RoutedFusion: Learning Real-time Depth Map Fusion

Silvan Weder
ETH Zurich

Johannes Schönberger
Microsoft

Marc Pollefeys
ETH Zurich/Microsoft

Martin R. Oswald
ETH Zurich

Abstract

The efficient fusion of depth maps is a key part of most state-of-the-art 3D reconstruction methods. Besides requiring high accuracy, these depth fusion methods need to be scalable and real-time capable. To this end, we present a novel real-time capable machine learning-based method for depth map fusion. Similar to the seminal depth map fusion approach by Curless and Levoy, we only update a local group of voxels to ensure real-time capability. Instead of a simple linear fusion of depth information, we propose a neural network that predicts non-linear updates to better account for typical fusion errors. Our network is composed of a 2D depth routing network and a 3D depth fusion network which efficiently handle sensor-specific noise and outliers. This is especially useful for surface edges and thin objects for which the original approach suffers from thickening artifacts. Our method outperforms the traditional fusion approach and related learned approaches on both synthetic and real data. We demonstrate the performance of our method in reconstructing fine geometric details from noise and outlier contaminated data on various scenes.

1. Introduction

Multi-view 3D reconstruction has been a central research topic in computer vision for many decades. Fusing depth maps from multiple camera viewpoints is an essential processing step in the majority of recent 3D reconstruction pipelines [59, 58, 27, 1, 44, 43, 12, 10], especially for real-time applications [21, 37, 55, 11]. We revisit the problem of 3D reconstruction via depth map fusion from a machine learning perspective. The major difficulty of this task is to deal with various amounts of noise, outliers, and missing data. The classical approach [9, 21] to fusing noisy depth maps is to average truncated signed distance functions (TSDF). This approach has many advantages: **1+)** The updates are local (truncated) and can be done in constant time for a fixed number of depth values. The high memory usage of voxel grids can be easily reduced with voxel hashing [37] or octrees [49]. **2+)** Online updates are simple to implement and noisy measurements are fused into a single surface with very few operations. **3+)** Due to local independent updates, the approach is computationally cheap and highly parallelizable.

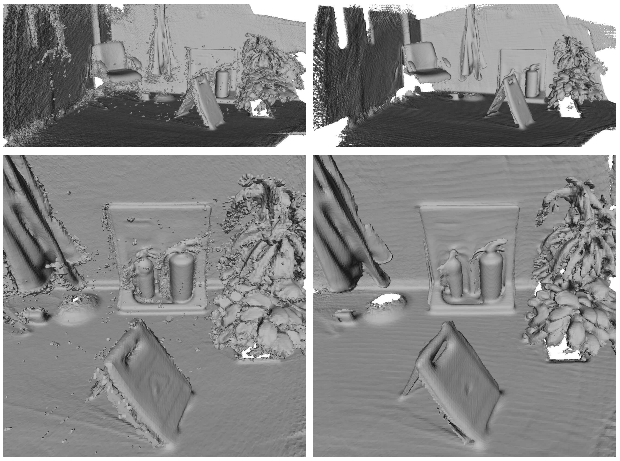


Figure 1: **Standard TSDF fusion vs. our learned depth map fusion approach** (on Kinect data [48]). Due to a more informed decision process, our approach better handles noise and fine geometric details.

However, the approach also has a number of shortcomings: **1-)** The average is only the optimal estimate for zero-mean Gaussian noise, but the real error distribution is typically non-Gaussian, non-centered and depth-dependent. **2-)** The updates are linear and a minimal thickness assumption of surfaces has to be made according to the expected noise level. Therefore, thickening artifacts become apparent along surface edges and for thin object structures. **3-)** This issue is even more severe when depth measurements of a thin object are made from opposite directions. Then the surface vanishes since the linear TSDF updates cancel each other out. **4-)** The linear fusion weights do not properly account for view directional dependencies during TSDF aggregation. The noise level along the viewing direction is typically very different from the one in orthogonal directions. **5-)** The fusion approach is unable to handle gross outliers. The depth map has to be pre-filtered or incorrect measurements will clutter the scene. **6-)** The fusion parameters must be tuned for specific scenes and sensors and it is often difficult to find a good trade-off between runtime and different aspects of reconstruction quality.

In this paper, we aim to tackle the above mentioned disadvantages while maintaining all the advantages of traditional approach with a reasonable amount of additional computation time to still meet real-time requirements. To this end, we propose a learned approach that fuses noisy and outlier contaminated measurements into a single surface, performs non-linear updates to better deal with object boundaries and thin structures, and is fast enough for real-time applications. Figure 1 shows example outputs of our approach. In sum, this paper's **contributions** are as follows:

- We present a learning-based method for real-time depth map fusion. Due to its compact architecture it requires only little training data, and is not prone to over-fitting.
- We propose a scalable and real-time capable neural architecture that is independent of the scene size. Therefore, it is applicable to a large set of real-world scenarios.
- We show significant improvement of standard TSDF fusion's shortcomings: 1) It better handles the fusion of anisotropic noise distributions that naturally arise from the multi-view setting, and 2) It mitigates the surface thickening effect on thin objects and surface boundaries by avoiding inconsistent updates.

2. Related Work

Volumetric Depth Map Fusion. With their seminal work, Curless and Levoy [9] proposed an elegant way for fusing noisy depth maps which later got adopted by numerous works like KinectFusion [21], more scalable generalizations like voxel hashing [37, 33], or hierarchical scene representations, such as voxel octrees [16, 49, 34] and hierarchical hashing [24]. Especially for SLAM pipelines like InfiniTAM [23], volumetric fusion became a standard approach due to its real-time capability. In this context, it was also extended to become more accurate and robust [8] as well as improve SLAM with additional surface registration of scene parts to account for pose drift as proposed in [55, 32, 11]. Approaches with additional median filtering [42, 34, 33] improve the robustness and are still real-time capable but with limited effectiveness. Global optimization approaches [59, 27] even better deal with noise and outliers if they further leverage semantic information [19, 6, 20, 44, 43], but they are computationally expensive and not real-time capable. In [65, 31], the authors propose methods for refinement of already fused SDF geometry based on shape-from-shading. The vast majority of these approaches directly fuse RGB-D images for which Zollhöfer *et al*. [66] provide a recent survey. All these methods handle noisy measurements by updating a wider band of voxels around the measured depth leading to thickening artifacts on thin geometry.

Surfel-based Fusion Methods. Surfel-based methods approximate the surface with local point samples, which can further encode additional local properties such as normal or texture information. Multiple methods have been proposed, *e.g*. MRSMap [50] uses an octree to store multi-resolution surfel data. The point-based fusion methods [25, 29] combine a surfel representation with probabilistic fusion discussed in the next paragraph. ElasticFusion [55] handles real-time loop closures and corrects all surface estimates online. Schöps *et al*. [47] proposed a depth fusion approach with real-time mesh construction. A disadvantage of surfel-based methods is the missing connectivity information among surfels. The unstructured neighborhood relationships can only be established with a nearest neighbor search or simplified with space partitioning data structures. In our work, we decided to rely on volumetric representation, but extending our approach to unstructured settings is an interesting avenue of future work.

Probabilistic Depth Map Fusion. To account for varying noise levels in the input depth maps and along different line-of-sight directions, the fusion problem can also be cast as probability density estimation [15] while typically assuming a Gaussian noise model. Keller *et al*. [25] propose a point-based fusion approach which directly updates a point cloud rather than a voxel grid. Lefloch *et al*. [29] extended this idea to anisotropic point-based fusion in order to account for different noise levels when a surface is observed from different incident angles. The mesh-based fusion approach by Zienkiewicz *et al*. [64] allows for depth fusion across various mesh resolutions for known fixed topology. The probabilistic fusion method by Woodford and Vogiatzis [56] incorporates long range visibility constraints. Similar ray-based visibility constraints were also used in [52, 51], but these methods are not real-time capable due to the complex optimization of ray potentials. Anisotropic depth map fusion methods additionally keep track of fusion covariances [57]. Similarly, PSDF Fusion [13] explicitly models directional dependent sensor noise. In contrast to our method, all these approaches assume particular noise distributions, primarily Gaussians, which often do not model the real sensor observations correctly.

Learning-based Reconstruction Approaches. Several learning-based methods have been proposed to fuse, estimate, or improve geometry. SurfaceNet [22] jointly estimates multi-view stereo depth maps and their volumetric fusion, but is extremely memory demanding as each camera view requires a full voxel grid. In [30], multi-view consistency is learned upon classical TSDF fusion. RayNet [39] models view dependencies along ray potentials with a Markov random field which is jointly learned with a view-invariant feature representation. 3DMV [10] combines 2D view information with a pre-fused TSDF scene to jointly optimize for shape and semantics. Riegler *et*

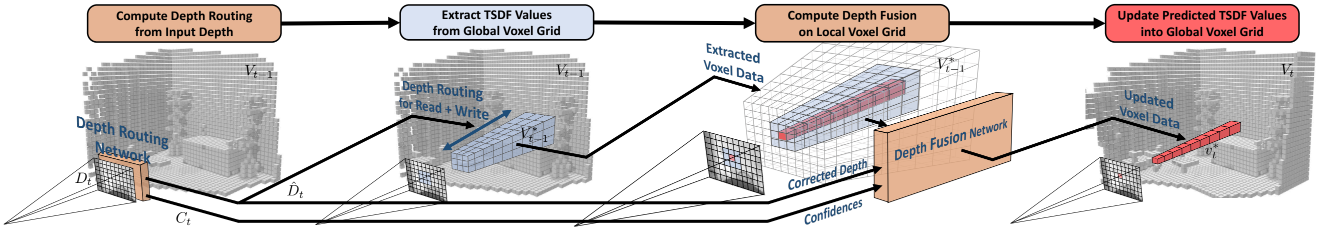

Compute Depth Routing from Input Depth
Extract TSDF Values from Global Voxel Grid
Compute Depth Fusion on Local Voxel Grid
Update Predicted TSDF Values into Global Voxel Grid
V_{t-1}
Depth Routing Network
D_t
C_t
$\hat{D}_t$
V_{t-1}
Depth Routing for Read + Write
V^*_{t-1}
Extracted Voxel Data
V^*_{t-1}
Depth Fusion Network
Corrected Depth
Confidences
V_t
Updated Voxel Data
v^*_t

Figure 2: **System overview for integrating depth maps into a global TSDF volume.** A 2D ***Depth Routing Network*** takes depth input and decides the update location for every ray within the TSDF volume. The network corrects for noise, outliers and missing values and further estimates per-ray confidence values. Then, for each ray, we extract a depth and view-dependent local voxel grid (light blue) which also includes neighboring rays. We sample S values along each ray, centered around the surface. A ***Depth Fusion Network*** then takes the local grid of existing TSDF values, the depth and confidences to predict adequate updates. The predicted TSDF values (red) are then written back into the global volume. Our method learns a robust weighting of input depths and performs non-linear updates to better handle noise, outliers, and thin objects.

al. [40] fuse depth maps using standard TSDF and subsequently post-process the fused model with a neural network. Moreover, hierarchical volumetric deep learning-based approaches [3, 7, 12] tackle the effects of noisy measurements, outliers, and missing data. All these approaches operate on a voxel grid with high memory demands and are not real-time capable. Further, there are several works that learn to predict 3D meshes based on input images [18, 17, 54].

Learned Scene Representations. Ladicky *et al.* [28] directly estimate an iso-surface from a point cloud via learned local point features with a random forest. In addition, there exist multiple proposals for methods that learn 3D reconstruction in an implicit space [35, 38, 5, 36]. These methods show promising results, but they operate only on a unit cube and are thus limited to single objects or small scenes and they are neither suited for online reconstruction. Contrary to all these methods, our method is independent of the scene's size and can thus also operate on large-scale scenes. Additionally, our method uses learning in an online process, which allows to leverage already fused information for fusing a new depth map. In [2, 60], the authors propose neural models that learn a compact and optimizable 2.5D depth representation for SLAM applications. DeepTAM [62] also addresses SLAM, but the mapping part heavily relies on hand-crafted photoconsistencies and corresponding weights to form a traditional cost volume for depth estimation. None of these methods address global model fusion.

3. Method

We first review the standard TSDF fusion approach to provide context and to introduce notation before we present our learned TSDF fusion method.

3.1. Review of Standard TSDF Fusion

Standard TSDF fusion integrates given depth maps $\boldsymbol{D}_{t=1,\ldots,T} \in \mathbb{R}^{W\times H}$ from known viewpoints $\boldsymbol{P}_t \in \mathrm{SE}(3)$ with camera intrinsics $\boldsymbol{K}_t$ into a discretized signed distance function $\boldsymbol{V}_t \in \mathbb{R}^{X\times Y\times Z}$ and weight function $\boldsymbol{W}_t \in \mathbb{R}^{X\times Y\times Z}$ defined over the entire scene. The fusion process is incremental, *i.e.* each depth map is integrated after one another for location $\boldsymbol{x}$ using the update equations introduced by Curless and Levoy [9] as

$$\boldsymbol{V}_t(\boldsymbol{x}) = \frac{\boldsymbol{W}_{t-1}(\boldsymbol{x}) \cdot \boldsymbol{V}_{t-1}(\boldsymbol{x}) + w_t(\boldsymbol{x}) \cdot v_t(\boldsymbol{x})}{\boldsymbol{W}_{t-1}(\boldsymbol{x}) + w_t(\boldsymbol{x})} \tag{1}$$

$$\boldsymbol{W}_t(\boldsymbol{x}) = \boldsymbol{W}_{t-1}(\boldsymbol{x}) + w_t(\boldsymbol{x}) \ \ , \tag{2}$$

starting from zero-initialized volumes $\boldsymbol{V}_0$ and $\boldsymbol{W}_0$. The signed distance update v_t and its corresponding weight w_t integrate the depth measurements of the next depth map $\boldsymbol{D}_t$ at time step t into the TSDF volume. These update functions are traditionally truncated before and after the surface in order to ensure efficient runtimes and robust reconstruction of fine-structured surfaces given noisy depth measurements.

The choice of the truncation distance parameter typically requires cumbersome hand-tuning to adapt to a specific scene and depth sensor as well as accounting for runtime. If the truncation distance is chosen too large, the reconstruction of thin structures becomes more difficult due to larger thickening artifacts and the fusion process gets slower since more voxels have to be updated for each ray. Contrary, a small truncation distance results in time efficient updates but cannot deal with larger noise in the depth measurements.

In this paper, we overcome this limitation by learning the function v_t automatically from data. Our system is based on the same above mentioned update equations and our learned functions have only little computational overhead compared to traditional TSDF fusion. As such, our method facilitates real-time depth map fusion and can be readily integrated into existing reconstruction systems. In the following, we describe our proposed method in more detail.

3.2. System Overview

Our method contains *two* network components: a ***depth routing network*** and a ***depth fusion network***. The pipeline consists of the following *four* essential processing steps which are also illustrated in Figure 2:

1. **Depth Routing:** The depth routing network takes a raw depth map $\boldsymbol{D}_t$ and estimates a denoised and outlier-corrected depth map $\hat{\boldsymbol{D}}_t$, and further estimates a corresponding confidence map $\boldsymbol{C}_t$. This network *routes* the depth location for reading and writing TSDF values along each viewing ray.
2. **TSDF Extraction:** Given the routed depth values $\hat{\boldsymbol{D}}_t$, we extract a local camera-aligned voxel grid with TSDF data $\boldsymbol{V}^*_{t-1}$ and weight $\boldsymbol{W}^*_{t-1}$ via trilinear interpolation from the corresponding global voxel grids $\boldsymbol{V}_{t-1}, \boldsymbol{W}_{t-1}$.
3. **Depth Fusion:** The depth fusion network takes the results of the previous processing steps $(\hat{\boldsymbol{D}}_t, \boldsymbol{C}_t, \boldsymbol{W}^*_{t-1}, \boldsymbol{V}^*_{t-1})$ and computes the local TSDF update v^*_t.
4. **TSDF Update Integration:** The predicted TSDF update v^*_t is transferred back into the global coordinate frame to get v_t which is then integrated into the global TSDF volumes $\boldsymbol{V}_t, \boldsymbol{W}_t$ using the TSDF updates in Eqs. (1), (2).

These processing steps are detailed in the next subsections.

3.3. Depth Routing

Using the depth routing network, we pre-process the depth maps before passing them to the depth fusion network with the main motivation of denoising and outlier correction. Towards this end, the network predicts denoised depth maps and also per-pixel confidence maps $\boldsymbol{C}_{t=1,\ldots,T} \in \mathbb{R}^{W \times H}$. Figure 3 illustrates our network architecture, which is using a fully-convolutional U-Net [41] with a joint encoder and separate decoders for confidence and depth prediction. Further, we do not use normalization layers since it negatively influences the depth prediction performance by adding a depth-dependent bias to the result. The depth map and the confidence map are processed by two separate decoders to which the output of the bottleneck layers serves as an input.

3.4. TSDF Extraction

Instead of processing each ray of a view t independently as in standard TSDF fusion, we deliberately choose to compute the TSDF updates based on the data of a larger neighborhood in order to make a more informed decision about the surface location. Further, the 2D input data also holds valuable information about surface locations as often indicated by depth discontinuities. We argue that the fusion network can best benefit from both 2D and 3D data sources when they are already in correspondence and therefore propose a view-aligned local neighborhood extraction. Then, the 3D TSDF data and the 2D input data can be easily concatenated and fed into the network. Hence, for efficient real-time updates of the global data $\boldsymbol{V}_{t-1}$, $\boldsymbol{W}_{t-1}$, we extract a local, view-dependent TSDF volume and corresponding weights $\boldsymbol{V}^*_{t-1}, \boldsymbol{W}^*_{t-1} \in \mathbb{R}^{W \times H \times S}$. The first two volume dimensions W, H correspond to the width and height of the depth map whereas the third dimension S represents the local depth-sampling dimension of the window sampled along the ray. This number S closely relates to the truncation distance in standard TSDF fusion. For each ray independently, the local windows are centered at their respective depth values $\hat{\boldsymbol{D}}_t$ and discretely sampled into a fixed number of S values from the volume $\boldsymbol{V}_{t-1}$. We choose the step size of the sampling according to the resolution of the scene and use trilinear interpolation to mitigate sampling artifacts.

The input $\boldsymbol{I}_t$ to the subsequent depth fusion is then a combination of all available local information, that is, corrected depth map $\hat{\boldsymbol{D}}_t$, confidence map $\boldsymbol{C}_t$ as well as the extracted TSDF values $\boldsymbol{V}^*_{t-1}$ and TSDF weights $\boldsymbol{W}^*_{t-1}$

$$\boldsymbol{I}_t = \begin{bmatrix} \hat{\boldsymbol{D}}_t & \boldsymbol{C}_t & \boldsymbol{W}^*_{t-1} & \boldsymbol{V}^*_{t-1} \end{bmatrix} \in \mathbb{R}^{W \times H \times (2S+2)} \ . \tag{3}$$

Before the subsequent update prediction step, we explicitly filter gross outliers where $\boldsymbol{C}_t < C_{\text{thr}}$ and set their corresponding feature values in $\boldsymbol{I}_t$ to zero.

3.5. Depth Fusion

Our depth fusion network takes the local 3D feature volume $\boldsymbol{I}_t$ as input and predicts the local TSDF update $v^*_t \in \mathbb{R}^{W \times H \times S}$. The architecture is fully convolutional in two dimensions and the channel dimension is along the camera viewing direction. Our network is relatively compact and thereby facilitates real-time computation.

Our depth fusion network operates in a two-stage approach, as shown in Figure 3. The first stage encodes local and global information in the viewing frustum. We sequentially pass the input 3D feature volume through encoding blocks of two consecutive convolutional layers with interleaved batch normalization, non-linear activation using leaky ReLUs, and a dropout layer. The output of every block is concatenated with its input and passed through the next block. With every block, the receptive field of the neural network increases. This sequential feature extraction results in a 100-dimensional feature vector for each ray in the viewing frustum.

The second network part takes the feature volume and predicts the TSDF updates along each ray. The number of features is sequentially reduced by passing them through convolutional blocks with two 1×1 convolutional layers interleaved with leaky ReLUs, batch normalization, and dropout layers. In the last block, we directly reduce from 40 features to 20 in the first layer and then to S TSDF values in the last convolutional layer, where we apply a tanh-activation on the output mapping it to the range $[-1, 1]$.

Note that predicted TSDF update values $\boldsymbol{v}^*_t$ can take any value. The network can decide to not update the TSDF at all, *e.g.*, in case of an outlier. Conversely, it can reduce the influence of existing TSDF values if they contained outliers.

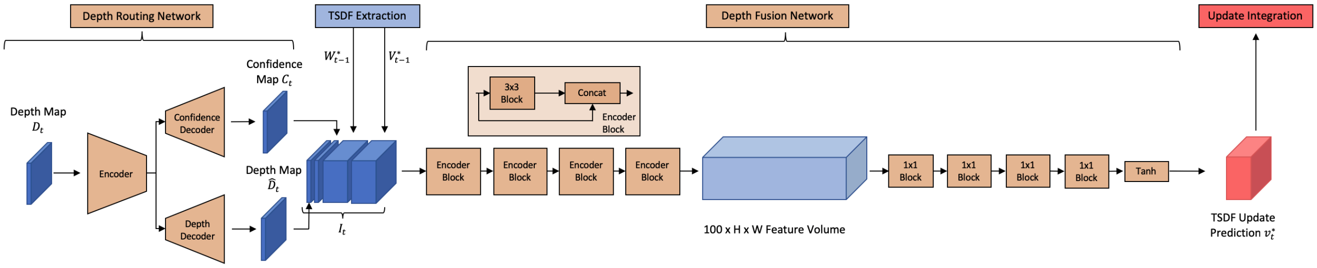


Figure 3: **Proposed network architecture.** Our depth routing network consists of a U-Net (depth one) with two separate decoders predicting a corrected depth map and a corresponding confidence map. The depth fusion network extracts in a series of encoding blocks 100 features along each ray. These features are then used to predict the TSDF updates along the ray.

3.6. TSDF Update Integration

In order to compute the updated global TSDF volume $\boldsymbol{V}_t$ we transform the predicted local TSDF updates v_t^* back into the global coordinate frame v_t. To this end, we apply the inverse operation of the previous extraction step, that is, we redistribute the values using the same trilinear interpolation weights. In fact, we actually repurpose the update weights w_t for this task, where $\boldsymbol{W}_t$ accumulates the splatting weights for each voxel in the scene. Moreover, we also use $\boldsymbol{W}_t$ for post-filtering extreme outliers[1].

3.7. Loss Function and Training Procedure

The two networks in our pipeline are trained in two steps. First, we train the depth routing network and then use the pre-trained routing output to train the fusion network.

Depth Routing Network. We train the depth prediction head in a supervised manner by computing the L1 loss on absolute depth values as well as on the depth map gradient, as proposed in [14]. For training the confidence head, we chose a self-supervised approach [26]. Therefore, the final loss function has the form

$$\mathcal{L}_{\text{2D}} = \sum_i c_i \mathcal{L}_1(y_i, \hat{y}_i) + c_i \mathcal{L}_1(\nabla y_i, \nabla \hat{y}_i) - \lambda \log c_i \quad (4)$$

where $y_i, \hat{y}_i$ are the predicted and ground-truth depth values at pixel i respectively and $c_i \in \boldsymbol{C}_t$ is the confidence value. The hyperparameter λ is empirically set to 0.015.

Depth Fusion Network. Despite the pre-processing of the routing network, the filtered depth map might still contain noise and outliers which should be further handled by the depth fusion network. Each global TSDF update step should a) integrate new information about the true geometry and b) not destroy valuable, previously fused surface information. We train the fusion network in a supervised manner by choosing random update steps at time t during the fusion and penalize differences between the updated local volume $\boldsymbol{V}_t^* = \frac{\boldsymbol{W}_{t-1}^* \cdot \boldsymbol{V}_{t-1}^* + w_t^* \cdot v_t^*}{\boldsymbol{W}_{t-1}^* + w_t^*} \in \mathbb{R}^{W \times H \times S}$ and the local ground-truth $\hat{\boldsymbol{V}}^* \in \mathbb{R}^{W \times H \times S}$. Therefore, we define the loss function over all rays i as

$$\mathcal{L}_{\text{3D}} = \sum_i \lambda_1 \mathcal{L}_1(\boldsymbol{V}_{ti}^*, \hat{\boldsymbol{V}}_i^*) + \lambda_C D_C(\boldsymbol{V}_{ti}^*, \hat{\boldsymbol{V}}_i^*) \quad (5)$$

Here, $\mathcal{L}_1$ denotes the L1 loss over raw TSDF values and D_C denotes the cosine distance between the signs of the TSDF values computed along each ray i. The goal of the first term is to preserve fine surface detail (through means of $\mathcal{L}_1$), while, the term D_C ensures that the surface is located at the zero-crossing of the signed distance field. The weights $\lambda_1 = 1$ and $\lambda_C = 0.1$ have been empirically found.

4. Experiments

In this section, we first present additional implementation details and our experimental setup. Next, we evaluate and discuss the efficacy of our approach on both synthetic and real-world data. We demonstrate that our approach outperforms traditional TSDF fusion and state-of-the-art learning-based approaches in terms of reconstruction accuracy with only little computational overhead.

4.1. Implementation Details

All networks were implemented in PyTorch and trained on an NVIDIA TITAN Xp GPU. We trained both networks using the RMSProp optimization algorithm with momentum 0.9 and initial learning rate 1e−5 for the depth routing network and 1e−3 for the depth fusion network. The dropout layers were set to a probability of 0.2. For all experiments, we trained our neural networks in a sequential process, where we first pre-trained the depth routing and then the depth fusion network. A joint end-to-end refinement did not lead to an improvement of the overall performance of the system. To train the depth routing network, we use 10K frames sampled from 100 ModelNet [61] or ShapeNet [4] objects and perturb them with artificial speckle noise. The data is packed into batches of size 4 and the gradient is accumulated across 8 batches before updating the routing network weights. Because of the incremental nature of the TSDF update equation, we must train our depth fusion network using a batch size of 1. However, each batch updates a very large number of voxels in the volume over which the

[1] See supplementary material for further information.

loss is defined and, together with batch normalization, we obtain robust convergence during training. Since our network has only very few parameters, it is hard to overfit and only little training data is required. In fact, we can train our entire network (given a pre-trained depth routing network) on only ten models from ModelNet [61] or ShapeNet [4] with a total of 1000 depth maps and it already generalizes robustly to other scenes. Furthermore, we can train the network from scratch in only 20 epochs (each epoch passes once over all 1000 frames). Unless otherwise specified, we used $S = 9$ and $C_{thr} = 0.9$ across all experiments[2]. For all experiments, we used a voxel size $0.008m$, corresponding to a grid resolution of 128^3 for ShapeNet and ModelNet.

Runtime. A forward pass through the depth routing network and the depth fusion network for one depth map ($W = 320, H = 240$) takes 0.9 ms and 1.8 ms, respectively while the full pipeline runs at 15 fps. These numbers can be improved with a more efficient implementation, but already meet real-time requirements.

4.2. Results

We evaluate our method on synthetic and real-world data comparing to traditional TSDF fusion [9] as a baseline as well as to the state-of-the-art PSDF fusion method presented by Dong *et al.* [13]. Moreover, we compare to state-of-the-art learning-based 3D reconstruction methods OccupancyNetworks [35] and DeepSDF [38].

Evaluation Metrics. For quantifying the performance of our method, we compute the following four metrics by comparing the estimated TSDF against the ground-truth.

- **MAD:** The mean absolute distance is computed over all TSDF voxels and measures the reconstruction performance on fine surface details.
- **MSE:** The mean squared error loss is computed over all TSDF voxels and measures the reconstruction performance on large surface deviations.
- **Accuracy:** We compare the actual reconstruction accuracy on the occupancy grid. We extract the occupancy grid in the ground-truth and the estimated TSDF by extracting all voxels with negative TSDF values.
- **Intersection over Union (IoU):** We compute the intersection-over-union on the occupancy grid, which is an alternative performance measure to the accuracy.

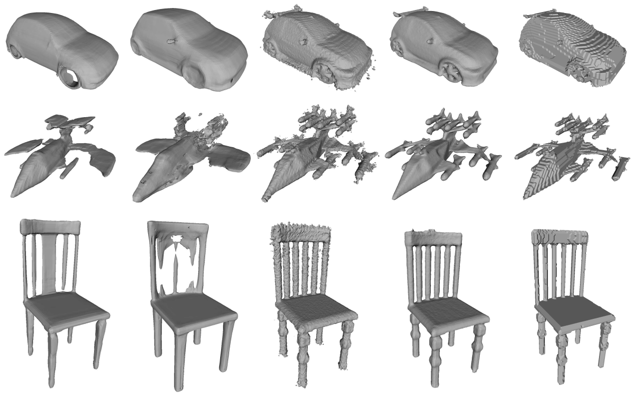
These metrics not only quantify how well our pipeline fuses depth maps into a TSDF, but also how well it performs in classifying the occupancy and reconstructing the geometry.

4.3. Synthetic Data

To evaluate our method's performance in fusing noisy synthetic data, we train and test it on the ModelNet [61] and ShapeNet [4] datasets using rendered ground-truth depth maps that are perturbed with an artificial depth-dependent multiplicative noise distribution. For both, ModelNet and ShapeNet, we randomly sample our training and test data from the official train-test split.

Method	MSE [e-05]	MAD	Acc. [%]	IoU [0, 1]
DeepSDF [38]	464.0	0.0499	66.48	0.538
OccupancyNetworks [35]	56.8	0.0166	85.66	0.484
TSDF Fusion [9]	11.0	0.0078	88.06	0.659
TSDF Fusion + Routing	27.0	0.0084	87.48	0.650
Ours w/o Routing	**5.9**	0.0051	93.91	0.765
Ours	**5.9**	**0.0050**	**94.77**	**0.785**

Table 1: **Quantitative Results on ShapeNet [4].** Our method outperforms TSDF fusion and other learning-based approaches in fusing noisy ($\sigma = 0.005$) depth-maps rendered from ShapeNet objects. The benefit of the routing network increases with higher noise levels (see Fig. 5).



DeepSDF [38] Occ.Net. [35] TSDF [9] Ours GT

Figure 4: **Qualitative Results on ShapeNet [4].** Our method is superior to all other methods in reconstructing fine details (see car wheels and spoiler) and produces smoother surfaces (input noise level $\sigma = 0.005$).

ShapeNet. The model trained on ShapeNet is then used to evaluate the performance of our method in comparison with other approaches. Therefore, we fuse noisy depth maps of 60 objects (10 per test class - plane, sofa, lamp, table, car, chair) from the test set, which have not been seen during training. For comparison, we use the provided pre-trained model for point cloud completion in the case of OccupancyNetworks. In the case of DeepSDF, we trained the model from scratch using the code provided by the authors and using ShapeNet as training data. The quantitative results of this evaluation are shown in Table 1. Our method consistently outperforms standard TSDF fusion as well as the pure learning-based approaches OccupancyNetworks [35] and DeepSDF [38] on all metrics. Our method significantly improves the accuracy of the fused implicit mesh as well as their IoU, MAD and MSE scores. The results also indicate the potential of our routing network. However, the full benefit of our routing network only becomes obvious when looking at the real-world data experiments and Figure 5.

[2] See supplementary material for further evaluation.

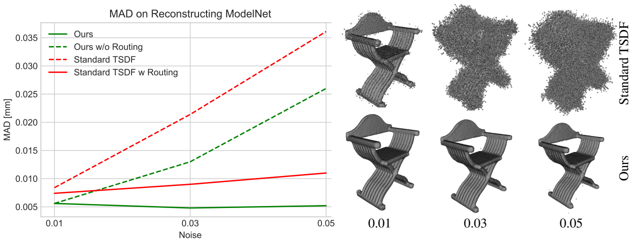


Figure 5: **Evaluation of different noise levels** σ. The left plot shows MAD for different noise levels $\sigma \in \{0.01, 0.03, 0.05\}$. Our routing network stabilizes both, our method as well as standard TSDF fusion, for high noise levels. On the right, we show corresponding qualitative results on ModelNet test data for Standard TSDF and our method. The figures show the denoising capability of our method, where standard TSDF fusion completely fails.

Figure 4 illustrates the strengths of our method in dealing with noise and in reconstructing thin structures. Flat surfaces in the ground-truth appear smoother in our results as compared to standard TSDF fusion. Furthermore, thin structures are better reconstructed and contain less thickening artifacts. The thickening artifacts are also visible on the car's rims, where our method yields accurate results and DeepSDF and OccupancyNetworks both fail. Both DeepSDF and OccupancyNetworks tend to over-smooth surface details less common in the training data, *e.g.*, the spoiler of the car or the details on the chair's legs.

ModelNet. In order to test our method's robustness against noise we trained and evaluated on various noise levels $\sigma \in \{0.01, 0.03, 0.05\}$ and compared it to standard TSDF fusion. We also analyze the effect of the depth routing network on the fusion result by omitting it in our pipeline and by testing it in combination with standard TSDF fusion. Figure 5 illustrates that our pipeline outperforms standard TSDF fusion for all tested noise levels. It also shows that our depth routing network stabilizes the fusion of data corrupted with extreme noise levels. When used for data preprocessing, our depth routing network also improves the results of standard TSDF fusion.

4.4. Real-World Data

We also evaluate on real-world datasets and compare to other state-of-the-art fusion methods. Due to lack of ground-truth data, we use the model trained on synthetic ModelNet data using an artificial and empirically chosen depth-dependent noise distribution with $\sigma = 0.01$. As such, we also show that our method must not necessarily be trained on real-world data but generalizes robustly to the real domain from being trained on noisy synthetic data only.

3D Scene Data [63]. To quantify the improvement of the reconstruction result, we evaluate our method compared to standard TSDF fusion on scenes provided by Zhou *et al*. [63]. Since there is no volumetric ground-truth available for these scenes, we fuse all frames of each scene using standard TSDF fusion and denoised the meshes. Then, we only fuse every 10th frame using standard TSDF fusion as well as our method for evaluation.

Method	Lounge	Copyroom	Stonewall	Cactusgarden	Burghers
TSDF	0.0095	0.0110	0.0117	0.0104	0.0126
Ours w/o routing	0.0055	0.0057	0.0047	0.0055	0.0071
Ours	**0.0051**	**0.0051**	**0.0043**	**0.0052**	**0.0067**

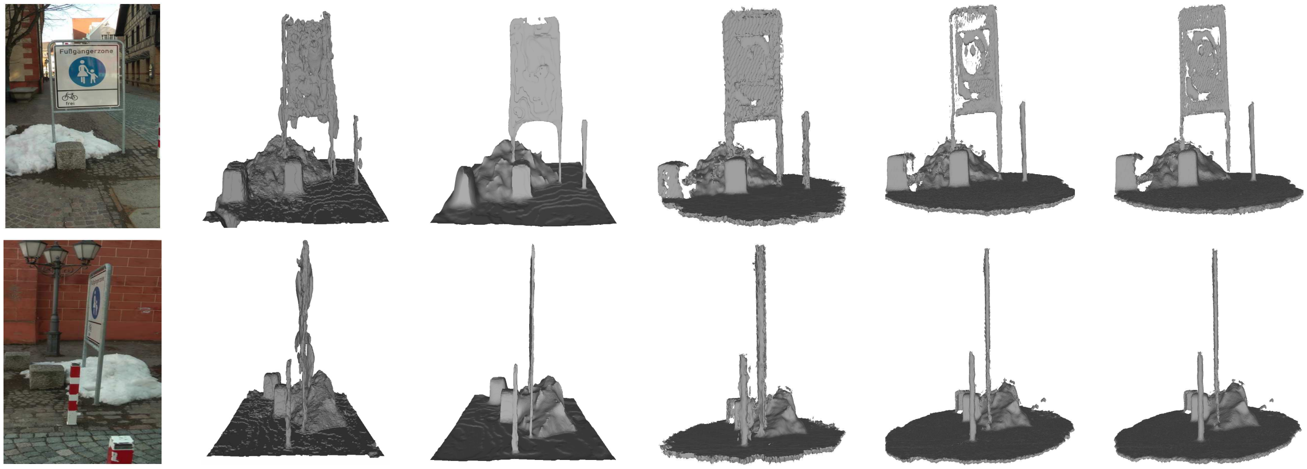
Table 2: **Quantitative evaluation (MAD [mm]) of our method on 3D Scene Data [63].** Our method is consistently better than standard TSDF fusion on 3D Scene Data. These experiment also shows the benefit of our routing network when applied to real-world data.

Table 2 shows the quantitative reconstruction results from fusing 5 scenes of the 3D scene dataset [63]. Our method significantly outperforms standard TSDF fusion on all scenes without being trained on real-world data.

We further show a qualitative comparison to standard TSDF as well as PSDF fusion [13] on the Burghers of Calais scene in Figure 8. The results illustrate that our method better reconstructs fine geometric details (hands, fingers and face) and produces smoother surfaces than standard TSDF fusion and PSDF fusion [13]. For more qualitative examples on this dataset, we refer to the supplementary material.

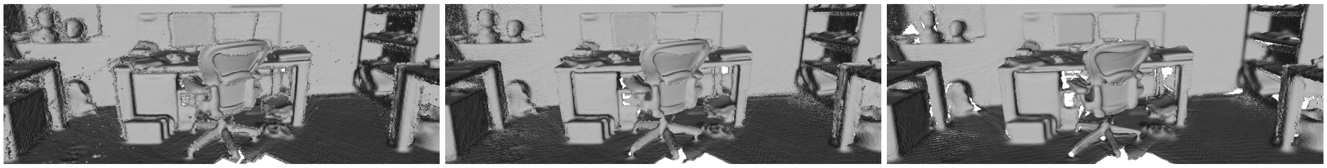
Street Sign Dataset [53]. To evaluate the performance of our method on thin structures, we also evaluate on the street sign dataset, again without fine-tuning the network. This dataset consists of 50 RGB frames and we use the COLMAP SfM pipeline [45, 46] to compute camera poses and depth maps. Qualitative results on this scene for different state-of-the-art methods are shown in Figure 6. Our method clearly outperforms TV-Flux [58] and standard TSDF, while producing comparable results with ray potentials [43]. The results also make the benefit of our routing network apparent. With routing, our method reconstructs with better completeness and less noise artifacts than without. Note that both TV-Flux and ray potentials involve an offline optimization with a smoothness prior to reduce noise and complete missing data. This prevents real-time application for these approaches, since ray potentials on this small scene runs for many hours on a cluster.

RGB-D Dataset 7-Scenes [48]. For qualitatively evaluating our method on Kinect data, we fuse the 7-Scenes [48] RGB-D dataset. For each scene, we have chosen the first trajectory and fused it using our pipeline as well as standard TSDF fusion. In Figure 7, we show that our method significantly reduces noise and mitigates the surface thickening effect compared to standard TSDF fusion. Notably, the chair leg and table edges are reconstructed with higher fidelity than it is done by standard TSDF fusion. Moreover, our method shows strong performance in denoising and removing outliers from the scene.



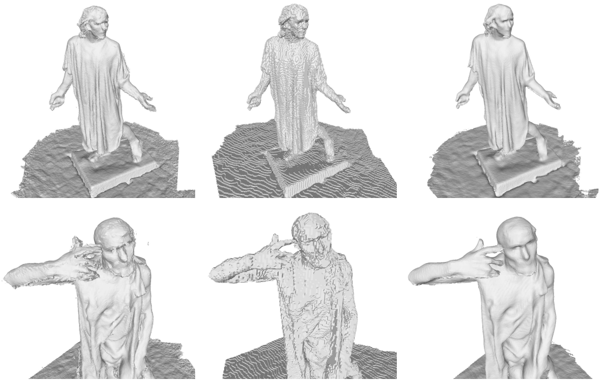
RGB TV-Flux [58] Ray Potentials [43] Standard TSDF [9] Ours w/o Routing Ours

Figure 6: **Qualitative results of our method on the Roadsign dataset [53].** Our method compares favorably to standard TSDF fusion as well as TV-Flux [58] in reconstructing thin surfaces while showing comparable performance with ray potentials [43]. Our method generalizes reasonably well since it was trained on ModelNet and never saw the noise and outlier statistics of stereo depth maps nor the shape statistics of this scene and therefore has a less complete output. ($C_{thr} = 0.5$)



Standard TSDF [9] Ours w/o Routing Ours

Figure 7: **Qualitative comparison on the heads scene of RGB-D Dataset 7-Scenes [48].** Our method significantly reduces noise artifacts and thickening effects - especially on the thin geometry of the chair's leg.



TSDF [9] PSDF [13] Ours

Figure 8: **Qualitative comparison on the Burghers of Calais scene [63].** Our method reconstructs hands and face geometry with much higher degree of detail than standard TSDF fusion and PSDF fusion.

5. Conclusion

We presented a novel real-time capable depth map fusion method tackling the common limitations of standard TSDF fusion [9]. Due to learned non-linear TSDF updates – rather than hand-crafted linear updates – our method mitigates inconsistent reconstruction results that occur at object edges and thin structures. The proposed split of our network architecture into a 2D depth routing network and a 3D depth fusion network allows to effectively handle noise and outliers at different processing stages. Moreover, sensor-specific noise distributions can be learned from small amounts of training data. Our approach outperforms competing methods on both synthetic and real data experiments. Due to its low computational requirements and compact architecture, our method has the potential to replace standard TSDF fusion in a variety of tasks and applications.

Acknowledgments. Special thanks go to Akihito Seki from Toshiba Japan for insightful discussions and comments that greatly improved the paper. This research was partially supported by Toshiba and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/IBC, or the U.S. Government.

References

[1] Maroš Bláha, Christoph Vogel, Audrey Richard, Jan D. Wegner, Thomas Pock, and Konrad Schindler. Large-scale semantic 3d reconstruction: an adaptive multi-resolution model for multi-class volumetric labeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual SLAM. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018. 3

[3] Yan-Pei Cao, Zheng-Ning Liu, Zheng-Fei Kuang, Leif Kobbelt, and Shi-Min Hu. Learning to reconstruct high-quality 3d shapes with cascaded fully convolutional networks. In *Proc. European Conference on Computer Vision (ECCV)*, pages 626–643, 2018. 3

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 5, 6

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[6] Ian Cherabier, Christian Häne, Martin R. Oswald, and Marc Pollefeys. Multi-label semantic 3d reconstruction using voxel blocks. In *International Conference on 3D Vision (3DV)*, 2016. 2

[7] Ian Cherabier, Johannes L. Schönberger, Martin R. Oswald, Marc Pollefeys, and Andreas Geiger. Learning priors for semantic 3d reconstruction. In *European Conference on Computer Vision (ECCV)*, September 2018. 3

[8] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. 2

[9] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1996, New Orleans, LA, USA, August 4-9, 1996*, pages 303–312, 1996. 1, 2, 3, 6, 8

[10] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 458–474, 2018. 1, 2

[11] 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 Trans. Graph.*, 36(3):24:1–24:18, 2017. 1, 2

[12] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 3

[13] 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 *Proc. European Conference on Computer Vision (ECCV)*, September 2018. 2, 6, 7, 8

[14] Simon Donné and Andreas Geiger. Learning non-volumetric depth fusion using successive reprojections. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7634–7643, 2019. 5

[15] Yong Duan, Mingtao Pei, and Yunde Jia. Probabilistic depth map fusion for real-time multi-view stereo. In *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11-15, 2012*, pages 368–371, 2012. 2

[16] Simon Fuhrmann and Michael Goesele. Fusion of depth maps with multiple scales. *ACM Trans. Graph.*, 30(6):148:1–148:8, 2011. 2

[17] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proc. International Conference on Computer Vision (ICCV)*, October 2019. 3

[18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018. 3

[19] Christian Häne, Christopher Zach, Andrea Cohen, Roland Angst, and Marc Pollefeys. Joint 3d scene reconstruction and class segmentation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, 2013. 2

[20] Christian Häne, Christopher Zach, Andrea Cohen, and Marc Pollefeys. Dense semantic 3d reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1730–1743, 2017. 2

[21] Shahram Izadi, Richard A. Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J. Davison, and Andrew W. Fitzgibbon. Kinectfusion: real-time dynamic 3d surface reconstruction and interaction. In *International Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2011, Vancouver, BC, Canada, August 7-11, 2011, Talks Proceedings*, page 23, 2011. 1, 2

[22] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proc. International Conference on Computer Vision (ICCV)*, pages 2326–2334, 2017. 2

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

[24] Olaf Kähler, Victor Adrian Prisacariu, Julien P. C. Valentin, and David W. Murray. Hierarchical voxel block hashing for efficient integration of depth images. *IEEE Robotics and Automation Letters*, 1(1):192–197, 2016. 2

[25] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision, 3DV 2013, Seattle, Washington, USA, June 29 - July 1, 2013*, pages 1–8, 2013. 2

[26] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5574–5584, 2017. 5

[27] Kalin Kolev, Maria Klodt, Thomas Brox, and Daniel Cremers. Continuous global optimization in multiview 3d reconstruction. *International Journal of Computer Vision*, 84(1):80–96, 2009. 1, 2

[28] Lubor Ladicky, Olivier Saurer, SoHyeon Jeong, Fabio Maninchedda, and Marc Pollefeys. From point clouds to mesh using regression. In *Proc. International Conference on Computer Vision (ICCV)*, pages 3913–3922, 2017. 3

[29] Damien Lefloch, Tim Weyrich, and Andreas Kolb. Anisotropic point-based fusion. In *18th International Conference on Information Fusion, FUSION 2015, Washington, DC, USA, July 6-9, 2015*, pages 2121–2128, 2015. 2

[30] Vincent Leroy, Jean-Sébastien Franco, and Edmond Boyer. Shape reconstruction using volume sweeping and learned photoconsistency. In *Proc. European Conference on Computer Vision (ECCV)*, pages 796–811, 2018. 2

[31] R. Maier, K. Kim, D. Cremers, J. Kautz, and M. Nießner. Intrinsic3d: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In *International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017. 2

[32] R. Maier, R. Schaller, and D. Cremers. Efficient online surface correction for real-time large-scale 3D reconstruction. In *British Machine Vision Conference (BMVC)*, London, United Kingdom, September 2017. 2

[33] Nico Marniok and Bastian Goldluecke. Real-time variational range image fusion and visualization for large-scale scenes using GPU hash tables. In *2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018, Lake Tahoe, NV, USA, March 12-15, 2018*, pages 912–920, 2018. 2

[34] Nico Marniok, Ole Johannsen, and Bastian Goldluecke. An efficient octree design for local variational range image fusion. In *Proc. German Conference on Pattern Recognition (GCPR)*, pages 401–412, 2017. 2

[35] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 6

[36] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. International Conference on Computer Vision (ICCV)*, October 2019. 3

[37] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Trans. Graph.*, 32(6):169:1–169:11, 2013. 1, 2

[38] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3, 6

[39] Despoina Paschalidou, Ali Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3897–3906, 2018. 2

[40] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *2017 International Conference on 3D Vision, 3DV 2017, Qingdao, China, October 10-12, 2017*, pages 57–66, 2017. 3

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 4

[42] M. Rothermel, N. Haala, and D. Fritsch. A median-based depthmap fusion strategy for the generation of oriented points. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume III-3, 2016. 2

[43] Nikolay Savinov, Christian Häne, Lubor Ladicky, and Marc Pollefeys. Semantic 3d reconstruction with continuous regularization and ray potentials using a visibility consistency constraint. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5460–5469, 2016. 1, 2, 7, 8

[44] Nikolay Savinov, Lubor Ladicky, Christian Hane, and Marc Pollefeys. Discrete optimization of ray potentials for semantic 3d reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5511–5518, 2015. 1, 2

[45] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[46] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 7

[47] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[48] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2013. 1, 7, 8

[49] Frank Steinbrücker, Christian Kerl, and Daniel Cremers. Large-scale multi-resolution surface reconstruction from

RGB-D sequences. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013*, pages 3264–3271, 2013. 1, 2

[50] Jörg Stückler and Sven Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *J. Visual Communication and Image Representation*, 25(1):137–147, 2014. 2

[51] Ali Osman Ulusoy, Michael J. Black, and Andreas Geiger. Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3280–3289, 2016. 2

[52] Ali Osman Ulusoy, Andreas Geiger, and Michael J. Black. Towards probabilistic volumetric reconstruction using ray potentials. In *2015 International Conference on 3D Vision, 3DV 2015, Lyon, France, October 19-22, 2015*, pages 10–18, 2015. 2

[53] Benjamin Ummenhofer and Thomas Brox. Point-based 3d reconstruction of thin objects. In *Proc. International Conference on Computer Vision (ICCV)*, pages 969–976, 2013. 7, 8

[54] Chao Wen, Yinda Zhang, Zhuwen Li, and Yanwei Fu. Pixel2mesh++: Multi-view 3d mesh generation via deformation. In *Proc. International Conference on Computer Vision (ICCV)*, October 2019. 3

[55] Thomas Whelan, Renato F. Salas-Moreno, Ben Glocker, Andrew J. Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense SLAM and light source estimation. *I. J. Robotics Res.*, 35(14):1697–1716, 2016. 1, 2

[56] Oliver J. Woodford and George Vogiatzis. A generative model for online depth fusion. In *Proc. European Conference on Computer Vision (ECCV)*, pages 144–157, 2012. 2

[57] Markus Ylimäki, Juho Kannala, and Janne Heikkilä. Accurate 3-d reconstruction with rgb-d cameras using depth map fusion and pose refinement. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1977–1982, 2018. 2

[58] Christopher Zach. Fast and high quality fusion of depth maps. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, 2008. 1, 7, 8

[59] Christopher Zach, Thomas Pock, and Horst Bischof. A globally optimal algorithm for robust tv-l1 range image integration. In *Proc. International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 1, 2

[60] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J. Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11776–11785, 2019. 3

[61] Zhirong Wu, S. Song, A. Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, June 2015. 5, 6

[62] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proc. European Conference on Computer Vision (ECCV)*, pages 851–868, 2018. 3

[63] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Trans. Graph.*, 32(4):112:1–112:8, 2013. 7, 8

[64] Jacek Zienkiewicz, Akis Tsiotsios, Andrew J. Davison, and Stefan Leutenegger. Monocular, real-time surface reconstruction using dynamic level of detail. In *Fourth International Conference on 3D Vision, 3DV 2016, Stanford, CA, USA, October 25-28, 2016*, pages 37–46, 2016. 2

[65] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics (TOG)*, 34(4), 2015. 2

[66] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb. State of the Art on 3D Reconstruction with RGB-D Cameras. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, 37(2), 2018. 2