ramezani et al. 2020b) to generate dense ground truth trajectories.

2.2 Datasets for Evaluating 3D Reconstruction

SLAM systems estimate both a robot/sensor trajectory and the map of their environment; however, many SLAM datasets only provide ground truth trajectory. Few datasets evaluate the accuracy of the map reconstruction, because accurate ground truth reconstructions are costly and laborious to obtain. Because of this, some SLAM datasets such as ICL-NUIM (Handa et al. 2014) create ground truth 3D models using simulation, and other datasets including Matterport3D (Chang et al. 2017) and ScanNet (Dai et al.

Table 1. Summary of related datasets for testing localisation and reconstruction methods. Oxford Spires is a large-scale outdoor SLAM dataset with three colour camera images, LiDAR as well as ground truth trajectories and 3D models. It can be used to evaluate tasks including localisation, reconstruction and novel-view synthesis. Features in other related datasets that are not suitable to our target domain (outdoor SLAM with colour reconstruction) are coloured with a grey background. These features include indoor scenes, greyscale camera images, short range (<10 m) depth sensing, and imprecise or missing ground truth 3D models.

Dataset	Scene	Camera			Depth Sensor	Ground Truth	
		Colour	Shutter	Resolution		Trajectory	3D Model
New College (Smith et al. 2009)	Outdoor	RGB/Grey	-	0.2 MP	2× LM2 291-S14	-	-
TUM-RGBD (Sturm et al. 2012)	Indoor	RGB	Rolling	0.3 MP	MS Kinect RGB-D	MoCap	-
KITTI (Geiger et al. 2013)	Outdoor	RGB/Grey	Global	1.4 MP	Velodyne HDL-64E	GPS RTK	-
NCLT (Carlevaris-Bianco et al. 2016)	Outdoor&Indoor	RGB	Global	1.9 MP	Velodyne HDL-32E	GPS RTK	-
EuROC (Burri et al. 2016)	Indoor	Greyscale	Global	0.4 MP	-	Leica&Vicon	Leica MS50
DTU (Aanæs et al. 2016)	Indoor	RGB	-	1.9 MP	-	Robot Arm	Structured light scanner
ScanNet (Dai et al. 2017)	Indoor	RGB	Rolling	1.3 MP	Structure Sensor RGB-D	RGB-D SLAM	RGB-D SLAM
ETH3D (Schops et al. 2017)	Outdoor&Indoor	RGB/Grey	Global	0.4/24 MP	-	COLMAP+ICP	FARO Focus X330
Tanks and Temples (Knapitsch et al. 2017)	Outdoor&Indoor	RGB	Rolling	8 MP	-	Mutual Information	FARO Focus X330
Complex Urban (Jeong et al. 2019)	Outdoor	RGB	Global	0.7 MP	Velodyne VLP-16	GPS+LiDAR SLAM	-
WoodScape (Yogamani et al. 2019)	Outdoor	RGB	Rolling	1 MP	Velodyne HDL-64E	GNSS-IMU	-
Newer College (Ramezani et al. 2020b)	Outdoor	Greyscale	Global	0.4 MP	Ouster OS1-64	ICP	Leica BLK360
Hilti-21 (Helmberger et al. 2022)	Outdoor&Indoor	Greyscale	Global	1.3MP	Ouster OS0-64	MoCap/Total station	-
Hilti-22 (Zhang et al. 2022)	Outdoor&Indoor	Greyscale	Global	0.4 MP	Hesai XT32	ICP+Reference Target	Z+F Imager 5016
LaMAR (Sarlin et al. 2022)	Outdoor&Indoor	RGB/Grey	Both	2.0 MP	HoloLens2/iPhone RGB-D	LiDAR SLAM+SfM	LiDAR SLAM+SfM
WHU-Helmet (Li et al. 2023)	Outdoor&Indoor	RGB	-	-	Livox Avia	LiDAR SLAM	LiDAR SLAM
RELLIS-3D (Jiang et al. 2021)	Outdoor	RGB	Global	2.3 MP	VLP-16&OS1-64	LiDAR SLAM	LiDAR SLAM
ROVER (Schmidt et al. 2024)	Outdoor	RGB/Gray	Both	0.7 MP	-	Visual SLAM	-
Hilti-23 (Nair et al. 2024)	Indoor	Greyscale	Global	1.0 MP	Robosense BPearl	Reference Target	Trimble X7
ScanNet++ (Yeshwanth et al. 2023)	Indoor	RGB	Rolling	2.8 MP	iPhone RGB-D	COLMAP	FARO Focus Premium
Hilti-24 (Sun et al. 2023)	Indoor	-	-	-	Matterport RGB-D	ICP	Matterport RGB-D
MARS-LVIG (Li et al. 2024)	Outdoor	RGB	Global	5.0 MP	Livox Avia	GNSS RTK	LiDAR-GNSS Mapping
VBR (Brizi et al. 2024)	Outdoor&Indoor	RGB	Global	0.9 MP	Ouster OS0-128	LiDAR-GNSS	-
BotanicGarden Liu et al. (2024)	Outdoor	RGB/Grey	Global	2.3 MP	VLP16+Livox Avia	ICP	Leica RTC360
FusionPortableV2 (Wei et al. 2024)	Outdoor&Indoor	RGB	Global	0.7 MP	Ouster OS1-128	Prism-GNSS-IMU	Leica RTC/BLK360
MCD (Nguyen et al. 2024)	Outdoor&Indoor	RGB	Global	1.0 MP	Ouster OS1 +Livox Mid70	ICP	Survey-grade map
Oxford Spires (ours)	Outdoor&Indoor	RGB	Global	1.6 MP	Hesai QT64	ICP	Leica RTC360

2017) actually use the output from a RGB-D SLAM system as ground truth. Replica (Straub et al. 2019) provides higher-quality 3D meshes than ScanNet and Matterport 3D, and the rendered images are photo-realistic. The RGB-D SLAM ground truth approach cannot be adapted to outdoor scenes due to the short range of depth cameras. LaMAR (Sarlin et al. 2022) includes outdoor sequences and a ground truth 3D model from a combination of VIO, SLAM and SfM. The ground truth 3D model obtained with SLAM is generally not as accurate as what TLS can produce. Survey-grade TLS achieves millimetre-level accuracy (A comparison can be found in ScanNet++ (Yeshwanth et al. 2023)).

Among the datasets that provide precise ground truth reconstruction (obtained from TLS), EuROC (Burri et al. 2016) and ScanNet++ (Yeshwanth et al. 2023) are captured from indoor environments, and hence LiDAR is not used. The only available outdoor SLAM datasets that include accurate ground truth 3D models (to the best of our knowledge) are Newer College (Ramezani et al. 2020b) and Hilti-Oxford-2022 (Zhang et al. 2022). Both datasets use relatively low-resolution greyscale cameras, and therefore are not suitable for colour 3D reconstruction. Compared to them, our dataset provides high-resolution colour images from three cameras, and is hence suitable for not only 3D reconstruction but also novel-view synthesis.

In the field of computer vision, datasets with accurate ground truth reconstruction exist for MVS research, but often they target small-scale indoor scenes. Middlebury (Seitz et al. 2006) was one of the early datasets with ground truth depth obtained using a structured light scanner. DTU (Aanæs et al. 2016) captured individual objects using a robotic arm in a controlled environment, with its ground truth also obtained using structured light scans. ETH3D (Schops et al. 2017) provides both high-resolution images (<80 per sequence) recorded by a DSLR camera, and low-resolution synchronised grey-scale images (∼1000 per sequence), with a ground truth 3D model obtained using a TLS. Tanks and Temples (Knapitsch et al. 2017) is another popular benchmark for 3D reconstruction with ground truth from TLS. It can be used to evaluate both SfM and MVS algorithms and uses a higher-quality camera for its video data.

2.3 Datasets for Evaluating Novel-view Synthesis

Radiance fields have emerged to the most promising representation for novel-view synthesis. The input images for radiance field methods are often co-registered using SfM methods such as COLMAP (Schönberger and Frahm 2016). In the original NeRF paper (Mildenhall et al. 2021), both synthetic datasets and real-world image sequences from LLFF (Mildenhall et al. 2019) were used. Subsequently Mip-NeRF 360 (Barron et al. 2022) included object-centric framed images taken in indoor and outdoor environments, and is popular in the radiance field community. Radiance field methods are typically evaluated using test set images that are sampled from an input trajectory and excluded from training. ScanNet++ (Yeshwanth et al. 2023) used a more challenging evaluation approach. The authors capture test images independently from the training sequence using a higher-quality DSLR camera in the indoor environment. In contrast, our dataset focuses on large-scale outdoor environments, and provides test images captured from

sequences with distant viewpoints. In this manner, we aim to advance the generalisation capability of existing radiance field methods.

3 Hardware

3.1 Handheld Perception Unit

Our perception unit, called Frontier, has three cameras, an IMU and a LiDAR. It is shown in Fig. 2 and Fig. 3 (right). The three colour fisheye cameras face forward, left, and right from a customised Alphasense Core Development Kit from Sevensense Robotics AG. Each camera has a Field-of-View of $126°\times92.4°$ with a resolution of 1440×1080 pixels. There is a cellphone-grade IMU in the Alphasense Core Development Kit, which is hardware-synchronised with the three cameras using an FPGA from Sevensense. The three cameras have around 36° overlap, which enables multi-camera calibration mentioned in Sec. 4.1. The cameras operate at 20 Hz, and the IMU operates at 400 Hz. Auto-exposure is enabled for the cameras to capture indoor and outdoor scenes with different lighting conditions. A 64-channel Hesai QT64 LiDAR operating at 10 Hz was mounted on top of the cameras, with a Field-of-View of 104°, and a maximum range of 60 m. The LiDAR's accuracy is ±3 cm (typical), and the relation between reflectance and intensity is $reflectance = \alpha \times intensity \times range^2$.

Multi-sensor synchronisation is achieved both on the hardware level and software level. In terms of software synchronisation, we synchronise the device clock of Alphasense Core Development Kit (with the three cameras and IMU) and the Hesai LiDAR to the host computer on the Frontier using the Precision Time Protocol (PTP) wihch achieves sub-microsecond accuracy. As for hardware synchronisation, the three cameras are synchronised with the IMU within the Alphasense Core Development Kit, and we synchronise a LiDAR point cloud with the cameras using motion undistortion. While the three cameras' exposure time are not identical for one shutter cycle due to the auto-exposure, the exposure intervals of the cameras are aligned in the middle*. This means that the timestamps of the mid-frame of the exposure interval of each camera are identical. This timestamp is used as the timestamp for the camera image. The Hesai QT64 LiDAR is a rolling shutter sensor, and the point cloud data is not triggered at a specific timestamp but obtained continuously. To obtain a synchronised LiDAR point cloud for a synchronised set of three camera images, we motion-correct a LiDAR point cloud with IMU preintegration using VILENS Wisth et al. (2023). The point cloud is undistorted to the same time as the next camera image frame. More information regarding the motion undistortion can be found in Wisth et al. (2021). In summary, each node in the SLAM pose graph has three camera images and an undistorted LiDAR point cloud with identical timestamps, which can be used for 3D reconstruction and novel-view synthesis.

3.2 Millimetre-accurate TLS

To obtain an accurate 3D reference model to benchmark localisation and reconstruction, we used a Leica RTC360 TLS (Fig. 3, left). It has a maximum range of 130 m and a Field-of-View of $360° \times 300°$. The final 3D point accuracy is 1.9 mm at 10 m and 5.3 mm at 40 m. The point clouds are coloured using 432 mega-pixel images captured by three cameras. Scans are registered in the field and re-optimised later using Leica's Cyclone REGISTER 360 Plus software. The average cloud-to-cloud error in our sites ranges from 3-7 mm. After merging all the scans, the resultant point cloud is very large (10GB), so for ease of use we downsampled the TLS scan to 1cm resolution. Nonetheless, we provide the original raw TLS scans in our dataset.

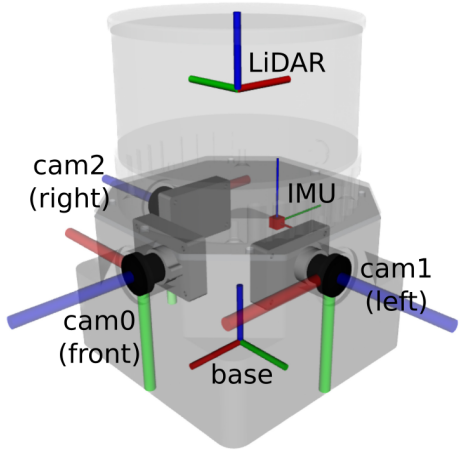


Figure 2. An isometric view of the sensor setup highlighting the coordinate frames of the cameras, the IMU and the LiDAR.



Figure 3. Leica RTC360 TLS and the Frontier device in Blenheim Palace (left) and Christ Church College (right).

4 Calibration

4.1 Multi-Camera Intrinsic and Extrinsic Parameters Calibration

Multi-camera sensor fusion requires an accurate camera projection modelling as well as accurate inter-camera extrinsic transforms. Given the wide Field-of-View and strong distortion of the fisheye camera lenses, we employ the equidistant distortion model (Kannala and Brandt 2006). Adjacent cameras in our setup share overlapping view frustums (approximately 36° horizontally), enabling the extrinsic calibration between the cameras via co-detection of known calibration target features. We calibrate both the

*https://github.com/sevensense-robotics/core_research_manual/

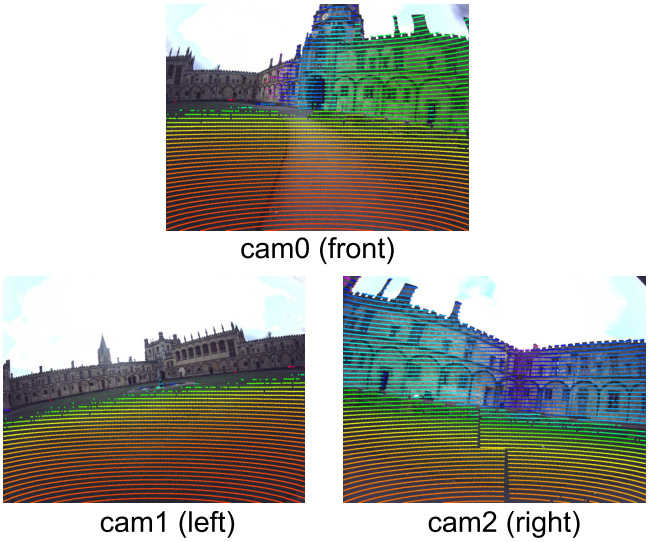
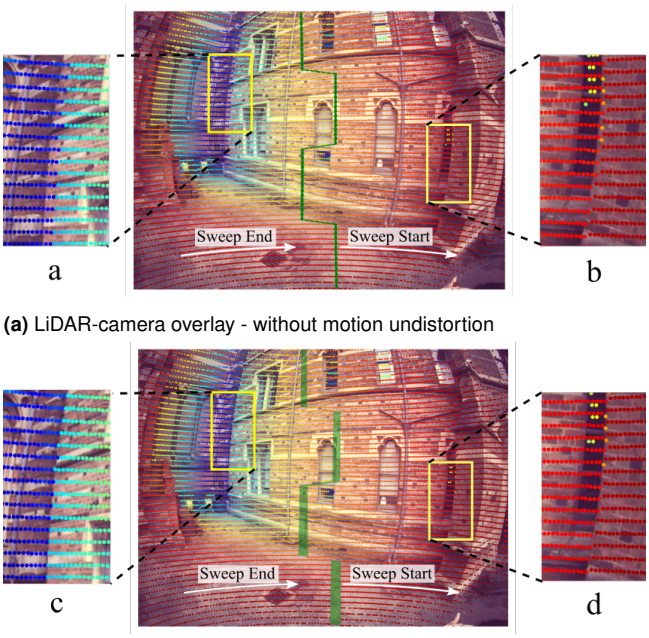


Figure 4. LiDAR point clouds overlaid on the camera images. This demonstrates the quality of camera intrinsics calibration and camera-LiDAR extrinsics calibration. In the left camera, the regions of the building without LiDAR points are due to the LiDAR's limited sensing range. Note that the motion undistortion process produces the jagged discontinuity in the LiDAR beam pattern in the right camera.



(a) LiDAR-camera overlay - without motion undistortion

(b) LiDAR-camera overlay - with motion undistortion

Figure 5. This images show LiDAR overlay on top of the right-facing camera (cam2) demonstrating the effects of accurate LiDAR ego-motion undistortion. The right camera overlaps the start and end of the LiDAR sweep (the seam shown in green). The top figure shows the unprocessed raw point cloud overlay. The LiDAR points are slightly misaligned closer to the start of the sweep (**b**), and significantly misaligned at the end of the LiDAR sweep (**a**). With IMU-only point cloud undistortion, we achieve consistent overlays (bottom figure) (**c**) and (**d**), demonstrating the accuracy of the multi-modal spatiotemporal calibration.

intrinsic and extrinsic parameters of the three cameras using the Kalibr open-source camera calibration toolbox (Rehder et al. 2016). The resultant calibration achieves sub-pixel reprojection error. We provide the camera calibration sequences with this dataset to facilitate experimentation with alternative calibration methods.

4.2 IMU Calibration

To facilitate visual and lidar odometry using the IMU, it is crucial to appropriately model the noise parameters of the IMU. For this, we measured the Allan variance parameters[†] of the IMU accelerometer and gyroscope using an eight-hour data sequence.

4.3 Camera-IMU Extrinsic Calibration

With these accurate camera intrinsic parameters and IMU noise process parameters, we then perform camera-to-IMU extrinsic calibration individually for each of the cameras using Kalibr (Rehder et al. 2016). We cross-validated the consistency of the camera-IMU calibration by measuring the variation in the estimated coordinates of the IMU, using the individual camera-IMU extrinsic parameters and the camera-camera extrinsic parameters.

4.4 Camera-LiDAR Extrinsic Calibration

Camera-LiDAR extrinsic parameters are calibrated in a bundled fashion with the inter-camera extrinsic parameters from 4.1 held constant; a single $SE(3)$ transform between the bundle of cameras and the LiDAR is calibrated. We perform this calibration using DiffCal (Fu et al. 2023). This method uses a differentiable representation of the checker pattern to align the point intensities observed by the LiDAR directly with the camera-detected checkerboard pattern. In Fig. 4, We show an example of the LiDAR point clouds overlaid on the camera images using the described calibration.

Calibration	Mean (px)	Median (px)	σ (px)
Camera/Camera			
cam0 (front)	0.22	0.21	0.14
cam1 (left)	0.23	0.22	0.15
cam2 (right)	0.22	0.21	0.14
Camera/IMU			
cam0 (front)	0.25	0.23	0.15

Table 2. Mean, median, and standard deviation of reprojection residuals for different calibration types.

5 Dataset

5.1 Data Format

The Oxford Spires Dataset consists of data collected in six sites in Oxford, UK, with multiple sequences taken at each site (Sec. 5.2). The data is originally collected as rosbags[‡]. We also provide raw sensor data (as individual files) as well as processed data. The processed data includes outputs from an example LiDAR SLAM system and a SfM system. Finally, we also provide the ground truth trajectories and reconstruction.

[†] https://github.com/ori-drs/allan_variance_ros.

[‡] We provide rosbag in ROS1 and ROS2 format

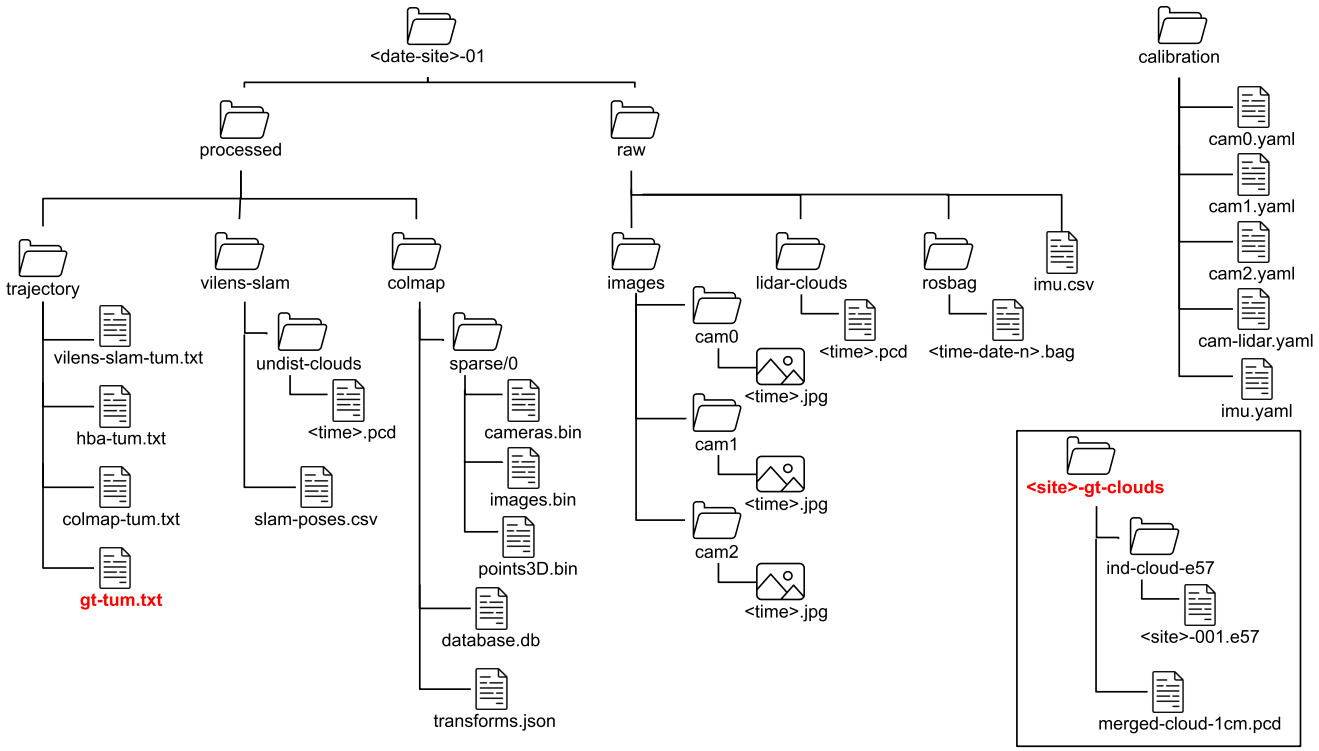


Figure 6. File structure of the Oxford Spires dataset: For *each sequence*, we provide the raw images and LiDAR point clouds, ground truth trajectory, LiDAR SLAM trajectory (including undistorted point clouds synchronised to images), and COLMAP trajectory. For *each site*, we provide TLS clouds as reconstruction ground truth. The ground truth systems are highlighted in red. We also provide calibration files for the camera intrinsics.

The following sections describe the raw data formats and the folder system which we provide for easy use of the data outside of ROS (Fig. 6).

5.1.1 Raw - Camera Images: The 20 Hz raw colour fisheye image streams from the three cameras of the Frontier are debayered and stored as 8-bit JPEG images. The three cameras are hardware-synchronised with each other, and hence the image triplets have the same timestamps. The images are stored as `<time>.jpg` under each camera folder, namely `cam0`, `cam1` and `cam2`, which correspond to the camera facing forward, left and right respectively. We debayer the raw image with bilinear interpolation, and white-balance the debayered image using the convolutional colour constancy method Barron and Tsai (2017). We provide tools to white-balance[§] the debayered images.

5.1.2 Raw - 3D LiDAR Point Clouds: 3D point clouds were collected using a Hesai QT64 LiDAR at 10Hz, and stored as `<time>.pcd`. Note that the unprocessed point clouds are raw measurements from the continuously scanning LiDAR, and the timestamp is the start time for each sweep. A subset of the LiDAR point clouds that were output by an example LiDAR SLAM system are also provided and described in Sec. 5.1.4.

5.1.3 Raw - IMU Measurements The linear acceleration and angular velocity measurements from the IMU are stored in `imu.csv`. Each row has the format of timestamp (seconds, nano-seconds), acceleration (x,y,z) and angular-velocity (x,y,z).

5.1.4 Processed - VILENS-SLAM Outputs: We provide the estimated trajectory and the motion undistorted point clouds output by LiDAR-inertial SLAM (VILENS-SLAM, Wisth et al. (2023); Ramezani et al. (2020a)). The trajectory is saved as `slam_poses.csv` with a $SE(3)$ pose estimate consisting of position (x,y,z) and quaternion (x,y,z,w) for each timestamp (seconds, nano-seconds). This data format can be directly used to test 3D reconstruction.

5.1.5 Processed - COLMAP Outputs: A solution to SfM is required as input to both MVS methods and radiance field methods (NeRF and 3D Gaussian Splatting). To facilitate researchers, we ran the state-of-the-art SfM method COLMAP (Schönberger and Frahm 2016) for each sequence and provide its outputs. Specifically, COLMAP provides camera information in `cameras.bin`, image information in `images.bin`, 3D feature points in `points3D.bin` and the database information in `database.db`.

Running SfM for all images captured at 20 Hz results in a large amount of output data and computation time. This is unnecessary because consecutive images are very close to each other and thus redundant. To keep the number of images manageable yet providing enough viewpoints for visual reconstruction, we selected images that are synchronised to the SLAM pose graph point clouds and spaced 1 m apart. We then ran COLMAP on this set of images. At walking speed, this results in a frequency of about 1 Hz. For each sequence, the total number of images was less than 2000 (for each of the three cameras). Using images aligned with corresponding LiDAR depth is also useful for methods that fuse LiDAR

[§]Tool based on https://github.com/leggedrobotics/raw_image_pipeline

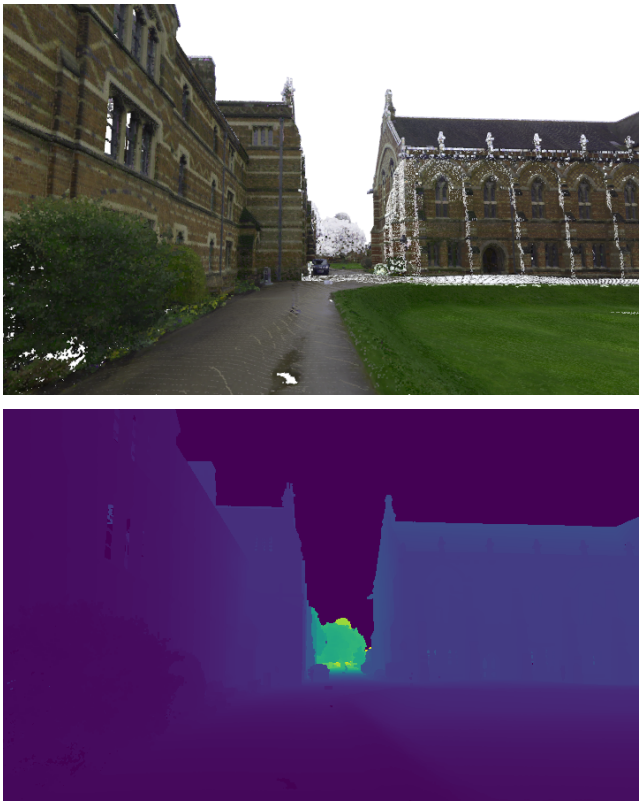


Figure 7. *Top*: Image rendered from TLS map in Keble College. *Bottom*: Depth image corresponding to the rendered image.

and vision, for example, colourising point clouds and depth-aided radiance fields such as Urban Radiance Field (Rematas et al. 2022). These aligned depth images are also provided in addition to the COLMAP outputs.

For compatibility with Nerfstudio (Tancik et al. 2023) (a popular open-sourced code base for state-of-the-art radiance field methods), we also convert the outputs from COLMAP into a `transforms.json` file. Specifically, `transforms.json` includes camera parameters (camera models, focal length, principle point, image size, distortion parameters), image file path and the corresponding $SE(3)$ pose estimate as a 4×4 transformation matrix.

Correcting the metric scale of vision-based 3D reconstructions produced MVS and radiance field methods is necessary to enable comparison to the metric ground truth. To estimate the scale, we used Umeyama's method[¶] to estimate a $Sim(3)$ transformation between the LiDAR trajectory and a COLMAP trajectory, and the results are saved in `evo_align_results.json`. We provide tools to compute the scale parameter and to rescale the MVS reconstruction and radiance field reconstruction to metric size.

5.1.6 Ground Truth - Reconstruction: We provide the registered individual TLS scans from Leica RTC360 for each site as the ground truth reconstruction. Each scan is saved as `<site-00x>.e57` under each site folder, and contains not only the point clouds but also the sensor origin, which is important in reconstruction methods such as occupancy mapping (Hornung et al. 2013). Moreover, we also provide the complete colourised TLS map at $1\,\mathrm{cm}$ resolution for each site (Fig. 1) by merging the individual RTC360 scans. The TLS scan registration reports from the Leica software are provided as `FinalizeReport.pdf` in each ground truth site folder, which specifies the number of scans, overlap, residual in the merging process, etc. Regarding the density, we tried to distribute the scans uniformly on each site.

5.1.7 Ground Truth - Localisation: The ground truth trajectory is computed by ICP registering each undistorted LiDAR point cloud (as described in Sec. 5.1.4) to the merged TLS map described in Sec. 5.1.6. To obtain the ground truth pose for each LiDAR scan, we use an offline version of VILENS (Wisth et al. 2023) with Iterative Closest Point (ICP) (Besl and McKay 1992) at its core. Running offline, we can allocate enough time to register the LiDAR to the colourised TLS map. The trajectory estimated through the described process is synchronised with cameras by the procedure described in Sec. 3.1. This is in the same manner as for Newer College (Ramezani et al. 2020b) and Hilti-2022 (Zhang et al. 2022). The accuracy of the ground truth trajectory is approximately 1-2 cm. We validated the ground truth trajectory by projecting the individual LiDAR scans into a map and comparing them to the TLS map. The trajectory is provided as `gt-tum.txt` in TUM (Sturm et al. 2012) format, with each line encoding timestamp, position (x,y,z) and quaternion (x,y,z,w).

5.1.8 Depth images from the TLS map: Additionally, we include ground truth depth images rendered from the TLS map using the ground truth sensor trajectories. In Fig. 7, we show an example of an image rendered from the Keble College site with its corresponding depth image. These images could be used for evaluating monocular depth estimation or novel view rendering.

5.2 Sequence Description

The dataset was recorded in six historic sites in Oxford, UK:

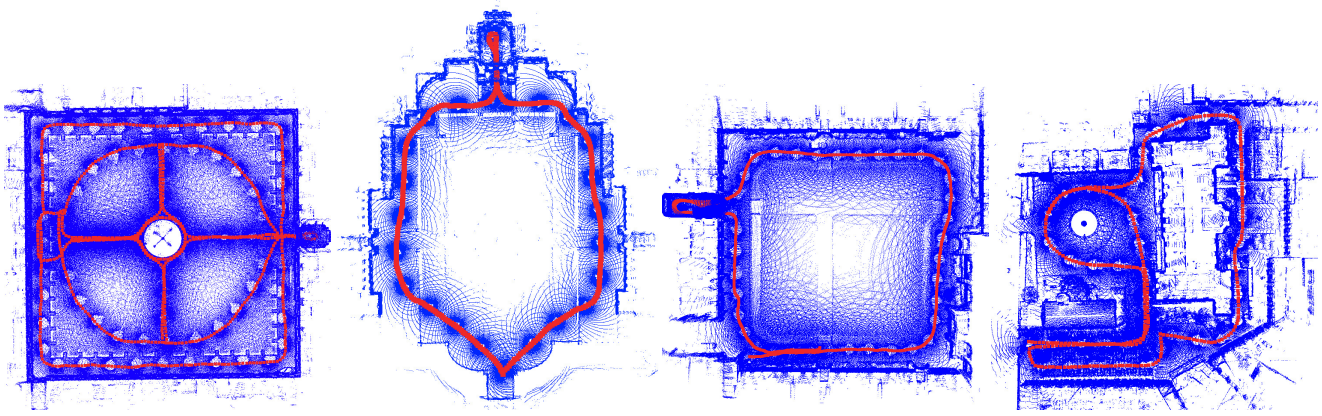
- Bodleian Library ($\sim 37\,000\,\mathrm{m}^2$)
- Blenheim Palace ($\sim 14\,000\,\mathrm{m}^2$)
- Christ Church College ($\sim 26\,000\,\mathrm{m}^2$)
- Keble College ($\sim 18\,000\,\mathrm{m}^2$)
- Radcliffe Observatory Quarter (ROQ) ($\sim 12\,000\,\mathrm{m}^2$)
- New College ($\sim 18\,000\,\mathrm{m}^2$)

Each sequence was collected by walking with the Frontier payload device mounted in a backpack as shown in Fig. 3.

5.2.1 Bodleian Library: This site consists of the area around the Bodleian Library, which includes Radcliffe Square, where the Radcliffe Camera and Oxford's University Church are located. It also reconstructs the outside of the Sheldonian Theatre and some of Broad Street. This part of the dataset contains the most iconic landmarks of the historic centre of Oxford (Fig. 1).

For this site, we provide two outdoor trajectories of walking through streets and squares around the described area. The recordings contain many details of the predominant medieval and Gothic buildings.

[¶]We used implementation from https://github.com/MichaelGrupp/evo



(a) Christ Church College **(b)** Blenheim Palace **(c)** Keble College **(d)** Radcliffe Observatory Quarter

Figure 8. Examples of SLAM trajectories (in red) and LiDAR point cloud maps (in blue) for four sequences from the dataset.

5.2.2 Blenheim Palace: This is one of England's largest houses and is notable as Sir Winston Churchill's ancestral home. Five trajectories were captured in the palace's main square, the principal hall, and rooms in the west wing, including the library. The trajectories have outdoor and indoor parts, including different-sized rooms and corridors. In Fig. 8 (b), we show an example of a sequence in the palace's main square.

5.2.3 Christ Church College: Founded in 1546, Christ Church is a constituent college of Oxford and one of the city's best-recognised locations. It contains Tom Quad, the largest square in Oxford, the college dining hall as well as Christ Church Cathedral and its cloister.

The sequences recorded in this site include outdoor areas with pavements and lawns as well as indoor parts with different lighting conditions and stairs accessing different levels, including the dining hall. One sequence is shown in Fig. 8 (a), which includes a complete loop of the perimeter of Tom Quad, which is challenging due to the limited range of the LiDAR sensor and repeating architecture.

5.2.4 Keble College: Keble is another constituent college of the University of Oxford. It comprises neo-Gothic-style buildings, including a hall and a church. Keble's buildings are distinctive because they are constructed of alternating red and white coloured bricks — which provides an interesting challenge to visual reconstruction. This differentiates it from the other sites which are mostly built from limestone.

The Keble sequences were recorded outdoors in the college's squares (Fig. 8 (c)), which includes lawns and trees, as well as some interior and exterior parts.

5.2.5 Radcliffe Observatory Quarter: The ROQ site consists of the Faculty of Philosophy, the Mathematical Institute, and St Luke's Chapel. This area is near the Oxford Robotics Institute, where the authors are affiliated. The two sequences recorded here contain squares with pavement, lawns, trees, narrow spaces between some buildings, and a fountain containing fine 3D details. In Fig. 8 (d), we show an example of a sequence through this site.

5.2.6 New College: New College is another constituent college of the University of Oxford and is located in the city's historic centre. It contains squares, a hall, a church and a cloister. We recorded four sequences in New College, including an oval lawn area at the centre of the main quad surrounded by medieval buildings. The sequences combine outdoor and indoor parts with abrupt changes in light conditions. Most of the 03 sequence was walking through the park, containing a lawn and many trees. Some parts were fully covered by tree canopies. This site corresponds to the earlier New College Dataset (Smith et al. 2009; Ramezani et al. 2020b).

6 Benchmarks and Results

In this section we describe three benchmarks we have created to demonstrate our dataset. The benchmarks compare state-of-the-art methods for localisation (Sec. 6.1), 3D reconstruction (Sec. 6.2), and novel view synthesis (Sec. 6.3).

6.1 Localisation Benchmark

In the localisation benchmark, we evaluate the trajectories estimated for each sequence using state-of-the-art LiDAR Inertial SLAM and SfM approaches:

- VILENS-SLAM: VILENS (Wisth et al. 2023) with pose graph optimisation Ramezani et al. (2020a) (online).
- Fast-LIO-SLAM (Kim et al. 2022): Fast-LIO2 (Xu et al. 2022) with pose graph optimisation and Scan Context loop closures (Kim and Kim 2018) (online).
- SC-LIO-SAM (Kim et al. 2022): LIO-SAM with Scan Context loop closures (online).
- ImMesh (Lin et al. 2023): LiDAR meshing with Fast-LIO2 odometry (online).
- HBA (Liu et al. 2023): LiDAR bundle adjustment using as input the VILENS-SLAM result (offline).
- COLMAP (Schönberger and Frahm 2016): Structure-from-Motion using only images (offline). For image matching, we used a sequential matcher with loop closure detection.

We evaluate sequences on the trajectories which were completely within the ground truth TLS map. We exclude Keble College 01, Blenheim Palace 03 and 04, Christ Church College 04 and 06, and Bodleian Library 01 from this analysis.

Table 3. Results in RMS (m) of the ATE using the provided ground truth. We mark the best results in blue using different tints. SC-LIO-SAM fails on some sequences. COLMAP gives incomplete results on some sequences.

Site	SEC.	Length (m)	VILENS-SLAM	Fast-LIO-SLAM	SC-LIO-SAM	ImMesh	HBA	COLMAP
Keble College	02	290	0.06	0.25	1.26	0.08	0.11	0.05
	03	280	0.14	0.11	4.02	0.14	0.12	0.05
	04	780	0.16	0.49	✗	3.67	0.12	0.07
	05	710	0.11	0.29	✗	0.13	0.13	0.09
Radcliffe Obs. Quarter	01	400	0.06	0.17	0.23	0.20	0.05	0.07
	02	390	0.09	0.24	0.14	0.27	0.08	0.08
Blenheim Palace	01	490	0.47	0.18	6.74	0.27	0.21	0.08
	02	390	0.16	0.12	4.41	0.36	0.08	0.05
	05	390	1.05	0.28	✗	0.22	0.14	0.26
Christ Church College	01	920	0.06	0.72	✗	0.19	0.07	0.06
	02	640	0.17	0.49	✗	1.70	0.12	0.15
	03	340	0.03	0.23	0.14	0.16	0.05	0.07
	05	820	0.17	0.30	✗	0.21	0.12	-
Bodleian Library	02	690	1.11	0.25	1.71	0.39	0.89	0.27

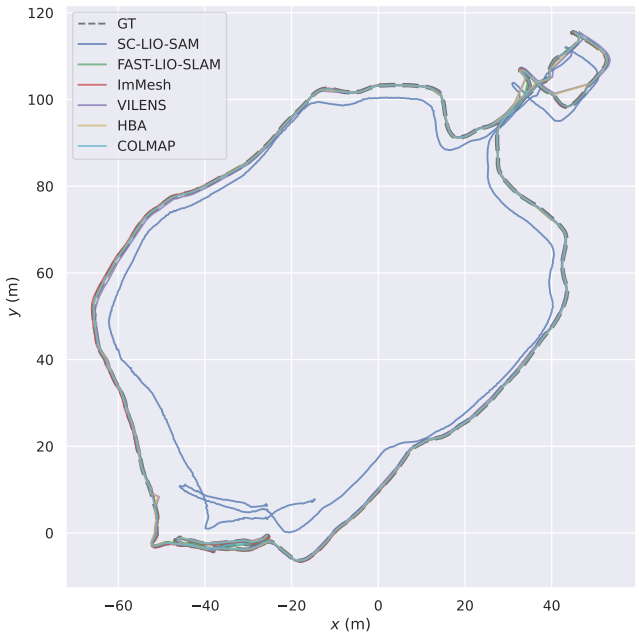


Figure 9. A top-down view showing a representative performance of the different systems for Sequence 01 at Blenheim Palace. The sequence starts and ends in the lower left. The environment where this sequence was collected can be seen in Fig. 8 (b).

6.1.1 Evaluation Metrics In the dataset tools we provide a Python script to compute Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics. We used ATE to compare the poses estimated by the SLAM and SfM methods as they should be globally consistent (Tab. 3). To transform the trajectories estimated by the methods to the ground truth coordinate frame (Sec. 5.1.7), we use the $SE(3)$ Umeyama alignment. For comparison between odometry systems, we recommend using RPE to measure local performance.

6.1.2 Experimental Results A comparison between the listed methods using the provided ground truth (Sec. 5.1.7) is presented in Tab. 3 for each sequence. We mark the best results in dark blue, while the second and third are highlighted with a lighter tint. The offline methods on the right side (HBA and COLMAP) can take advantage of all available data and provide the most accurate results. While one might expect that HBA should provide better accuracy (as a LiDAR bundle adjustment approach), we achieved the best performance with COLMAP. We believe this is due to the overlapping multicamera configuration, which provides abundant view constraints and allows features to be seen from different perspectives with proper light conditions in general. Furthermore, the prior map we provide to HBA is imperfect which means that the undistorted point clouds can be imperfect, which would then lead in turn to inaccuracies in HBA. However, note also that COLMAP was unable to solve some sequences (with poor lighting) at all and instead produces multiple disconnected sub-models. This is usually due to insufficient visual features being matched in the area where two sub-models ought to connect, which can be due to there being insufficient visual overlap or there being challenging lighting conditions (e.g., The dining hall in Christ Church College is relatively dark). In comparison, the LiDAR SLAM systems are invariant to lighting conditions and the present visual features.

The second most accurate method is HBA, which further refines VILENS-SLAM's trajectory estimation with LiDAR bundle adjustment in post-processing. Of the online methods, VILENS-SLAM gave the best performance. ImMesh and Fast-LIO-SLAM use Fast-LIO2 (Xu et al. 2022) as their core odometry module, which achieved accurate trajectory estimation. SC-Fast-LIO performs loop closure correction using Scan Context (Kim and Kim 2018), While ImMesh can recover from drifts while generating the mesh map. SC-LIO-SAM produces satisfactory results on some sequences using Scan Context (Kim and Kim 2018) as an appearance-based place recognition module. However, it adds incorrect loop closures in sequences with large loops and repeated building patterns, such as Christ Church College and Blenheim Palace. We note that these methods could potentially perform better with further parameter tuning.

In Fig. 9, we show a representative example of the performance of the evaluated methods using Sequence 01 of Blenheim Palace. All of the methods produce an accurate trajectory except for SC-LIO-SAM, which incorporates an incorrect loop closure when closing the large loop.

Table 4. Quantitative evaluation of the 3D reconstructions from VILENS-SLAM, OpenMVS and Nerfacto. We indicate the best results with a dark blue background.

Site	SEC.	Method	Accuracy↓	Completeness↓	5cm Precision	5cm Recall	5cm F-score	10cm Precision	10cm Recall	10cm F-score
Blenheim Palace	05	VILENS-SLAM	0.070	0.506	0.670	0.392	0.495	0.867	0.661	0.750
		OpenMVS	0.126	1.045	0.451	0.251	0.323	0.574	0.381	0.458
		Nerfacto	0.302	0.676	0.232	0.094	0.134	0.388	0.257	0.309
Christ Church College	02	VILENS-SLAM	0.082	3.296	0.540	0.250	0.342	0.794	0.408	0.539
		OpenMVS	0.046	5.381	0.771	0.201	0.319	0.886	0.266	0.410
		Nerfacto	0.219	4.435	0.328	0.157	0.212	0.532	0.254	0.343
Keble College	04	VILENS-SLAM	0.067	0.342	0.527	0.527	0.527	0.816	0.779	0.797
		OpenMVS	0.050	0.409	0.766	0.606	0.677	0.918	0.718	0.806
		Nerfacto	0.137	0.150	0.418	0.484	0.449	0.654	0.709	0.680
Radcliffe Obs. Quarter	01	VILENS-SLAM	0.047	0.233	0.708	0.536	0.610	0.909	0.806	0.854
		OpenMVS	0.048	0.622	0.745	0.470	0.577	0.902	0.618	0.734
		Nerfacto	0.197	0.398	0.415	0.395	0.405	0.587	0.598	0.592

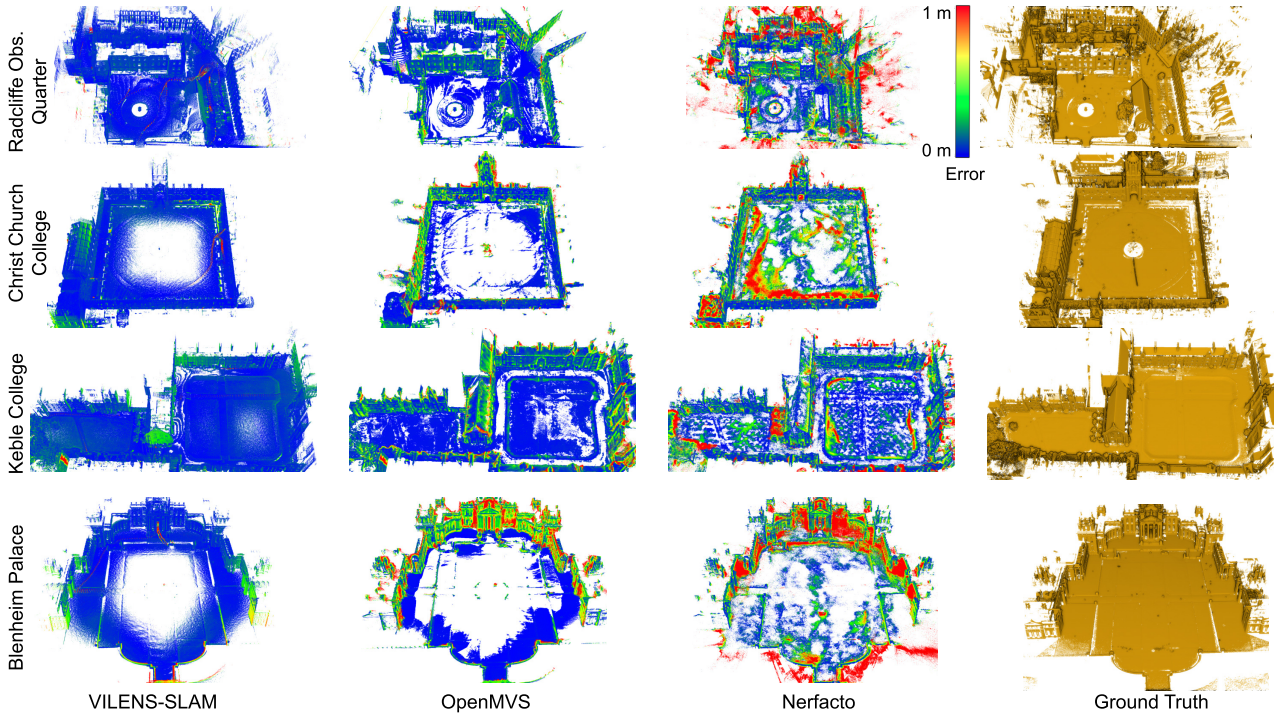


Figure 10. Comparison between the reconstructions achieved by the different methods. The reconstructions in the first three columns are coloured by point-to-point distance to the ground truth model.

6.2 3D Reconstruction Benchmark

The reconstruction benchmark evaluates outputs from the systems that use vision or LiDAR. Specifically, we evaluate the following systems:

- VILENS-SLAM: Merged LiDAR point clouds using poses obtained using the LiDAR SLAM system described in Sec. 6.1 (online).
- OpenMVS[‖]: an MVS system which uses input from COLMAP (Schönberger and Frahm 2016) (offline).
- Nerfacto: The default and recommended method from Nerfstudio[**] (Tancik et al. 2023) that combines features from MipNeRF-360 (Barron et al. 2022), Instant-NGP (Müller et al. 2022) and others. It uses input from COLMAP, as with OpenMVS (offline).

The outputs from each system are all in the form of 3D point clouds. For Nerfacto, the point cloud is generated from the trained model by calculating the expected depth and colour for the training rays, and projecting the depth points into 3D.

We selected example trajectories that are completely within the ground truth reconstruction from Blenheim Palace, Christ Church College, Keble College and Radcliffe Observatory Quarter.

6.2.1 Evaluation Metrics: We use the F-score as the primary metric for reconstruction. The F-score is calculated as the harmonic mean of precision and recall, thus it considers both aspects of the reconstruction: accuracy and completeness. To calculate precision and recall, we consider a point to be a true positive (TP) if the distance from it to

[‖] https://cdcseacave.github.io/openMVS

[**] We used version 1.1.4

the closest ground truth point is within a certain threshold. We report results using $5\,\mathrm{cm}$ and $10\,\mathrm{cm}$ thresholds. False positives (FP) are reconstructions that are further from the ground truth and thus inaccurate. False negatives (FN) are regions in the ground truth that have no neighbouring points in the reconstruction, and are thus incomplete. Specifically, precision and recall are defined by

$$\text{Precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

The F-score is then calculated as

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

We also report the point-to-point distances to measure accuracy and completeness following the conventions of the DTU dataset (Aanæs et al. 2016). Specifically, accuracy is measured as the point-to-point distance from the reconstruction to the ground truth map and measures the reconstruction quality (the lower, the better). Completeness is the distance from the point-wise reference map to the reconstruction and indicates how much of the ground truth surface has been captured by the reconstruction (the lower, the better). However, these metrics are more sensitive to outliers and non-overlapping regions, which may skew the results in practice. Thus, we do not use them as the primary metrics.

6.2.2 Reconstruction Filtering: In practice, the reconstruction and ground truth reference models will contain regions which were not mutually scanned. If unaccounted for, this would lead to erroneous false positives and false negatives. In turn, this would result in precision and recall measures which do not reflect the true quality of the reconstruction. For fairer comparison, we filter out points in the reconstruction that fall outside the reconstructed ground truth region, i.e. the regions not reconstructed in the ground truth model. In particular, for Nerfacto the sky must be specifically removed because, as a dense representation, it attempts to reconstruct it using available depth cues. We filter these sky point clouds so that its results focus on the reconstruction of the physical environment itself.

6.2.3 Experimental Results: The quantitative evaluation of the reconstructions are presented in Tab. 4, and the qualitative results are shown in Fig. 10. VILENS-SLAM reconstruction achieves the best F-score in most experiments except Keble-04. This is a reasonable result given LiDAR's accurate depth measurements. The remaining inaccuracy mostly comes from trajectory errors in long sequences and the presence of dynamic objects. The reconstruction completeness is limited by the short sensor range in Christ Church College and Blenheim where the central region of the large squares is not reconstructed.

Reconstructions from OpenMVS are accurate in regions with abundant view constraints and distinct texture, but it is not able to reconstruct surfaces with uniform texture such as the ground in Blenheim Palace and the lawn in Christ Church College. The error distribution in MVS cloud is not uniform and tends to appear at surface boundaries where occlusion is an issue.

Although both OpenMVS and Nerfacto are purely vision-based reconstruction methods, Nerfacto point clouds are generally less precise. This is because MVS filters uncertain points (by checking photo-consistency), but the NeRF approach instead optimises a continuous radiance field without an explicit notion of uncertainty. For regions with insufficient view constraints and uniform texture, Nerfacto estimates incorrect depth values which leads to uneven ground reconstructions. In comparison, OpenMVS filters some of the reconstruction there, which leads to better precision and accuracy.

The reconstruction quality is determined not only by the reconstruction method, but also by the accuracy of the input trajectory. Both precision and recall can be affected by an imperfect trajectory estimation. In clouds produced by VILENS-SLAM contain surfaces with high error that are the result of incorrectly registered LiDAR scans. Meanwhile, for Christ Church College, both the OpenMVS and Nerfacto reconstructions do not contain the dining hall (bottom left in the corresponding reconstructions from Fig. 10). This is because the dining hall could not be registered with the outdoor square by COLMAP (partly due to the poor lighting conditions as explained in 6.1.2).

6.3 Novel View Synthesis

We evaluate the quality of novel-view synthesis using the radiance field methods. Specifically, we evaluate:

- Nerfacto (Tancik et al. 2023) which is described in Sec. 6.2.
- Splatfacto (Ye et al. 2024), an implementation of 3D Gaussian Splatting (Kerbl et al. 2023) with quality comparable to the original implementation.

We also include results using the above methods with increased representation capability, namely Nerfacto-big (Nerfacto with larger hash grid size and proposal network size, and more ray samples) and Splatfacto-big (Splatfacto with lower thresholds for densifying and culling 3D Gaussians, which results in more Gaussians being used). All methods are trained for 5000 iterations. We select one in every ten images as the in-sequence evaluation images.

6.3.1 Evaluation Metrics: We measure the quality of the rendered images using the Peak Signal-to-noise Ratio (PSNR), Structural Similarity (SSIM) (Wang et al. 2004) and Learned Perceptual Image Patch Similarity (LPIPS) (Zhang et al. 2018) metrics, as commonly used in the literature (Mildenhall et al. 2021; Barron et al. 2022; Tancik et al. 2023).

6.3.2 Out-of-Sequence Novel View Synthesis: When evaluating radiance field methods, methods often use test poses that are close to the training poses. This is typically because the test poses and training poses are sampled from a common input trajectory. In downstream applications, the ability to render photorealistic images from viewpoints that are quite different from the training poses is crucial. To facilitate research in this direction, we generate challenging test sets whose viewpoints are very different from the

Table 5. Quantitative evaluation of Novel View Synthesis. The best results are coloured in blue using different tints. The test images are selected from the input trajectory (In-Sequence) as well as a separate trajectory with viewpoints far from the input trajectory (Out-of-Sequence).

Sequence	Method	In-Sequence			Out-of-Sequence		
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Observatory Quarter	Nerfacto	23.40	0.807	0.336	21.25	0.786	0.370
	Nerfacto-big	20.66	0.807	0.292	19.38	0.787	0.317
	Splatfacto	22.76	0.791	0.373	19.47	0.736	0.445
	Splatfacto-big	23.54	0.811	0.347	20.26	0.761	0.413
Blenheim Palace	Nerfacto	18.42	0.716	0.506	17.09	0.682	0.537
	Nerfacto-big	17.93	0.724	0.445	17.09	0.695	0.493
	Splatfacto	19.34	0.726	0.589	16.02	0.668	0.659
	Splatfacto-big	19.77	0.733	0.576	16.20	0.671	0.643
Keble College	Nerfacto	21.10	0.731	0.397	20.29	0.748	0.368
	Nerfacto-big	19.71	0.749	0.326	18.15	0.736	0.381
	Splatfacto	20.47	0.651	0.514	19.92	0.658	0.500
	Splatfacto-big	21.36	0.688	0.478	20.86	0.707	0.434

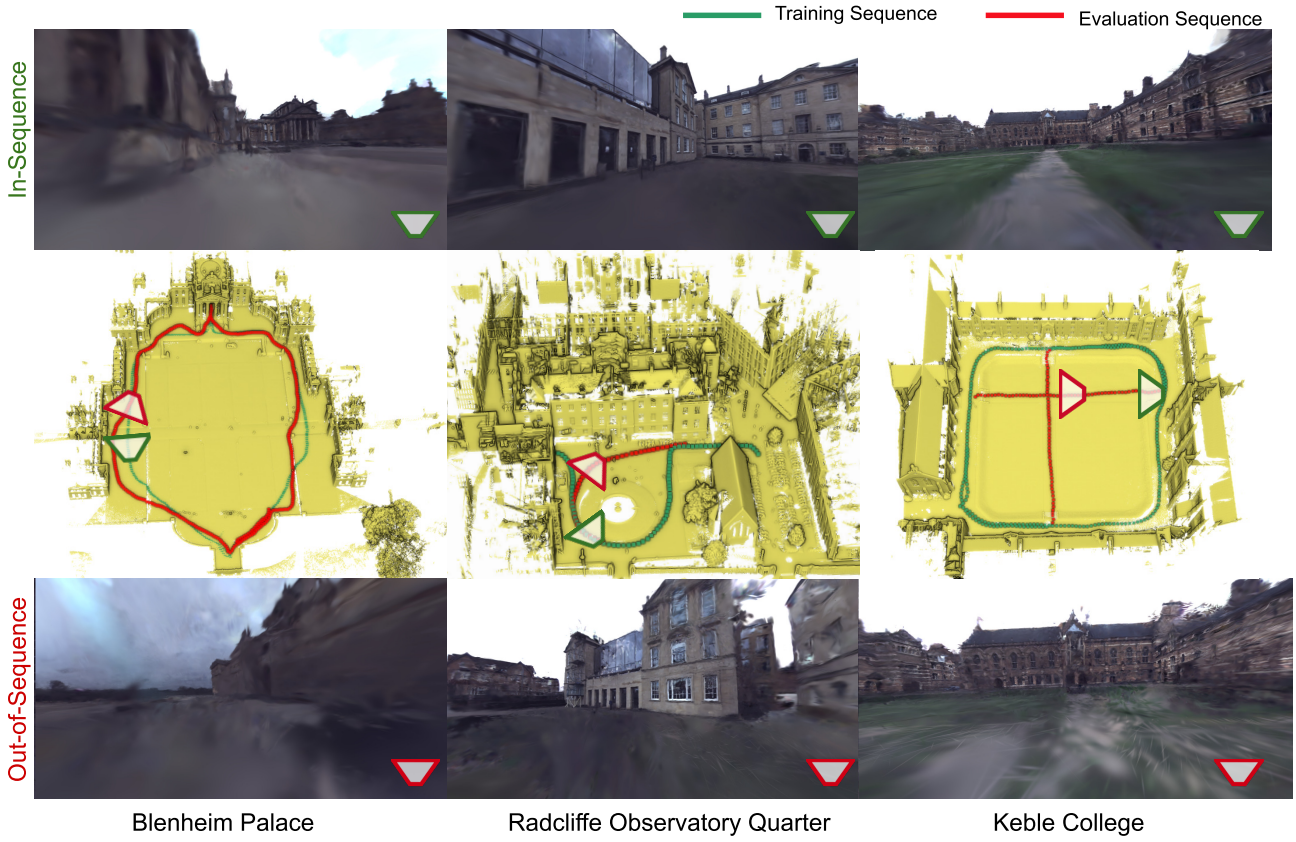


Figure 11. Illustrative results of Splatfacto-big when evaluated using in-sequence (green) and out-of-sequence (red) trajectories. When the rendering viewpoint is quite different from the training trajectory, the rendered images exhibit many more artefacts. The in-sequence and out-of-sequence trajectories in Radcliffe Observatory Quarter and Blenheim Palace are in different directions, while the trajectories in Keble College have similar viewing directions but are from distant positions. From our test, we found that Splatfacto-big generates more visual artefacts than Nerfacto-big.

training sets. Specifically, we merged images from different sequences taken in the same site using COLMAP. Then, we manually selected training and test set images that are far away apart or have very different view directions. We describe the images that are selected from the input trajectory as "in-sequence" and images from a separate trajectory with different viewpoints as "out-of-sequence".

6.3.3 Experimental Results: We present quantitative results in Tab. 5. Of particular interest, one can see that the quality of novel view synthesis falls significantly when moving from the in-sequence trajectory to the out-of-sequence trajectory. Compared to Nerfacto, Splatfacto (and its big version) generalise worse in the out-of-sequence setting, and we show qualitative results of Splatfacto-big in Fig. 11. The generalisation issue is particularly evident in Radcliffe Observatory Quarter and Keble College where the renderings are almost photorealistic from an angle close to the training data, but exhibits severe artefacts when rendered from a different location. Some of the artefacts have the

wrong 3D geometry, and a typical issue is there being elongated 3D Gaussians along the training view angles as mentioned in Matsuki et al. (2024). Other artefacts, such as the black artefact on the ground from Radcliffe Observatory Quarter, are due to the modelling of view-dependent colour used in radiance field methods. View-dependent colour is commonly modelled by a neural network Mildenhall et al. (2021) or spherical harmonics (Kerbl et al. 2023). When the training viewing angles are limited (which is common in robotics applications), the optimised neural network or spherical harmonics can be overfit which leads to unexpected colours when rendering from a novel viewing angle. This is a limitation of state-of-the-art radiance fields method which is under-explored in the literature.

For the methods we tested, we found them all to be are capable of generating reasonably photo-realistic images when rendering from in-sequence poses. A key difference between Nerfacto and Splatfacto is the rendering speed at test time: Both Splatfacto and Splatfacto-big render at 3.5 Hz on average, while Nerfacto renders at 1.25 Hz and Nerfacto-big at 0.57 Hz. When using the "big" version for Nerfacto and Splatfacto, the rendering quality is generally better with LPIPS increased by 9.6% and SSIM by 2% on average. This improvement is not always reflected in the PSNR measure, because it is also affected the per-frame appearance difference (e.g. lighting) (Martin-Brualla et al. 2021). For this reason, we give more consideration to changes in LPIPS and SSIM.

7 CONCLUSIONS

We present a large-scale dataset with colour images and LiDAR scans paired with high-quality ground truth 3D models and sensor trajectories. We demonstrate that the dataset is suitable for evaluating a variety of tasks in robotics and computer vision including LiDAR SLAM, Structure-from-Motion, Multi-View Stereo, Neural Radiance Field and 3D Gaussian Splatting. The scale of the provided data sequences and the quality of the ground truth trajectory and reconstruction makes it suitable for evaluating large-scale localisation and 3D reconstruction methods in a outdoor environment. In addition, the colour cameras used in our dataset make it suitable for evaluating radiance field approaches, and encourages the development of SLAM systems integrated with radiance field representations. In particular, we demonstrate that state-of-the-art radiance field methods require further development to be applicable in the robotics context, namely inaccurate 3D geometry and limited generalisation capability when tested with poses distant from the training sequence.

Acknowledgements

The authors would like to thank Tobit Flatscher for helping with the GPS sensor, Prof. Ayoung Kim, Matias Mattamala and Christina Kassab for discussions and proofreading, Haedam Oh and Jianeng Wang for helping with dataset collection and calibration, Dongjae Lee for helping with post-processing.

Declaration of Conflicting Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This project has been partly funded by the Horizon Europe project Digiforest (101070405) and the Korean National Research Foundation project Harmonic Space (R94525). Maurice Fallon is supported by a Royal Society University Research Fellowship (URF\R\221016). Miguel Ángel Muñoz-Bañón is supported by the Valencian Community Government and the European Union through the CIBEST/2023/44 fellowship and PROMETEO/2021/075 project.

References

Aanæs H, Jensen RR, Vogiatzis G, Tola E and Dahl AB (2016) Large-scale data for multiple-view stereopsis. *Intl. J. of Computer Vision* 120: 153–168.

Barron JT, Mildenhall B, Verbin D, Srinivasan PP and Hedman P (2022) Mip-NeRF 360: Unbounded anti-aliased neural radiance fields. *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)* : 5470–5479.

Barron JT and Tsai YT (2017) Fast fourier color constancy. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 886–894.

Besl PJ and McKay ND (1992) Method for registration of 3-d shapes. In: *Sensor fusion IV: control paradigms and data structures*, volume 1611. Spie, pp. 586–606.

Brizi L, Giacomini E, Di Giammarino L, Ferrari S, Salem O, De Rebotti L and Grisetti G (2024) VBR: A vision benchmark in Rome. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*.

Burri M, Nikolic J, Gohl P, Schneider T, Rehder J, Omari S, Achtelik MW and Siegwart R (2016) The EuRoC micro aerial vehicle datasets. *Intl. J. of Robotics Research* 35(10): 1157–1163.

Carlevaris-Bianco N, Ushani AK and Eustice RM (2016) University of Michigan North Campus long-term vision and lidar dataset. *Intl. J. of Robotics Research* 35(9): 1023–1035.

Chang A, Dai A, Funkhouser T, Halber M, Niessner M, Savva M, Song S, Zeng A and Zhang Y (2017) Matterport3D: Learning from RGB-D data in indoor environments. In: *IEEE Intl. Conf. on 3D Vision*. pp. 667–676.

Dai A, Chang AX, Savva M, Halber M, Funkhouser T and Nießner M (2017) ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 5828–5839.

Fu LFT, Chebrolu N and Fallon M (2023) Extrinsic calibration of camera to LIDAR using a differentiable checkerboard model. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 1825–1831.

Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets robotics: The KITTI dataset. *Intl. J. of Robotics Research* 32(11): 1231–1237.

Handa A, Whelan T, McDonald J and Davison AJ (2014) A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. IEEE, pp. 1524–1531.

Helmberger M, Morin K, Berner B, Kumar N, Cioffi G and Scaramuzza D (2022) The Hilti SLAM challenge dataset. *IEEE Robotics and Automation Letters* 7(3): 7518–7525.

Hornung A, Wurm KM, Bennewitz M, Stachniss C and Burgard W (2013) OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots* 34: 189–206.

Jeong J, Cho Y, Shin YS, Roh H and Kim A (2019) Complex urban dataset with multi-level sensors from highly diverse urban environments. *Intl. J. of Robotics Research* 38(6): 642–657.

Jiang P, Osteen P, Wigness M and Saripalli S (2021) Rellis-3d dataset: Data, benchmarks and analysis. In: *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, pp. 1110–1116.

Kannala J and Brandt S (2006) A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Trans. Pattern Anal. Machine Intell.* 28(8): 1335–1340.

Kerbl B, Kopanas G, Leimkühler T and Drettakis G (2023) 3D Gaussian Splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)* 42(4): 1–14.

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

Kim G, Yun S, Kim J and Kim A (2022) SC-LiDAR-SLAM: a front-end agnostic versatile LiDAR SLAM system. In: *International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, pp. 1–6.

Knapitsch A, Park J, Zhou QY and Koltun V (2017) Tanks and Temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (TOG)* 36(4): 1–13.

Li H, Zou Y, Chen N, Lin J, Liu X, Xu W, Zheng C, Li R, He D, Kong F, Cai Y, Liu Z, Zhou S, Xue K and Zhang F (2024) MARS-LVIG dataset: A multi-sensor aerial robots SLAM dataset for LiDAR-visual-inertial-GNSS fusion. *Intl. J. of Robotics Research* 43(8): 1114–1127.

Li J, Wu W, Yang B, Zou X, Yang Y, Zhao X and Dong Z (2023) Whu-helmet: a helmet-based multisensor slam dataset for the evaluation of real-time 3-d mapping in large-scale gnss-denied environments. *IEEE Transactions on Geoscience and Remote Sensing* 61: 1–16.

Lin J, Yuan C, Cai Y, Li H, Ren Y, Zou Y, Hong X and Zhang F (2023) ImMesh: An immediate LiDAR localization and meshing framework. *IEEE Trans. Robotics* 39(6): 4312–4331.

Liu X, Liu Z, Kong F and Zhang F (2023) Large-scale LiDAR consistent mapping using hierarchical LiDAR bundle adjustment. *IEEE Robotics and Automation Letters* 8(3): 1523–1530.

Liu Y, Fu Y, Qin M, Xu Y, Xu B, Chen F, Goossens B, Sun PZ, Yu H, Liu C, Chen L, Tao W and Zhao H (2024) BotanicGarden: A high-quality dataset for robot navigation in unstructured natural environments. *IEEE Robotics and Automation Letters* 9(3): 2798–2805.

Martin-Brualla R, Radwan N, Sajjadi MSM, Barron JT, Dosovitskiy A and Duckworth D (2021) NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 7210–7219.

Matsuki H, Murai R, Kelly PH and Davison AJ (2024) Gaussian splatting SLAM. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 18039–18048.

Mildenhall B, Srinivasan PP, Ortiz-Cayon R, Kalantari NK, Ramamoorthi R, Ng R and Kar A (2019) Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)* 38(4): 1–14.

Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R and Ng R (2021) NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* 65(1): 99–106.

Müller T, Evans A, Schied C and Keller A (2022) Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)* 41(4): 1–15.

Nair AD, Kindle J, Levchev P and Scaramuzza D (2024) Hilti SLAM Challenge 2023: Benchmarking single + multi-session slam across sensor constellations in construction. *IEEE Robotics and Automation Letters* 9(8): 7286–7293.

Nguyen TM, Yuan S, Nguyen TH, Yin P, Cao H, Xie L, Wozniak M, Jensfelt P, Thiel M, Ziegenbein J and Blunder N (2024) MCD: Diverse large-scale multi-campus dataset for robot perception. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 22304–22313.

Ramezani M, Tinchev G, Iuganov E and Fallon M (2020a) Online LiDAR-SLAM for legged robots with robust registration and deep-learned loop closure. In: *IEEE Robotics and Automation Letters*. pp. 4158–4164.

Ramezani M, Wang Y, Camurri M, Wisth D, Mattamala M and Fallon M (2020b) The Newer College dataset: Handheld lidar, inertial and vision with ground truth. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, pp. 4353–4360.

Rehder J, Nikolic J, Schneider T, Hinzmann T and Siegwart R (2016) Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. pp. 4304–4311.

Rematas K, Liu A, Srinivasan PP, Barron JT, Tagliasacchi A, Funkhouser T and Ferrari V (2022) Urban radiance fields. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 12932–12942.

Sarlin PE, Dusmanu M, Schönberger JL, Speciale P, Gruber L, Larsson V, Miksik O and Pollefeys M (2022) LaMAR: Benchmarking localization and mapping for augmented reality. In: *Eur. Conf. on Computer Vision (ECCV)*. pp. 686–704.

Schmidt F, Blessing C, Enzweiler M and Valada A (2024) Rover: A multi-season dataset for visual slam. *arXiv preprint arXiv:2412.02506* .

Schönberger JL and Frahm JM (2016) Structure-from-Motion revisited. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 4104–4113.

Schops T, Schonberger JL, Galliani S, Sattler T, Schindler K, Pollefeys M and Geiger A (2017) A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 3260–3269.

Schubert D, Goll T, Demmel N, Usenko V, Stueckler J and Cremers D (2018) The TUM VI benchmark for evaluating visual-inertial odometry. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 1680–1687.

Seitz SM, Curless B, Diebel J, Scharstein D and Szeliski R (2006) A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE, pp. 519–528.

Smith M, Baldwin I, Churchill W, Paul R and Newman P (2009) The New College vision and laser data set. *Intl. J. of Robotics Research* 28(5): 595–599.

Straub J, Whelan T, Ma L, Chen Y, Wijmans E, Green S, Engel JJ, Mur-Artal R, Ren C, Verma S, Clarkson A, Yan M, Budge B, Yan Y, Pan X, Yon J, Zou Y, Leon K, Carter N, Briales J, Gillingham T, Mueggler E, Pesqueira L, Savva M, Batra D, Strasdat HM, Nardi RD, Goesele M, Lovegrove S and Newcombe R (2019) The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797* .

Sturm J, Engelhard N, Endres F, Burgard W and Cremers D (2012) A benchmark for the evaluation of RGB-D SLAM systems. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. pp. 573–580.

Sucar E, Liu S, Ortiz J and Davison AJ (2021) iMAP: Implicit mapping and positioning in real-time. In: *Intl. Conf. on Computer Vision (ICCV)*. pp. 6229–6238.

Sun T, Hao Y, Huang S, Savarese S, Schindler K, Pollefeys M and Armeni I (2023) Nothing Stands Still: A spatiotemporal benchmark on 3D point cloud registration under large geometric and temporal change. *arXiv preprint arXiv:2311.09346* .

Tancik M, Weber E, Ng E, Li R, Yi B, Wang T, Kristoffersen A, Austin J, Salahi K, Ahuja A, Mcallister D, Kerr J and Kanazawa A (2023) Nerfstudio: A modular framework for neural radiance field development. In: *SIGGRAPH*. pp. 1–12.

Wang Z, Bovik A, Sheikh H and Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4): 600–612.

Wei H, Jiao J, Hu X, Yu J, Xie X, Wu J, Zhu Y, Liu Y, Wang L and Liu M (2024) FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments. *Intl. J. of Robotics Research* .

Wisth D, Camurri M, Das S and Fallon M (2021) Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry. *IEEE Robotics and Automation Letters* 6(2): 1004–1011.

Wisth D, Camurri M and Fallon M (2023) VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots. *IEEE Trans. Robotics* 39(1): 309–326.

Xu W, Cai Y, He D, Lin J and Zhang F (2022) Fast-LIO2: Fast direct lidar-inertial odometry. *IEEE Trans. Robotics* 38(4): 2053–2073.

Ye V, Li R, Kerr J, Turkulainen M, Yi B, Pan Z, Seiskari O, Ye J, Hu J, Tancik M and Kanazawa A (2024) gsplat: An open-source library for Gaussian splatting. *arXiv preprint arXiv:2409.06765* .

Yeshwanth C, Liu YC, Nießner M and Dai A (2023) ScanNet++: A high-fidelity dataset of 3D indoor scenes. In: *Intl. Conf. on Computer Vision (ICCV)*. pp. 12–22.

Yogamani S, Hughes C, Horgan J, Sistu G, Varley P, O'Dea D, Uricár M, Milz S, Simon M, Amende K et al. (2019) Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. In: *Intl. Conf. on Computer Vision (ICCV)*. pp. 9308–9318.

Zhang L, Helmberger M, Fu LFT, Wisth D, Camurri M, Scaramuzza D and Fallon M (2022) Hilti-Oxford dataset: A millimeter-accurate benchmark for simultaneous localization and mapping. *IEEE Robotics and Automation Letters* 8(1): 408–415.

Zhang R, Isola P, Efros AA, Shechtman E and Wang O (2018) The unreasonable effectiveness of deep features as a perceptual metric. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 586–595.

Zhu Z, Peng S, Larsson V, Xu W, Bao H, Cui Z, Oswald MR and Pollefeys M (2022) NICE-SLAM: Neural implicit scalable encoding for SLAM. In: *IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR)*. pp. 12786–12796.

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	The Oxford Spires dataset: Benchmarking large-scale LiDAR-visual localisation, reconstruction and radiance field methods
Publication Status	□Published ☒Accepted for Publication □ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Y. Tao, M. Á. Muñoz-Bañón, L. Zhang, J. Wang, L. F. T. Fu, and M. Fallon, "The Oxford Spires dataset: Benchmarking large-scale LiDAR-visual localisation, reconstruction and radiance field methods," International Journal of Robotics Research, 2025

Student Confirmation

Student Name:	Yifu Tao		
Contribution to the Paper	• Contributed to development of the idea and lead the project • Collected more than half of the data • Conducted reconstruction benchmark and NeRF benchmark experiments • Wrote the majority of the paper		
Signature.	陶逸夫	Date	22nd January 2025

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: Maurice Fallon			
Supervisor comments Yifu was the lead investigator, implementer and researcher on this project and wrote the bulk of the paper			
Signature	Maurice Fallon	Date	22nd January 2025

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

7.2 Technical Remarks

In this section, we provide additional details and clarification in response to first round of feedback from the journal reviewers.

Auto-Exposure

One question asked is the reason for using camera's auto-exposure module, and how it affects the mapping results, especially how the colour inconsistency might affect the reconstruction.

Fixed exposure time can reduce photometric variation, which is useful for training and testing radiance field methods. For example, Mip-NeRF 360 [27] captures several sequences with fixed exposure time and consistent lighting conditions. However, many of our sequences contain areas of different lighting conditions. In this case, fixed-exposure time can only allow the camera with limited dynamic range to capture information in certain regions, with the rest part of the environment under/over-exposed. For example, in Fig. 7.1, we need different exposure times to capture the floor (blue) and the outdoor building (red). By using auto-exposure, we can capture scenes with different lighting conditions given the cameras' limited dynamic range.



Figure 7.1: Images of the Bodleian library with different exposure times. The low dynamic range camera is only able to capture either the dark or the bright region in the scene.

While we acknowledge that the auto-exposure could result in ßcolour inconsistency in the camera images, we believe that solving this issue algorithmically is actually an important research question. For example, HDR-NeRF [198] can

recover high dynamic range (HDR) images from multiple (low dynamic range) images captured with different exposure times. We encourage future systems to embrace the challenge and tackle colour consistency in their textured mapping systems. In addition, the variance in lighting conditions and exposure also affects the evaluation of novel-view synthesis. In this chapter, we prioritise SSIM and LPIPS as the main evaluation metrics since they are more invariant to appearance changes than PSNR. PSNR evaluation can be given more weight when the exposure is properly controlled or compensated for. Alternatively, methods such as test-time appearance encoding optimisation from NeRF-W [126] might be useful.

7.3 Discussion

One key outcome from our initial evaluations using the reconstruction benchmark is that the 3D reconstruction quality of NeRF and 3DGS are in fact much less competitive to a classical MVS algorithm. In Chap. 6, we also tested a SDF-based method NeuS [136] which also does not perform well, especially in textureless areas. One reason that makes visual 3D reconstruction difficult is that the images are captured using a linear motion in an outdoor environment, while radiance fields are typically evaluated in datasets which only contain object-centric trajectories. A valuable extension of this study would be to assess more recent radiance field methods that claim to provide precise geometry such as 2DGS [133] using our datasets. Additionally, incorporating novel-view synthesis results from classical textured 3D representations could provide further insight into their comparative performance. For example, textured mesh (e.g. R3LIVE [16]) and dense colourised point cloud from MVS or the TLS ground truth can be used for novel-view synthesis benchmarking. In particular, these methods may demonstrate competitive generalisation capability while producing fewer artefacts compared to radiance field approaches.

Another important aspect to investigate is the ground truth poses of the camera images. Currently, the ground truth trajectory is computed using the lidar point cloud with the ICP algorithm. Precise camera poses are needed for visual reconstruction methods such as NeRF, and they are typically estimated by

SfM methods. Given that our dataset includes colourised TLS point clouds, it is possible to obtain the ground truth camera poses directly. Fig. 7.2 provides an example rendering of the TLS map. A potential method for computing the ground truth camera poses within the TLS map is to perform image alignment between the real camera image and the TLS rendering. This approach may offer a more effective way to evaluate real-time SLAM systems which are designed to provide accurate trajectories for training radiance field methods. Furthermore, a direct comparison between the image-derived and lidar-derived ground truth poses provides a means to evaluate their consistency and accuracy. In particular, computing the relative transformations between the two sets of ground truth poses and benchmarking them with the checkerboard-based calibration can serve as a rigorous validation of sensor synchronisation and calibration.



Figure 7.2: Colourised TLS cloud projected onto a 2D image plane.

8 Conclusion

Contents

8.1 Summary 129
8.2 Future Work 131
8.2.1 Learnt Priors with Radiance Fields 131
8.2.2 Spatial Temporal Scalability for Long-term Mapping . . 131
8.2.3 Graph-based Scene Representations 132

This thesis presented two mapping systems that integrate learning-based depth completion and radiance fields into lidar SLAM systems. We developed sensor fusion strategies to leverage the complementary strengths of lidar and visual data, while also quantifying and utilising the uncertainty in these learning-based approaches to enhance reconstruction accuracy. In this chapter, we summarise the contributions, and discuss some potential avenues for future work.

8.1 Summary

In Chap. 4, we presented an online mapping system based on occupancy mapping. This system fuses lidar and camera by using a probabilistic depth completion network. This approach fuses the sequential observations by using a Bayes filter within an occupancy mapping pipeline. The fusion of the temporal data stream is

loosely-coupled, as the depth completion only operates on individual monocular camera images and a lidar depth image, and does not enforce temporal geometric consistency across frames. In terms of uncertainty quantification, this system focuses on aleatoric uncertainty in the neural network's outputs. This is compatible with the occupancy mapping framework which assumes a sensor model with a predefined uncertainty characteristic. The final occupancy value of each voxel can be interpreted as the epistemic uncertainty.

We then present an offline mapping system based on NeRF in Chap. 5. This system fuses lidar and camera by optimising the map given the sensor observation models of both visual and lidar data. Specifically, we can render camera images and lidar depths and surface normals from the map via volumetric rendering, provided that the pixel-ray correspondence has been obtained from sensor extrinsics (camera poses) and intrinsics (e.g. focal lengths, distortion parameters). Compared to the system in Chap. 4, this fusion is much more tightly-coupled, and all sensor measurements are fused together in the optimisation process. This is also a map-centric approach, compared to the view-centric approach in Chap. 4 where the dense 3D information is first obtained per frame, and then probabilistically combined. In terms of uncertainty quantification, this work focuses on epistemic uncertainty of the final NeRF map. The technique developed in Chap. 6 represents a principle way of decoupling the epistemic uncertainty resulting from the visual ambiguity and the limited lidar measurements. The dataset curated alongside the development of the lidar-visual NeRF system is presented in Chap. 7. The sensor setup of this dataset facilitates comparison across different areas of research, including lidar/visual SLAM, SfM, MVS, and radiance field methods including NeRF and 3D Gaussian Splatting.

A central approach in this thesis is leveraging uncertainty estimation when integrating learning-based methods into real-world robot mapping systems. Neural networks are prone to overconfidence and hallucination [199], a limitation that is particularly critical in robotic applications. Bayesian modelling [193] offers a principled approach to mitigate this issue by providing uncertainty estimates. In the context of large-scale outdoor mapping, we demonstrate that these uncertainty

estimates can be used to identify inaccurate predictions from learning-based methods and enhance 3D reconstruction quality. Overall, this work represents a step toward next-generation mapping systems that produce dense, geometrically and photometrically accurate reconstructions which unlock new applications in robot simulation, reality capture and navigation.

8.2 Future Work

In this section, we discuss a few emerging aspects of robot perception which show promise for developing the next generation of mapping systems.

8.2.1 Learnt Priors with Radiance Fields

The proposed mapping system based on radiance field in Chap. 5 uses the photometric information in the image and the depth information in the lidar data. The introduction of surface normal represents a geometric prior which encourages smoothness in the reconstruction surface. This approach is suitable when there are abundant sensor observations available for training, and the reconstruction is well-conditioned. However, when there are only sparse data available, additional priors are necessary. It would be interesting to incorporate the learnt priors from e.g. monocular depth estimation, deep MVS networks such as MVSNet [23] and TANDEM [66], or the recent DUSt3R [24].

8.2.2 Spatial Temporal Scalability for Long-term Mapping

Scaling up 3D reconstruction to large-scale environments is an important task. In Sec. 6.3, we have discussed how the submapping strategies can be further improved. In particular, we discussed aspects of modelling the scene complexity or the information content in the map rather than just the physical scale. Orthogonal to spatial scalability, temporal scalability is another crucial direction for long-term mapping systems. This requires a map representation that can model 4D data, e.g. sequences captured at different times, or a sequence with dynamic objects. The first type of input can be used for change detection [200], while the second type

of input can be used to model dynamic scenes [201]. These directions can help build a scalable mapping system suitable for long-term operations.

8.2.3 Graph-based Scene Representations

Many advancements in 3D reconstruction come from novel scene representations. Neural representations such as NeRF are highly compact, but less interpretable which makes downstream tasks more challenging. For example, it is not trivial to perform loop closure inside a NeRF map, and one cannot perform local map updates by changing the MLP parameters without affecting the rest of the scene. Instant-NGP [28] improves this by introducing explicit representations of voxel grids into the data structure.

The scene representation should be able to represent a full volumetric map with geometry and colour, yet it has to be compact. The compactness essentially comes from the fact that there exists a structure in the real-world 3D scenes among the full function space for a 3D field: the map elements are not randomly distributed in the space; instead, they often form distinct objects. Also, a significant portion of the scene consists of empty space. In fact, multi-resolution data structures such as octrees exploit this in the design of the data structure.

A graph is a general data structure that can represent complex data ranging from DNAs to social networks. The graph allows one to encode the relationship between the objects by the edges. In the context of 3D mapping, a node in a graph can represent an object, and the edges in the graph can be used to represent the grouping of objects. Scene graphs, which describe the environments as a graph of spatial concepts including objects, rooms and buildings, have been used in recent perception systems such as Hydra [202]. The low-level map which includes geometry and colour information can potentially be modelled by a local radiance field representation, by a MLP or 3D Gaussians [29]. This could be a compact yet interpretable scene representation which facilitates multiple robot perception tasks. Firstly, a graph representation can easily incorporate SLAM loop closures; Secondly, the high-level semantics can guide a robot for navigation; Thirdly, the

graph representation allows the modelling of temporal data: a node can be used to represent a dynamic object, and also to represent a change in the scene for a longitudinal sequence. The use of this kind of graph representation presents exciting opportunities for next-generation robot perception systems.

References

[1] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1366–1373.

[2] Tao Sun, Yan Hao, Shengyu Huang, Silvio Savarese, Konrad Schindler, Marc Pollefeys, and Iro Armeni. "Nothing Stands Still: A spatiotemporal benchmark on 3D point cloud registration under large geometric and temporal change". In: *arXiv preprint arXiv:2311.09346* (2023).

[3] Richard A. Newcombe, Steven J. Lovegrove, and Andrew J. Davison. "DTAM: Dense tracking and mapping in real-time". In: *International Conference on Computer Vision (ICCV)*. 2011, pp. 2320–2327.

[4] Jakob Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-scale direct monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. Springer. 2014, pp. 834–849.

[5] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. "Pixelwise view selection for unstructured multi-view stereo". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 501–518.

[6] Ji Zhang, Sanjiv Singh, et al. "LOAM: Lidar odometry and mapping in real-time." In: *Robotics: Science and Systems (RSS)*. Vol. 2. 9. Berkeley, CA. 2014, pp. 1–9.

[7] Emanuele Vespa, Nikolay Nikolov, Marius Grimm, Luigi Nardi, Paul H. J. Kelly, and Stefan Leutenegger. "Efficient Octree-Based Volumetric SLAM Supporting Signed-Distance and Occupancy Mapping". In: *IEEE Robotics and Automation Letters* 3.2 (Apr. 2018), pp. 1144–1151.

[8] Victor Reijgwart, Alexander Millane, Helen Oleynikova, Roland Siegwart, Cesar Cadena, and Juan Nieto. "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps". In: *IEEE Robotics and Automation Letters* 5.1 (2019), pp. 227–234.

[9] Ji Zhang and Sanjiv Singh. "Visual-lidar odometry and mapping: Low-drift, robust, and fast". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 2174–2181.

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

[11] Young-Sik Shin, Yeong Sang Park, and Ayoung Kim. "DVL-SLAM: Sparse depth enhanced direct visual-LiDAR SLAM". In: *Autonomous Robots* 44.2 (2020), pp. 115–130.

[12] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. "Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5692–5698.

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

[14] Jiarong Lin, Chunran Zheng, Wei Xu, and Fu Zhang. "R^{2}LIVE: A Robust, Real-Time, LiDAR-Inertial-Visual Tightly-Coupled State Estimator and Mapping". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 7469–7476.

[15] Kamak Ebadi, Lukas Bernreiter, Harel Biggie, Gavin Catt, Yun Chang, Arghya Chatterjee, Christopher E Denniston, Simon-Pierre Deschênes, Kyle Harlow, Shehryar Khattak, et al. "Present and future of slam in extreme environments: The darpa subt challenge". In: *IEEE Transactions on Robotics* 40 (2023), pp. 936–959.

[16] Jiarong Lin and Fu Zhang. "R^{3}LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2022, pp. 10672–10678.

[17] Chunran Zheng, Wei Xu, Zuhao Zou, Tong Hua, Chongjian Yuan, Dongjiao He, Bingyang Zhou, Zheng Liu, Jiarong Lin, Fangcheng Zhu, et al. "FAST-LIVO2: Fast, direct LiDAR-inertial-visual odometry". In: *IEEE Transactions on Robotics* (2024).

[18] Jiarong Lin, Chongjian Yuan, Yixi Cai, Haotian Li, Yunfan Ren, Yuying Zou, Xiaoping Hong, and Fu Zhang. "ImMesh: An Immediate LiDAR Localization and Meshing Framework". In: *IEEE Transactions on Robotics* 39.6 (2023), pp. 4312–4331.

[19] David Eigen, Christian Puhrsch, and Rob Fergus. "Depth map prediction from a single image using a multi-scale deep network". In: *Advances in Neural Information Processing Systems (NeurIPS)* 27 (2014).

[20] Fangchang Ma and Sertac Karaman. "Sparse-to-Dense: Depth prediction from sparse depth samples and a single image". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 4796–4803.

[21] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P. Srinivasan, and Jonathan T. Barron. "NeRF in the Dark: High Dynamic Range View Synthesis from Noisy Raw Images". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[22] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. "Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024.

[23] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. "MVSNet: Depth inference for unstructured multi-view stereo". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 767–783.

[24] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. "DUSt3R: Geometric 3D vision made easy". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 20697–20709.

[25] 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: *Communications of the ACM* 65.1 (2021), pp. 99–106.

[26] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5855–5864.

[27] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 5470–5479.

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant neural graphics primitives with a multiresolution hash encoding". In: *ACM Transactions on Graphics (TOG)* 41.4 (2022), pp. 1–15.

[29] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering". In: *ACM Transactions on Graphics (TOG)* 42.4 (July 2023), pp. 1–14.

[30] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. "Plenoxels: Radiance Fields Without Neural Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5501–5510.

[31] Arunkumar Byravan, Jan Humplik, Leonard Hasenclever, Arthur Brussee, Francesco Nori, Tuomas Haarnoja, Ben Moran, Steven Bohez, Fereshteh Sadeghi, Bojan Vujatovic, et al. "Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 9362–9369.

[32] Jie Tang, Fei-Peng Tian, Boshi An, Jian Li, and Ping Tan. "Bilateral propagation network for depth completion". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 9763–9772.

[33] Amir Atapour-Abarghouei and Toby P Breckon. "Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2800–2810.

[34] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. "CodeSLAM—learning a compact, optimisable representation for dense visual SLAM". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2560–2568.

[35] Yifu Tao, Marija Popović, Yiduo Wang, Sundara Tejaswi Digumarti, Nived Chebrolu, and Maurice Fallon. "3D Lidar Reconstruction with Probabilistic Depth Completion for Robotic Navigation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 5339–5346.

[36] Yifu Tao, Yash Bhalgat, Lanke Frank Tarimo Fu, Matias Mattamala, Nived Chebrolu, and Maurice Fallon. "SiLVR: Scalable Lidar-Visual Reconstruction with Neural Radiance Fields for Robotic Inspection". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024.

[37] Kevin P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023.

[38] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

[39] OpenCV. *Camera Calibration and 3D Reconstruction*. Accessed: 2025-03-02. 2024. URL: `https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html`.

[40] Juho Kannala and Sami S Brandt. "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.8 (2006), pp. 1335–1340.

[41] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees". In: *Autonomous Robots* 34 (2013), pp. 189–206.

[42] Lee Alan Westover. *Splatting: a parallel, feed-forward volume rendering algorithm*. The University of North Carolina at Chapel Hill, 1991.

[43] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. "Real-time 3D reconstruction at scale using voxel hashing". In: *ACM Transactions on Graphics (ToG)* 32.6 (2013), pp. 1–11.

[44] Simon T O'Callaghan and Fabio T Ramos. "Gaussian process occupancy maps". In: *International Journal of Robotics Research* 31.1 (2012), pp. 42–62.

[45] Nelson Max. "Optical models for direct volume rendering". In: *IEEE Transactions on Visualization and Computer Graphics* 1.2 (1995), pp. 99–108.

[46] James T Kajiya and Brian P Von Herzen. "Ray tracing volume densities". In: *SIGGRAPH* 18.3 (1984), pp. 165–174.

[47] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. "MonoSLAM: Real-time single camera SLAM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.6 (2007), pp. 1052–1067.

[48] Georg Klein and David Murray. "Parallel Tracking and Mapping for Small AR Workspaces". In: *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. 2007, pp. 225–234.

[49] Hauke Strasdat, José MM Montiel, and Andrew J Davison. "Visual SLAM: why filter?" In: *Image and Vision Computing* 30.2 (2012), pp. 65–77.

[50] Hauke Strasdat, JMM Montiel, and Andrew J Davison. "Scale Drift-Aware Large Scale Monocular SLAM." In: *Robotics: Science and Systems (RSS)*. Vol. 2. 3. 2010, p. 5.

[51] Hauke Strasdat, Andrew J Davison, JM Martìnez Montiel, and Kurt Konolige. "Double window optimisation for constant time visual SLAM". In: *International Conference on Computer Vision (ICCV)*. IEEE. 2011, pp. 2352–2359.

[52] Dorian Galvez-López and Juan D. Tardos. “Bags of Binary Words for Fast Place Recognition in Image Sequences”. In: *IEEE Transactions on Robotics* 28.5 (2012), pp. 1188–1197.

[53] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. “ORB-SLAM: a versatile and accurate monocular SLAM system”. In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163.

[54] Raul Mur-Artal and Juan D Tardós. “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras”. In: *IEEE Transactions on Robotics* 33.5 (2017), pp. 1255–1262.

[55] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. “ORB-SLAM3: An accurate open-source library for visual, visual-inertial, and multimap SLAM”. In: *IEEE Transactions on Robotics* 37.6 (2021), pp. 1874–1890.

[56] Jakob Engel, Vladlen Koltun, and Daniel Cremers. “Direct sparse odometry”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.3 (2017), pp. 611–625.

[57] Christian Forster, Zichao Zhang, Michael Gassner, Manuel Werlberger, and Davide Scaramuzza. “SVO: Semidirect visual odometry for monocular and multicamera systems”. In: *IEEE Transactions on Robotics* 33.2 (2016), pp. 249–265.

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

[59] Raúl Mur-Artal and Juan D Tardós. “Visual-inertial monocular SLAM with map reuse”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 796–803.

[60] Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang. “OpenVINS: A research platform for visual-inertial estimation”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 4666–4672.

[61] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. “Kimera: an open-source library for real-time metric-semantic localization and mapping”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 1689–1696.

[62] Stefan Leutenegger. “OKVIS2: Realtime scalable visual-inertial SLAM with loop closure”. In: *arXiv preprint arXiv:2202.09199* (2022).

[63] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. “CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 6243–6252.

[64] Gwangbin Bae and Andrew J Davison. “Rethinking inductive biases for surface normal estimation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 9535–9545.

[65] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. "DeepFactors: Real-time probabilistic dense monocular SLAM". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 721–728.

[66] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. "TANDEM: Tracking and dense mapping in real-time using deep multi-view stereo". In: *Conference on Robot Learning (CoRL)*. 2022, pp. 34–45.

[67] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2043–2050.

[68] Zachary Teed and Jia Deng. "DROID-SLAM: Deep visual SLAM for monocular, stereo, and RGB-D cameras". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), pp. 16558–16569.

[69] Riku Murai, Eric Dexheimer, and Andrew J Davison. "MASt3R-SLAM: Real-Time Dense SLAM with 3D Reconstruction Priors". In: *arXiv preprint arXiv:2412.12392* (2024).

[70] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. "Grounding image matching in 3D with MASt3R". In: *European Conference on Computer Vision (ECCV)*. Springer. 2024, pp. 71–91.

[71] Zheng Liu and Fu Zhang. "BALM: Bundle adjustment for lidar mapping". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3184–3191.

[72] Zheng Liu, Xiyuan Liu, and Fu Zhang. "Efficient and consistent bundle adjustment on lidar point clouds". In: *IEEE Transactions on Robotics* (2023).

[73] Xiyuan Liu, Zheng Liu, Fanze Kong, and Fu Zhang. "Large-Scale LiDAR Consistent Mapping Using Hierarchical LiDAR Bundle Adjustment". In: *IEEE Robotics and Automation Letters* 8.3 (2023), pp. 1523–1530.

[74] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking". In: *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Ieee. 2011, pp. 127–136.

[75] Brian Curless and Marc Levoy. "A volumetric method for building complex models from range images". In: *SIGGRAPH*. 1996, pp. 303–312.

[76] Thomas Whelan, Michael Kaess, Hordur Johannsson, Maurice Fallon, John J Leonard, and John McDonald. "Real-time large-scale dense RGB-D SLAM with volumetric fusion". In: *International Journal of Robotics Research* 34.4-5 (2015), pp. 598–626.

[77] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu. "Octree-based fusion for realtime 3D reconstruction". In: *Graphical Models* 75.3 (2013), pp. 126–136.

[78] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. "Very high frame rate volumetric integration of depth images on mobile devices". In: *IEEE Transactions on Visualization and Computer Graphics* 21.11 (2015), pp. 1241–1250.

[79] Olaf Kähler, Victor A Prisacariu, and David W Murray. "Real-time large-scale dense 3D reconstruction with loop closure". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 500–516.

[80] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. "BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration". In: *ACM Transactions on Graphics (TOG)* (2017).

[81] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. "Surfels: Surface elements as rendering primitives". In: *SIGGRAPH*. 2000, pp. 335–342.

[82] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. "Real-time 3D reconstruction in dynamic scenes using point-based fusion". In: *IEEE International Conference on 3D Vision*. IEEE. 2013, pp. 1–8.

[83] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. "ElasticFusion: Dense SLAM without a pose graph." In: *Robotics: Science and Systems (RSS)*. Vol. 11. Rome, Italy. 2015, p. 3.

[84] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. "BAD SLAM: Bundle Adjusted Direct RGB-D SLAM". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[85] Tixiao Shan and Brendan Englot. "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 4758–4765.

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

[87] 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 Robotics and Automation Letters* 8.2 (2023), pp. 1029–1036.

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

[89] Giseop Kim, Sunwook Choi, and Ayoung Kim. "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments". In: *IEEE Transactions on Robotics* 38.3 (2021), pp. 1856–1874.

[90] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. "LoGG3D-Net: Locally guided global descriptor learning for 3D place recognition". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2215–2221.

[91] Haoyang Ye, Yuying Chen, and Ming Liu. "Tightly Coupled 3D Lidar Inertial Odometry and Mapping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2019, pp. 3144–3150.

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

[93] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. "Fast-LIO2: Fast direct lidar-inertial odometry". In: *IEEE Transactions on Robotics* 38.4 (2022), pp. 2053–2073.

[94] Nils Funk, Juan Tarrio, Sotiris Papatheodorou, Marija Popović, Pablo F Alcantarilla, and Stefan Leutenegger. "Multi-resolution 3D mapping with explicit free space representation for fast and accurate mobile robot motion planning". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 3553–3560.

[95] Yiduo Wang, Nils Funk, Milad Ramezani, Sotiris Papatheodorou, Marija Popović, Marco Camurri, Stefan Leutenegger, and Maurice Fallon. "Elastic and efficient LiDAR reconstruction for large-scale exploration tasks". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5035–5041.

[96] Xingguang Zhong, Yue Pan, Jens Behley, and Cyrill Stachniss. "Shine-mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 8371–8377.

[97] Junyuan Deng, Qi Wu, Xieyuanli Chen, Songpengcheng Xia, Zhen Sun, Guoqing Liu, Wenxian Yu, and Ling Pei. "Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping". In: *International Conference on Computer Vision (ICCV)*. 2023, pp. 8218–8227.

[98] Yue Pan, Xingguang Zhong, Louis Wiesmann, Thorbjörn Posewsky, Jens Behley, and Cyrill Stachniss. "PIN-SLAM: LiDAR SLAM using a point-based implicit neural representation for achieving global map consistency". In: *IEEE Transactions on Robotics* (2024).

[99] Ji Zhang, Michael Kaess, and Sanjiv Singh. "Real-time depth enhanced monocular odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2014, pp. 4973–4980.

[100] Xiao Liang, Haoyao Chen, Yanjie Li, and Yunhui Liu. "Visual laser-SLAM in large-scale indoor environments". In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2016, pp. 19–24.

[101] Gaurav Pandey, Silvio Savarese, James R McBride, and Ryan M Eustice. "Visually bootstrapped generalized ICP". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011, pp. 2660–2667.

[102] Xingxing Zuo, Patrick Geneva, Woosik Lee, Yong Liu, and Guoquan Huang. "Lic-fusion: Lidar-inertial-camera odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 5848–5854.

[103] Anastasios I. Mourikis and Stergios I. Roumeliotis. "A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2007, pp. 3565–3572.

[104] David Wisth, Marco Camurri, Sandipan Das, and Maurice Fallon. "Unified multi-modal landmark tracking for tightly coupled lidar-visual-inertial odometry". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1004–1011.

[105] David Wisth, Marco Camurri, and Maurice Fallon. "VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots". In: *IEEE Transactions on Robotics* 39.1 (2023), pp. 309–326.

[106] Jiarong Lin and Fu Zhang. "R^{3}LIVE++: A Robust, Real-time, Radiance Reconstruction Package with a Tightly-coupled LiDAR-Inertial-Visual State Estimator". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[107] Jason Ku, Ali Harakeh, and Steven L Waslander. "In defense of classical image processing: Fast depth completion on the cpu". In: *Conference on Computer and Robot Vision (CRV)*. IEEE. 2018, pp. 16–22.

[108] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[109] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. "Self-supervised Sparse-to-Dense: Self-supervised depth completion from lidar and monocular camera". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3288–3295.

[110] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer. 2015, pp. 234–241.

[111] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. "DeepLiDAR: Deep Surface Normal Guided Depth Prediction for Outdoor Scene from Sparse LiDAR Data and Single Color Image". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 3313–3322.

[112] Maximilian Jaritz, Raoul de Charette, Emilie Wirbel, Xavier Perrotton, and Fawzi Nashashibi. "Sparse and Dense Data with CNNs: Depth Completion and Semantic Segmentation". In: *IEEE International Conference on 3D Vision*. 2018, pp. 52–60.

[113] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. "Sparsity Invariant CNNs". In: *Intl. Conf. on 3D Vision (3DV)* (2017), pp. 11–20.

[114] Hans Knutsson and C-F Westin. "Normalized and differential convolution". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 1993, pp. 515–523.

[115] Jiashen Hua and Xiaojin Gong. "A Normalized Convolutional Neural Network for Guided Sparse Depth Upsampling". In: *International Joint Conference on Artificial Intelligence* (2018), pp. 2283–2290.

[116] Abdelrahman Eldesokey, Michael Felsberg, and Fahad Shahbaz Khan. "Confidence Propagation through CNNs for Guided Sparse Depth Regression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.10 (2020), pp. 2423–2436.

[117] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Mikael Persson. "Uncertainty-Aware CNNs for Depth Completion: Uncertainty from Beginning to End". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[118] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. "Learning affinity via spatial propagation networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[119] Xinjing Cheng, Peng Wang, and Ruigang Yang. "Depth estimation via affinity learned with convolutional spatial propagation network". In: *European Conference on Computer Vision (ECCV)*. 2018, pp. 103–119.

[120] Xinjing Cheng, Peng Wang, Chenye Guan, and Ruigang Yang. "CSPN++: Learning Context and Resource Aware Convolutional Spatial Propagation Networks for Depth Completion". In: *AAAI Conference on Artificial Intelligence*. Vol. 34. 07. Apr. 2020, pp. 10615–10622.

[121] Mu Hu, Shuling Wang, Bin Li, Shiyu Ning, Li Fan, and Xiaojin Gong. "PENet: Towards Precise and Efficient Image Guided Depth Completion". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2021.

[122] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. "Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.3 (2022).

[123] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. "Fourier features let networks learn high frequency functions in low dimensional domains". In: *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), pp. 7537–7547.

[124] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. "NeRF++: Analyzing and improving neural radiance fields". In: *arXiv preprint arXiv:2010.07492* (2020).

[125] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5855–5864.

[126] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. "NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 7210–7219.

[127] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. "Plenoctrees for real-time rendering of neural radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5752–5761.

[128] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. "Zip-NeRF: Anti-aliased grid-based neural radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2023, pp. 19697–19705.

[129] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. "Point-NeRF: Point-based neural radiance fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 5438–5448.

[130] Yiming Gao, Yan-Pei Cao, and Ying Shan. "SurfelNeRF: Neural surfel radiance fields for online photorealistic reconstruction of indoor scenes". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 108–118.

[131] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. "Mip-Splatting: Alias-free 3D gaussian splatting". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 19447–19456.

[132] Antoine Guédon and Vincent Lepetit. "SuGaR: Surface-aligned Gaussian Splatting for efficient 3D mesh reconstruction and high-quality mesh rendering". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 5354–5363.

[133] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. "2D Gaussian Splatting for Geometrically Accurate Radiance Fields". In: *SIGGRAPH*. Association for Computing Machinery, 2024.

[134] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. "Volume rendering of neural implicit surfaces". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), pp. 4805–4815.

[135] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. "Implicit Geometric Regularization for Learning Shapes". In: *Machine Learning and Systems*. 2020, pp. 3569–3579.

[136] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. "NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

[137] Michael Oechsle, Songyou Peng, and Andreas Geiger. "UNISURF: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5589–5599.

[138] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. "NeuS2: Fast learning of neural implicit surfaces for multi-view reconstruction". In: *International Conference on Computer Vision (ICCV)*. 2023, pp. 3295–3306.

[139] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. "Neuralangelo: High-Fidelity Neural Surface Reconstruction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[140] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. "iMAP: Implicit mapping and positioning in real-time". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 6229–6238.

[141] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. "NICE-SLAM: Neural implicit scalable encoding for SLAM". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12786–12796.

[142] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. "Neural RGB-D Surface Reconstruction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 6290–6301.

[143] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. "Depth-supervised NeRF: Fewer views and faster training for free". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12882–12891.

[144] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. "Nerfingmvs: Guided optimization of neural radiance fields for indoor multi-view stereo". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5610–5619.

[145] Barbara Roessle, Jonathan T Barron, Ben Mildenhall, Pratul P Srinivasan, and Matthias Nießner. "Dense depth priors for neural radiance fields from sparse input views". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12892–12901.

[146] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. "MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction". In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022).

[147] Hidenobu Matsuki, Riku Murai, Paul HJ Kelly, and Andrew J Davison. "Gaussian splatting slam". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 18039–18048.

[148] Haithem Turki, Deva Ramanan, and Mahadev Satyanarayanan. "Mega-NeRF: Scalable construction of large-scale NeRF for virtual fly-throughs". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12922–12931.

[149] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. "Block-NeRF: Scalable large scene neural view synthesis". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 8248–8258.

[150] Jiaqi Lin, Zhihao Li, Xiao Tang, Jianzhuang Liu, Shiyong Liu, Jiayue Liu, Yangdi Lu, Xiaofei Wu, Songcen Xu, Youliang Yan, et al. "VastGaussian: Vast 3D Gaussians for large scene reconstruction". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 5166–5175.

[151] Yang Liu, Chuanchen Luo, Lue Fan, Naiyan Wang, Junran Peng, and Zhaoxiang Zhang. "CityGaussian: Real-time high-quality large-scale scene rendering with Gaussians". In: *European Conference on Computer Vision (ECCV)*. Springer. 2024, pp. 265–282.

[152] Ruilong Li, Sanja Fidler, Angjoo Kanazawa, and Francis Williams. "NeRF-XL: Scaling NeRFs with multiple gpus". In: *European Conference on Computer Vision (ECCV)*. Springer. 2024, pp. 92–107.

[153] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David Mcallister, Justin Kerr, and Angjoo Kanazawa. "Nerfstudio: A modular framework for neural radiance field development". In: *SIGGRAPH*. 2023, pp. 1–12.

[154] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. "A benchmark for the evaluation of RGB-D SLAM systems". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 573–580.

[155] 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: *International Journal of Robotics Research* 35.10 (2016), pp. 1157–1163.

[156] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stueckler, and D. Cremers. "The TUM VI Benchmark for Evaluating Visual-Inertial Odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Oct. 2018, pp. 1680–1687.

[157] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. "Vision meets robotics: The KITTI dataset". In: *International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[158] Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. "Complex urban dataset with multi-level sensors from highly diverse urban environments". In: *International Journal of Robotics Research* 38.6 (2019), pp. 642–657.

[159] Michael Helmberger, Kristian Morin, Beda Berner, Nitish Kumar, Giovanni Cioffi, and Davide Scaramuzza. "The Hilti SLAM Challenge Dataset". In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 7518–7525.

[160] Milad Ramezani, Yiduo Wang, Marco Camurri, David Wisth, Matias Mattamala, and Maurice Fallon. "The Newer College Dataset: Handheld lidar, inertial and vision with ground truth". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 4353–4360.

[161] 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 Robotics and Automation Letters* 8.1 (2022), pp. 408–415.

[162] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 1524–1531.

[163] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5828–5839.

[164] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. "Matterport3D: Learning from RGB-D Data in Indoor Environments". In: *IEEE International Conference on 3D Vision*. 2017, pp. 667–676.

[165] Thien-Minh Nguyen, Shenghai Yuan, Thien Hoang Nguyen, Pengyu Yin, Haozhi Cao, Lihua Xie, Maciej Wozniak, Patric Jensfelt, Marko Thiel, Justin Ziegenbein, and Noel Blunder. "MCD: Diverse Large-Scale Multi-Campus Dataset for Robot Perception". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024, pp. 22304–22313.

[166] Hexiang Wei, Jianhao Jiao, Xiangcheng Hu, Jingwen Yu, Xupeng Xie, Jin Wu, Yilong Zhu, Yuxuan Liu, Lujia Wang, and Ming Liu. "FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments". In: *International Journal of Robotics Research* (2024).

[167] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. "Large-scale data for multiple-view stereopsis". In: *International Journal of Computer Vision* 120 (2016), pp. 153–168.

[168] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. "A multi-view stereo benchmark with high-resolution images and multi-camera videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3260–3269.

[169] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. "Tanks and Temples: Benchmarking large-scale scene reconstruction". In: *ACM Transactions on Graphics (TOG)* 36.4 (2017), pp. 1–13.

[170] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. "DeepVoxels: Learning Persistent 3D Feature Embeddings". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

[171] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. "Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), pp. 1–14.

[172] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. "ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes". In: *International Conference on Computer Vision (ICCV)*. 2023, pp. 12–22.

[173] Yohann Cabon, Naila Murray, and Martin Humenberger. *Virtual KITTI 2*. 2020. arXiv: `2001.10773 [cs.CV]`.

[174] Lanke Frank Tarimo Fu, Nived Chebrolu, and Maurice Fallon. "Extrinsic Calibration of Camera to LIDAR Using a Differentiable Checkerboard Model". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2023, pp. 1825–1831.

[175] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *International Conference on Machine Learning (ICML)*. 2016, pp. 1050–1059.

[176] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *Advances in Neural Information Processing Systems (NeurIPS)* 30 (2017).

[177] Siddharth Ancha, Philip R. Osteen, and Nicholas Roy. "Deep Evidential Uncertainty Estimation for Semantic Segmentation under Out-Of-Distribution Obstacles". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2024, pp. 6943–6951.

[178] Konstantinos Rematas, Andrew Liu, Pratul P Srinivasan, Jonathan T Barron, Andrea Tagliasacchi, Thomas Funkhouser, and Vittorio Ferrari. "Urban radiance fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 12932–12942.

[179] Joowan Kim, Younggun Cho, and Ayoung Kim. "Proactive camera attribute control using bayesian optimization for illumination-resilient visual navigation". In: *IEEE Transactions on Robotics* 36.4 (2020), pp. 1256–1271.

[180] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. "Global Structure-from-Motion Revisited". In: *European Conference on Computer Vision (ECCV)*. 2024.

[181] Johannes Lutz Schönberger and Jan-Michael Frahm. "Structure-from-Motion Revisited". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 4104–4113.

[182] Lin Yen-Chen, Pete Florence, Jonathan T Barron, Alberto Rodriguez, Phillip Isola, and Tsung-Yi Lin. "iNeRF: Inverting neural radiance fields for pose estimation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 1323–1330.

[183] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. "BARF: Bundle-adjusting neural radiance fields". In: *International Conference on Computer Vision (ICCV)*. 2021, pp. 5741–5751.

[184] Wenjing Bian, Zirui Wang, Kejie Li, Jia-Wang Bian, and Victor Adrian Prisacariu. "NOPE-NeRF: Optimising neural radiance field with no pose prior". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 4160–4169.

[185] Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. "Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 5044–5053.

[186] Eric Brachmann, Jamie Wynn, Shuai Chen, Tommaso Cavallari, Áron Monszpart, Daniyar Turmukhambetov, and Victor Adrian Prisacariu. "Scene coordinate reconstruction: Posing of image collections via incremental learning of a relocalizer". In: *European Conference on Computer Vision (ECCV)*. Springer. 2024, pp. 421–440.

[187] Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. "Laplace redux-effortless bayesian deep learning". In: *Advances in Neural Information Processing Systems (NeurIPS)* 34 (2021), pp. 20089–20103.

[188] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. "Bayes' Rays: Uncertainty Quantification in Neural Radiance Fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2024.

[189] Wen Jiang, Boshu Lei, and Kostas Daniilidis. "FisherRF: Active view selection and uncertainty quantification for radiance fields using fisher information". In: *European Conference on Computer Vision (ECCV)*. 2024.

[190] Christopher Nemeth and Paul Fearnhead. "Stochastic gradient markov chain monte carlo". In: *Journal of the American Statistical Association* 116.533 (2021), pp. 433–450.

[191] Andrew G Wilson and Pavel Izmailov. "Bayesian deep learning and a probabilistic perspective of generalization". In: *Advances in Neural Information Processing Systems (NeurIPS)* 33 (2020), pp. 4697–4708.

[192] Tim GJ Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. "Tractable function-space variational inference in bayesian neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)* 35 (2022), pp. 22686–22698.

[193] Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, Aliaksandr Hubin, Alexander Immer, Theofanis Karaletsos, Mohammad Emtiyaz Khan, Agustinus Kristiadi, Yingzhen Li, Stephan Mandt, Christopher Nemeth, Michael A. Osborne, Tim G. J. Rudner, David Rügamer, Yee Whye Teh, Max Welling, Andrew Gordon Wilson, and Ruqi Zhang. "Position paper: Bayesian deep learning in the age of large-scale AI". In: *arXiv e-prints* (2024), arXiv–2402.

[194] Alex Kendall and Yarin Gal. "What uncertainties do we need in Bayesian deep learning for computer vision?" In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.

[195] Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. "Optimizing the latent space of generative networks". In: *International Conference on Machine Learning (ICML)*. 2018.

[196] Rowan Border, Nived Chebrolu, Yifu Tao, Jonathan D. Gammell, and Maurice Fallon. "Osprey: Multisession Autonomous Aerial Mapping With LiDAR-Based SLAM and Next Best View Planning". In: *IEEE Transactions on Field Robotics* 1 (2024), pp. 113–130.

[197] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. "Towards internet-scale multi-view stereo". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2010, pp. 1434–1441.

[198] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, Xuan Wang, and Qing Wang. "HDF-NeRF: High dynamic range neural radiance fields". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 18398–18408.

[199] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. "On calibration of modern neural networks". In: *International Conference on Machine Learning (ICML)*. PMLR. 2017, pp. 1321–1330.

[200] Giseop Kim and Ayoung Kim. "LT-mapper: A modular framework for lidar-based lifelong mapping". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 7995–8002.

[201] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. "Dynamic 3D Gaussians: Tracking by persistent dynamic view synthesis". In: *IEEE International Conference on 3D Vision*. IEEE. 2024, pp. 800–809.

[202] Nathan Hughes, Yun Chang, and Luca Carlone. "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization". In: *Robotics: Science and Systems (RSS)*. 2022.