UncLe-SLAM: Uncertainty Learning for Dense Neural SLAM

Author(s):
Sandström, Erik; Ta, Kevin; Van Gool, Luc; Oswald, Martin R.

Publication date:
2023

Permanent link:
https://doi.org/10.3929/ethz-b-000643412

Originally published in:
https://doi.org/10.1109/ICCVW60793.2023.00488

UncLe-SLAM: Uncertainty Learning for Dense Neural SLAM

Erik Sandström[1*] Kevin Ta[1*] Luc Van Gool[1,2] Martin R. Oswald[1,3]
[1]ETH Zürich, Switzerland [2]KU Leuven, Belgium [3]University of Amsterdam, Netherlands

Abstract

We present an uncertainty learning framework for dense neural simultaneous localization and mapping (SLAM). Estimating pixel-wise uncertainties for the depth input of dense SLAM methods allows re-weighing the tracking and mapping losses towards image regions that contain more suitable information that is more reliable for SLAM. To this end, we propose an online framework for sensor uncertainty estimation that can be trained in a self-supervised manner from only 2D input data. We further discuss the advantages of the uncertainty learning for the case of multi-sensor input. Extensive analysis, experimentation, and ablations show that our proposed modeling paradigm improves both mapping and tracking accuracy and often performs better than alternatives that require ground truth depth or 3D. Our experiments show that we achieve a 38% and 27% lower absolute trajectory tracking error (ATE) on the 7-Scenes and TUM-RGBD datasets respectively. On the popular Replica dataset using two types of depth sensors, we report an 11% F1-score improvement on RGBD SLAM compared to the recent state-of-the-art neural implicit approaches. Source code: https://github.com/kev-in-ta/UncLe-SLAM.

1. Introduction

Neural scene representations have taken over the 3D reconstruction field by storm [47, 41, 12, 42] and have recently also been built into SLAM systems [67, 81, 78] with excellent results for geometric reconstruction, hole filling, and novel view synthesis. However, their camera tracking performance is typically inferior to the one of traditional sparse methods [9] that rely on feature point matching [81, 78]. A major difference to sparse methods which focus on a small set of points is that the rendering loss in most dense methods treats all pixels equally although it is plausible that they differ in their amount of useful information for SLAM, due to sensor noise. In the context of RGBD-cameras, it is well-known that several factors such as surface material type, texture *etc.*, often affect the sensor's raw output, leading to noisy measurements [23, 4]. Introducing pixel-wise uncertainties into a dense SLAM approach allows us to model non-uniform weights to focus on tracking and mapping suitable scene parts in a continuous manner. This is akin to the discrete selection of features points in traditional sparse approaches. Currently, the majority of dense neural SLAM approaches employ a uniform weighting for all pixels during mapping [81, 78, 37, 80] and tracking [81, 78, 67, 80]. Some efforts have been made to construct more informed pixel sampling strategies via active resampling or rejection based on the re-rendering loss for mapping [67] and tracking [37], but these approaches are ultimately limited by simple heuristics. In this paper, we therefore tackle the task of learning aleatoric depth sensor uncertainty on the fly to weigh scene parts in a non-uniform manner based on the estimated confidence. Furthermore, mobile devices are often equipped with more than one depth sensing modality and it is often observed that different modalities complement each other [58]. With these aspects in mind, we design our implicit SLAM system to perform dense SLAM with one or more depth sensors. Additionally, existing depth fusion methods that model single

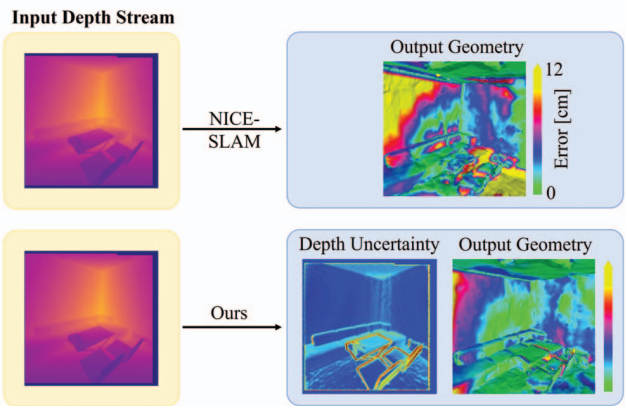

Figure 1: **UncLe-SLAM benefit.** Our proposed method learns depth uncertainty on the fly in a self-supervised way. We show that our approach yields more accurate 3D mapping and tracking than other dense neural implicit SLAM methods, like NICE-SLAM [81] which does not model depth uncertainty.

*Equal contribution.

sensor depth uncertainty [59, 56, 54, 71] or fuse multiple depth sensors [58] require access to ground-truth depth or 3D at train time. Hence, these methods may not be robust to domain shifts at test time. On the contrary, we learn sensor-agnostic uncertainty online in a self-supervised way without requiring ground truth depth or 3D. For that, we assume a Laplacian error distribution on the depth sensor and derive the corresponding loss function.

Our method, dubbed UncLe-SLAM, jointly learns the aleatoric depth uncertainty and the scene geometry by passing cheaply available 2D features from the depth sensor as input to a small uncertainty decoder, meaning that we stay within real-time run constraints. Our approach thus guides the mapping and tracking process with the implicitly learned uncertainty, see Fig. 1. Moreover, we showcase that our formulation generalizes well to the multi-sensor setting where two depth sensors with varying noise distributions are fused into the same 3D representation. Our contributions are:

- A robust approach for estimating aleatoric depth uncertainty for the single and multi-sensor case is proposed. The introduced framework is robust, accurate and can be directly integrated into a dense SLAM system without the need for ground truth depth or 3D.
- In the single depth sensor case, we show that our uncertainty-driven approach often improves on standard performance metrics regarding geometric reconstruction and tracking accuracy. In the multi-sensor case, we show for various sensor combinations that our method extracts results that are consistently better than those obtained from the individual sensors.

2. Related Work

The approach proposed in this paper covers a wide range of research topics such as SLAM, sensor fusion, sensor modeling, uncertainty modeling, etc. All of these topics are well-studied with an exhaustive list of literature. Therefore, we narrow our related work discussion to the relevant methods that better helps expose our contributions.

2.1. Single-Sensor Depth Fusion and Dense SLAM

Curless and Levoy's seminal work [14] is the basis for many dense depth mapping approaches [43, 71]. Subsequent developments include scalable techniques with voxel hashing [45, 28, 46], octrees [63], and pose robustness [8]. Further advancements led to dense SLAM, such as [44, 61, 67, 81], which can also handle loop closures such as BundleFusion [15]. To address the issue with noisy depth maps, RoutedFusion [71] learns a fusion network that outputs the TSDF update of the volumetric grid. Other works such as NeuralFusion [72] and DI-Fusion [27] extend this concept by learning the scene representation, resulting in better outlier handling. Lately, the work on continuous neural mapping [74] learns the scene representation using continual mapping from a sequence of depth maps. Yet, none of the above-mentioned approaches explicitly study multiple depth modalities or their uncertainty and their fusion in a neural SLAM framework. Further, their extensions to multiple sensor fusion are often not trivial. Nevertheless, by treating all sensors alike, they can be used as simple baselines.

2.2. Multi-Sensor Depth Fusion

The fusion of at least two types of depth-sensing devices has been studied in the past. Notably, the fusion of raw depth maps from two different sensors, such as RGB stereo and time-of-flight (ToF) [70, 13, 2, 21, 16, 38, 3, 17], RGB stereo and Lidar [36], RGB and Lidar [55, 48, 50], RGB stereo and monocular depth [40] and the fusion of multiple RGB stereo algorithms [53] is well-studied and explored. Yet, these methods study specific sensors and are not inherently equipped with 3D reasoning. Few works consider 3D reconstruction with multiple sensors [57, 31, 7, 76, 77, 24], but these do not consider the online mapping setting. Conceptually, more closely related to our work is SenFuNet [58], which is an online mapping method for multi-sensor depth fusion. Still, contrary to our approach, [58] requires access to ground truth 3D data at train time. It does not predict explicit uncertainty per sensor but requires multi-sensor input to weigh the sensors against each other.

2.3. Uncertainty Modeling for Depth

Uncertainty modeling for depth estimation has been studied extensively in the past, specifically for multiview stereo (MVS) [33, 73, 79, 66] and binocular stereo [54, 62, 69, 30]. In addition to the popular Gaussian distribution to model sensor noise [10], the Laplacian noise model has also been employed to analyse depth uncertainty. For instance, Klodt *et al.* [32] assume, like our approach, a Laplacian noise model to explore the advantage of depth uncertainty modeling from short sequences of RGB images. Likewise, Yang *et al.* [75] uses a Laplacian model for monocular depth estimation [75]. Furthermore, some works propose self-supervised frameworks for monocular depth estimation, such as [52, 75]. Aleatoric uncertainty estimation has also been applied for surface normal estimation from RGB [5]. This technique was recently used to refine depth estimated from a monocular RGB camera [6]. Closer to our setting, RoutedFusion [71] trains an encoder-decoder style network to refine depth maps and predict a measure of confidence. Nevertheless, unlike our approach, they require access to ground truth depth for training. Despite impressive progress in depth uncertainty modeling, there has been little focus on uncertainty estimation of the 3D surface. DI-fusion [27] proposed a technique to do this by imposing a

Gaussian assumption on the signed distance function. Yet, unlike our approach, it needs ground truth 3D for training.

Regarding uncertainty modeling, our method is related to the treatment of probabilistic depth fusion methods [19, 20, 34, 18, 10]. As studied and observed by several methods, As studied and observed by several methods, explicit uncertainty modeling is helpful[1]. In the context of SLAM, Cao *et al.* [10] introduced a probabilistic framework via a Gaussian mixture model for dense visual SLAM based on surfels to address uncertainties in the observed depth. However, it is well-known that Gaussian noise modeling has its practical limitations [49].

Overall, to the best of our knowledge, none of the state-of-the-art neural SLAM methods for dense online SLAM consider aleatoric uncertainty modeling along with multiple sensors. Moreover, none of the above works consider estimating uncertainty in an online self-supervised way with implicit neural SLAM.

3. Preliminaries

To perform online neural implicit SLAM from a sequence of RGBD images, it is necessary to have a 3D representation. Furthermore, due to the self-supervision from the incoming sensor frames, a rendering technique is needed that connects the 3D representation to the 2D observations. By using the 3D representation and 2D rendering technique, the mapping and tracking processes can be constructed. In this paper, we focus on solid (non-transparent) surface reconstruction. We first present background information on implicit surface and volumetric radiance representations, which is then used to develop our online uncertainty modeling approach.

3.1. Scene Representation

Convolutional Occupancy Networks [51] proposes to learn the occupancy $\mathrm{o} \in [0,1]$ using an encoded 3D grid of features that can be passed, after trilinear interpolation, through an MLP decoder to acquire the occupancy. NICE-SLAM [81] utilizes this idea and encodes the scene in hierarchical voxel grids of features. For any sampled 3D coordinate $\mathbf{p}_i \in \mathbb{R}^3$, feature vectors can be extracted from these voxel grids. The features can then be fed, in a coarse-to-fine manner, through MLP decoders to extract the occupancy of the given point.

The geometry is encoded in two feature grids - middle and fine[2]. Each feature grid ϕ_θ^l has an associated pretrained decoder f^l, where $l \in \{1,2\}$ and θ describes the optimizable features. We denote a trilinearly interpolated feature vector at point $\mathbf{p}_i$ as $\phi_\theta^l(\mathbf{p}_i)$. Additionally, the color is encoded in a fourth feature grid ψ_ω (parameters ω) with decoder g_ξ (parameters ξ), and is used for further scene refinement after initial stages of geometric optimization. The observed scene geometry is reconstructed from the middle and fine resolution feature grids, with the fine feature grid output residually added to the middle grid occupancy. In summary, the occupancy o_i and color $\mathbf{c}_i$ are predicted as

$$\begin{aligned} \mathrm{o}_i &= f^1\big(\mathbf{p}_i, \phi_\theta^1(\mathbf{p}_i)\big) + f^2\big(\mathbf{p}_i, \phi_\theta^2(\mathbf{p}_i), \phi_\theta^1(\mathbf{p}_i)\big) \\ \mathbf{c}_i &= g_\xi\big(\mathbf{p}_i, \psi_\omega(\mathbf{p}_i)\big). \end{aligned} \tag{1}$$

3.2. Depth and Image Rendering

To link the 3D representation with supervision using 2D RGBD observations, NICE-SLAM uses volume rendering of depth maps and RGB images. This process involves sampled points $\mathbf{p}_i \in \mathbb{R}^3$ at depth $d_i \in \mathbb{R}^1$ along a ray $\mathbf{r} \in \mathbb{R}^3$ cast from origin $\mathbf{O} \in \mathbb{R}^3$, as

$$\mathbf{p}_i = \mathbf{O} + d_i \mathbf{r}, \quad i \in \{1, ..., N\}. \tag{2}$$

The occupancies are evaluated along the ray according to Eq. (1) and volume rendering constructs a weighting function w_i using Eq. (3). This weight represents the discretized probability that the ray terminates at that particular point.

$$w_i = \mathrm{o}_i \prod_{j=1}^{i-1} (1 - \mathrm{o}_j) \tag{3}$$

The rendered depth is computed as the weighted average of the depth values along each ray, and equivalently for the color following Eq. (4) as defined below.

$$\hat{D} = \sum_{i=1}^{N} w_i d_i, \quad \hat{I} = \sum_{i=1}^{N} w_i \mathbf{c}_i \tag{4}$$

This volume rendering method also provides variance from the discretized selection of points. By taking the depth differences with respect to the sensor depth multiplied by the weighting function, a measure of variance can be extracted that is a composite of the model uncertainty and sampling uncertainty, as defined in Eq. (5).

$$\hat{S}_D = \sqrt{\sum_{i=1}^{N} w_i \big(\hat{D} - d_i\big)^2} \tag{5}$$

4. Method

This section details how we introduce aleatoric uncertainty modeling based on the preliminaries covered in Section 3. The rest of our methodology section is arranged as follows: We first present our theoretical assumptions which form the basis for our loss function derivation. Then,

[1] For a review on uncertainty estimation in deep learning we refer to [1]

[2] There is an additional coarse grid, but it is not used for mapping, and despite claims from the authors, when looking at the source code, it is neither used for tracking. Thus, we do not consider it.

we explain how our framework elegantly supports multi-sensor fusion with additional depth sensors and RGBD fusion without relying on heuristic hyperparameters. Finally, we describe our architecture and implementation. For an overview, see Fig. 2.

4.1. Theoretical Assumptions

We motivate our formulation of sensor noise under the assumption of a Laplacian noise distribution on a per-ray basis which was found to perform better on vision tasks than a Gaussian assumption by [29]. Further, we assume that the noise is heteroscedastic meaning that the noise variance is a variable for each pixel. That is, each pixel m in the captured depth sensor is treated independently. Consequently, the measured depth is sampled from the probability density function

$$P(D_m) = \frac{1}{2\beta_m} \exp\left(-\frac{|D_m - \hat{D}_m|_1}{\beta_m}\right) . \tag{6}$$

We take $\hat{D}_m$ to be the true depth and $\sqrt{2}\beta_m$ to be the standard deviation of the depth reading of a specific pixel, parameterised by some function with parameters τ. When we aggregate all depth sensor information, we get the joint density of the per-ray depth observations

$$P(D_1, ..., D_M) = \prod_{m=1}^{M} \frac{1}{2\beta_m} \exp\left(-\frac{|D_m - \hat{D}_m|_1}{\beta_m}\right) ,$$

where M is the total number of pixel readings. The best estimate of the depth can thus be determined via maximum likelihood estimation

$$\begin{aligned} \arg\max_{\theta,\tau} P(D_1, ..., D_M) &= \arg\min_{\theta,\tau} -\log\left(P(D_1, ..., D_M)\right) \\ &= \arg\min_{\theta,\tau} \sum_{m=1}^{M} \frac{|D_m - \hat{D}_m|_1}{\beta_m} + \log(\beta_m). \end{aligned} \tag{7}$$

4.2. Mapping

Mapping is performed equivalently to [81], but with the revised loss function

$$\mathcal{L}_{map} = \sum_{m=1}^{M} \frac{|D_m - \hat{D}_m(\theta)|_1}{\beta_m(\tau)} + \log\left(\beta_m(\tau)\right) \tag{8}$$

A database of keyframes is utilized to regularize the mapping loss. Keyframes are added at a regular frame interval and sampled for each mapping phase to have a significant overlap with the viewing frustum of the current frame. Pixels are then sampled from the keyframes along with the current frame to optimize the map. In terms of optimization, a two-stage approach is taken. For each mapping phase, the middle grid is first optimized and then, once converged, the fine grid is included for further refinement. For more details, we refer to [81].

4.3. Tracking

Tracking is performed equivalently to [81], but with the revised mapping loss function

$$\mathcal{L}_{\text{track}} = \frac{1}{M_t} \sum_{m=1}^{M_t} \frac{|D_m - \hat{D}_m(\theta)|_1}{\hat{S}_D(\theta) + \beta_m(\tau)}, \tag{9}$$

which additionally takes the aleatoric sensor uncertainty into account. M_t is the number of pixels that are sampled during tracking. We optimize the camera extrinsics $\{\mathbf{R}, \mathbf{t}\}$.

4.4. Multi-Sensor Depth Fusion and RGBD Fusion

The methods described so far have encompassed implicitly learning uncertainty given a single sensor. We extend this single-sensor approach to incorporate a second sensor. If we again assume that each depth observation is I.I.D., the joint likelihood we wish to maximize is the product of the probability distributions for each pixel in each sensor.

Given two synchronized and aligned sensors, we can sample a set of pixels $m \in \{1, ..., M\}$ from two depth sensors yielding the generalized loss function

$$\mathcal{L} = \sum_{m=1}^{M} \sum_{i=1}^{2} \frac{|D_{m,i} - \hat{D}_m|_1}{\beta_{m,i}} + \log(\beta_{m,i}). \tag{10}$$

One interpretation of this objective function is that the pipeline implicitly learns the weighting between the two sensor observations. The loss function penalizes large uncertainties via the log terms, and implicitly learns the uncertainty for both sets of observations as the model depth is optimized. In an analogous fashion, RGBD fusion can be achieved via the loss function

$$\mathcal{L}_{rgbd} = \mathcal{L}_{geo} + \mathcal{L}_{rgb} \tag{11}$$

$$\mathcal{L}_{geo} = \sum_{m=1}^{M} \frac{|D_m - \hat{D}_m|_1}{\beta_{m,d}} + \log(\beta_{m,d}) \tag{12}$$

$$\mathcal{L}_{rgb} = \sum_{m=1}^{M} \frac{|I_m - \hat{I}_m|_1}{\beta_{m,r}} + \log(\beta_{m,r}), \tag{13}$$

where $\beta_{m,d}$ and $\beta_{m,r}$ denote the per pixel sensor uncertainty for the depth and rgb sensor respectively. This modeling is different to NICE-SLAM where the color and geometry losses are weighted by a heuristic hyperparameter.

4.5. Design Choices and Architecture Details

The per-pixel depth and variance is rendered according to Eq. (4) and Eq. (5) respectively.

The variance from Eq. (5) could naively be applied to Eqs. (8) and (9) with the rendered variance $\hat{S}_D$ representing $2\beta^2$. Unfortunately, such an approach is poorly motivated as this calculated variance is related to the model confidence, as opposed to the sensor-specific noise. In practice,

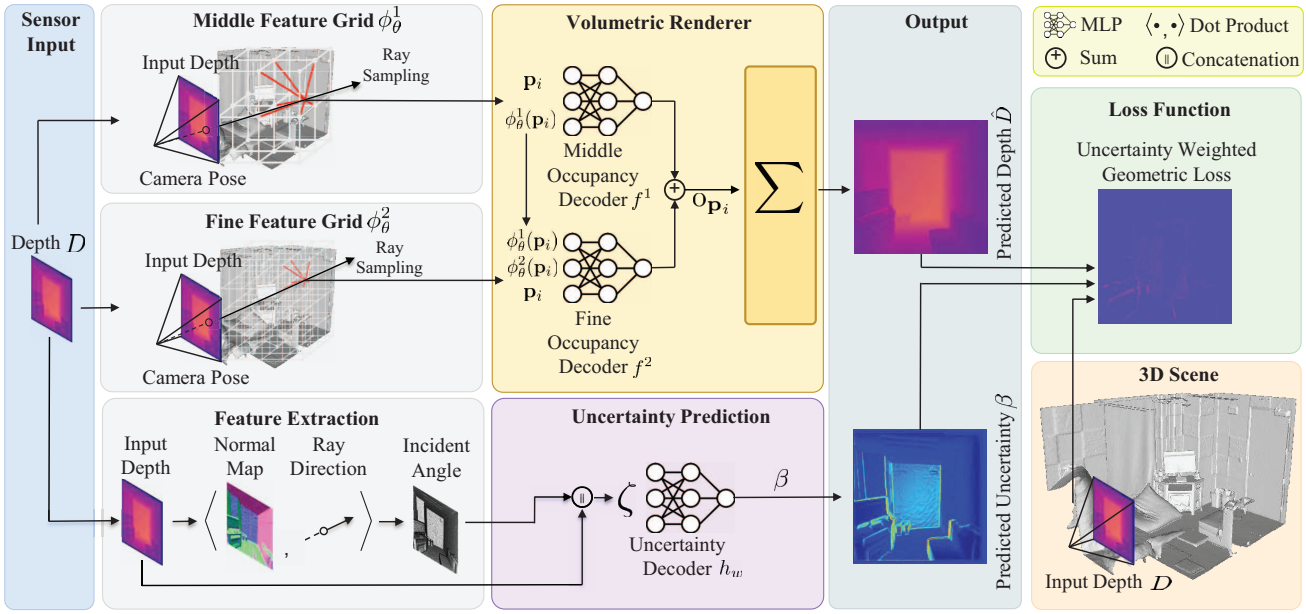

Figure 2: **UncLe-SLAM Architecture.** Given an input depth map from an estimated camera pose, mapping and tracking is performed by minimizing a re-rendering loss, by optimizing either the grid features θ and network parameters w or the camera extrinsics respectively. The depth is estimated using point samples $\mathbf{p}_i$ along rays with a volumetric renderer which decodes geometric multi-scale features $\phi_\theta^1(\mathbf{p}_i)$ and $\phi_\theta^2(\mathbf{p}_i)$ into occupancies. The uncertainty is estimated by feeding informative features through an uncertainty decoder h_w. The architecture can be extended to a multi-sensor setting or with RGB by adding additional uncertainty MLPs. We build the architecture on top of NICE-SLAM [81].

the uncertainty we strive to model is aleatoric uncertainty and should be distinct from the model confidence. One interpretation of the variance from Eq. (5) is as the epistemic uncertainty. With an increasing number of observations, the epistemic uncertainty should shrink, driving the model towards sharp bounds. We instead seek a separate process to extract aleatoric uncertainty. We take the concept of implicitly learned aleatoric uncertainty from the work of Kendall and Gal [29] and design a patch-based MLP. Our approach takes in spatial information from the specific depth frame to generate uncertainty β, distinct and decoupled from the rendered variance $\hat{S}_D$.

An additional concern within the framework is the computational overhead. Volume rendering is one of the more intensive operations and an additional rendering for each sensor may be prohibitively expensive. Consequently, we propose a simpler approach to derive a ray-specific uncertainty through the use of 2D features that contain relevant information. We can leverage cheaply available metadata, as was done in *e.g.* [60], to capture sensor noise. We investigate plausible per-pixel (per-ray) features and end up with the following inputs to estimate depth uncertainty: the measured depth $D_m \in \mathbb{R}$ and the incident angle $\theta \in \mathbb{R}$ between the local ray direction and the surface normal, computed as in [43] from the depth map through central difference after bilateral filtering [68]. For RGB uncertainty, we feed the color instead of the depth and incident angle. Instead of only feeding the features from a single pixel observation, we feed the features from a 5×5 patch, effectively expanding the receptive field of the ray. This patch of pixels gives local context and local correlation of uncertainty for areas near edges or with high frequency content. We denote the concatenation of the features ζ.

The MLP network, denoted h_w, is similar in architecture to the MLPs f^l used for the occupancy decoders. We use a network with 5 intermediate layers with 32 nodes each, activated via ReLU, except for the last layer. Inspired by NeRF-W [39], we apply a softplus activation with a minimum uncertainty value $\beta_{\min}$. The output $\tilde{y}_m \in \mathbb{R}$ from the last layer is thus processed as

$$\beta_m = h_w(\zeta) = \beta_{\min} + \log\left(1 + \exp\left(\tilde{y}_m\right)\right) \tag{14}$$

The addition of a minimum uncertainty changes the bound of the uncertainty to $(\beta_{\min}, \infty)$, and mitigates numerical instability during optimization. Finally, we only update h_w during the fine stage of optimization *i.e.* in the middle stage, we use the same loss as [81].

5. Experiments

We first describe our experimental setup and then report results on single and multi-sensor experiments. We evaluate our method on the Replica dataset [64] as well as the real-world 7-Scenes [22] and TUM-RGBD [65] datasets. All reported results are averages over the respective test scenes and over ten runs, unless otherwise stated. Further experiments and details are in the supplementary material.

Implementation Details. We leave many of the hyperparameters from [81] as is *e.g.* we use 0.32 m and 0.16 m voxel size for the middle and fine resolution respectively. The ray sampling strategy remains the same, with 32 points uniformly sampled along the ray and 16 points sampled uniformly near the depth reading. The feature grids store 32-dimensional features and we use the same occupancy decoders and color decoders as [81]. We leave the learning rates for feature grid optimization under the same schedule—*i.e.* 0.1 for the middle stage and 0.005 for the fine stage. On Replica, we map every 5th frame and use 5K pixels uniformly sampled during mapping and tracking. We use 10 tracking iterations and 60 mapping iterations and inlude the fine grid optimization after 60 % of the total mapping iterations. These parameters were not tuned and may be optimized to further improve performance. Specifically, the learning rates may be adjusted under the new loss formulation to improve stability.

Evaluation Metrics. The meshes, produced by marching cubes [35] from the occupancy grids, are evaluated using the F-score which is the harmonic mean of the Precision (P) and Recall (R). We further provide the mean precision and mean recall along with the depth L1 metric as in [81]. For tracking accuracy, we use ATE RMSE [65].

Baseline Methods. We compare our proposed method to existing state-of-the-art online dense neural SLAM methods. The most natural baseline is NICE-SLAM [81], which treats all depth observations equally, followed by SenFuNet [58], which performs multi-sensor depth fusion. SenFuNet does not explicitly model per sensor uncertainty, but fuses two depth sensors with a learned weighting network. In the multi-sensor setting, we also compare to VoxFusion [78] by weighting all depth readings equally. Additionally, we pretrain a 2D confidence prediction network from the raw depth maps using a slightly modified version of the network proposed by Weder *et al.* [71]. The per pixel learned confidences are used at runtime in NICE-SLAM to scale the importance in the mapping and tracking loss function. We call this baseline "NICE-SLAM+Pre". Details are provided in the supplementary material.

Datasets. The Replica dataset [64] comprises high-quality 3D reconstructions of a variety of indoor scenes. We utilize the publicly available dataset collected by Sandström *et al.* [58], which provides trajectories from a simulated structured light (SL) sensor [25], depth from stereo with semi-global matching [26] (SGM) and from a learning-based approach called PSMNet [11] as well as color.

The 7-Scenes [22] and TUM-RGBD [65] datasets comprise a set of RGBD scenes captured with an active depth camera along with ground truth poses.

Model ↓ |Metric →	Depth L1↓ [cm]	mP↓ [cm]	mR↓ [cm]	P↑ [%]	R↑ [%]	F↑ [%]	ATE↓ [cm]
Depth + Ground Truth Poses							
NICE-SLAM [81]	2.64	2.65	2.35	88.75	88.20	88.45	-
NICE-SLAM+Pre	2.67	2.65	2.31	89.00	88.62	88.78	-
Ours	**2.42**	**2.58**	**2.29**	**89.14**	**88.70**	**88.89**	-
Depth + Tracking							
NICE-SLAM [81]	10.65	10.04	7.17	48.46	51.43	49.80	27.90
NICE-SLAM+Pre	9.90	13.99	6.84	52.43	57.72	54.54	36.95
Ours	**7.39**	**6.56**	**6.20**	**57.30**	**57.57**	**57.41**	**19.36**
RGB-D + Tracking							
NICE-SLAM [81]	8.11	7.81	6.77	51.81	53.56	52.63	20.25
Ours	**6.49**	**6.43**	**5.93**	**58.89**	**59.39**	**59.09**	**18.92**

Table 1: **Reconstruction Performance on Replica [64]: PSMNet [11].** Our model outperforms the baseline methods in the mapping only setting as well as with tracking enabled and when color is available. Best results are highlighted as **first**, second, and third.

5.1. Single Sensor Evaluation

Replica. We provide experimental evaluations on two depth sensors in three different settings: 1. Depth with ground truth poses *i.e.* pure mapping from noisy depth. 2. Depth with estimated camera poses (*i.e.* with tracking) and 3. RGBD with tracking. In Table 1 for the PSMNet [11] sensor, our model shows consistent improvements on all metrics in all three settings. For the SGM [26] sensor (in Table 2) we find consistent improvements in the settings where tracking is enabled. In the mapping only setting, the pretrained confidence model performs marginally better for the SGM sensor. Fig. 4 shows the reconstruction results for two scenes from the Replica dataset with the two sensors. Compared to NICE-SLAM [81], we find that UncLe-SLAM on average reconstructs more accurate geometries.

Uncertainty Visualization. To gain insights about the estimated uncertainties that our model produces, we visualize the estimated uncertainties for our two depth sensors in Fig. 3. For reference, we also plot the absolute ground truth depth error. Compared to the uncertainties produced by the pretrained network, we find that our model produces sharper estimates, see *e.g.* the last row where our model can replicate the error pattern more accurately. This is likely a result of our restricted receptive field while the pretrained model employs a fully convolutional network model with a larger receptive field. Moreover, our model seems to be able to replicate some errors better than the pretrained model, see *e.g.* the red patch for the PSMNet sensor where our model

Model ↓ |Metric →	Depth L1↓ [cm]	mP↓ [cm]	mR↓ [cm]	P↑ [%]	R↑ [%]	F↑ [%]	ATE↓ [cm]
Depth + Ground Truth Poses							
NICE-SLAM [81]	2.35	2.55	2.12	89.54	91.07	90.29	-
NICE-SLAM+Pre	**2.25**	**2.49**	**2.08**	**89.86**	**91.42**	**90.62**	-
Ours	2.27	2.56	2.10	89.59	91.24	90.40	-
Depth + Tracking							
NICE-SLAM [81]	12.03	10.21	7.75	46.00	50.58	48.10	30.73
NICE-SLAM+Pre	18.96	16.35	6.90	48.92	57.60	52.54	39.14
Ours	**10.60**	**9.38**	**6.58**	**52.62**	**57.21**	**54.72**	**29.11**
RGB-D + Tracking							
NICE-SLAM [81]	9.91	**10.37**	6.82	50.12	54.51	52.00	**26.56**
Ours	**7.79**	11.01	**5.80**	**56.10**	**61.16**	**58.19**	27.41

Table 2: **Reconstruction Performance on Replica [64]: SGM [26].** Our model outperforms the baseline methods in most settings while being marginally worse than the model using pretrained uncertainties in the mapping only setting.

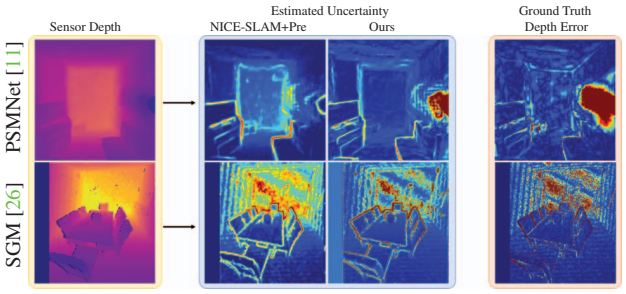

Figure 3: **Uncertainty Visualization.** Each row shows a depth map from a specific sensor with the associated uncertainty estimation from the pretrained network model and ours. As reference, the ground truth absolute depth error is shown in the last column. We find our model reproduces the error map with less smoothing than the pretrained model while capturing more details, *e.g.* the red patch from the PSMNet sensor. Blue: low uncertainty, red: high uncertainty.

can capture the error while the pretrained model struggles. We believe this is due the the ability of our model to adapt to test time constraints through runtime optimization. Moreover, our network h_w contains only 5409 parameters while the pretrained network contains 360 241.

7-Scenes. In Table 3, we evaluate our framework on the 7-Scenes dataset [22]. We use sequence 1 for all scenes. We find that NICE-SLAM [81] consistently yields worse tracking results suggesting the effectiveness of our depth uncertainty when it comes to maintaining robust camera pose tracking. On average, our method yields a 38 % gain in terms of the mean ATE.

TUM-RGBD. In Table 4, we evaluate our framework on the real-world TUM-RGBD dataset [65]. Our conclusions on this dataset is similar to the 7-Scenes dataset. On average,

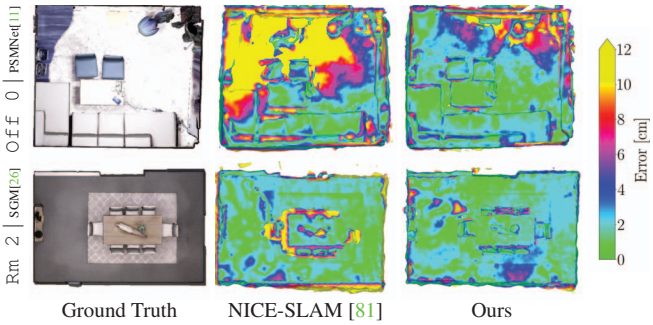

Figure 4: **Single Sensor Reconstruction on Replica [64].** We show that our uncertainty modeling on average helps to achieve more accurate reconstructions when noisy depth sensors are provided as input. The `office 0` scene uses only depth as input while the `room 2` scene is provided RGBD input. Tracking is enabled for all experiments. The colorbar displays the deviation from the ground truth mesh.

Method	Chess	Fire	Head	Off.	Pump.	Kitch.	Stairs	Avg.
NICE-SLAM [81]	40.30	47.67	20.55	8.49	33.11	24.39	9.18	24.24
Ours	**14.85**	**25.47**	**13.12**	**7.83**	**29.32**	**6.21**	**8.53**	**15.05**

Table 3: **Tracking Evaluation on 7-Scenes.** We report the average ATE RMSE [cm] over 5 runs for each scene. With our depth uncertainty modeling, we achieve significantly better tracking compared to NICE-SLAM. On average, our method yields a 38 % gain in terms of the mean ATE.

Method	fr1/ desk	fr1/ desk2	fr1/ xyz	Avg.
NICE-SLAM [81]	40.40	47.81	5.11	31.11
Ours	**29.04**	**36.57**	**2.71**	**22.77**

Table 4: **Tracking Evaluation on TUM-RGBD.** We report the average ATE RMSE [cm] by mapping every 2nd frame.

camera pose tracking is greatly benefited by our uncertainty aware strategy.

5.2. Multi-Sensor Evaluation

We conduct experiments in the multi-sensor setting. We compare to Vox-Fusion [78], a dense neural SLAM system and SenFuNet [58], which is a mapping only framework. To learn sensor specific uncertainties, we use one uncertainty decoder h_w per sensor. In Table 5 we show for SGM+PSMNet fusion that we are able to consistently improve over the single-sensor reconstructions in isolation and over SenFuNet [58] and VoxFusion [78]. When ground truth poses are provided, we find that original NICE-SLAM performs very similar to our proposed uncertainty aware model. On a closer look, the PSMNet and SGM sensors are

Model ↓ |Metric →	Depth L1↓ [cm]	mP↓ [cm]	mR↓ [cm]	P↑ [%]	R↑ [%]	F↑ [%]	ATE↓ [cm]
Single Sensor Ours: Depth + Ground Truth Poses							
PSMNet [11]	2.42	2.58	2.29	89.14	88.70	88.89	-
SGM [26]	2.27	2.56	2.10	89.59	**91.24**	90.40	-
Multi-Sensor: Depth + Ground Truth Poses							
NICE-SLAM [81]	2.03	**2.34**	**1.99**	**90.57**	90.86	**90.69**	-
SenFuNet [58]	23.49	15.62	12.66	32.74	28.32	30.22	-
Vox-Fusion [78]	6.52	48.76	30.72	28.01	49.36	35.65	-
NICE-SLAM+Pre	2.19	2.44	2.01	89.93	90.76	90.31	-
Ours	**1.97**	2.36	2.01	90.15	90.76	90.42	-
Single Sensor Ours: Depth + Tracking							
PSMNet [11]	7.39	6.56	6.20	57.30	57.57	57.41	**19.36**
SGM [26]	10.60	9.38	6.58	52.62	57.21	54.72	29.11
Multi-Sensor: Depth + Tracking							
NICE-SLAM [81]	13.58	16.76	7.84	51.19	55.45	52.81	40.37
NICE-SLAM+Pre	11.29	13.59	6.12	62.02	65.95	63.30	35.55
Ours	**4.13**	**4.60**	**4.35**	**70.30**	**69.30**	**69.76**	19.88

Table 5: **Reconstruction Performance on Replica [64]: SGM [26]+PSMNet [11].** Our multi-sensor reconstruction performance improves over the single sensor results in isolation and we outperform most of the baseline methods. The experiment was conducted in the depth only setting with known camera poses.

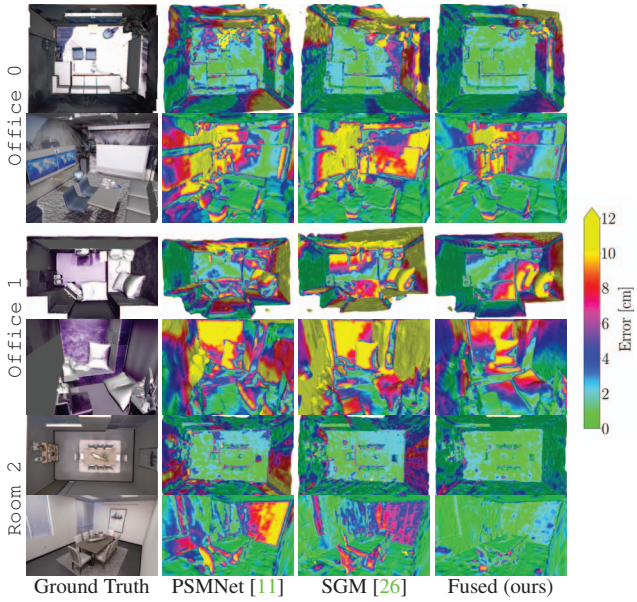

Figure 5: **Multi-Sensor Reconstruction on Replica [64].** The two middle columns show single sensor reconstructions while the rightmost column shows the result when both sensors are jointly fused into the same geometry using our proposed UncLe-SLAM. Our uncertainty modeling helps on average to achieve more accurate reconstructions in the multi-sensor setting compared to the single sensor reconstructions. The colorbar displays the deviation from the ground truth mesh.

quite similar and we believe that when both sensors yield similar depth characteristics, simple averaging works well, *i.e.* putting equal weight to both sensors as done by NICE-SLAM. We find, however, that uncertainty modeling is very important to obtain robust tracking which greatly improves the reconstruction accuracy. Finally, Fig. 5 shows visualizations of the reconstruction accuracy comparing the single sensor reconstructions to the geometry attained by UncLe-SLAM. We find that the most accurate sensor is on average favored. For more results, see the supplementary material.

5.3. Memory and Runtime

Due to the low number of parameters in our uncertainty MLP h_w (5409), we add 43 kB to the already allocated 421 kB for the decoders in NICE-SLAM. This is negligible in comparison to the 95.86 MB allocated for the dense grids for the `office 0` scene. We report a 15 % increase in runtime over NICE-SLAM which can be compared to the average gain of 38% and 27% in terms of ATE RMSE on the 7-Scenes and TUM-RGBD datasets respectively and 11% and 32% in terms of the F1-score on single sensor RGBD SLAM and multi-sensor depth SLAM.

5.4. Limitations

Our framework uses patch based modeling of uncertainty which may not hold in the general case along with the cheaply available features we feed as input to the uncertainty decoder. Simply using a more expressive model with learned features is not straight forward though, as shown by our results with the pretrained model and we leave this as future work. Finally, we believe that the relatively large voxel size we use can prevent efficient uncertainty learning from fine geometric details due to the high degree of averaging. We believe that our method can benefit from a scene representation that allows for resolving finer details.

6. Conclusion

The paper presents a way to learn per pixel depth uncertainties for dense neural SLAM. This allows the mapping and tracking re-rendering losses to be re-weighted such that trustworthy sensor readings are used to track the camera and to update the map. We believe this is a useful instrument in closing the gap in tracking accuracy to traditional sparse SLAM methods. We show that modeling depth uncertainty generally results in improvements both in terms of mapping and tracking accuracy and often performs better than alternatives that require ground truth depth or 3D. The paper also provides one of the initial solutions that utilizes more than one depth sensing modality for dense neural SLAM.

Acknowledgements. This work was supported by a VIVO collaboration project on real-time scene reconstruction, as well as by a research grants from FIFA. We thank Suryansh Kumar for fruitful discussions.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. 3

[2] Gianluca Agresti, Ludovico Minto, Giulio Marin, and Pietro Zanuttigh. Deep learning for confidence information in stereo and tof data fusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 697–705, 2017. 2

[3] Gianluca Agresti, Ludovico Minto, Giulio Marin, and Pietro Zanuttigh. Stereo and tof data fusion by learning from synthetic data. *Information Fusion*, 49:161–173, 2019. 2

[4] Michael Riis Andersen, Thomas Jensen, Pavel Lisouski, Anders Krogh Mortensen, Mikkel Kragh Hansen, Torben Gregersen, and PJAU Ahrendt. Kinect depth sensor evaluation for computer vision applications. *Aarhus University*, pages 1–37, 2012. 1

[5] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and Exploiting the Aleatoric Uncertainty in Surface Normal Estimation. In *the proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. IronDepth: Iterative Refinement of Single-View Depth using Surface Normal and its Uncertainty. In *the proceedings of the British Machine Vision Conference (BMVC)*, 2022. 2

[7] Erik Bylow, Robert Maier, Fredrik Kahl, and Carl Olsson. Combining depth fusion and photometric stereo for fine-detailed 3d models. In *Scandinavian Conference on Image Analysis*, pages 261–274. Springer, 2019. 2

[8] E. Bylow, C. Olsson, and F. Kahl. Robust online 3d reconstruction combining a depth sensor and sparse feature points. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3709–3714, 2016. 2

[9] Carlos Campos, Richard Elvira, Juan J.Gomez Rodriguez, Jose M.M. Montiel, and Juan D. Tardos. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021. 1

[10] Yan-Pei Cao, Leif Kobbelt, and Shi-Min Hu. Real-time high-accuracy three-dimensional reconstruction with consumer rgb-d cameras. *ACM Transactions on Graphics (TOG)*, 37(5):1–16, 2018. 2, 3

[11] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 6, 7, 8

[12] Zhiqin Chen. Im-net: Learning implicit fields for generative shape modeling. 2019. 1

[13] Ouk Choi and Seungkyu Lee. Fusion of time-of-flight and stereo for disambiguation of depth measurements. In Kyoung Mu Lee, Yasuyuki Matsushita, James M. Rehg, and Zhanyi Hu, editors, *Computer Vision - ACCV 2012, 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part IV*, volume 7727 of *Lecture Notes in Computer Science*, pages 640–653. Springer, 2012. 2

[14] Brian Curless and Marc Levoy. Volumetric method for building complex models from range images. In *the proceedings of the SIGGRAPH Conference on Computer Graphics*. ACM, 1996. 2

[15] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017. 2

[16] Carlo Dal Mutto, Pietro Zanuttigh, and Guido Maria Cortelazzo. Probabilistic tof and stereo data fusion based on mixed pixels measurement models. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2260–2272, 2015. 2

[17] Yong Deng, Jimin Xiao, and Steven Zhiying Zhou. Tof and stereo data fusion using dynamic search range stereo matching. *IEEE Transactions on Multimedia*, 24:2739–2751, 2021. 2

[18] Wei Dong, Qiuyuan Wang, Xin Wang, and Hongbin Zha. Psdf fusion: Probabilistic signed distance function for on-the-fly 3d data fusion and scene reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 701–717, 2018. 3

[19] Yong Duan, Mingtao Pei, and Yucheng Wang. Probabilistic depth map fusion of kinect and stereo in real-time. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2317–2322. IEEE, 2012. 3

[20] Yong Duan, Mingtao Pei, Yucheng Wang, Min Yang, Iameng Qin, and Yunde Jia. A unified probabilistic framework for real-time depth map fusion. *J. Inf. Sci. Eng.*, 31(4):1309–1327, 2015. 3

[21] Georgios D Evangelidis, Miles Hansard, and Radu Horaud. Fusion of range and stereo data for high-resolution scene-modeling. *IEEE transactions on pattern analysis and machine intelligence*, 37(11):2178–2192, 2015. 2

[22] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179. IEEE, 2013. 6, 7

[23] S Burak Gokturk, Hakan Yalcin, and Cyrus Bamji. A time-of-flight depth sensor-system description, issues and solutions. In *2004 conference on computer vision and pattern recognition workshop*, pages 35–35. IEEE, 2004. 1

[24] Panlong Gu, Fengyu Zhou, Dianguo Yu, Fang Wan, Wei Wang, and Bangguo Yu. A 3d reconstruction method using multisensor fusion in large-scale indoor scenes. *Complexity*, 2020, 2020. 2

[25] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE international conference on Robotics and automation (ICRA)*, pages 1524–1531. IEEE, 2014. 6

[26] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern*

analysis and machine intelligence, 30(2):328–341, 2007. 6, 7, 8

[27] Jiahui Huang, Shi-Sheng Huang, Haoxuan Song, and Shi-Min Hu. Di-fusion: Online implicit 3d reconstruction with deep priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8932–8941, 2021. 2

[28] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip H. S. Torr, and David William Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE Trans. Vis. Comput. Graph.*, 21(11):1241–1250, 2015. 2

[29] Alex Kendall and Yarin Gal. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *the proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*. NeurIPS Foundation, 2017. 4, 5

[30] Sunok Kim, Seungryong Kim, Dongbo Min, and Kwanghoon Sohn. Laf-net: Locally adaptive fusion networks for stereo confidence estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 205–214, 2019. 2

[31] Young Min Kim, Christian Theobalt, James Diebel, Jana Kosecka, Branislav Miscusik, and Sebastian Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *2009 IEEE 12th international conference on computer vision workshops, ICCV workshops*, pages 1542–1549. IEEE, 2009. 2

[32] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–713, 2018. 2

[33] Andreas Kuhn, Christian Sormann, Mattia Rossi, Oliver Erdler, and Friedrich Fraundorfer. Deepc-mvs: Deep confidence prediction for multi-view stereo reconstruction. In *2020 International Conference on 3D Vision (3DV)*, pages 404–413. Ieee, 2020. 2

[34] Damien Lefloch, Tim Weyrich, and Andreas Kolb. Anisotropic point-based fusion. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 2121–2128. IEEE, 2015. 3

[35] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 6

[36] Will Maddern and Paul Newman. Real-time probabilistic fusion of sparse 3d lidar and dense stereo. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2181–2188. IEEE, 2016. 2

[37] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. *arXiv e-prints*, pages arXiv–2211, 2022. 1

[38] Giulio Marin, Pietro Zanuttigh, and Stefano Mattoccia. Reliable fusion of tof and stereo depth driven by confidence measures. In *European Conference on Computer Vision*, pages 386–401. Springer, 2016. 2

[39] Ricardo Martin-Brualla, Noha Radwan, Mehdi S.M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *the proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE/CVF, 2021. 5

[40] Diogo Martins, Kevin Van Hecke, and Guido De Croon. Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 849–856. IEEE, 2018. 2

[41] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019. 1

[42] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *the proceedings of the European Conference on Computer Vision (ECCV)*. CVF, 2020. 1

[43] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, volume 11, pages 127–136, 2011. 2, 5

[44] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. 2011. 2

[45] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32, 11 2013. 2

[46] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan I. Nieto. Voxblox: Incremental 3d euclidean signed distance fields for on-board MAV planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, pages 1366–1373. IEEE, 2017. 2

[47] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019. 1

[48] Kihong Park, Seungryong Kim, and Kwanghoon Sohn. High-precision depth estimation with the 3d lidar and stereo fusion. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2156–2163. IEEE, 2018. 2

[49] Sangwoo Park, Erchin Serpedin, and Khalid Qaraqe. Gaussian assumption: The least favorable but the most useful [lecture notes]. *IEEE Signal Processing Magazine*, 30(3):183–186, 2013. 3

[50] Vaishakh Patil, Wouter Van Gansbeke, Dengxin Dai, and Luc Van Gool. Don't forget the past: Recurrent depth estimation from monocular video. *IEEE Robotics and Automation Letters*, 5(4):6813–6820, 2020. 2

[51] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional Occupancy Networks. In *the proceedings of the European Conference Computer Vision (ECCV)*. CVF, 2020. 3

[52] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 2

[53] Matteo Poggi and Stefano Mattoccia. Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 138–147. IEEE, 2016. 2

[54] Matteo Poggi and Stefano Mattoccia. Learning from scratch a confidence measure. In *Bmvc*, volume 2, page 4, 2016. 2

[55] Jiaxiong Qiu, Zhaopeng Cui, Yinda Zhang, Xingdi Zhang, Shuaicheng Liu, Bing Zeng, and Marc Pollefeys. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3313–3322. Computer Vision Foundation / IEEE, 2019. 2

[56] Malcolm Reynolds, Jozef Doboš, Leto Peel, Tim Weyrich, and Gabriel J Brostow. Capturing time-of-flight data with confidence. In *CVPR 2011*, pages 945–952. IEEE, 2011. 2

[57] Denys Rozumnyi, Ian Cherabier, Marc Pollefeys, and Martin R. Oswald. Learned semantic multi-sensor depth map fusion. In *Internatioal Conference on Computer Vision Workshop (ICCVW), Workshop on 3D Reconstruction in the Wild, 2019*, Seoul, South Korea, 2019. 2

[58] Erik Sandström, Martin R. Oswald, Suryansh Kumar, Silvan Weder, Fisher Yu, Cristian Sminchisescu, and Luc Van Gool. Learning Online Multi-Sensor Depth Fusion. In *the proceedings of the European Conference Computer Vision (ECCV)*. CVF, 2022. 1, 2, 6, 7, 8

[59] Wojciech Sankowski, M Włodarczyk, Damian Kacperski, and Kamil Grabowski. Estimation of measurement uncertainty in stereo vision system. *Image and Vision Computing*, 61:70–81, 2017. 2

[60] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *European Conference on Computer Vision*, pages 1–19. Springer, 2022. 5

[61] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. 2019. 2

[62] Akihito Seki and Marc Pollefeys. Patch based confidence prediction for dense disparity map. In *BMVC*, volume 2, page 4, 2016. 2

[63] F. Steinbrucker, C. Kerl, D. Cremers, and J. Sturm. Large-scale multi-resolution surface reconstruction from rgb-d sequences. In *2013 IEEE International Conference on Computer Vision*, pages 3264–3271, 2013. 2

[64] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6, 7, 8

[65] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *the proceedings of the International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2012. 6, 7

[66] Wanjuan Su, Qingshan Xu, and Wenbing Tao. Uncertainty guided multi-view stereo network for depth estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11):7796–7808, 2022. 2

[67] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J. Davison. iMAP: Implicit Mapping and Positioning in Real-Time. In *the proceedings of the International Conference on Computer Vision (ICCV)*. IEEE/CVF, 2021. 1, 2

[68] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE, 1998. 5

[69] Fabio Tosi, Matteo Poggi, Antonio Benincasa, and Stefano Mattoccia. Beyond local reasoning for stereo confidence estimation with deep learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 319–334, 2018. 2

[70] Jeroen Van Baar, Paul Beardsley, Marc Pollefeys, and Markus Gross. Sensor fusion for depth estimation, including tof and thermal sensors. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 472–478. IEEE, 2012. 2

[71] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. Routedfusion: Learning real-time depth map fusion. *ArXiv*, abs/2001.04388, 2020. 2, 6

[72] Silvan Weder, Johannes L Schonberger, Marc Pollefeys, and Martin R Oswald. Neuralfusion: Online depth fusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3162–3172, 2021. 2

[73] Hongbin Xu, Zhipeng Zhou, Yali Wang, Wenxiong Kang, Baigui Sun, Hao Li, and Yu Qiao. Digging into uncertainty in self-supervised multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6078–6087, 2021. 2

[74] Zike Yan, Yuxin Tian, Xuesong Shi, Ping Guo, Peng Wang, and Hongbin Zha. Continual neural mapping: Learning an implicit scene representation from sequential observations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15782–15792, October 2021. 2

[75] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020. 2

[76] Sheng Yang, Beichen Li, Yan-Pei Cao, Hongbo Fu, Yu-Kun Lai, Leif Kobbelt, and Shi-Min Hu. Noise-resilient reconstruction of panoramas and 3d scenes using robot-mounted unsynchronized commodity rgb-d cameras. *ACM Transactions on Graphics (TOG)*, 39(5):1–15, 2020. 2

[77] Sheng Yang, Beichen Li, Minghua Liu, Yu-Kun Lai, Leif Kobbelt, and Shi-Min Hu. Heterofusion: Dense scene re-

construction integrating multi-sensors. *IEEE transactions on visualization and computer graphics*, 26(11):3217–3230, 2019. 2

[78] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 1, 6, 7, 8

[79] Wang Zhao, Shaohui Liu, Yi Wei, Hengkai Guo, and Yong-Jin Liu. A confidence-based iterative solver of depths and surface normals for deep multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6168–6177, 2021. 2

[80] Zihan Zhu, Songyou Peng, Viktor Larsson, Zhaopeng Cui, Martin R Oswald, Andreas Geiger, and Marc Pollefeys. Nicer-slam: Neural implicit scene encoding for rgb slam. *arXiv preprint arXiv:2302.03594*, 2023. 1

[81] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 1, 2, 3, 4, 5, 6, 7, 8