Learning Online Multi-sensor Depth Fusion

Erik Sandström[1(✉)], Martin R. Oswald[1,2], Suryansh Kumar[1], Silvan Weder[1], Fisher Yu[1], Cristian Sminchisescu[3,5], and Luc Van Gool[1,4]

[1] ETH Zürich, Zürich, Switzerland
esandstroem@ee.ethz.ch
[2] University of Amsterdam, Amsterdam, Netherlands
[3] Lund University, Lund, Sweden
[4] KU Leuven, Leuven, Belgium
[5] Google Research, Sunnyvale, USA

Abstract. Many hand-held or mixed reality devices are used with a single sensor for 3D reconstruction, although they often comprise multiple sensors. Multi-sensor depth fusion is able to substantially improve the robustness and accuracy of 3D reconstruction methods, but existing techniques are not robust enough to handle sensors which operate with diverse value ranges as well as noise and outlier statistics. To this end, we introduce SenFuNet,- a depth fusion approach that learns sensor-specific noise and outlier statistics and combines the data streams of depth frames from different sensors in an online fashion. Our method fuses multi-sensor depth streams regardless of time synchronization and calibration and generalizes well with little training data. We conduct experiments with various sensor combinations on the real-world CoRBS and Scene3D datasets, as well as the Replica dataset. Experiments demonstrate that our fusion strategy outperforms traditional and recent online depth fusion approaches. In addition, the combination of multiple sensors yields more robust outlier handling and more precise surface reconstruction than the use of a single sensor. The source code and data are available at https://github.com/tfy14esa/SenFuNet.

1 Introduction

Real-time online 3D reconstruction has become increasingly important with the rise of applications like mixed reality, autonomous driving, robotics, or live 3D content creation via scanning. The majority of 3D reconstruction hardware platforms like phones, tablets, or mixed reality headsets contain a multitude of sensors, but few algorithms leverage them jointly to increase their accuracy, robustness, and reliability. For instance, the HoloLens2 has four tracking cameras and a depth camera for mapping. Neither is its depth camera used for tracking, nor are the tracking cameras used for mapping. Fusing the data from multiple sensors

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19824-3_6.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13692, pp. 87–105, 2022.
https://doi.org/10.1007/978-3-031-19824-3_6

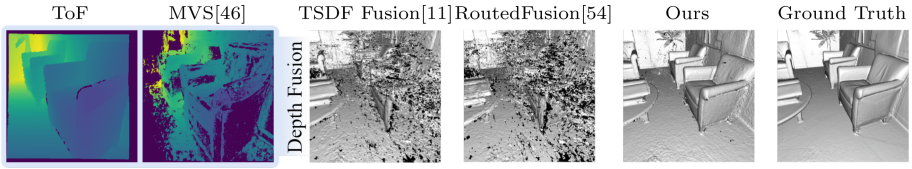


Fig. 1. Online multi-sensor depth map fusion. We fuse depth streams from a time-of-flight (ToF) camera and multi-view stereo (MVS) depth. Compared to competitive depth fusion methods such as TSDF Fusion and RoutedFusion, our learning-based approach can handle multiple depth sensors and significantly reduces the amount of outliers without loss of completeness.

is challenging since different sensors typically operate in different domains, have diverse value ranges as well as noise and outlier statistics. This diversity is, however, what motivates sensor fusion. For example, RGB stereo cameras typically have a larger field of view and higher resolution than time-of-flight (ToF) cameras, but typically struggle on homogeneously textured surfaces. ToF cameras perform well regardless of texture, but show performance drops around edges. Fig. 1 shows the online fusion result of a ToF camera and a multi-view stereo (MVS) depth sensor. Both traditional and recent learning-based techniques such as TSDF Fusion [11] and RoutedFusion [54] respectively, reveal a high degree of noise and outliers when fusing multi-sensor depth. Although other recent works tackle depth map fusion [23,50,55,56] with learning techniques, there is yet no work that considers multiple sensors for online dense reconstruction.

In this paper, we present an approach for sensor fusion (SenFuNet) which jointly learns (1) the iterative online fusion of depth maps from a single sensor and (2) the effective fusion of depth data from multiple different sensors. During training, our method learns relevant sensor properties which impact the reconstruction accuracy to locally emphasize the better sensor for particular input and geometry configurations (see Fig. 1). We demonstrate with multiple sensor combinations that the learned sensor weighting is generic and can also be used as an expert system, *e.g.* for fusing the results of different stereo methods. In this case, our method predicts which algorithm performs better on which part of the scene. Since our approach handles time asynchronous sensor inputs, it is also applicable to collaborative multi-agent reconstruction. Our contributions are:

- Our approach learns location-dependent fusion weights for the individual sensor contributions according to learned sensor statistics. For various sensor combinations our method extracts multi-sensor results that are consistently better than those obtained from the individual sensors.
- Our pipeline is end-to-end trained in an online manner, is light-weight, real-time capable and generalizes well even for small amounts of training data.
- In contrast to early fusion approaches which directly fuse depth values and thus generally assume a time synchronized sensor setup, our approach is flexible and can fuse the recovered scene reconstruction from asynchronous sensors. Our system is therefore more robust (compared to early fusion) to sensor differences such as sampling frequency, pose and resolution differences.

2 Related Work

In this section, we discuss dense online 3D reconstruction, multi-sensor depth fusion and multi-sensor dense 3D reconstruction.

Dense Online 3D Scene Reconstruction. The foundation for many volumetric online 3D reconstruction methods via truncated signed distance functions (TSDF) was laid by Curless and Levoy [11]. Popular extensions of this seminal work are KinectFusion [24] and scalable generalizations with voxel hashing [25,38,39], octrees [48], or increased pose robustness via sparse image features [5]. Further extensions include tracking for Simultaneous Localization and Mapping (SLAM) [37,47,50,60] which potentially also handle loop closures, *e.g.* BundleFusion [12]. To account for greater depth noise, RoutedFusion [54] learns online updates of the volumetric grid. NeuralFusion [55] extends this idea by additionally learning the scene representation which significantly improves robustness to outliers. DI-Fusion [23], similarly to [55], learns the scene representation, but additionally decodes a confidence of the signed distance per voxel. Continual Neural Mapping [56] learns a continuous scene representation through a neural network from sequential depth maps. Several recent works do not require depth input and instead perform online reconstruction from RGB-cameras such as Atlas [35], VolumeFusion [9], TransformerFusion [4] and NeuralRecon [51]. None of these approaches consider multiple sensors and their extensions to sensor-aware data fusion is often by no means straightforward. Nevertheless, by treating all sensors equally, they can be used as baseline methods.

The majority of the aforementioned traditional methods do not properly account for varying noise and outlier levels for different depth values, which are better handled by probabilistic fusion methods [15–17,28]. Cao *et al.* [7] introduced a probabilistic framework via a Gaussian mixture model into a surfel-based reconstruction framework to account for uncertainties in the observed depth. For a recent survey on online RGB-D 3D scene reconstruction, readers are referred to [61]. Overall, none of the state-of-the-art methods for dense online 3D scene reconstruction consider multiple sensors.

Multi-sensor Depth Fusion. The task of fusing depth maps from diverse sensors has been studied extensively. Many works study the fusion of a specific set of sensors. For example, RGB stereo and time-of-flight (ToF) [1,2,10,13,14,18,33,52], RGB stereo and Lidar [31], RGB and Lidar [40,41,44], RGB stereo and monocular depth [34] and the fusion of multiple RGB stereo algorithms [42]. All these methods only study a specific set of sensors, while we do not enforce such a limitation. Few works study the fusion of arbitrary depth sensors [43]. Contrary to our method, all methods performing depth map fusion assume time synchronized sensors, which is hard, if not impossible, to achieve with realistic multi-sensor equipment.

Multi-sensor Dense 3D Reconstruction. Some works consider the problem of offline multi-sensor dense 3D reconstruction. For example, depth map fusion for semantic 3D reconstruction [45], combining multi-view stereo with a ToF sensor in a probabilistic framework [27], the combination of a depth sensor

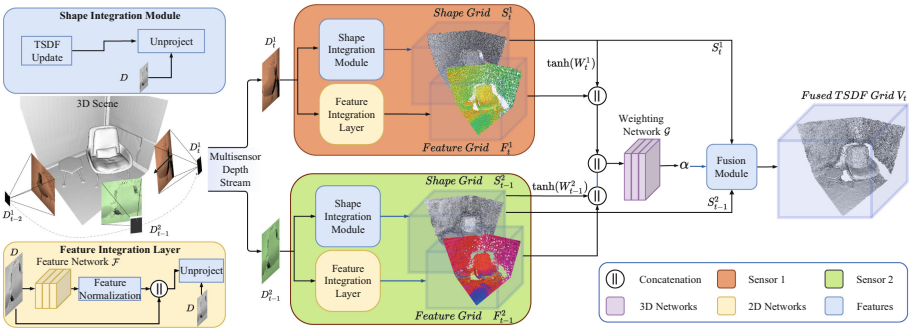


Fig. 2. SenFuNet Architecture. Given a depth stream D_t^i, with known camera poses, our method fuses each frame at time t from sensor i into global sensor-specific shape S_t^i and feature F_t^i grids. The **Shape Integration Module** fuses the frames into $S_t^i = \{V_t^i, W_t^i\}$ consisting of a TSDF grid V_t^i and a weight counter grid W_t^i. In parallel, the **Feature Integration Layer** extracts features from the depth maps using a 2D Feature Network $\mathcal{F}^i$ and integrates them into the feature grid F_t^i. Next, S_t^i and F_t^i are combined and decoded through a 3D **Weighting Network** $\mathcal{G}$ into a sensor weighting $\alpha \in [0, 1]$. Together with S_t^i and α, the **Fusion Module** computes the fused grid V_t at each voxel location.

with photometric stereo [6] and large scene reconstruction using unsynchronized RGBD cameras mounted on an indoor robot [57]. These offline methods do not address the online problem setting that we are concerned with. Some works use sensor fusion to achieve robust pose estimation in an online setting [20,58]. In contrast to our method, these works do not leverage sensor fusion for mapping. Ali *et al.* [3] present an online framework which perhaps is most closely related to our work. They take Lidar and stereo depth maps as input and fuse the TSDF signals of both sensors with a linear average before updating the global grid using TSDF Fusion [11]. To reduce noise further, they optimize a least squares problem which encourages surface smoothing. Contrary to our method, no learning is used and their system is only designed to fuse stereo depth with Lidar.

3 Method

Overview. Given multiple noisy depth streams $D_t^i : \mathbb{R}^2 \rightarrow \mathbb{R}$ from different sensors with known camera calibration, *i.e.* extrinsics $P_t^i \in \mathbb{SE}(3)$ and intrinsics $K^i \in \mathbb{R}^{3\times3}$, our method integrates each depth frame at time $t \in \mathbb{N}$ from sensor $i \in \{1, 2\}$ into a globally consistent shape S_t^i and feature F_t^i grid. Through a series of operations, we then decode S_t^i and F_t^i into a fused TSDF grid $V_t \in \mathbb{R}^{X\times Y\times Z}$, which can be converted into a mesh with marching cubes [29]. Our overall framework can be split into four components (see Fig. 2). First, the **Shape Integration Module** integrates depth frames D_t^i successively into the zero-initialized shape representation $S_t^i = \{V_t^i, W_t^i\}$. S_t^i consists of a TSDF grid $V_t^i \in \mathbb{R}^{X\times Y\times Z}$ and a corresponding weight grid $W_t^i \in \mathbb{N}^{X\times Y\times Z}$, which keeps track

of the number of updates to each voxel. In parallel, the **Feature Integration Layer** extracts features from the depth maps using a 2D feature network $\mathcal{F}^i : D_t^i \in \mathbb{R}^{W\times H\times 1} \rightarrow f_t^i \in \mathbb{R}^{W\times H\times n}$, with n being the feature dimension. We use separate feature networks per sensor to learn sensor-specific depth dependent statistics such as shape and edge information. The extracted features f_t^i are then integrated into the zero-initialized feature grid $F_t^i \in \mathbb{R}^{X\times Y\times Z\times n}$. Next, S_t^i and F_t^i are combined and decoded through a 3D **Weighting Network** $\mathcal{G}$ into a location-dependent sensor weighting $\alpha \in [0, 1]$. Together with S_t^i and α, the **Fusion Module** fuses the information into V_t at each voxel location. Key to our approach is the separation of per sensor information into different representations along with the successive aggregation of shapes and features in the 3D domain. This strategy enables $\mathcal{G}$ to learn a fusion strategy of the incoming multi-sensor depth stream. Our method is able to fuse the sensors in a spatially dependent manner from a smooth combination to a hard selection as illustrated in Fig. 3. Our scheme hence avoids the need to perform post-outlier filtering by thresholding with the weight W_t^i, which is difficult to tune and is prone to reduce scene completion [54]. Another popular outlier filtering technique is free-space carving[1], but this can be computationally expensive and is not required by our method. Instead, we use the learned α as part of an outlier filter at test time, requiring no manual tuning. Next, we describe each component in detail.

(a) Shape Integration Module. For each depth map D_t^i and pixel, a full perspective unprojection of the depth into the world coordinate frame yields a 3D point $\mathbf{x}_w \in \mathbb{R}^3$. Along each ray from the camera center, centered at $\mathbf{x}_w$, we sample T points uniformly over a predetermined distance l. The coordinates are then converted to the voxel space and a local shape grid $S_{t-1}^{i,*}$ is extracted from S_{t-1}^i through nearest neighbor extraction. To incrementally update the local shape grid, we follow the moving average update scheme of TSDF Fusion [11]. For numerical stability, the weights are clipped at a maximum weight $\omega_{\max}$. For more details, see the suppl. material.

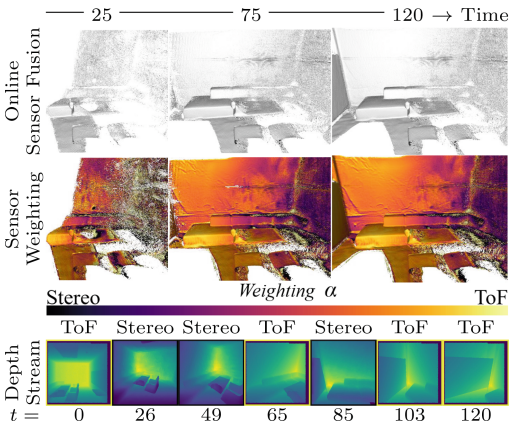


Fig. 3. Overview. Left to right: Sequential fusion of a multi-sensor noisy depth stream. Our method integrates each depth frame at time t and produces a sensor weighting which fuses the sensors in a spatially dependent manner. For example, areas in yellow denote high trust of the ToF sensor.

[1] Enforcing free space for voxels along the ray from the camera to the surface [36]. Note that outliers behind surfaces are not removed with this technique.

(b) Feature Integration Layer. Each depth map D_t^i is passed through a 2D network $\mathcal{F}^i$ to extract context information f_t^i, which can be useful during the sensor fusion process. When fusing sensors based on the stereo matching principle, we provide the RGB frame as additional input channels to $\mathcal{F}^i$. The network is fully convolutional and comprises 7 network blocks, each consisting of the following operations, 1) a 3×3 convolution with zero padding 1 and input channel dimension 4 and output dimension 4 (except the first block which takes 1 channel as input when only depth is provided), 2) a tanh activation, 3) another 3×3 convolution with zero padding 1 outputting 4 channels and 4) a tanh activation. The output of the first six blocks is added to the output of the next block via residual connections. Finally, we normalize the feature vectors at each pixel location and concatenate the input depth.

Next, we repeat the features T times along the direction of the viewing ray from the camera, $f_t^i \xrightarrow[\text{T times}]{\text{Repeat}} f_t^{i,T} \in \mathbb{R}^{W \times H \times T \times n}$. The local feature grid $F_{t-1}^{i,*}$ is then updated using the precomputed update indices from the Shape Integration Module with a moving average update: $F_t^{i,*} = \frac{W_{t-1}^{i,*} F_{t-1}^{i,*} + f_t^{i,T}}{W_{t-1}^{i,*}+1}$. For all update locations the grid F_t^i is replaced with $F_t^{i,*}$.

(c) Weighting Network. The task of the weighting network $\mathcal{G}$ is to predict the optimal fusion strategy of the surface hypotheses V_t^i. The input to the network is prepared by first concatenating the features F_t^i and the tanh-transformed weight counters W_t^i and second by concatenating the resulting vectors across the sensors. Due to memory constraints, the entire scene cannot be fit onto the GPU, and hence we use a sliding-window approach at test time to feed $\mathcal{G}$ chunks of data. First, the minimum bounding grid of the measured scene (*i.e.* where $W_t^i > 0$) is extracted from the global grids. Then, the extracted grid is parsed using chunks of size $d \times d \times d$ through $\mathcal{G} : \mathbb{R}^{d \times d \times d \times 2(n+1)} \rightarrow \alpha \in \mathbb{R}^{d \times d \times d \times 1}$ into $\alpha \in [0, 1]$. To avoid edge effects, we use a stride of $d/2$ and update the central chunk of side length $d/2$. The architecture of $\mathcal{G}$ combines 2 layers of 3D-convolutions with kernel size 3 and replication padding 1 interleaved with ReLU activations. The first layer outputs 32 and the second layer 16 channels. Finally, the 16-dimensional features are decoded into the sensor weighting α by a 1×1×1 convolution followed by a sigmoid activation.

(d) Fusion Module. The task of the fusion module is to combine α with the shapes S_t^i. In the following, we define the set of voxels where only sensor 1 integrates as $C^1 = \{W_t^1 > 0,\ W_{t-1}^2 = 0\}$, the set where only sensor 2 integrates as $C^2 = \{W_t^1 = 0,\ W_{t-1}^2 > 0\}$ and the set where both sensors integrate as $C^{12} = \{W_t^1 > 0,\ W_{t-1}^2 > 0\}$. Let us also introduce $\alpha_1 = \alpha$ and $\alpha_2 = 1 - \alpha$. The fusion module computes the fused grid V_t as

$$V_t = \begin{cases} \alpha_1 V_t^1 + \alpha_2 V_{t-1}^2 & \text{if } C^{12} \\ V_t^1 & \text{if } C^1 \\ V_{t-1}^2 & \text{if } C^2. \end{cases} \tag{1}$$

Depending on the voxel set, V_t is computed either as a weighted average of the two surface hypotheses or by selecting one of them. With only one sensor observation, a weighted average would corrupt the result. Hence, the single observed surface is selected. At test time, we additionally apply a learned **Outlier Filter** which utilizes α_i and W_t^i. The filter is formulated for sensors 1 and 2 as

$$\hat{W}_t^1 = \mathbb{1}_{\{C^1,\ \alpha_1 > 0.5\}} W_t^1, \quad \hat{W}_{t-1}^2 = \mathbb{1}_{\{C^2,\ \alpha_2 > 0.5\}} W_{t-1}^2, \tag{2}$$

where $\mathbb{1}_{\{.\}}$ denotes the indicator function[2]. When only one sensor is observed at a certain voxel, we remove the observation if α_i, which could be interpreted as a confidence, is below 0.5. This is done by setting the weight counter to 0.

Loss Function. The full pipeline is trained end-to-end using the overall loss

$$\mathcal{L} = \mathcal{L}_f + \lambda_1 \sum_{i=1}^{2} \mathcal{L}_{C^i}^{in} + \lambda_2 \sum_{i=1}^{2} \mathcal{L}_{C^i}^{out}. \tag{3}$$

The term $\mathcal{L}_f$ computes the mean L_1 error to the ground truth TSDF masked by C^{12} (4). To supervise the voxel sets C^1 and C^2, we introduce two additional terms, which penalize L_1 deviations from the optimal α. The purpose of these terms is to provide a training signal for the outlier filter. If the L_1 TSDF error is smaller than some threshold η, the observation is deemed to be an inlier, and the corresponding confidence α_i should be 1, otherwise 0. The loss is computed as the mean L_1 error to the optimal α:

$$\mathcal{L}_f = \frac{1}{N_{C^{12}}} \sum \mathbb{1}_{\{C^{12}\}} |V_t - V^{GT}|_1,\ \mathcal{L}_{C^i}^{in} = \frac{1}{N_{C^i}^{in}} \sum \mathbb{1}_{\{C^i,\ |V_t - V^{GT}|_1 < \eta\}} |\alpha_i - 1|_1,$$
$$\mathcal{L}_{C^i}^{out} = \frac{1}{N_{C^i}^{out}} \sum \mathbb{1}_{\{C^i,\ |V_t - V^{GT}|_1 > \eta\}} |\alpha_i|_1, \tag{4}$$

where the normalization factors are defined as

$$N_{C^{12}} = \sum \mathbb{1}_{\{C^{12}\}},\ N_{C^i}^{in} = \sum \mathbb{1}_{\{C^i,\ |V_t - V^{GT}|_1 < \eta\}},$$
$$N_{C^i}^{out} = \sum \mathbb{1}_{\{C^i,\ |V_t - V^{GT}|_1 > \eta\}}. \tag{5}$$

Training Forward Pass. After the integration of a new depth frame D_t^i into the shape and feature grids, the update indices from the Shape Integration Module are used to compute the minimum bounding box of update voxels in S_t^i and F_t^i. The update box varies in size between frames and cannot always fit on the GPU. Due to this and for the sake of training efficiency, we extract a $d \times d \times d$ chunk from within the box volume. The chunk location is randomly selected using a uniform distribution along each axis of the bounding box. If the bounding box volume is smaller along any dimension than d, the chunk shrinks to the minimum size along the affected dimension. To maximize the number of voxels that are used to train the networks F^i, we sample a chunk until we find one with at least 2000

[2] See supplementary material for a definition.

update indices. At most, we make 600 attempts. If not enough valid indices are found, the next frame is integrated. The update indices in the chunk are finally used to mask the loss. We randomly reset the shape and feature grids with a probability of 0.01 at each frame integration to improve training robustness.

4 Experiments

We first describe our experimental setup and then evaluate our method against state-of-the-art online depth fusion methods on Replica, the real-world CoRBS and the Scene3D datasets. All reported results are averages over the respective test scenes. Further experiments and details are in the supplementary material.

Implementation Details. We use $\omega_{\max} = 500$ and extract $T = 11$ points along the update band $l = 0.1$ m. We store $n = 5$ features at each voxel location and use a chunk side length of $d = 64$. For the loss (3) $\lambda_1 = 1/60$, $\lambda_2 = 1/600$ and $\eta = 0.04$ m. In total, the networks of our model comprise 27.7K parameters, where 24.3K are designated to $\mathcal{G}$ and the remaining parameters are split equally between $\mathcal{F}^1$ and $\mathcal{F}^2$. For our method and all baselines, the image size is $W = H = 256$, the voxel size is 0.01 m and we mask the 10 pixel border of all depth maps to avoid edge artifacts, *i.e.* pixels belonging to the mask are not integrated into 3D. Since our TSDF updates cannot be larger than 0.05 m, we truncate the ground truth TSDF grid at $l/2 = 0.05$ m.

Evaluation Metrics. The TSDF grids are evaluated using the Mean Absolute Distance (MAD), Mean Squared Error (MSE), Intersection over Union (IoU) and Accuracy (Acc.). The meshes, produced by marching cubes [29] from the TSDF grids, are evaluated using the F-score which is the harmonic mean of the Precision (P) and Recall (R).

Baseline Methods. Since there is no other multi-sensor online 3D reconstruction method that addresses the same problem, we define our own baselines by generalizing single sensor fusion pipelines to multiple sensors. TSDF Fusion [11] is the gold standard for fast, dense mapping of posed depth maps. It generalizes to the multi-sensor setting effortlessly by integrating all depth frames into the same TSDF grid at runtime. RoutedFusion [54] extends TSDF Fusion by learning the TSDF mapping. We generalize RoutedFusion to multiple sensors by feeding all depth frames into the same TSDF grid, but each sensor is assigned a separate fusion network to account for sensor-dependent noise[3]. NeuralFusion [55] extends RoutedFusion for better outlier handling, but despite efforts and help from the authors, the network did not converge during training due to heavy loss oscillations caused by integrating different sensors. DI-Fusion [23] learns the scene representation and predicts the signed distance value as well as the uncertainty σ per voxel. We use the provided model from the authors and integrate all frames from both sensors into the same volumetric grid. In the following, we refer

[3] Additionally, we tweak the original implementation to get rid of outliers. See supplementary material.

to each multi-sensor baseline by using the corresponding single sensor name. For additional comparison, when time synchronized sensors with ground truth depth are available, we train a so-called "Early Fusion" baseline by fusing the 2D depth frames of both sensors. The fusion is performed with a modified version of the 2D denoising network proposed by Weder *et al.* [54] followed by TSDF Fusion to attain the 3D model (see supplementary material). This baseline should be interpreted as a light-weight alternative to our proposed SenFuNet, but assumes synchronized sensors, which SenFuNet does not. Finally, for the single-sensor results, we evaluate the TSDF grids V_t^i. To make the comparisons fair, we do not use weight counter thresholding as post-outlier filter for any method. For DI-Fusion, we filter outliers by thresholding the learned voxel uncertainty. The default value provided in the implementation is used.

Table 1. Replica Dataset. ToF+PSMNet Fusion. (a) Our method outperforms the baselines as well as both of the sensor inputs and sets a new state-of-the-art for multi-sensor online depth fusion. (b) The denoising network mitigates outliers along planar regions, compare to Table 1a. Our method even outperforms the Early Fusion baseline, which assumes synchronized sensors.

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor							
PSMNet [8]	7.30	1.95	0.664	83.23	56.20	43.10	81.34
ToF [21]	7.48	1.99	0.664	83.65	58.52	45.84	84.85
Multi-Sensor Fusion							
TSDF Fusion [11]	8.20	2.11	0.669	84.09	49.58	35.33	**85.44**
RoutedFusion [54]	5.62	1.66	0.735	87.22	61.08	49.51	79.92
DI-Fusion [23] σ=0.15	-	-	-	-	48.39	34.24	85.29
SenFuNet (Ours)	**4.65**	**1.54**	**0.753**	**88.05**	**69.29**	**62.05**	79.81

(a) Without depth denoising.

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor							
PSMNet [8]	6.35	1.77	0.673	84.54	60.28	48.26	80.41
ToF [21]	5.08	1.58	0.709	87.32	68.93	59.01	83.08
Multi-Sensor Fusion							
TSDF Fusion [11]	6.40	1.80	0.681	85.31	52.93	38.95	**84.60**
RoutedFusion [54]	6.04	1.68	0.644	85.10	62.67	51.75	79.52
Early Fusion	6.40	1.40	0.760	89.02	74.60	67.46	83.47
DI-Fusion [23] σ=0.15	-	-	-	-	55.66	41.49	85.33
SenFuNet (Ours)	**3.49**	**1.31**	**0.761**	**89.61**	**76.47**	**73.58**	79.77

(b) With depth denoising

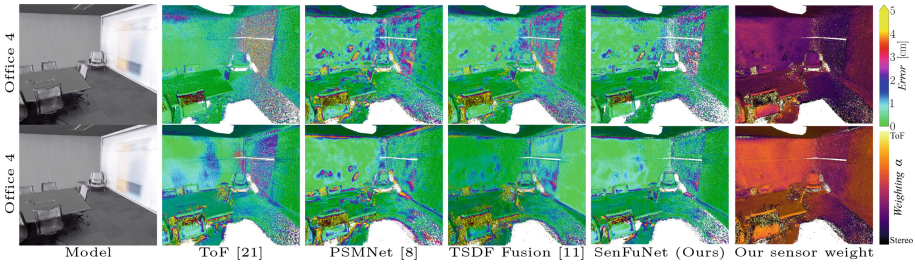


Fig. 4. Replica Dataset. Our method fuses the sensors consistently better than the baselines. Concretely, our method learns to detect and remove outliers much more effectively (best viewed on screen). **Top row:** ToF+PSMNet Fusion without denoising. See also Table 1a. **Bottom row:** ToF+PSMNet Fusion with denoising. See also Table 1b.

4.1 Experiments on the Replica Dataset

The Replica dataset [49] comprises high-quality 3D reconstructions of a variety of indoor scenes. We collect data from Replica to create a multi-sensor dataset

suitable for depth map fusion. To prepare ground truth signed distance grids, we first make the 3D meshes watertight using screened Poisson surface reconstruction [26]. The meshes are then converted to signed distance grids using a modified version of mesh-to-sdf[4] to accommodate non-cubic voxel grids. Ground truth depth and an RGB stereo pair are extracted using AI Habitat [32] along random trajectories. In total, we collected 92698 frames. We use 7 training and 3 test scenes. We simulate a depth sensor by adding noise to the GT depth of the left stereo view. Correspondingly, from the RGB stereo pairs a left stereo view depth map can be predicted using (optionally multi-view) stereo algorithms. In the following, we construct two sensor combinations and evaluate our model.

ToF+PSMNet Fusion. We simulate a ToF sensor by adding realistic noise to the ground truth depth maps[5] [21]. To balance the two sensors, we increase the noise level by a factor 5 compared to the original implementation. We simulate another depth sensor from the RGB stereo pair using PSMNet [8]. We train the network on the Replica train set and keep it fixed while training our pipeline. Table 1a shows that our method outperforms both TSDF Fusion, RoutedFusion and DI-Fusion on all metrics except Recall with at least 13% on the F-score. Additionally, the F-score improves with a minimum of 18% compared to the input sensors. Specifically, note the absence of outliers (colored yellow) in Fig. 4 *Top row* when comparing our method to TSDF Fusion. Also note the sensor weighting *e.g.* we find lots of noise on the right wall of the ToF scene and thus, our method puts more weight on the stereo sensor in this region.

Weder *et al.* [54] showed that a 2D denoising network (called routing network in the paper) that preprocesses the depth maps can improve performance when noise is present in planar regions. To this end, we train our own denoising network on the Replica train set and train a new model which applies a fixed denoising network. According to Table 1, this yields a gain of 10% on the F-score of the fused model compared to without using a denoising network, see also Fig. 4 *Bottom row*. Early Fusion is a strong alternative to our method when the sensors are synchronized. We want to highlight, however, that the resource overhead of our method is worthwhile since we outperform Early Fusion even in the synchronized setting.

Time Asynchronous Evaluation. RGB cameras often have higher frame rates than ToF sensors which makes Early Fusion more challenging as one sensor might lack new data. We simulate this setting by giving the PSMNet sensor twice the sampling rate of the ToF sensor, *i.e.* we drop every second ToF frame. To provide a correspond-

Table 2. Time Asynchronous Evaluation. SenFuNet outperforms Early Fusion for sensors with different sampling frequencies. *With depth denoising.

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ %	P↑ [%]	R↑ [%]
Early Fusion	7.66	1.99	0.642	84.65	61.34	48.47	**83.63**
SenFuNet (Ours)	4.21	1.45	0.755	88.26	73.04	69.13	78.43
SenFuNet (Ours)*	**3.15**	**1.23**	**0.760**	**89.52**	**79.26**	**79.91**	78.79

[4] https://github.com/marian42/mesh_to_sdf.
[5] http://redwood-data.org/indoor/dataset.html.

ing ToF frame for Early Fusion, we reproject the latest observed ToF frame into the current view of the PSMNet sensor. As demonstrated in Table 2 the gap between our SenFuNet late fusion approach and Early Fusion becomes even larger (cf. Table 1 (b)).

Table 3. Replica Dataset. SGM+PSMNet Fusion. Our method does not assume a particular sensor pairing and works well for all tested sensors. The gain from the denoising network is marginal with a 2% F-score improvement since there are few outliers on planar regions of the stereo depth maps and the denoising network over-smooths the depth along discontinuities. Our method outperforms Early Fusion (which generally assumes synchronized sensors) on most metrics even without depth denoising.

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor							
PSMNet [8]	7.30	1.95	0.664	83.23	56.20	43.10	81.34
SGM [22]	8.90	2.17	0.610	81.48	57.71	44.40	84.08
Multi-Sensor Fusion							
TSDF Fusion [11]	9.17	2.24	0.634	82.62	47.75	33.39	85.10
RoutedFusion [54]	7.11	1.82	0.671	84.63	60.31	48.47	80.21
DI-Fusion [23] σ=0.15	-	-	-	-	47.29	32.92	**85.14**
SenFuNet (Ours)	**4.77**	**1.56**	**0.738**	**87.62**	**69.83**	**63.20**	79.12

(a) Without depth denoising

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor							
PSMNet [8]	6.35	1.77	0.673	84.54	60.28	48.26	80.41
SGM [22]	6.60	1.80	0.659	84.29	60.79	49.78	78.13
Multi-Sensor Fusion							
TSDF Fusion [11]	7.28	1.93	0.669	84.74	54.09	40.45	81.65
RoutedFusion [54]	8.09	2.05	0.580	80.18	59.88	47.13	82.12
Early Fusion	4.99	1.51	0.707	86.99	69.40	61.07	80.36
DI-Fusion [23] σ=0.15	-	-	-	-	52.65	38.50	**83.62**
SenFuNet (Ours)	**4.04**	**1.41**	**0.737**	**88.11**	**71.18**	**66.81**	76.27

(b) With depth denoising

SGM+PSMNet Fusion. Our method does not assume a particular sensor pairing. We show this, by replacing the ToF sensor with a stereo sensor acquired using semi-global matching [22]. In Table 3, we show state-of-the-art sensor fusion performance both with and without a denoising network. The denoising network tends to over-smooth depth discontinuities, which negatively affects performance when few outliers exist. Additionally, even without using a denoising network, we outperform Early Fusion on most metrics. TSDF Fusion, RoutedFusion and DI-Fusion aggregate outliers across the sensors leading to worse performance than the single sensor results.

4.2 Experiments on the CoRBS Dataset

The real-world CoRBS dataset [53] provides a selection of reconstructed objects with very accurate ground truth 3D and camera trajectories along with a consumer-grade RGBD camera. We apply our method to the dataset by training a model on the desk scene and testing it on the human scene. The procedure to create the ground truth signed distance grids is identical to the Replica dataset. We create an additional depth sensor along the ToF depth using MVS with COLMAP [46][6]. Fig. 5 shows that our model can fuse very imbalanced sensors while the baseline methods fail severely. Even if one sensor (MVS) is significantly worse, our method still improves on most metrics. This confirms that our method learns to meaningfully fuse the sensors even if one sensor adds very little.

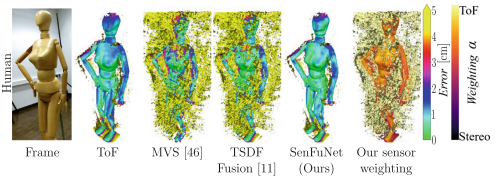
[6] Unfortunately, no suitable public real 3D dataset exists, which comprises binocular stereo pairs, and an active depth sensor, as well as ground truth geometry.

4.3 Experiments on the Scene3D Dataset

We demonstrate that our framework can fuse imbalanced sensors on room-sized real-world scenes using the RGBD Scene3D dataset [59]. The Scene3D dataset comprises a collection of 3D models of indoor and outdoor scenes. We train our model on the stonewall scene and test it on the copy room scene. To create the ground truth training grid, we follow the steps outlined previously except that it was not necessary to make the mesh watertight. We fuse every 10th frame during train and test time. Equivalent to the study on CoRBS, we create an MVS depth sensor using COLMAP and perform ToF and MVS fusion. We only integrate MVS depth in the interval $[0.5, 3.0]$ m. Table 4a along with Fig. 6 *Top row* shows that our method yields a fused result better than the individual input sensors and the baseline methods. Further, Fig. 1 shows our method in comparison with TSDF Fusion [11] and RoutedFusion [54] on the lounge scene.

Model \ Metric	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor							
MVS [46]	15.39	3.25	0.263	74.84	17.73	9.77	95.96
ToF	7.11	**1.52**	0.486	83.13	72.98	57.85	98.82
Multi-Sensor Fusion							
TSDF Fusion [11]	15.09	3.15	0.290	75.96	18.13	9.98	**99.19**
RoutedFusion [54]	23.09	4.40	0.222	49.42	2.36	1.27	16.43
DI-Fusion [23] σ=0.15	-	-	-	-	28.19	16.71	90.15
SenFuNet (Ours)	**6.53**	1.55	**0.510**	**84.88**	**74.56**	**59.74**	99.16

(a)



(b)

Fig. 5. CoRBS Dataset. ToF+MVS Fusion. Our model can find synergies between very imbalanced sensors. (a) The numerical results show that our fused model is better than the individual depth sensor inputs and significantly better than any of the baseline methods. (b) Contrary to our method, the baseline methods cannot handle the high degree of outliers from the MVS sensor.

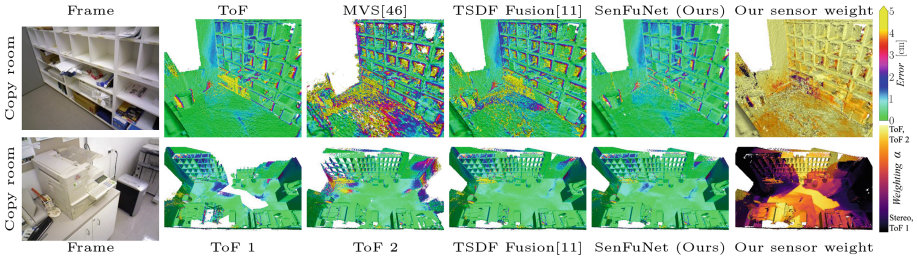


Fig. 6. Top row: Our method effectively fuses the ToF and MVS sensors. Note specifically the absence of yellow outliers in the corner of the bookshelf. See also Table 4a. **Bottom row:** Multi-Agent ToF Reconstruction. Our method is flexible and can perform Multi-Agent reconstruction. Note that our model learns where to trust each agent at different spatial locations for maximum completeness, while also being noise aware. See for instance the left bottom corner of the bookshelf where both agents integrate, but the noise-free agent is given a higher weighting. The above scene is taken from the Scene3D Dataset [59] (best viewed on screen). See also Table 4b.

Multi-agent ToF Fusion. Our method is not exclusively applicable to sensor fusion, but more flexible. We demonstrate this by formulating a Multi-Agent reconstruction problem, which assumes that two identical ToF sensors with different camera trajectories are provided. The task is to fuse the reconstructions from the two agents. This requires an understanding of when to perform sensor selection for increased completeness and smooth fusion when both sensors have registered observations. Note that this formulation is different from typical works on collaborative 3D reconstruction *e.g.* [19] where the goal is to align 3D reconstruction fragments to produce a complete model. In our Multi-Agent setting, the alignment is given and the task is instead to perform data fusion on the 3D fragments. No modification of our method is required to perform this task. We set $\lambda_1 = 1/1200$ and $\lambda_2 = 1/12000$ and split the original trajectory into 100 frame chunks that are divided between the agents. Table 4b and Fig. 6 *Bottom row* show that our method effectively fuses the incoming data streams and yields a 4% F-score gain on the TSDF Fusion baseline.

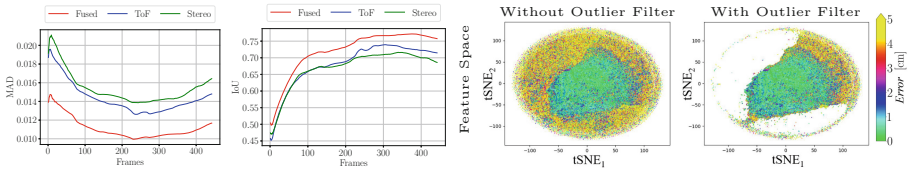
Table 4. Scene3D Dataset. (a) **ToF+MVS Fusion.** Our method outperforms the baselines on real-world data on a room-sized scene. (b) **Multi-Agent ToF Reconstruction.** Our method is flexible and can perform collaborative sensor fusion from multiple sensors with different camera trajectories.

Model	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor			
MVS [46]	44.26	32.08	71.33
ToF	90.73	85.53	96.61
Multi-Sensor Fusion			
TSDF Fusion [11]	54.77	38.11	97.29
RoutedFusion [54]	53.98	37.73	94.86
DI-Fusion [23] σ=0.15	48.84	32.50	**98.24**
SenFuNet (Ours)	**93.39**	**91.28**	95.60

(a)

Model	F↑ [%]	P↑ [%]	R↑ [%]
Single Sensor			
ToF 1	84.68	89.92	80.01
ToF 2	77.77	79.65	75.97
Multi-Sensor Fusion			
TSDF Fusion [11]	90.73	85.53	96.61
RoutedFusion[54]	84.11	74.16	97.14
DI-Fusion [23] σ=0.15	86.31	77.27	**97.74**
SenFuNet (Ours)	**93.73**	**91.56**	96.00

(b)



(a) (b)

Fig. 7. (a) **Performance over Camera Trajectory.** Our fused output outperforms the single sensor reconstructions (V_t^i) for all frames along the camera trajectory. The above results are evaluated on the Replica *office 0* scene using {ToF, PSMNet} with depth denoising. Note that the results get slightly worse after 300 frames. This is due to additional noise from the depth sensors when viewing the scene from further away. (b) **Effect of Learned Outlier Filter.** The learned filter is crucial for robust outlier handling. Erroneous outlier predictions shown in yellow are effectively removed by our approach while keeping the correct green-colored predictions. (Color figure online)

4.4 More Statistical Analysis

Performance over Camera Trajectory. To show that our fused output is not only better at the end of the fusion process, we visualize the quantitative performance across the accumulated trajectory. In Fig. 7a, we evaluate the office

0 scene on the sensors {ToF, PSMNet} with depth denoising. Our fused model consistently improves on the inputs.

Architecture Ablation. We perform a network ablation on the Replica dataset with the sensors {SGM, PSMNet} without depth denoising. In Table 5, we investigate the number of layers with kernel size 3 in the Weighting Network $\mathcal{G}$. Two layers yield optimal performance which amounts to a receptive field of $9 \times 9 \times 9$. This is realistic given that the support for a specific sensor is local by nature.

Table 5. Architecture Ablation. We vary the number of 3D convolutional layers with kernel size 3. Performance is optimal at 2 layers, equivalent to a receptive field of 9^3.

# Layers	0	1	2	3	4
F1	48.57	68.47	**69.86**	69.45	68.21

Generalization Capability. Table 6 shows our model's generalization ability for {SGM, PSMNet} fusion when evaluated against a model trained and tested on the *office 0* scene. Our model generalizes well and performs almost on par with one which is only trained on the *office 0* scene. The generalization capability is not surprising since $\mathcal{G}$ has a limited receptive field of $9 \times 9 \times 9$.

Table 6. Generalization Capability. Our model generalizes well when evaluated on the *office 0* scene. We conclude this by training a model only on *office 0*.

Train Set	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ %	P↑ [%]	R↑ [%]
Full Train Set	**4.51**	1.54	0.748	87.78	68.70	**59.49**	81.28
Office 0	4.54	**1.53**	**0.752**	**87.97**	**69.23**	59.45	**82.86**

Effect of Learned Outlier Filter. To show the effectiveness of the filter, we study the feature space on the input side of $\mathcal{G}$. Specifically, we consider the *hotel 0* scene and sensors {ToF, PSMNet} with depth denoising. First, we concatenate both feature grids and flatten the resulting grid. Then, we reduce the observations of the 12-dim feature space to a 2-dim representation using tSNE [30]. We then colorize each point with the corresponding signed distance error at the original voxel position. We repeat the visualization with and without the learned outlier filter. Fig. 7b shows that the filter effectively removes outliers while keeping good predictions.

Loss Ablation. Table 7 shows the performance difference when the model is trained only with the term $\mathcal{L}_f$ compared to the full loss (3). We perform {SGM, PSMNet} fusion on the Replica dataset. The extra terms of the full loss clearly help improve overall performance and specifically to filter outliers.

Table 7. Loss Ablation. When only the term $\mathcal{L}_f$ is used, we observe a significant performance drop compared to the full loss. Note, however, that only with the term $\mathcal{L}_f$, our model still improves on the single sensor input metrics compared to Table 3a.

Loss $\mathcal{L}$	MSE↓ *e-04	MAD↓ *e-02	IoU↑ [0,1]	Acc.↑ [%]	F↑ [%]	P↑ [%]	R↑ [%]
Only $\mathcal{L}_f$	6.04	1.78	0.710	86.20	62.65	50.88	**82.17**
Full Loss	**4.77**	**1.56**	**0.738**	**87.62**	**69.83**	**63.20**	79.12

Limitations. Our framework currently supports two sensors. Its extension to a k-sensor setting is straightforward, but the memory footprint grows linearly

with the number of sensors. However, few devices have more than two or three depth sensors. While our method generates better results on average, some local regions may not improve and our method struggles with overlapping outliers from both sensors. For qualitative examples, see the supplementary material. Ideally, outliers can be filtered and the data fused with a learned scene representation as in [55], but our efforts to make [55] work with multiple sensors suggests that this is a harder learning problem which deserves attention in future work.

5 Conclusion

We propose a machine learning approach for online multi-sensor 3D reconstruction using depth maps. We show that a fusion decision on 3D features rather than directly on 2D depth maps generally improves surface accuracy and outlier robustness. This also holds when 2D fusion is straightforward, *i.e.* for time synchronized sensors, equal sensor resolution and calibration. The experiments demonstrate that our model handles various sensors, can scale to room-sized real-world scenes and produce a fused result that is quantitatively and qualitatively better than the single sensor inputs and the compared baselines methods.

Acknowledgements. This work was supported by the Google Focused Research Award 2019-HE-318, 2019-HE-323, 2020-FS-351, 2020-HS-411, as well as by research grants from FIFA and Toshiba. We further thank Hugo Sellerberg for helping with video editing.

References

1. Agresti, G., Minto, L., Marin, G., Zanuttigh, P.: Deep learning for confidence information in stereo and ToF data fusion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 697–705 (2017)
2. Agresti, G., Minto, L., Marin, G., Zanuttigh, P.: Stereo and ToF data fusion by learning from synthetic data. Inf. Fusion **49**, 161–173 (2019)
3. Ali, M.K., Rajput, A., Shahzad, M., Khan, F., Akhtar, F., Börner, A.: Multi-sensor depth fusion framework for real-time 3d reconstruction. IEEE Access **7**, 136471–136480 (2019)
4. Božič, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: monocular RGB scene reconstruction using transformers. arXiv preprint arXiv:2107.02191 (2021)
5. Bylow, E., Olsson, C., Kahl, F.: Robust online 3d reconstruction combining a depth sensor and sparse feature points. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3709–3714 (2016)
6. Bylow, E., Maier, R., Kahl, F., Olsson, C.: Combining depth fusion and photometric stereo for fine-detailed 3d models. In: Felsberg, M., Forssén, P.-E., Sintorn, I.-M., Unger, J. (eds.) SCIA 2019. LNCS, vol. 11482, pp. 261–274. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20205-7_22
7. Cao, Y.P., Kobbelt, L., Hu, S.M.: Real-time high-accuracy three-dimensional reconstruction with consumer RGB-D cameras. ACM Trans. Graph. (TOG) **37**(5), 1–16 (2018)

8. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5410–5418 (2018)
9. Choe, J., Im, S., Rameau, F., Kang, M., Kweon, I.S.: VolumeFusion: deep depth fusion for 3d scene reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 16086–16095, October 2021
10. Choi, O., Lee, S.: Fusion of time-of-flight and stereo for disambiguation of depth measurements. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012. LNCS, vol. 7727, pp. 640–653. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37447-0_49
11. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, pp. 303–312 (1996)
12. Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., Theobalt, C.: BundleFusion: real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. ACM Trans. Graph. (ToG) **36**(4), 1 (2017)
13. Dal Mutto, C., Zanuttigh, P., Cortelazzo, G.M.: Probabilistic TOF and stereo data fusion based on mixed pixels measurement models. IEEE Trans. Pattern Anal. Mach. Intell. **37**(11), 2260–2272 (2015)
14. Deng, Y., Xiao, J., Zhou, S.Z.: TOF and stereo data fusion using dynamic search range stereo matching. IEEE Trans. Multimedia **24**, 2739–2751 (2021)
15. Dong, W., Wang, Q., Wang, X., Zha, H.: 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), pp. 701–717 (2018)
16. Duan, Y., Pei, M., Wang, Y.: Probabilistic depth map fusion of kinect and stereo in real-time. In: 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2317–2322. IEEE (2012)
17. Duan, Y., Pei, M., Wang, Y., Yang, M., Qin, I., Jia, Y.: A unified probabilistic framework for real-time depth map fusion. J. Inf. Sci. Eng. **31**(4), 1309–1327 (2015)
18. Evangelidis, G.D., Hansard, M., Horaud, R.: Fusion of range and stereo data for high-resolution scene-modeling. IEEE Trans. Pattern Anal. Mach. Intell. **37**(11), 2178–2192 (2015)
19. Golodetz, S., Cavallari, T., Lord, N.A., Prisacariu, V.A., Murray, D.W., Torr, P.H.: Collaborative large-scale dense 3d reconstruction with online inter-agent pose optimisation. IEEE Trans. Visual. Comput. Graph. **24**(11), 2895–2905 (2018)
20. Gu, P., et al.: A 3d reconstruction method using multisensor fusion in large-scale indoor scenes. Complexity **2020** (2020)
21. Handa, A., Whelan, T., McDonald, J., Davison, A.J.: A benchmark for RGB-D visual odometry, 3d reconstruction and slam. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 1524–1531. IEEE (2014)
22. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. **30**(2), 328–341 (2007)
23. Huang, J., Huang, S.S., Song, H., Hu, S.M.: Di-fusion: online implicit 3d reconstruction with deep priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8932–8941 (2021)
24. Izadi, S., et al.: KinectFusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM Symposium on User Interface Software and Technology, pp. 559–568. ACM (2011)
25. Kähler, O., Prisacariu, V.A., Ren, C.Y., Sun, X., Torr, P.H.S., Murray, D.W.: Very high frame rate volumetric integration of depth images on mobile devices. IEEE Trans. Vis. Comput. Graph. **21**(11), 1241–1250 (2015)

26. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. (ToG) **32**(3), 1–13 (2013)
27. Kim, Y.M., Theobalt, C., Diebel, J., Kosecka, J., Miscusik, B., Thrun, S.: Multi-view image and TOF sensor fusion for dense 3d reconstruction. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV workshops, pp. 1542–1549. IEEE (2009)
28. Lefloch, D., Weyrich, T., Kolb, A.: Anisotropic point-based fusion. In: 2015 18th International Conference on Information Fusion (Fusion), pp. 2121–2128. IEEE (2015)
29. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3d surface construction algorithm. ACM siggraph Comput. Graph. **21**(4), 163–169 (1987)
30. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. J. Mach. Learn. Res. **9**(11), 2579–2605 (2008)
31. Maddern, W., Newman, P.: Real-time probabilistic fusion of sparse 3d lidar and dense stereo. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2181–2188. IEEE (2016)
32. Savva, M., et al.: Habitat: a platform for embodied AI Research. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
33. Marin, G., Zanuttigh, P., Mattoccia, S.: Reliable fusion of ToF and stereo depth driven by confidence measures. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 386–401. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_24
34. Martins, D., Van Hecke, K., De Croon, G.: Fusion of stereo and still monocular depth estimates in a self-supervised learning context. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 849–856. IEEE (2018)
35. Murez, Z., van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: end-to-end 3D scene reconstruction from posed images. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12352, pp. 414–431. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58571-6_25
36. Newcombe, R.A., et al.: KinectFusion: real-time dense surface mapping and tracking. In: ISMAR, vol. 11, pp. 127–136 (2011)
37. Newcombe, R.A., Lovegrove, S.J., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: ICCV (2011)
38. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Trans. Graph. (TOG) **32** (2013). https://doi.org/10.1145/2508363.2508374
39. Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., Nieto, J.I.: 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, 24–28 September 2017, pp. 1366–1373. IEEE (2017). https://doi.org/10.1109/IROS.2017.8202315
40. Park, K., Kim, S., Sohn, K.: High-precision depth estimation with the 3d lidar and stereo fusion. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2156–2163. IEEE (2018)
41. Patil, V., Van Gansbeke, W., Dai, D., Van Gool, L.: Don't forget the past: Recurrent depth estimation from monocular video. IEEE Robot. Autom. Lett. **5**(4), 6813–6820 (2020)
42. Poggi, M., Mattoccia, S.: Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In: 2016 Fourth International Conference on 3D Vision (3DV), pp. 138–147. IEEE (2016)

43. Pu, C., Song, R., Tylecek, R., Li, N., Fisher, R.B.: SDF-MAN: semi-supervised disparity fusion with multi-scale adversarial networks. Remote Sens. **11**(5), 487 (2019)
44. Qiu, J., et al.: 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, 16–20 June 2019, pp. 3313–3322. Computer Vision Foundation/IEEE (2019). https://doi.org/10.1109/CVPR.2019.00343, https://openaccess.thecvf.com/content_CVPR_2019/html/Qiu_DeepLiDAR_Deep_Surface_Normal_Guided_Depth_Prediction_for_Outdoor_Scene_CVPR_2019_paper.html
45. Rozumnyi, D., Cherabier, I., Pollefeys, M., Oswald, M.R.: Learned semantic multi-sensor depth map fusion. In: International Conference on Computer Vision Workshop (ICCVW), Workshop on 3D Reconstruction in the Wild, 2019. Seoul, South Korea (2019)
46. Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 501–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31
47. Schops, T., Sattler, T., Pollefeys, M.: BAD SLAM: bundle adjusted direct RGB-D SLAM. In: CVPR (2019)
48. Steinbrucker, F., Kerl, C., Cremers, D., Sturm, J.: Large-scale multi-resolution surface reconstruction from RGB-D sequences. In: 2013 IEEE International Conference on Computer Vision, pp. 3264–3271 (2013)
49. Straub, J., et al.: The replica dataset: a digital replica of indoor spaces. arXiv preprint arXiv:1906.05797 (2019)
50. Sucar, E., Liu, S., Ortiz, J., Davison, A.: iMAP: implicit mapping and positioning in real-time. In: Proceedings of the IEEE International Conference on Computer Vision (2021)
51. Sun, J., Xie, Y., Chen, L., Zhou, X., Bao, H.: NeuralRecon: real-time coherent 3d reconstruction from monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 15598–15607 (2021)
52. Van Baar, J., Beardsley, P., Pollefeys, M., Gross, M.: Sensor fusion for depth estimation, including TOF and thermal sensors. In: 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission, pp. 472–478. IEEE (2012)
53. Wasenmüller, O., Meyer, M., Stricker, D.: Corbs: comprehensive RGB-D benchmark for slam using kinect v2. In: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 1–7. IEEE (2016)
54. Weder, S., Schönberger, J.L., Pollefeys, M., Oswald, M.R.: RoutedFusion: learning real-time depth map fusion. ArXiv abs/2001.04388 (2020)
55. Weder, S., Schonberger, J.L., Pollefeys, M., Oswald, M.R.: NeuralFusion: online depth fusion in latent space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3162–3172 (2021)
56. Yan, Z., Tian, Y., Shi, X., Guo, P., Wang, P., Zha, H.: Continual neural mapping: learning an implicit scene representation from sequential observations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 15782–15792, October 2021
57. Yang, S., et al.: Noise-resilient reconstruction of panoramas and 3d scenes using robot-mounted unsynchronized commodity RGB-D cameras. ACM Trans. Graph. (TOG) **39**(5), 1–15 (2020)

58. Yang, S., Li, B., Liu, M., Lai, Y.K., Kobbelt, L., Hu, S.M.: HeteroFusion: dense scene reconstruction integrating multi-sensors. IEEE Trans. Visual. Comput. Graph. **26**(11), 3217–3230 (2019)
59. Zhou, Q.Y., Koltun, V.: Dense scene reconstruction with points of interest. ACM Trans. Graph. (ToG) **32**(4), 1–8 (2013)
60. Zhu, Z., et al.: Nice-slam: neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12786–12796 (2022)
61. Zollhöfer, M., et al.: State of the art on 3d reconstruction with RGB-D cameras. In: Computer Graphics Forum, vol. 37, pp. 625–652. Wiley Online Library (2018)