



# SiLVR: Scalable Lidar-Visual Radiance Field Reconstruction With Uncertainty Quantification

Yifu Tao , *Graduate Student Member, IEEE*, and Maurice Fallon , *Senior Member, IEEE*

**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 modeling uniform texture surfaces which contain ambiguous visual reconstruction cues. A key contribution of this work is a novel method to quantify the epistemic uncertainty of the lidar-visual NeRF reconstruction by estimating the spatial variance of each point location in the radiance field given the sensor observations from the cameras and lidar. This provides a principled approach to evaluate the contribution of each sensor modality to the final reconstruction. In this way, reconstructions that are uncertain (due to, e.g., uniform visual texture, limited observation viewpoints, or little lidar coverage) can be identified and removed. Our system is integrated with a real-time pose-graph lidar simultaneous localization and mapping (SLAM) system, which is used to bootstrap a structure-from-motion reconstruction procedure. It also helps to properly constrain the overall metric scale, which is essential for the lidar depth loss. The refined SLAM trajectory can then be divided into submaps using spectral clustering to group sets of covisible images together. This submapping approach is more suitable for visual reconstruction than distance-based partitioning. Our uncertainty estimation is particularly effective when merging submaps, as their boundaries often contain artifacts due to limited observations. We demonstrate the reconstruction system using a multicamera, 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<sup>2</sup>, including multiple university buildings and an aerial survey of a multistorey building. Quantitative evaluation is provided by comparing with maps produced by a commercial tripod scanner.

**Index Terms**—Mapping, neural radiance field (NeRF), RGB-D perception, sensor fusion, uncertainty estimation.

## I. INTRODUCTION

**D**ENSE 3-D reconstruction is a core component of a range of robotics applications such as industrial inspection and autonomous navigation. Common sensors used for reconstruction

Received 16 June 2025; accepted 8 September 2025. Date of publication 17 November 2025; date of current version 11 December 2025. This work was supported in part by the Horizon Europe project Digiforest under Grant 101070405, in part by Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/Z531212/1, and in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) under Grant RS-2024-00461409. This article was recommended for publication by Associate Editor G. Cioffi and Editor J. Civera upon evaluation of the reviewers' comments. (*Corresponding author: Yifu Tao.*)

The authors are with the Oxford Robotics Institute, Department of Engineering Science, University of Oxford, OX2 6NN Oxford, U.K. (e-mail: yifu@robots.ox.ac.uk).

Digital Object Identifier 10.1109/TRO.2025.3633844

include cameras and lidars. Camera-based reconstruction systems use techniques including structure-from-motion (SfM) [1] and multiview 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 color, which is important for applications such as virtual reality and 3-D asset generation.

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

As with traditional vision-based reconstruction methods, NeRF struggles to estimate accurate geometry in locations where there is limited multiview input (i.e., when images are only taken from a single direction) or sparse texture. Autonomous systems commonly encounter these situations—for example, an inspection robot moving only in a forward direction might only obtain visual observations from parallel viewpoints, which makes triangulation of visual features more difficult. In addition, a robot operating in man-made environments often encounters objects such as textureless walls, which are difficult to reconstruct using only vision. As a result, radiance field reconstruction of featureless regions (e.g., the ground reconstruction using Nerfacto in Fig. 6; the sky and clouds reconstructed as floating points on the top of the scene [5]) and objects with limited viewpoints (e.g., reconstruction of a wall viewed from a single direction shown in Fig. 12) are often inaccurate. These challenges affect not only volume-density-based fields such as NeRFs, but also SDF-based radiance fields such as NeuS [6] and 3-D Gaussian representations [4] (e.g., NeuSfacto reconstruction in Fig. 6 and elongated Gaussians described in MonoGS [7]). In addition, the implicit representation used in NeRF, while providing tremendous model size compression compared to explicit representations such

as 3DGS [4], can generate reconstruction artifacts in unobserved regions of space. This is due to NeRF formulation as a continuous volumetric field over  $\mathbb{R}^3$ , which enables it to produce outputs even in areas lacking observation. Unlike classical visual SLAM/MVS methods (such as MonoSLAM [8] and OpenMVS) that estimate the uncertainties for the visual reconstructions to tackle reconstruction artifacts, the NeRF representation itself has no notion of uncertainties. These factors limit the use of radiance fields in real-world robotic applications where 3-D reconstructions have to be reliable.

The challenges of 3-D reconstruction using only visual sensing can be mitigated by fusing range sensing devices such as lidar that provides accurate geometric measurements regardless of object texture. Several recent reconstruction systems [9], [10], [11] have adopted a lidar-visual radiance field approach, since both RGB color and depth measurements can be obtained (in a differentiable manner) from the radiance field, and then, supervised by lidar and camera data. However, lidar measurements are sparse, and have limited range and field-of-view (FOV). As a result, not all parts of the scene can be scanned by a lidar sensor. For example, the lidar in Fig. 10 has a limited FOV, and the top of the buildings are not scanned by it. As another example, the ground shown in the bottom left of Fig. 4 is beyond the lidar's sensing range. For these regions, a lidar-visual radiance field reconstruction relies mainly on visual data, and the accuracy then depends on the conditioning of visual reconstruction as discussed in the previous paragraph. Because of this, it is crucial to properly quantify the individual contributions of visual and lidar information in the resultant lidar-visual radiance field reconstruction. However, this issue is relatively underexplored in the literature.

The reconstruction task becomes even more challenging when the scene is large-scale (e.g., urban districts). It is time consuming to estimate sensor trajectories by running an incremental SfM algorithm such as COLMAP [1] for a large-scale scene. In addition, SfM is not guaranteed to succeed in registering all input images (especially if parts of the scene have poor lighting), which could then 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 [12] or the number of 3-D 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 [13] or evenly partitioning the scene using a grid [14]. The limitation of a simple grid-based partition is that the view orientation and visibility are not considered. These factors are, however, important to consider when 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.

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 3-D reconstructions which can also synthesize photorealistic novel views. SiLVR builds upon existing NeRF research

and the Nerfacto implementation [15], which utilizes hash encoding [12] that is significantly faster than multilayer perceptron (MLP)-based NeRF [3] (it takes less than 5 min 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 modeling featureless areas where geometry cannot be accurately reconstructed using 3-D SfM features [16]. 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 [17]. Two examples of large-scale reconstructions from SiLVR are shown in Fig. 1.

Compared to prior NeRF-based reconstruction systems that incorporate lidar [9] or depth cameras [17], [18], 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 view selection [19] and navigation within an NeRF map [20]. After training, SiLVR computes the epistemic uncertainty of the NeRF with the Laplace approximation (LA) [21] 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 [22] and use the spatial variance of the field as the measure of epistemic uncertainty. The estimated uncertainty is used to filter out reconstruction artifacts that improves the final reconstruction accuracy. This is particularly important when merging NeRF submaps since these submaps often contain artifacts caused by limited observation at the submap boundaries. In addition, we adapt a previously developed lidar SLAM algorithm [23], [24] to bootstrap the SfM component and to enforce an accurate metric scale. For mapping large-scale building complexes or a city block, we adopt a submapping approach and apply a graph partitioning algorithm [25] 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 artifacts at their boundaries compared to methods that only consider spatial information [9].

In summary, our contributions are as follows.

- 1) An uncertainty-aware NeRF reconstruction system fusing lidar and vision that can reconstruct large-scale outdoor environments.
- 2) Epistemic uncertainty quantification of the multimodal NeRF pipeline as a principled approach to quantify the contribution of lidar and visual information to a 3-D reconstruction, which can be used to improve reconstruction accuracy, especially at submap boundaries. Our method extends the vision-only uncertainty estimation framework proposed in BayesRays [22] 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 that have also not been scanned by lidar).

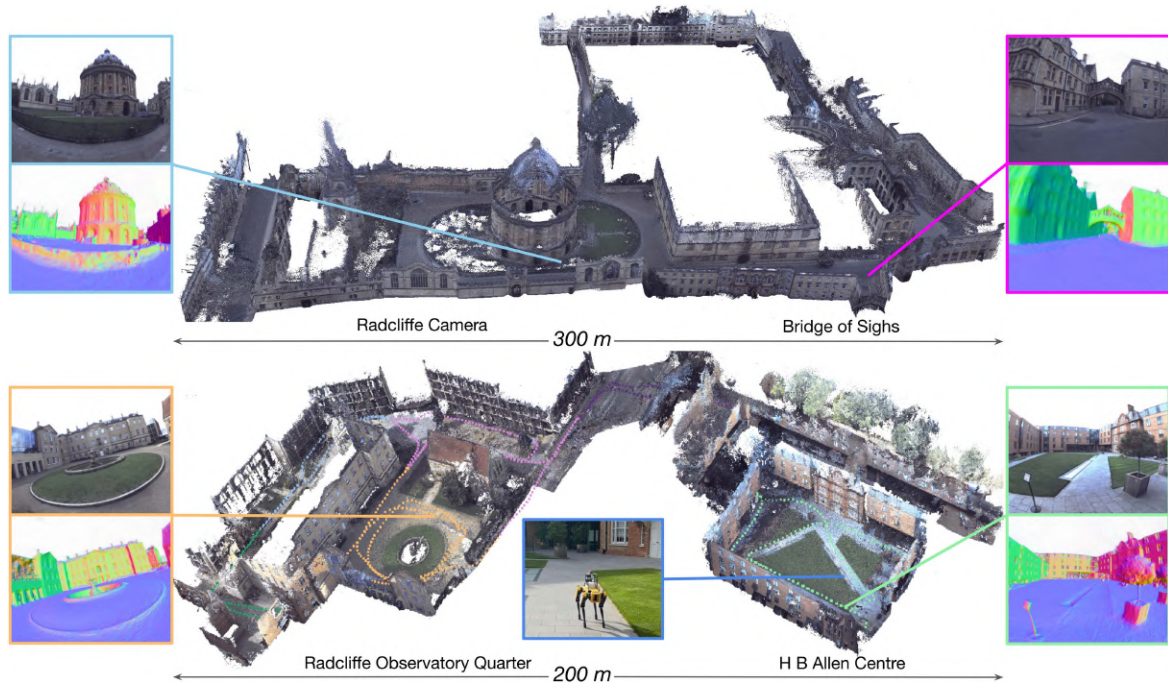


Fig. 1. Two large-scale reconstructions generated by SiLVR. Rendered RGB and surface normal images from the reconstructions are shown on each side. SiLVR combines visual and lidar information to create geometrically accurate maps with photorealistic textures, while considering sensor uncertainty. SiLVR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

- 3) A submapping strategy that leverages per-image visibility information. Compared to the distance-based clustering method [9], we develop a visibility-based clustering method which reduces visual overlap between submaps and in turn creates fewer artifacts at submap boundaries.
- 4) Large-scale evaluation using two large-scale datasets from the Oxford Spires dataset [26] with quantitative results from millimeter-accurate 3-D ground truth. Further comparison is made to baseline radiance field methods that use SDF [6] and 3-D Gaussians [4] representations.

## II. RELATED WORK

In this section, we first review classical 3-D reconstruction methods based on lidar sensors or cameras, followed by more recent approaches using radiance field representations. We then discuss methods to quantify the uncertainty of an NeRF reconstruction, and techniques that can extend the NeRF methods to large-scale environments.

### A. Classical 3-D Reconstruction

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

Lidar is the dominant sensor used for accurate 3-D reconstruction of large-scale outdoor environments [27], [28]. It actively transmits laser pulses to measure ranges, and as a result, is

accurate even at ranges beyond 100 m. With these distance measurements, Lidar odometry typically uses a variant of iterative closest point, and often integrates high-frequency inertial measurement unit (IMU) measurements [23], [29], [30], [31]. 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 optimization [32] that 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 [33], [34], voxels [35], [36], [37], [38], and wavelets [39]. Despite its advantages, lidar has its own limitations. Lidar sensors are much more expensive than cameras, and their measurements are typically much sparser than camera images. The measurements have inherent noise with ranging errors in the order of centimeters, which makes it difficult to reconstruct small objects accurately in indoor scenes. Finally, lidar data contains no texture or color, so the final reconstruction is only geometric and cannot be used for applications such as view synthesis, which requires texture.

Alternatively, textured reconstructions can be recovered from camera images alone via SfM. Given the correspondences between images, an SfM system [1] can optimize a set of camera poses, camera intrinsics, and 3-D sparse feature points. This can then be used by an MVS system [40] to compute dense depth for each frame and in turn to create a dense 3-D point cloud. Compared to lidar, cameras are much more affordable, and also provide texture and color. 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, poor lighting conditions, and non-Lambertian materials.

When the scene to be reconstructed is large-scale (e.g., urban districts or multiroom indoor environments), computer memory becomes a limiting factor. Attempting to map a large scene while constraining the output map size might result in a lower resolution reconstruction lacking detail. A common strategy to extend dense reconstructions to large-scale areas is to divide the scene into submaps [41]. For visual reconstruction with many thousands of images [42], a submapping approach can significantly reduce computation time and memory requirements. One approach used in large-scale MVS is the submap partitioning [43], 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 truncated signed distance function (TSDF)]. These systems construct submaps that are attached to a pose graph [44], [45], [46], and can deform each submap by reoptimizing the pose graph upon loop closure.

### B. Radiance Field Representation

NeRFs were first proposed in the seminal paper from Mildenhall et al. [3]. The technique uses an MLP to represent a continuous radiance field, and uses differentiable volume rendering to reconstruct novel views. It minimizes the photometric error between the rendered image and the input image, which implicitly achieves multiview consistency. NeRF and its many variants use frequency encoding [47] 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 [12], [48] and more recently 3-D Gaussians [4], are shown to accelerate the training, at the cost of being more memory intensive. Octree or sparse-grid structures [12], [49] can reduce memory usage by pruning grid-features in empty space. Our work is built upon Nerfacto from the open-sourced Nerfstudio project [15]. It incorporates the main features from other NeRF works [12], [50], [51] that have been found to work well with real data.

While NeRFs excel at high-quality view synthesis, obtaining a 3-D surface reconstruction of similar quality remains challenging, mainly due to the flexible volumetric representation being underconstrained by the limited multiview inputs. One approach to improve the reconstruction is to impose depth regularization [5], [16] or surface normal regularization [17], 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 SDF [52], [53] and 2-D Gaussians [54] to enforce a surface reconstruction output, although the novel view synthesis quality might be compromised [6] with this approach. Our method uses a volume density representation, which is extended with depth [16] and surface normal [17] regularizations from lidar measurements instead of using SfM [16] or learnt priors [17]. This can significantly improve the reconstruction quality in texture-less areas with smooth surfaces.

Neural field representations have been used for lidar-based mapping [55], [56], [57], 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 multiview geometry constraints. Because of this, it can reconstruct regions outside of the lidar's FOV.

### C. Uncertainty in NeRFs

The standard formulation of NeRF has no notion of uncertainty. The lack of uncertainty makes it difficult to apply them in robotics applications because an NeRF reconstruction could contain artifacts. From a Bayesian machine learning perspective, one could model the data uncertainty (aleatoric uncertainty) and model uncertainty (epistemic uncertainty) [58] 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 [59] and appearance encoding [51] 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 initializations [60]. BayesRays [22] proposes to model the uncertainty of a perturbation field instead, and estimates the uncertainty with the LA. 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 NeRFs

Submapping has been used in NeRF representations for city-scale reconstruction. There are several partitioning strategies that are based on grid partitioning [14] or using road intersections [13]. Merging NeRF submaps is difficult, since each NeRF submap's boundary can be ambiguous, and the appearance encoding of each submap can be different [51]. Block-NeRF [13] merges submaps by first selecting submap candidates based on distance and visibility, and combines submaps in the 2-D image space with interpolation and test-time appearance matching. These methods either require manual submap partitioning [13], or partition the scene into 2-D grids [14]. Our work adopts the submapping approach, and develops partitioning strategies based on visibility, which is advantageous compared to 2-D 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 that 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 3-D location  $\mathbf{p} = (p_x, p_y, p_z)$  and 2-D viewing direction  $\mathbf{d} = (\phi, \psi)$ , the output is an emitted color  $\mathbf{c} = (r, g, b)$  and a volume density  $\sigma$ . To render a novel view using an 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-color by integrating over the set of points sampled along the ray. The pixel color  $\hat{\mathbf{C}}(\mathbf{r})$  is computed by the volume rendering integral, which is approximated using the quadrature rule [61], [62] as

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

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

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

where  $\delta_i$  is the distance between adjacent samples, and  $\sigma_i$  is the volume density of  $\mathbf{p}_i$ .

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

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

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

#### 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 nontextured 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 FOV compared to a surface that is densely scanned. Quantifying these uncertainties associated with the NeRF reconstruction allows one to identify the unreliable reconstructions and filter them out accordingly, hence improving reconstruction accuracy. It can also enable downstream tasks, such as view selection for more accurate and complete mapping.

The Bayesian probability theory provides useful tools for quantifying the uncertainties in neural models [58], which can benefit the NeRF reconstruction. For a regression task with the dataset  $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$  (where  $x_n, y_n$  are the  $n$ th pair of the input and output) and model parameters  $\theta$ , 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 minimize the total training loss  $\mathcal{L}(\mathcal{D}; \theta)$  with respect to the NeRF parameters  $\theta$  (e.g., using the photometric loss from (3) if only vision is provided). This is equivalent to computing the maximum a posteriori (MAP) estimate  $\hat{\theta}$

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

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 regularization that corresponds to the log-prior.

It can be seen from (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 unnormalized posterior  $p(\mathcal{D}|\theta)p(\theta)$  as follows:

$$p(\theta | \mathcal{D}) = \frac{1}{Z} p(\mathcal{D} | \theta) p(\theta) = \frac{1}{Z} \exp(-\mathcal{L}(\mathcal{D}; \theta)) \quad (5)$$

where the normalizing constant  $Z$  is the normalizing constant, and is defined as

$$Z = \int p(\mathcal{D} | \theta) p(\theta) d\theta. \quad (6)$$

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

#### C. Laplace Approximation (LA)

LA is a technique to estimate the otherwise intractable posterior distribution as a Gaussian function. This allows one to efficiently approximate the posterior distribution, and has been used widely in Bayesian deep learning literature [63]. In this section, we describe the details of the LA, following the presentation by Daxberger et al. [21].

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

$$\mathcal{L}(\mathcal{D}; \theta) \approx \mathcal{L}(\mathcal{D}; \hat{\theta}) + \frac{1}{2} (\theta - \hat{\theta})^{\top} \mathbf{H} (\theta - \hat{\theta}) \quad (7)$$

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

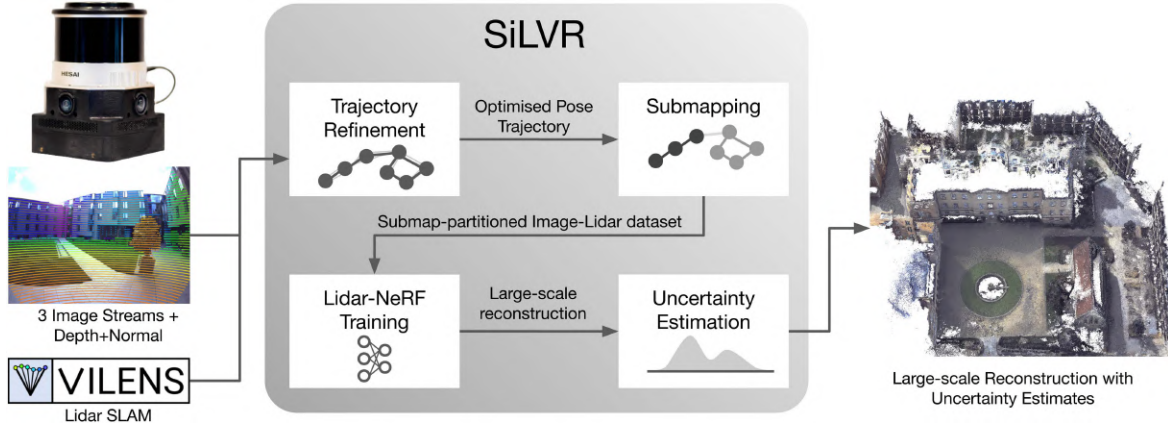


Fig. 2. System overview: SiLVR builds large-scale reconstructions using images and lidar data, and a pose trajectory estimated by a separate odometry system. The sensor streams are provided by the *Frontier*, our custom perception payload carrying three fisheye color cameras, IMU measurements, and a 3-D lidar. When collecting the data, we used VILENS [23] to estimate the trajectory of the sensor, which is refined in postprocessing using COLMAP [1] and partitioned into submaps. The camera image, lidar depth, and a derivative normal image are used to train an NeRF to achieve a final 3-D 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.

Substituting the approximation in (7) into (6), we obtain

$$\begin{aligned}
 Z &= \int p(\mathcal{D} | \theta) p(\theta) d\theta \\
 &= \int \exp(-\mathcal{L}(\mathcal{D}; \theta)) d\theta \\
 &\approx \exp(-\mathcal{L}(\mathcal{D}; \hat{\theta})) \int \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^\top \mathbf{H}(\theta - \hat{\theta})\right) d\theta \\
 &= \exp(-\mathcal{L}(\mathcal{D}; \hat{\theta})) \frac{(2\pi)^{\frac{D}{2}}}{(\det \mathbf{H})^{\frac{1}{2}}} \quad (8)
 \end{aligned}$$

where  $D$  denotes the dimensionality of the parameters  $\theta$ .

We then substitute the Taylor expansion (7) and expression of the normalization constant (8) into the posterior (5)

$$p(\theta | \mathcal{D}) \approx \frac{(\det \mathbf{H})^{\frac{1}{2}}}{(2\pi)^{\frac{D}{2}}} \exp\left(-\frac{1}{2}(\theta - \hat{\theta})^\top \mathbf{H}(\theta - \hat{\theta})\right) \quad (9)$$

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

By using LA, the uncertainty estimation problem can be formulated as estimating the Hessian matrix  $\mathbf{H}$  at the MAP estimate  $\hat{\theta}$ . We describe how we apply this to the NeRF reconstruction problem in Section IV-C2, and techniques to further approximate the Hessian matrix in Section IV-C3.

#### IV. METHOD

In this section, we present SiLVR, a 3-D reconstruction system based on an NeRF representation. An overview of our system is presented in Fig. 2. SiLVR takes in as input synchronized triplets of a camera image, a lidar depth image, and the corresponding surface normals, as well as the sensor trajectory estimated by an online Lidar SLAM system. This trajectory is refined and partitioned into submaps. For each submap, we train an NeRF that combines both vision and lidar data. After training the NeRF,

SiLVR estimates the epistemic uncertainty of the NeRF. The final point cloud reconstruction is extracted from the NeRF based on rendered depth, and filtered using the per-point uncertainty estimates.

We describe our approach to extend the NeRF formulation to include lidar data in Section IV-A, and how to remove the sky reconstruction in Section IV-B. We then introduce the epistemic uncertainty estimation method in Section IV-C. Finally, we present how SiLVR achieves scalability by partitioning large-scale poses into submaps in Section IV-D.

##### A. Geometric Constraints From Lidar Measurements

Image-based 3-D reconstruction with NeRF becomes challenging when a surface has a 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 Section 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<sup>1</sup> to the nearest image's timestamp, and hence, synchronized with the image data. Example overlays can be found in Fig. 4.

1) *Lidar Depth Constraints*: We follow the depth regularization approach proposed by DS-NeRF [16] for RGB-D cameras. We define the rendering weight distribution  $w(t)$  as a discrete probability distribution, with probabilities given by  $w_i$ , the weights of the ray samples defined in (2). The depth regularization term is obtained by minimizing the Kullback-Leibler (KL) divergence between a narrow normal distribution centered at the lidar depth measurement  $\mathbf{D}$  and the rendering weight

<sup>1</sup>See implementations at [https://github.com/ethz-asl/lidar\\_undistortion](https://github.com/ethz-asl/lidar_undistortion)

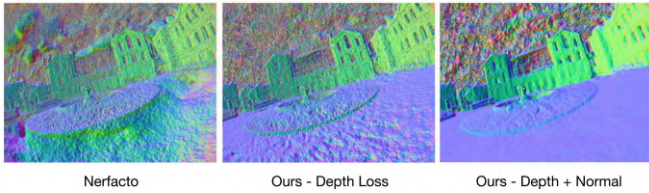


Fig. 3. Comparison of surface normal renderings of the Maths Institute. Incorporating normal constraints in addition to depth from lidar improves the smoothness of the reconstruction. (Right) Smooth reconstruction of the ground portion highlights this improvement.

distribution  $w(t)$  as follows:

$$\mathcal{L}_{\text{Depth-KL}} = \sum_{\mathbf{r} \in \mathcal{D}_d} \text{KL}[\mathcal{N}(\mathbf{D}, \hat{\sigma}) \| w(t)]. \quad (10)$$

During training, we apply this regularization using a coarse-to-fine schedule by progressively reducing the covariance  $\hat{\sigma}$  of the target Normal distribution. This encourages the rendering weight distribution  $w(t)$  to approach a Dirac delta function. As a result, the density along the ray is encouraged to be concentrated near the lidar depth.

2) *Surface Normal Constraints From Lidar Measurements:* While the depth loss improves 3-D reconstruction, we found that the surface it produces will still contain wavy artifacts in regions where it is expected to be smooth (see Fig. 3). To mitigate this, we regularize 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  $\sigma$ . 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 regularization loss in the NeRF training, inspired by [17]

$$\mathcal{L}_{\text{Normal}} = \sum_{\mathbf{r} \in \mathcal{D}_n} \|\hat{N}(\mathbf{r}) - \bar{N}(\mathbf{r})\|_1 + \left\| 1 - \hat{N}(\mathbf{r})^\top \bar{N}(\mathbf{r}) \right\|_1. \quad (11)$$

Compared to learning-based surface normal estimation from the camera image used in [17], our surface normal is estimated from the 3-D lidar point cloud and does not suffer from generalization issues. The effect of the surface normal regularization can be seen in Fig. 3.

### 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 artifacts in the final reconstruction. To remove these “sky points” from the training procedure, we used a semantic segmentation network<sup>2</sup> to obtain a sky mask that is used to regularize 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 minimize the weights of all samples on these rays, similar to [5]

$$\mathcal{L}_{\text{Sky}} = \sum_{\mathbf{r} \in \mathcal{D}_s} \sum_i w_i^2. \quad (12)$$

Combining the loss terms introduced in (3), (10), (11), and (12), our overall training loss is

$$\mathcal{L} = \mathcal{L}_{\text{color}} + \mathcal{L}_{\text{Depth-KL}} + \mathcal{L}_{\text{Normal}} + \mathcal{L}_{\text{Sky}}. \quad (13)$$

### C. Epistemic Uncertainty of the NeRF Reconstruction

We aim to obtain an explicit metric of uncertainty of our NeRF reconstruction. Particularly, following Section 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 LA. We first describe the reformulation of the parameters  $\theta$ , 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  $\theta$  do not directly correspond to the 3-D 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 3-D location. This is in contrast to other approaches, such as 3DGS [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 an NeRF, we adopt the perturbation field formulation introduced in BayesRays [22]. Specifically, we construct the perturbation field  $\mathcal{P}$  of every 3-D coordinate  $\mathbf{x}$ . We define the perturbation field as  $\mathcal{P}_{\theta_P}(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , where  $\theta_P$  denotes the parameters in the perturbation field. A 3-D coordinate’s perturbation can be obtained using trilinear interpolation

$$\mathcal{P}_{\theta_P}(\mathbf{x}) = \text{Trilinear}(\mathbf{x}, \theta_P). \quad (14)$$

When estimating the uncertainty, we modify the volume rendering procedure by adding the perturbation field to the point samples  $\mathbf{p}_i$  along the ray  $\mathbf{r}$  to produce the new point sample  $\mathbf{p}'_i = \mathbf{p}_i + \mathcal{P}_{\theta_P}(\mathbf{p}_i)$ . The new point color and volume density  $(\mathbf{c}'_i, \sigma'_i) = R(\mathbf{p}'_i, \mathbf{d})$ , and the new pixel color  $\hat{\mathbf{C}}'(\mathbf{r})$  can then be computed using (1) and (2).

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

2) *Epistemic Uncertainty Estimation Using LA:* We estimate the epistemic uncertainty of the NeRF reconstruction by estimating the covariance of the posterior  $p(\theta|\mathcal{D})$ . Using the LA technique described in Section III-C, we estimate the otherwise intractable posterior distribution as a Gaussian function, and then, use its covariance as the estimated uncertainty of our reconstruction.

<sup>2</sup>We used Detectron2 from <https://github.com/facebookresearch/detectron2>

Specifically, the mean of the Gaussian  $\hat{\theta}$  is the MAP estimate of the parameters  $\theta$ . Here, the parameters  $\theta$  that we are estimating are the perturbation field  $\theta_P$  introduced in Section IV-C1. Assuming that the NeRF reconstruction has converged to a local minima after training, a small perturbation should not cause the reconstruction to deteriorate, 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

$$\begin{aligned} \mathbf{H} &= \nabla_{\theta}^2 \mathcal{L}(\mathcal{D}; \theta)|_{\hat{\theta}} \\ &= -\gamma^{-2} \mathbf{I} - \sum_{n=1}^N \nabla_{\theta}^2 \log p(y_n | f_{\theta}(x_n))|_{\hat{\theta}}. \end{aligned} \quad (15)$$

3) *Approximation of the Hessian Matrix*: While the second term in the Hessian from (15) is usually intractable, it can be approximated by the Fisher information matrix [64]

$$\mathbf{H} \approx -\gamma^{-2} \mathbf{I} - \sum_{n=1}^N \mathbf{J} \mathbf{J}^{\top} \quad (16)$$

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 the others, and use a diagonal approximation to the Hessian

$$\mathbf{H} \approx -\gamma^{-2} \mathbf{I} - \text{diag}(\mathbf{J} \mathbf{J}^{\top}). \quad (17)$$

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

$$H_{ii} \approx \sum_{j=1}^n \left( \frac{\partial \ell_j}{\partial \theta_i} \right)^2 + \gamma^{-2}. \quad (18)$$

The Hessian matrix is initialized as all zeroes; i.e., infinite covariance. After accumulating the gradients, the prior term  $\gamma^{-2}$  in (18) ensures that the final Hessian matrix is positive definite and the covariances are bounded. 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 color and density in some locations, leading to “hallucinated” reconstruction points.

4) *Uncertainty for Heterogeneous Sensors*: The NeRF model is trained with a total training loss function  $\mathcal{L}(\mathcal{D}; \hat{\theta})$  that contains the photometric loss  $\mathcal{L}_{\text{Color}}$  and the depth loss  $\mathcal{L}_{\text{Depth-KL}}$ . This means that the Jacobian  $\mathbf{J}$  can be decoupled into a color component  $\mathbf{J}_{\text{Color}}$  and depth component  $\mathbf{J}_{\text{Depth-KL}}$ , from which we can then approximate the Hessian that corresponds only to the visual information  $\mathbf{H}_{\text{Color}}$ , and the Hessian that

TABLE I  
DETAILS OF REAL-WORLD DATASETS THAT ARE USED FOR EVALUATION

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

corresponds to only to the lidar depth information  $\mathbf{H}_{\text{Depth-KL}}$ . 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  $\mathcal{L}_{\text{Normal}}$ . In our study, we focus on just the photometric loss  $\mathcal{L}_{\text{Color}}$  and depth loss  $\mathcal{L}_{\text{Depth-KL}}$ . 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 color. As later shown in Fig. 8, “low” visual uncertainty corresponds to distinct visual features and high-frequency areas. “High” visual uncertainty typically corresponds to areas with uniform texture, since perturbing a point in an area with similar colors can lead to small changes in the rendered color. 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 FOV.

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) SfM 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 h (as shown in Table IV)];
- 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 [23]. 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* artifacts in the images rendered by the NeRF model. Several works [13], [18] use noisy pose inputs, and then, jointly refine the poses within the NeRF optimization to generate better results. However, as shown later in Table IV, our experiments showed that this pose-refinement approach can produce results that 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 SfM 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 initialization. For a mission spanning over 20 min, 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 Section IV-A1 are also metric. The depth regularization 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 [9]. 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 covisibility metric. Two images are covisible if some feature points on one image can be viewed from the other image. Specifically, the covisibility 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 normalized cuts algorithm [25] 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 covisible 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 color and depth using the training data rays, and we filter out points that have high uncertainty. Rendering at the submap intersections can be obtained by combining two submap renderings in the image space where the weights are determined based on the distances to the neighboring submap, following Block-NeRF [13].

## 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 fisheye color<sup>3</sup> Alphasense cameras (from Sevensense Robotics AG) on three sides of the device, as well as a synchronized IMU. The three-camera setup enables omnidirectional NeRF mapping from a single walking pass through a test site. We installed a Hesai QT64 lidar (104° FOV, 60-m 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 [65]), or simply handheld (using the Oxford Spires dataset [26]). Some data were 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 millimeter-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, U.K. The large-scale sites, ROQ and the Bodleian Library, cover areas of 5000 m<sup>2</sup> and 15 000 m<sup>2</sup>, respectively. The hardware details are in Table I, and some sample lidar-camera overlays are shown in Fig. 4.

### B. Implementation Details

1) *Data Collection and Processing*: When collecting the data, we use VILENS [23], 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 optimized using bundle adjustment as described in Section IV-D1. This improves the visual reconstruction quality, as shown later in Table IV. Individual lidar scans are projected to form a sparse depth image coinciding with the camera image (i.e., the same camera pose

<sup>3</sup>To produce RGB images, we debayer and white-balance the raw bayered images using [https://github.com/leggedrobotics/raw\\_image\\_pipeline](https://github.com/leggedrobotics/raw_image_pipeline)

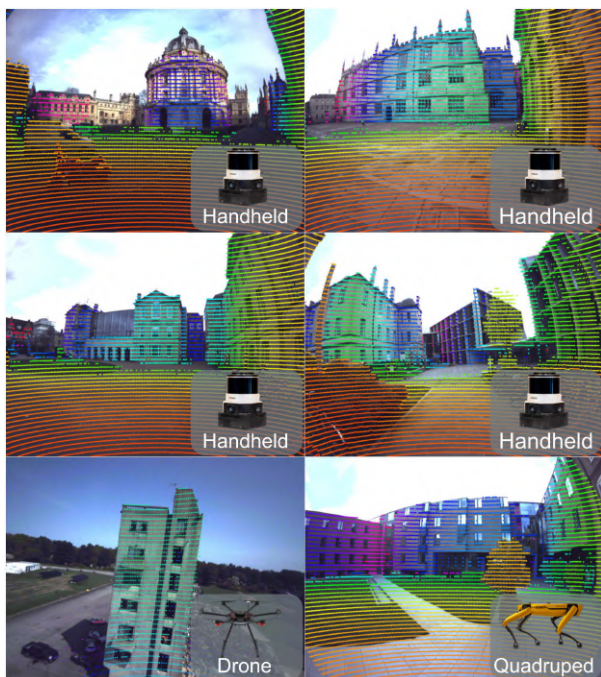


Fig. 4. Sample data from our diverse robotic datasets. Here, each image is overlaid with a projected lidar point cloud to demonstrate the accuracy of the sensor calibration.

and intrinsic parameters). Surface normals of the lidar points are estimated as the surface normal of the polygon formed by the current point’s neighboring points in the lidar range image using Newell’s method, and projected as a sparse normal image. We use the calibrations provided by the Oxford Spires dataset [26]. In this dataset, the intrinsics and extrinsics of the set of cameras are estimated using Kalibr [66], and a single extrinsic transformation between the three cameras and the lidar is estimated using DiffCal [67]. When running COLMAP, we further optimize the camera intrinsics produced by Kalibr.

2) *NeRF Reconstruction*: Our NeRF reconstruction system extends Nerfacto, which is a specific vision-only pipeline implemented within the Nerfstudio framework [15]. Nerfacto’s rendering quality is comparable to state-of-the-art methods such as MipNeRF-360 [50] while achieving a substantial acceleration in reconstruction speed as it also incorporates efficient hash encoding, which was proposed by the authors of Instant-NGP [12]. The scene contraction, proposed in [50], is also used to improve memory efficiency and to represent scenes with high-resolution content near the input camera locations. The contraction function nonlinearly 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 are often large variations in exposure and lighting conditions. Because of this, we use a per-frame appearance encoding for each image, similar to [5] and [51]. To train the NeRF model, we used an Nvidia RTX 4090. One training iteration takes 4 096 rays.

### C. Evaluation Details

1) *3-D Reconstruction Metrics*: To evaluate the geometry of the reconstruction, we report *Accuracy* and *Completeness*

following the conventions of the DTU dataset [68]. Accuracy is measured as the point-to-point distance from the reconstruction to the (ground truth) reference 3-D model and indicates the reconstruction quality. Completeness is the distance from the pointwise 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 predefined error threshold. A point in the reconstruction, which is below this threshold can be considered to be a true positive. We use both 5 and 10 cm for the threshold following [26].

2) *Map Filtering for Fair Evaluation*: Perfect accuracy and completeness scores (of zero in both cases) would be achieved if the two point clouds were identical, and any deviation is penalized 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 that 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* as shown in Fig. 7.

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* as shown in Fig. 7, and the completeness score would again be artificially higher than it should be.

These overestimated error measures are typically much higher than the errors that 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 nonoverlapping 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 [35]. 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) [69], 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 3-D Reconstruction

We perform a quantitative evaluation of our method using real-world datasets captured by different robotic platforms. Of

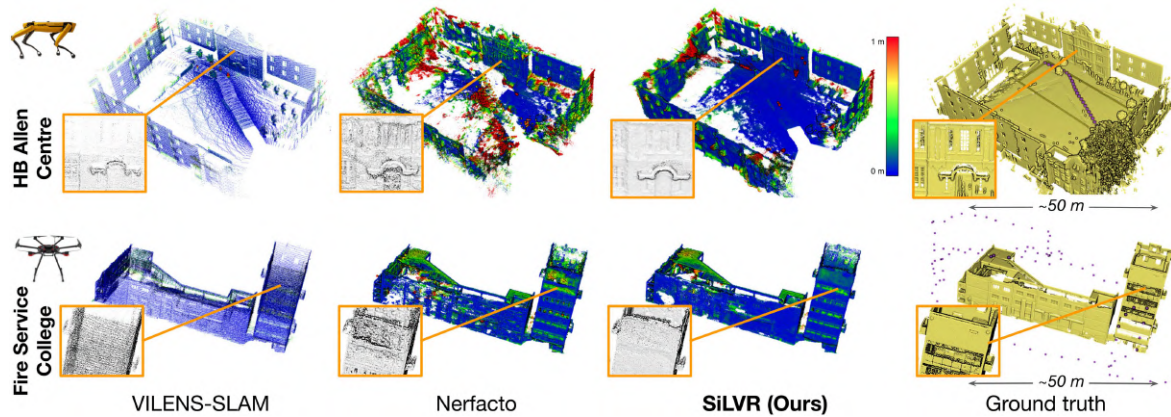


Fig. 5. Comparison of reconstruction quality of VILENS-SLAM, Nerfacto (vision-only) and our approach in small-scale scenes. Reconstructions are colored using the point-to-point distance between the respective reconstructions and the ground-truth scan with increasing error from blue (0 m) to red (1 m). 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.

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 [23] and pose graph optimization [24].
- 2) *Poisson Reconstruction* [70]: Surface reconstruction using point clouds from VILENS-SLAM.
- 3) *OpenMVS<sup>4</sup>*: Multiview stereo reconstruction.
- 4) *Nerfacto* [15]: Vision-only radiance field reconstruction using volume density.
- 5) *NeuSfacto* [6]: Vision-only radiance field reconstruction using SDF.
- 6) *Splatafacto* [71]: Vision-only radiance field reconstruction using 3-D Gaussians, which are initialized 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, Splatafacto, and SiLVR).

We summarize the quantitative results in Tables II and III, and show the 3-D reconstructions in Figs. 5 and 6. Among all methods, the 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

<sup>4</sup>Available at <https://github.com/cdseacave/openMVS>

TABLE II  
EVALUATION OF 3-D RECONSTRUCTION QUALITY OF SMALL SCENES

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

described in Section 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 3-D Gaussians exported by Splatafacto also cannot reconstruct the ground accurately. These 3-D Gaussians are located mostly on the visual features of the sites—since they are initialized 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. 6), this advantage is less prominent.

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

TABLE III  
EVALUATION OF 3-D RECONSTRUCTION QUALITY

Method	Accuracy↓	Completeness↓	Precision	5cm Recall	F-Score	Precision	10cm Recall	F-Score	PSNR↑		SSIM↑	LPIPS↓
	(m)	(m)							train	test	test	
Radcliffe Observatory Quarter (ROQ)												
VILENS-SLAM	<b>0.077</b>	<b>1.214</b>	<b>0.552</b>	<b>0.367</b>	<b>0.441</b>	<b>0.832</b>	<b>0.625</b>	<b>0.714</b>	/	/	/	/
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	<b>25.71</b>	<b>20.92</b>	<b>0.714</b>	<b>0.490</b>
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
<b>SiLVR (Ours)</b>	<b>0.095</b>	<b>1.803</b>	<b>0.416</b>	<b>0.150</b>	<b>0.221</b>	<b>0.699</b>	<b>0.344</b>	<b>0.461</b>	24.73	20.90	0.653	0.551
Bodleian Library												
VILENS-SLAM	1.017	<b>0.736</b>	<b>0.324</b>	0.098	0.150	<b>0.518</b>	0.290	<b>0.372</b>	/	/	/	/
Poisson Reconstruction	1.256	1.230	0.239	0.104	0.145	0.400	<b>0.334</b>	0.364	/	/	/	/
OpenMVS	<b>0.955</b>	2.257	0.223	<b>0.129</b>	<b>0.163</b>	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	<b>28.92</b>	<b>23.03</b>	0.827	<b>0.715</b>
Splatfacto 3-cam	13.532	1.275	0.020	0.004	0.007	0.044	0.027	0.033	23.92	22.19	<b>0.850</b>	0.748
NeuSfacto 3-cam	2.656	<b>1.074</b>	0.015	0.007	0.010	0.035	0.042	0.038	24.00	20.61	0.619	0.731
<b>SiLVR (Ours)</b>	<b>1.292</b>	1.532	<b>0.129</b>	<b>0.041</b>	<b>0.063</b>	<b>0.276</b>	<b>0.170</b>	<b>0.211</b>	28.00	22.94	0.754	0.779

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.

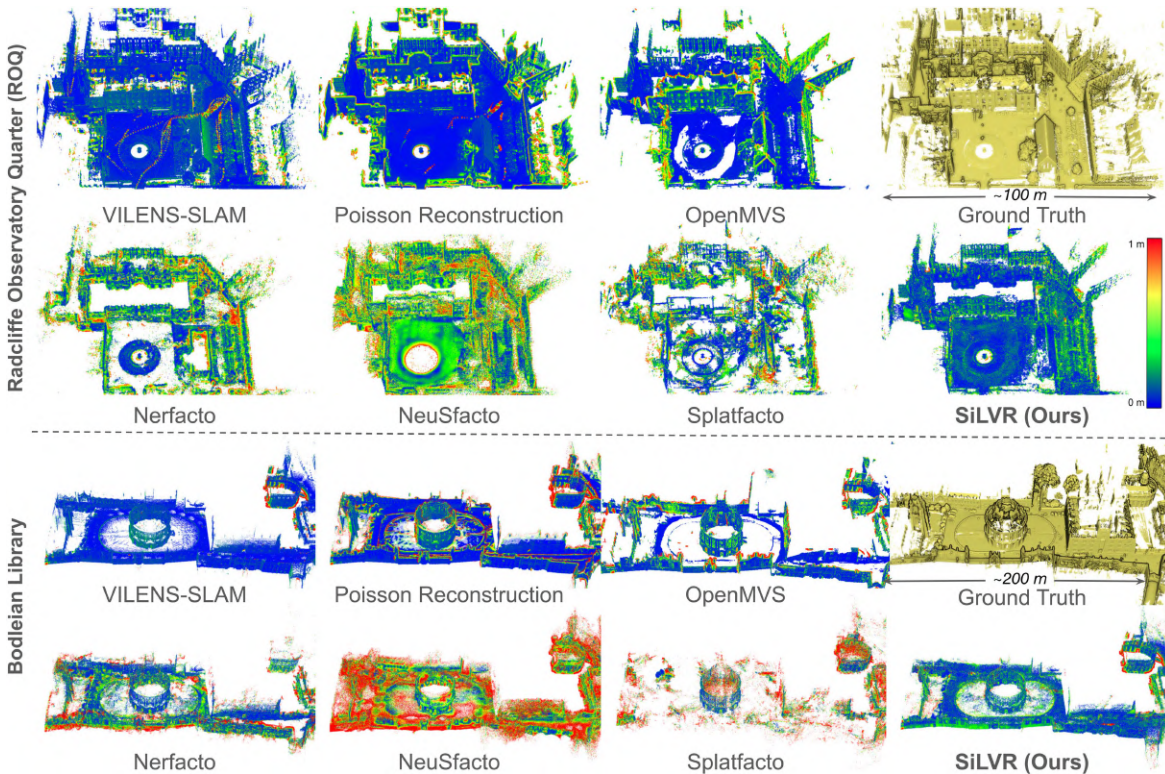


Fig. 6. Comparison of reconstruction quality of our method and other baseline methods in two real-world large-scale scenes. Among the radiance field baseline methods, SiLVR’s reconstruction is the most accurate and complete, especially on the ground where there are insufficient visual constraints.

achieves a balance between the rendering quality and the 3-D 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 are generated using the CARLA simulator [72], 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 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 [22], [73], [74]. 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

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

Fig. 7. Classification of different occupancy categories for the reconstruction and reference models.

TABLE IV  
ABLATION: EFFECT OF BOOTSTRAPPING W/ SLAM POSES

Method	Features	Prior	Traj.	Regis- tered (%)	PSNR $\uparrow$		SSIM $\uparrow$	Time (s)
					Train	Test	Test	
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	<b>20.6</b>	<b>0.74</b>	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	<b>26.6</b>	19.9	0.72	37067.5
	8192	✓		100.0	26.3	20.4	<b>0.74</b>	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.

points are calculated to form a graph. If the uncertainty estimates align perfectly with the actual errors, then the reconstruction points with 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 that focuses on the error of the *remaining* reconstructions, we also analyze 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 normalized (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 (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.

When merging NeRF submaps, uncertainty-based filtering is particularly important. As shown in Fig. 9, the submaps can contain reconstruction artifacts, which interfere with neighboring submaps due to limited observation at the submap boundary. These artifacts can be however identified as they tend to have high epistemic uncertainty, and can be filtered to improve the merged reconstruction accuracy.

The advantage of the disentangled uncertainty estimates is also demonstrated in Fig. 10 where we use a narrow vertical FOV lidar with wider vertical FOV cameras. Here, the lidar is not able to scan the upper part of the two buildings highlighted in Fig. 10, 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 [9]. As shown in Fig. 11, 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 submap's partial

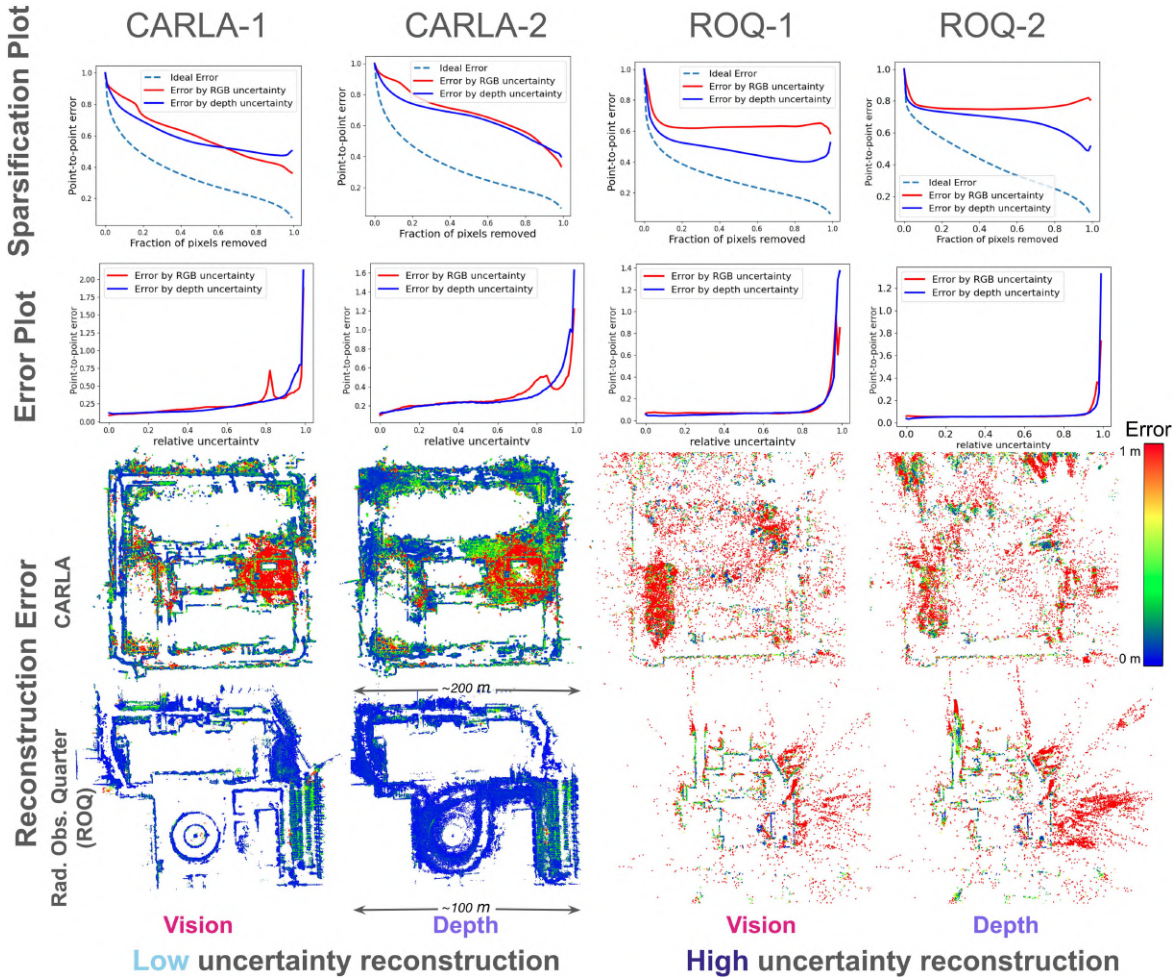


Fig. 8. Qualitative and quantitative evaluation of epistemic uncertainty estimates using both synthetic (CARLA) and real-world (ROQ) datasets. The sparsification plot shows how the (normalized) reconstruction error decreases as more uncertain points are removed, and how close it is to the oracle error reduction curve. The error plot additionally shows the distribution of error at different uncertainty estimates. Low visual uncertainty corresponds to visual features that 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.

reconstruction of that building have a lower quality. In addition, distance-based submapping puts 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. Multicamera Setup Ablation Study

The advantage of our multicamera sensor setup is demonstrated qualitatively in Fig. 12. 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 good-quality rendering only at the input viewing angle. The reconstruction using the front-only camera in Fig. 12 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.

#### 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 [15], SLAM poses refined using COLMAP’s [1] 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 [75].

The results are summarized in Table 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 initialization. Our SLAM prior poses can also register all the images in the trajectory; meanwhile, COLMAP on its own only registers only 55%–95%

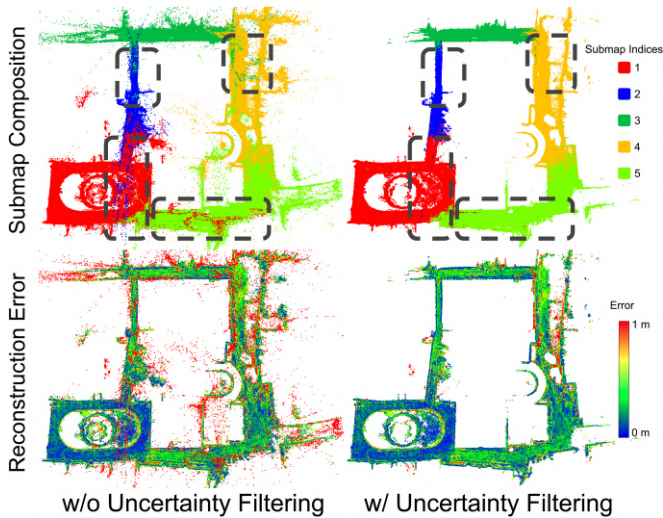


Fig. 9. Comparison between two merged NeRF submaps with and without uncertainty filtering. On the left, the submaps contain artifacts, which extend to neighboring submaps and reduce overall reconstruction quality. These artifacts are mainly due to insufficient observations at the submap boundaries. On the right, we show how uncertainty filtering can be used to remove these artifacts, which leads to a more accurate merged reconstruction.

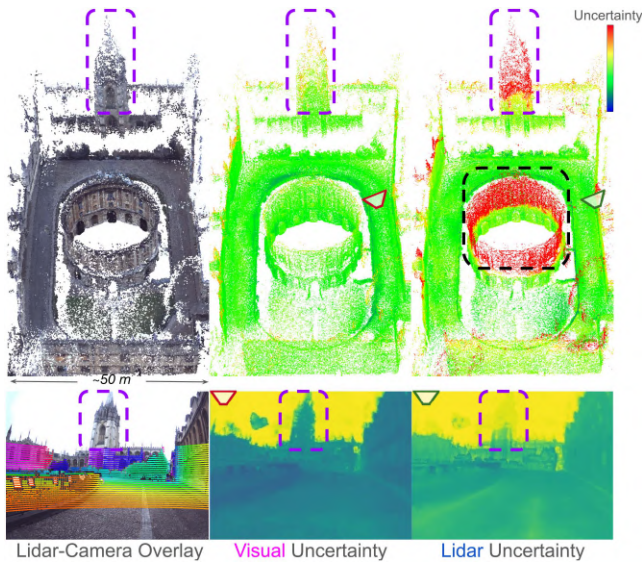


Fig. 10. Comparison between visual and lidar uncertainty. The reconstructions at the top are colored by uncertainty estimates (red is high uncertainty and green is low uncertainty). Here, the lidar used was a Hesai XT32, which has a narrow FOV and cannot scan the upper part of the buildings, and the depth uncertainty estimates can identify such regions (red point clouds). This indicates that only visual information is used when reconstructing these areas.

images. Extracting more visual features per image (from 1 024 to 8 192) 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.

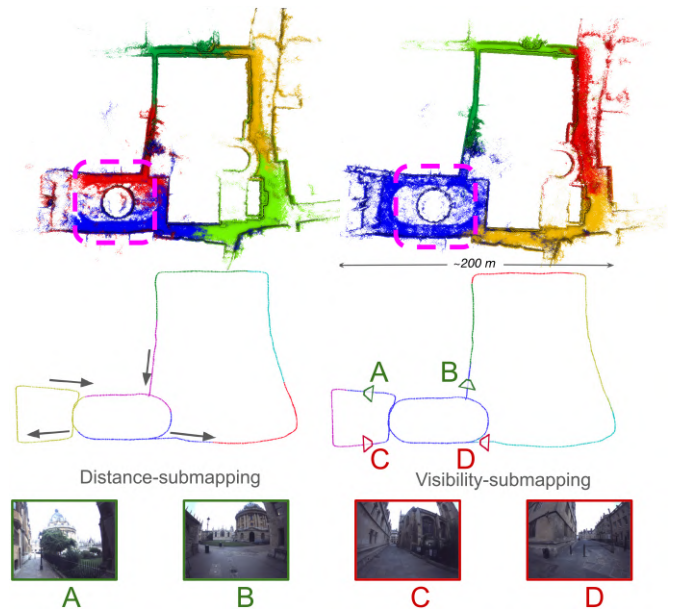


Fig. 11. Comparison of two submapping strategies. Visibility information guides the clustering algorithm to group images looking at the same object together, which then leads to a more accurate and complete reconstruction of that object than algorithms that only consider distances.

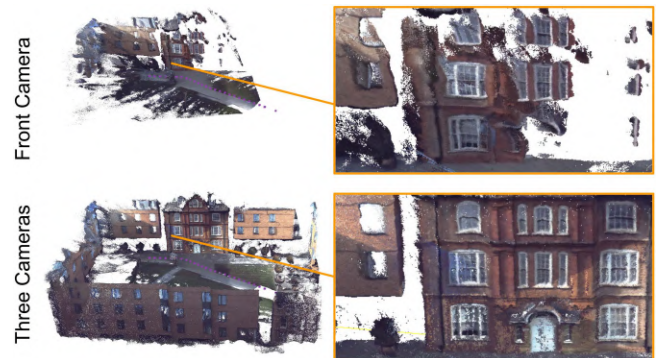


Fig. 12. Comparison of reconstruction of HBAC building using the front camera only versus using all three cameras. The three-camera setup generates more complete and accurate reconstructions compared to using only a single front-facing camera. The multicamera setting is important in robotic applications where it would be infeasible to actively scan the entire scene to obtain strong viewpoint constraints.

## VII. CONCLUSION

In summary, we proposed a large-scale 3-D reconstruction system fusing both lidar and vision in an NeRF. The proposed approach combines the advantages of the two sensor modalities and generates reconstructions with both photorealistic 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 artifacts 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.

## ACKNOWLEDGMENT

The authors would like to thank M. Mattamala for discussion, proofreading, and helping with figures. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) license to any Accepted Manuscript version arising. The work of Maurice Fallon is supported by a Royal Society University Research Fellowship.

## REFERENCES

- [1] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4104–4113.
- [2] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixelwise view selection for unstructured multi-view stereo," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 105–118.
- [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," *Commun. 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 Trans. Graph.*, vol. 42, no. 4, pp. 139–1, Jul. 2023.
- [5] K. Rematas et al., "Urban radiance fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12932–12942.
- [6] 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," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 27171–27183.
- [7] H. Matsuki, R. Murai, P. H. Kelly, and A. J. Davison, "Gaussian splatting slam," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18039–18048.
- [8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.
- [9] 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 *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 17983–17989.
- [10] H. Zhao, W. Guan, and P. Lu, "LVI-GS: Tightly coupled LiDAR-visual-inertial SLAM using 3D Gaussian splatting," *IEEE Trans. Instrum. Meas.*, vol. 74, Mar. 2025, Art. no. 7504810.
- [11] S. Hong et al., "GS-LIVO: Real-time LiDAR, inertial, and visual multi-sensor fused odometry with Gaussian mapping," *IEEE Trans. Robot.*, vol. 41, pp. 4253–4268, Jun. 2025.
- [12] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022.
- [13] M. Tancik et al., "Block-NeRF: Scalable large scene neural view synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8248–8258.
- [14] H. Turki, D. Ramanan, and M. Satyanarayanan, "Mega-NeRF: Scalable construction of large-scale NeRF for virtual fly-throughs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12922–12931.
- [15] M. Tancik et al., "Nerfstudio: A modular framework for neural radiance field development," in *Proc. ACM SIGGRAPH Conf.*, 2023, pp. 1–12.
- [16] K. Deng, A. Liu, J.-Y. Zhu, and D. Ramanan, "Depth-supervised NeRF: Fewer views and faster training for free," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12882–12891.
- [17] Z. Yu, S. Peng, M. Niemeyer, T. Sattler, and A. Geiger, "MonoSDF: Exploring monocular geometric cues for neural implicit surface reconstruction," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 25018–25032.
- [18] D. Azinović, R. Martin-Brualla, D. B. Goldman, M. Nießner, and J. Thies, "Neural RGB-D surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 6290–6301.
- [19] M. Strong, B. Lei, A. Swann, W. Jiang, K. Daniilidis, and M. Kennedy III, "Next best sense: Guiding vision and touch with FisherRF for 3D Gaussian splatting," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 3204–3210.
- [20] M. Adamkiewicz et al., "Vision-only robot navigation in a neural radiance world," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4606–4613, Apr. 2022.
- [21] E. Daxberger, A. Kristiadi, A. Immer, R. Eschenhagen, M. Bauer, and P. Hennig, "Laplace redux-effortless Bayesian deep learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 20089–20103.
- [22] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi, "Bayes' Rays: Uncertainty quantification in neural radiance fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 20061–20070.
- [23] D. Wisth, M. Camurri, and M. Fallon, "VILENS: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 309–326, Feb. 2023.
- [24] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, "Online LiDAR-SLAM for legged robots with robust registration and deep-learned loop closure," in *Proc. IEEE Robot. Automat. Lett.*, May 2020, pp. 4158–4164.
- [25] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [26] 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," *Int. J. Robot. Res.*, 2025, Art. no. 02783649251369905.
- [27] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot., Sci. Syst.*, 2018, pp. 59–68.
- [28] J. Lin and F. Zhang, "R3LIVE: A robust, real-time, RGB-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 10672–10678.
- [29] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proc. Robot., Sci. Syst.*, 2014, vol. 2, pp. 1–9.
- [30] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8729–8736.
- [31] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "FAST-LIO2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, Aug. 2022.
- [32] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, no. 5-6, pp. 403–429, 2006.
- [33] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.
- [34] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic LiDAR fusion: Dense map-centric continuous-time SLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1206–1213.
- [35] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [36] R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2011, pp. 127–136.
- [37] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1366–1373.
- [38] 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 Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1144–1151, Apr. 2018.
- [39] V. Reijgwart, C. Cadena, R. Siegwart, and L. Ott, "Efficient volumetric mapping of multi-scale environments using wavelet-based compression," in *Proc. Robot., Sci. Syst.*, 2023, p. 65.
- [40] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.
- [41] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller, "An Atlas framework for scalable mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, vol. 2, pp. 1899–1906.
- [42] S. Agarwal et al., "Building Rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, 2011.
- [43] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1434–1441.
- [44] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusunur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2175–2182.
- [45] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robot. Automat. Lett.*, vol. 5, no. 1, pp. 227–234, Jan. 2020.

- [46] Y. Wang, M. Ramezani, M. Mattamala, S. T. Digumarti, and M. Fallon, "Strategies for large scale elastic and semantic LiDAR reconstruction," *J. Robot. Auton. Syst.*, vol. 155, 2022, Art. no. 104185.
- [47] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.
- [48] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5501–5510.
- [49] A. Yu, R. Li, M. Tancik, H. Li, R. Ng, and A. Kanazawa, "PlenOctrees for real-time rendering of neural radiance fields," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 5752–5761.
- [50] J. T. Barron, B. Mildenhall, D. Verbin, P.P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded anti-aliased neural radiance fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5470–5479.
- [51] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7210–7219.
- [52] Z. Li et al., "Neuralangelo: High-fidelity neural surface reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8456–8465.
- [53] L. Yariv, J. Gu, Y. Kasten, and Y. Lipman, "Volume rendering of neural implicit surfaces," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 4805–4815.
- [54] B. Huang, Z. Yu, A. Chen, A. Geiger, and S. Gao, "2D gaussian splatting for geometrically accurate radiance fields," in *Proc. ACM SIGGRAPH 2024 Conf. Papers*, 2024, pp. 1–11.
- [55] X. Zhong, Y. Pan, J. Behley, and C. Stachniss, "SHINE-Mapping: Large-scale 3D mapping using sparse hierarchical implicit neural representations," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 8371–8377.
- [56] J. Deng et al., "NeRF-LOAM: Neural implicit representation for large-scale incremental LiDAR odometry and mapping," in *Proc. Int. Conf. Comput. Vis.*, 2023, pp. 8218–8227.
- [57] 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 Trans. Robot.*, vol. 40, pp. 4045–4064, Jul. 2024.
- [58] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5580–5590.
- [59] S. Sabour, S. Vora, D. Duckworth, I. Krasin, D. J. Fleet, and A. Tagliasacchi, "RobustNeRF: Ignoring distractors with robust losses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 20626–20636.
- [60] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6405–6416.
- [61] N. Max, "Optical models for direct volume rendering," *IEEE Trans. Vis. Comput. Graph.*, vol. 1, no. 2, pp. 99–108, Jun. 1995.
- [62] J. T. Kajiya and B. P. Von Herzen, "Ray tracing volume densities," *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 165–174, 1984.
- [63] T. Papamarkou et al., "Position: Bayesian deep learning is needed in the age of large-scale AI," in *Proc. Int. Conf. Mach. Learn.*, 2024, pp. 39556–39586.
- [64] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
- [65] 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 Trans. Field Robot.*, vol. 1, pp. 113–130, Jul. 2024.
- [66] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1280–1286.
- [67] L. F. T. Fu, N. Chebrolu, and M. Fallon, "Extrinsic calibration of camera to lidar using a differentiable checkerboard model," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 1825–1831.
- [68] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *Int. J. Comput. Vis.*, pp. 1–16, 2016.
- [69] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [70] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurographics Symp. Geometry Process.*, 2006, pp. 61–70.
- [71] V. Ye, "gsplat: An open-source library for Gaussian splatting," *J. Mach. Learn. Res.*, vol. 26, no. 34, pp. 1–17, 2025.
- [72] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.
- [73] E. Ilg et al., "Uncertainty estimates and multi-hypotheses networks for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 652–667.
- [74] A. Eldesokey, M. Felsberg, K. Holmquist, and M. Persson, "Uncertainty-aware CNNs for depth completion: Uncertainty from beginning to end," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12014–12023.
- [75] 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 *Proc. Asian Conf. Comput. Vis.*, 2017, pp. 321–337.



**Yifu Tao** (Graduate Student Member, IEEE) received the M.Eng. and Ph.D. degrees in engineering science from the University of Oxford, Oxford, U.K., in 2020 and 2025, respectively.

He is currently a Postdoctoral Researcher with the Oxford Robotics Institute, University of Oxford. His research interests include 3-D reconstruction using visual and lidar sensors and deep learning methods.



**Maurice Fallon** (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from University College Dublin, Dublin, Ireland, in 2004, and the Ph.D. degree in acoustic source tracking from the University of Cambridge, Cambridge, U.K., in 2008.

From 2008 to 2012, he was a Postdoc and a Research Scientist with MIT Marine Robotics Group working on SLAM. Later, he was the Perception Lead of MIT's team in the DARPA Robotics Challenge. Since 2017, he has been a Royal Society University

Research Fellow and a Professor with the University of Oxford, Oxford, U.K. He leads the Dynamic Robot Systems Group in Oxford Robotics Institute.. His research interests include probabilistic methods for localization, mapping, multisensor fusion, and robot navigation.

Dr. Fallon has been the recipient or finalist for best paper awards at six IEEE conferences (IEEE International Conference on Robotics and Automation, IEEE-RAS International Conference on Humanoid Robots, and IEEE Intelligent Vehicles Symposium).