NeuralFusion: Online Depth Fusion in Latent Space

Silvan Weder[1]   Johannes L. Schönberger[2]   Marc Pollefeys[1,2]   Martin R. Oswald[1]

[1]Department of Computer Science, ETH Zurich
[2]Microsoft Mixed Reality and AI Zurich Lab

Abstract

We present a novel online depth map fusion approach that learns depth map aggregation in a latent feature space. While previous fusion methods use an explicit scene representation like signed distance functions (SDFs), we propose a learned feature representation for the fusion. The key idea is a separation between the scene representation used for the fusion and the output scene representation, via an additional translator network. Our neural network architecture consists of two main parts: a depth and feature fusion subnetwork, which is followed by a translator sub-network to produce the final surface representation (e.g. TSDF) for visualization or other tasks. Our approach is an online process, handles high noise levels, and is particularly able to deal with gross outliers common for photometric stereo-based depth maps. Experiments on real and synthetic data demonstrate improved results compared to the state of the art, especially in challenging scenarios with large amounts of noise and outliers. The source code will be made available at https://github.com/weders/NeuralFusion.

1. Introduction

Reconstructing the geometry of a scene is a central component of many applications in 3D computer vision. Awareness of the surrounding geometry enables robots to navigate, augmented and mixed reality devices to accurately project the information into the user's field of view, and serves as the basis for many 3D scene understanding tasks.

In this paper, we consider online surface reconstruction by fusing a stream of depth maps with known camera calibration. The fundamental challenge in this task is that depth maps are typically noisy, incomplete, and contain outliers. Depth map fusion is a key component in many 3D reconstruction methods, like KinectFusion [40], VoxelHashing [42], InfiniTAM [23], and many others. The vast majority of methods builds upon the concept of averaging truncated signed distance functions (TSDFs), as proposed in the pioneering work by Curless and Levoy [9]. This approach is so popular due to its simple, highly parallelizable, and real-time capable

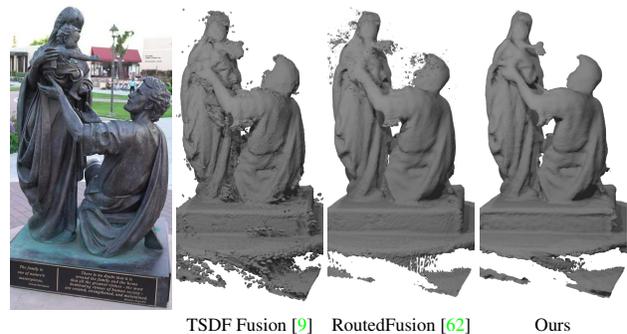TSDF Fusion [9]   RoutedFusion [62]   Ours

Figure 1. **Results of our end-to-end depth fusion** on real-world MVS data [29]. Our method learns to separate outliers and true geometry without the need of filtering heuristics.

way of fusing noisy depth maps into a surface. However, it has difficulties with handling outliers and thin geometry, which can be mainly attributed to the local integration of depth values in the TSDF volume.

To tackle this fundamental limitation, existing methods use various heuristics to filter outliers in decoupled pre- or post-processing steps. Such filtering techniques entail the usual trade-off in terms of balancing accuracy against completeness. Especially in an online fusion system, striking this balance is extremely challenging in the pre-filtering stage, because it is difficult to distinguish between a first surface measurement and an outlier. Consequently, to achieve complete surface reconstructions, one must use conservative pre-filtering, which in turn requires careful post-filtering of outliers by non-local reasoning on the TSDF volume or the final mesh. This motivates our key idea to use different scene representations for the fusion and final method output, while all prior works perform depth fusion and filtering directly in the output representation. In contrast, we perform the fusion step in a latent scene representation, implicitly learned to encode features like confidence information or local scene information. A final translation step simultaneously filters and decodes this learned scene representation into the final output relevant to downstream applications.

In summary, we make the following **contributions:**

- We propose a novel, end-to-end trainable network architecture, which separates the scene representations for depth fusion and the final output into two different modules.

- The proposed latent representation yields more accurate and complete fusion results for larger feature dimensions allowing to balance accuracy against resource demands.

- Our network architecture allows for end-to-end learnable outlier filtering within a translation step that significantly improves outlier handling.

- Although fully trainable, our approach still only performs very localized updates to the global map which maintains the online capability of the overall approach.

2. Related Work

Representation Learning for 3D Reconstruction. Many works proposed learning-based algorithms using implicit shape representations. Ladicky *et al.* [31] learn to regress a mesh from a point cloud using a random forest. The concurrent works OccupancyNetworks [38], DeepSDF [43], and IM-NET [5] encode the object surface as the decision boundary of a trained classification or regression network. These methods only use a single feature vector to encode an entire scene or object within a unit cube and are, unlike our approach, not suitable for large-scale scenes. This was improved in [3, 7, 21, 45] which use multiple features to encode parts of the scene. [8, 19, 25, 50, 51] extract image features and learn scene occupancy labels from single or multi-view images. DISN [65] works in a similar fashion but regresses SDF values. Chiyu *et al.* [22] propose a local implicit grid representation for 3D scenes. However, they encode the scene through direct optimization of the latent representation through a pre-trained neural network. More recently, Scene Representation Networks [58] learn to reconstruct novel views from a single RGB image. Liu *et al.* [33] learn implicit 3D shapes from 2D images in a self-supervised manner. These works also operate within a unit cube and are difficult to scale to larger scenes. DeepVoxels [57] encodes visual information in a feature grid with a neural network to generate novel, high-quality views onto an observed object. Nevertheless, the work is not directly applicable to a 3D reconstruction task. Our method combines the concept of learned scene representations with data fusion in a learned latent space. Recently, several works proposed a more local scene representation [1, 15, 16] that allows larger scale scenes and multiple objects. Further, [34, 41] learn implicit representations with 2D supervision via differentiable rendering. Overall, none of all mentioned works consider online updates of the shape representation as new information becomes available and adding such functionality is by no means straightforward.

Classic Online Depth Fusion Approaches. The majority of depth map fusion approaches are built upon the seminal "TSDF Fusion" work by Curless and Levoy [9], which fuses depth maps using an *implicit representation* by averaging TSDFs on a dense voxel grid. This approach especially became popular with the wide availability of low-cost depth sensors like the Kinect and has led to works like KinectFusion [40] and related works like sparse-sequence fusion [66], BundleFusion [12], or variants on sparse grids like VoxelHashing [42], InfiniTAM [46], Voxgraph [47], octree-based approaches [14, 35, 60], or hierarchical hashing [24]. However, the output of these methods usually contains typical noise artifacts, such as surface thickening and outlier blobs. Another line of works uses a *surfel-based representation* and directly fuses depth values into a sparse point cloud [27, 32, 36, 56, 61, 63]. This representation perfectly adapts to inhomogeneous sampling densities, requires low memory storage, but also lacks connectivity and topological surface information. An overview of depth fusion approaches is given in [73].

Classic Global Depth Fusion Approaches. While online approaches only process one depth map at a time, global approaches use all information at once and typically apply additional smoothness priors like total variation [30, 67], its variants including semantic information [6, 17, 18, 52, 53], or refine surface details using color information [72]. Consequently, their high compute and memory requirements prevent their application in online scenarios unlike ours.

Learned Global Depth Fusion Approaches. Octnet [49] and its follow-up OctnetFusion [48] fuse depth maps using TSDF fusion into an octree and then post-processes the fused geometry using machine learning. RayNet [44] uses a learned Markov random field and a view-invariant feature representation to model view dependencies. SurfaceNet [20] jointly estimates multi-view stereo depth maps and the fused geometry, but requires to store a volumetric grid for each input depth map. 3DMV [11] optimizes shape and semantics of a pre-fused TSDF scene of given 2D view information. Contrary to these methods, we learn online depth fusion and can process an arbitrary number of input views.

Learned Online Depth Fusion Approaches. In the context of simultaneous localization and mapping, CodeSLAM [2], SceneCode [69] and DeepFactors [10] learn a 2.5D depth representation and its probabilistic fusion rather than fusing into a full 3D model. The DeepTAM [70] mapping algorithm builds upon traditional cost volume computation with hand-crafted photoconsistency measures, which are fed into a neural network to estimate depth, but full 3D model fusion is not considered. DeFuSR [13] refines depth maps by improving cross-view consistency via reprojection, but it is not real-time capable. Similar to our approach, Weder *et al.* [62] perform online reconstruction and learn the fusion updates and weights. In contrast to our work,

all information is fused into an SDF representation which requires handcrafted pre- or post-filtering to handle outliers, which is not end-to-end trainable. The recent ATLAS [39] method fuses features from RGB input into a voxel grid and then regresses a TSDF volume. While our method learns the fusion of features, they use simple weighted averaging. Their large ResNet50 backbone limits real-time capabilities.

Sequence to Vector Learning. On a high-level, our method processes a variable length sequence of depth maps and learns a 3D shape represented by a fixed length vector. It is thus loosely related to areas like video representation learning [59] or sentiment analysis [37, 68] which processes text, audio or video to estimate a single vectorial value. In contrast to these works, we process 3D data and explicitly model spatial geometric relationships.

3. Method

Overview. Given a stream of input depth maps $D^t : \mathbb{R}^2 \to \mathbb{R}$ with known camera calibration for each time step $t \in \mathbb{N}$, we aim to fuse all surface information into a globally consistent scene representation $g : \mathbb{R}^3 \to \mathbb{R}^N$ while removing noise and outliers as well as complete potentially missing observations. The final output of our method is a TSDF map $s : \mathbb{R}^3 \to \mathbb{R}$, which can be processed into a mesh with standard iso-surface extraction methods as well as an occupancy map $o : \mathbb{R}^3 \to \mathbb{R}$. Figure 2 provides an overview of our method. The key idea is to decouple the scene representation for geometry fusion from the output scene representation. This decoupling is motivated by the difficulty for existing methods to handle outliers within an online fusion method. Therefore, we propose to fuse geometric information into a latent feature space without any preliminary outlier pre-filtering. A subsequent translator network then decodes the latent feature space into the output scene representation (*e.g.*, a TSDF grid). This approach allows for better and end-to-end trainable handling of outliers and avoids any handcrafted post-filtering, which is inherently difficult to tune and typically decreases the completeness of the reconstruction. Furthermore, the learned latent representation also enables to capture complex and higher resolution shape information, leading to more accurate reconstruction results.

Our feature fusion pipeline consists of four key stages depicted as networks in Figure 2. The first stage extracts the current state of the global feature volume into a local, view-aligned feature volume using an affine mapping defined by the given camera parameters. After the extraction, this local feature volume is passed together with the new depth measurement and the ray directions through a feature fusion network. This feature fusion network predicts optimal updates for the local feature volume, given the new measurement and its old state. The updates are integrated back into the global feature volume using the inverse affine mapping defined in the first stage. These three stages form the core of

the fusion pipeline and are executed iteratively on the input depth map stream. An additional fourth stage translates the feature volume into an application-specific scene representation, such as a TSDF volume, from which one can finally render a mesh for visualization. We detail our pipeline in the following and we refer to the supplementary material for additional low-level architectural details.

Feature Extraction. The goal of iteratively fusing depth measurements is to **(a)** fuse information about previously unknown geometry, **(b)** increase the confidence about already fused geometry, and **(c)** to correct wrong or erroneous entries in the scene. Towards these goals, the fusion process takes the new measurements to update the previous scene state g^{t-1}, which encodes all previously seen geometry. For a fast depth integration, we extract a local view-aligned feature subvolume v^{t-1} with one ray per depth measurement centered at the measured depth via nearest neighbor search in the grid positions. Each ray of features in the local feature volume is concatenated with the ray direction and the new depth measurement. This feature volume is then passed to the fusion network.

Feature Fusion. The fusion network fuses the new depth measurements D^t into the existing local feature representation v^{t-1}. Therefore, we pass the feature volume through four convolutional blocks to encode neighborhood information from a larger receptive field. Each of these encoding blocks, consists of two convolutional layers with a kernel size of three. These layers are followed by layer normalization and tanh activation function. We found layer normalization to be crucial for training convergence. The output of each block is concatenated with its input, thereby generating a successively larger feature volume with increasing receptive field. The decoder then takes the output of the four encoding blocks to predict feature updates. The decoder consists of four blocks with two convolutional layers and interleaved layer normalization and tanh activation. The output of the final layer is passed through a single linear layer. Finally, the predicted feature updates are normalized and passed as v^t to the feature integration.

Feature Integration. The updated feature state is integrated back into the global feature grid by using the inverse global-local grid correspondences of the extraction mapping. Similar to the extraction, we write the mapped features into the nearest neighbor grid location. Since this mapping is not unique, we aggregate colliding updates using an average pooling operation. Finally, the pooled features are combined with old ones using a per-voxel running average operation, where we use the update counts as weights. This residual update operation ensures stable training and a homogeneous latent space, as compared to direct prediction of the global features. Both the feature extraction and integration steps are inspired by [62], but they use tri-linear interpolation instead of nearest-neighbor sampling. When extracting and integrat-
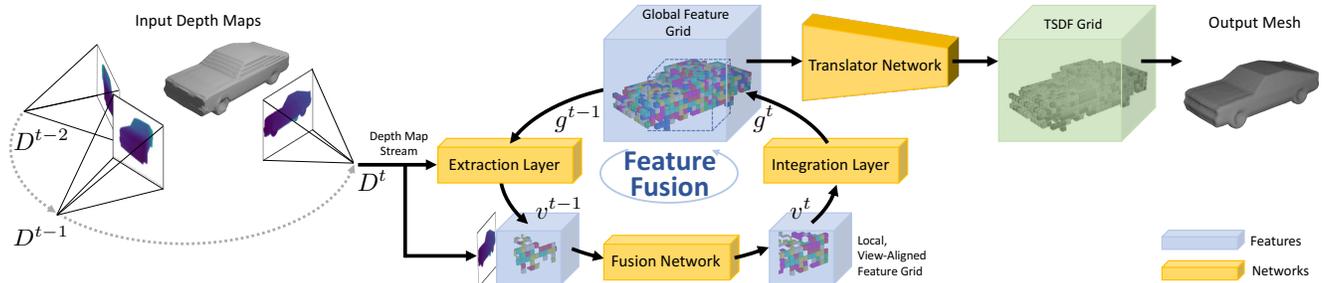
Figure 2. **Proposed online reconstruction approach.** Our pipeline consists of two main parts: **1)** A fusion network with its extraction and integration layers, and **2)** A translator network that translates the feature representation into an interpretable TSDF representation. For any new depth map D^t a local, view-aligned feature grid v^{t-1} is extracted from the previous global feature grid g^{t-1}. The fusion network updates the local feature grid v^t which is then integrated back into an updated global feature grid g^t. The translator network is independent of the fusion process and can be used asynchronously for an efficient fusion process.

ing features instead of SDF values, we empirically found that nearest-neighbor interpolation produces better results and leads to more stable convergence during training.

Feature Translation. In the final and possibly asynchronous step, we translate the latent scene representation g^t into a representation usable for visualization of the scene (*e.g.*, signed distance field or occupancy grid). The network architecture in this step is inspired by IM-Net [5]. For efficient and complete translation, we sample a regular grid of world coordinates. Then, at each of these sampled points p_i, the translator aggregates the information stored in the features of the local neighborhood and predicts the TSDF $s(p_i)$ as well as occupancy $o(p_i)$ for this specific grid location. To this end, the translator concatenates the feature vectors of the $5 \times 5 \times 5$ neighborhood and compresses them into a single feature vector using a linear layer followed by tanh activation. Next, the so combined features are concatenated with the query point feature $g^t(p_i)$ and passed through the remaining translation network, which consists of four linear layers interleaved with tanh activations and channel-wise dropout preventing the network from overfitting to a single feature channel. According to the desired output ranges, the TSDF head is activated using tanh, while the occupancy head uses a sigmoid activation. After each layer, we concatenate the output with the original query point feature $g^t(p_i)$.

Training Procedure and Loss Function. All networks are jointly trained end-to-end. In each training epoch, we randomly shuffle the input depth maps and iteratively fuse them one by one into the corresponding latent feature grid $g : \mathbb{R}^3 \rightarrow \mathbb{R}^N$. After integrating the depth map into the latent feature grid, we query the translator network, where the latent feature grid was just updated, and render the TSDF $s : \mathbb{R}^3 \rightarrow \mathbb{R}$ and occupancy $o : \mathbb{R}^3 \rightarrow [0,1]^3$. The entire pipeline is optimized using the following loss function:

$$\mathcal{L} = \frac{1}{n} \sum_i \lambda_1 \mathcal{L}_1(s_i, \hat{s}_i) + \lambda_2 \mathcal{L}_2(s_i, \hat{s}_i)$$
$$+ \lambda_o \mathcal{L}_o(o_i, \hat{o}_i) + \lambda_g \overline{\sigma_{ch}^2(g)} \ , \tag{1}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ denote the L_1 and L_2 norms, and $\mathcal{L}_o$ is the binary cross-entropy on the predicted occupancy. The $\mathcal{L}_2$ loss is helpful with outliers, whereas the $\mathcal{L}_1$ loss improves the reconstruction of fine details. In each step, n denotes the number of all updated feature grid locations. When training with outlier contaminated data, we found that setting n equal to all visited feature grid locations yields the best results. Therefore, n is a crucial hyperparameter when training the pipeline. Moreover, $\hat{s}_i$ and $\hat{o}_i$ denote the ground-truth TSDF and occupancy value, respectively. To avoid large deviations for a single feature in the latent space, we regularize the feature grid g by penalizing the mean of the channel-wise variance by $\overline{\sigma_{ch}^2(g)}$. We empirically set the loss weights to $\lambda_1 = 1., \lambda_2 = 10., \lambda_o = 0.01$, and $\lambda_g = 0.05$.

4. Experiments

We first discuss implementation details and evaluation metrics before evaluating our method on synthetic and real-world data in comparison to other methods. We further analyze our method for varying numbers of features N in an ablation study. We provide additional experiments and results in the supplementary material.

Implementation Details. Our pipeline is implemented in PyTorch and trained on an NVIDIA RTX 2080. All networks were trained using the Adam optimizer [28] with an initial learning rate of 0.01, which was adapted using an exponential learning rate scheduler at a rate of 0.998. For momentum and beta, we empirically found the default parameters to yield the best results. We trained all networks on synthetic data being augmented with artificial noise and outliers. The batch-size is set to one due to the nature of the sequential fusion process. However, we accumulate the gradients across 8 scene update steps and then update the
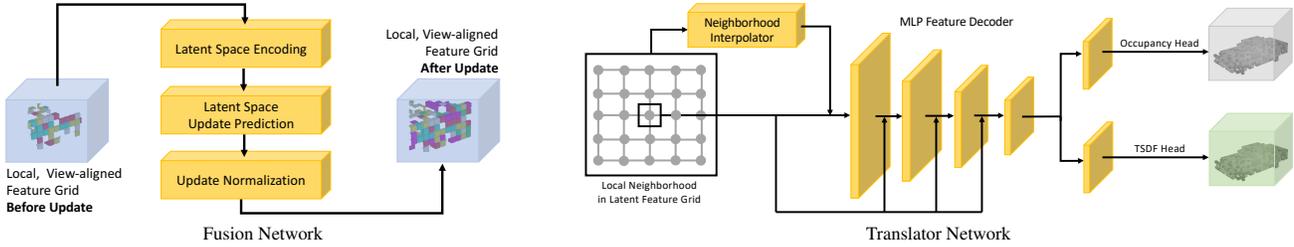
Figure 3. Left: The feature **fusion network** consists of a latent space encoder that fuses information from neighboring rays. This is followed by a latent updated predictor that predicts the updates for the latent space. Finally, the predicted features are normalized along the feature vector dimension. **Right:** The **translator network** consists of a series of neural blocks with linear layers, channel-wise dropout, and tanh activations. The first block extracts neighborhood information that is concatenated with the central feature vector. From the concatenated features the TSDF value is then predicted. All joining arrows correspond to a concatenation operation.

network parameters. Our un-optimized implementation runs at ~ 7 frames per seconds with a depth map resolution of 240×320 on an NVIDIA RTX 2080. This demonstrates the real-time applicability of our approach.

Evaluation Metrics. We use the following evaluation metrics to quantify the performance of our approach: Mean Squared Error (**MSE**), Mean Absolute Distance (**MAD**), Accuracy (**Acc.**), Intersection-over-Union (**IoU**), Mesh Completeness (**M.C.**) Mesh Accuracy (**M.A.**), and **F1** score. Further details are in the supplementary material.

4.1. Results on Synthetic Data

Datasets. We used the synthetic ShapeNet [4] and ModelNet [64] datasets for performance evaluation. From ShapeNet, we selected 13 classes for training and evaluate on the same test set as [62] consisting of 60 objects from six classes, for which pretrained models [43] are available. For ModelNet [64], we trained and tested on 10 classes using the train-test split from [62]. We first generated watertight models using the mesh-fusion pipeline used in [38] and computed TSDFs using the mesh-to-sdf[1] library. Additionally, we render depth frames for 100 randomly sampled camera views for each mesh. These depth maps are the input to our pipeline and existing methods. For both datasets, we found that training on one single object per class is sufficient for generalization to any other object and class.

Comparison to Existing Methods. For performance comparisons, we fuse depth maps and augment them with artificial depth-dependent noise as in [48]. We compare to state-of-the-art learned scene representation methods DeepSDF [43], OccupancyNetworks [38], and IF-Net [7], as well as to the online fusion methods TSDF Fusion [9] and RoutedFusion [62]. We further implemented two additional baselines to demonstrate the benefits of a fully learned scene representation for depth map fusion: **(1)** one baseline performs a learned 2D noise filtering before fusing the frames using TSDF Fusion [9], and **(2)** a baseline that post-

[1] https://github.com/marian42/mesh_to_sdf

Method	MSE↓ [e-05]	MAD↓ [e-02]	Acc.↑ [%]	IoU↑ [0,1]	F1↑ [0,1]
DeepSDF [43]	464.0	4.99	66.48	0.538	0.66
Occ.Net. [38]	56.8	1.66	85.66	0.484	0.62
IF-Net [7]	6.2	0.47	93.16	0.759	0.86
TSDF Fusion [9]	11.0	0.78	88.06	0.659	0.79
TSDF + 2D denoising	27.0	0.84	87.48	0.650	0.78
TSDF + 3D denoising	8.2	0.61	94.76	0.816	0.89
RoutedFusion [62]	5.9	0.50	94.77	0.785	0.87
Ours	**2.9**	**0.27**	**97.00**	**0.890**	**0.94**

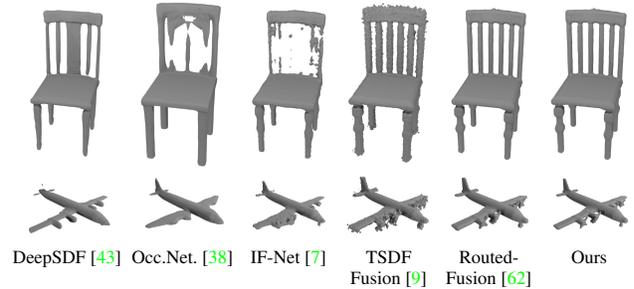DeepSDF [43] Occ.Net. [38] IF-Net [7] TSDF Fusion [9] Routed-Fusion [62] Ours

Figure 4. Quantitative and qualitative results on ShapeNet [4]. Our fusion approach consistently outperforms all baselines and state of the art in both, scene representation and depth map fusion. The performance differences to [62] are also visualized in Figure 5.

processes models fused by TSDF Fusion using a simplification of our translation network - the principle is similar to OctnetFusion [48], but on a dense grid. Further details on these baselines is given in the supplementary material. We compare all baselines on the test set of Weder *et al*. [62] in Figure 4. For input data augmentation, we used the same scale 0.005 as in [62]. Figure 4 shows that our method significantly outperforms all existing depth map fusion as well as learned scene representations. We especially emphasize the increase in IoU by more than 10%. This significant increase is due to many fine-grained improvements, where Routed-Fusion [62] wrongly predicts the sign, as shown in Figure 5. In all experiments, we set the truncation distance of TSDF Fusion to 4cm, which is similar to the receptive field of our fusion network.

Figure 5. **Mesh Accuracy (M.A.) visualization on ShapeNet meshes.** Our method consistently reconstructs more accurate meshes than the baseline depth fusion methods. Especially thin geometries (table/chair legs, lamp cable) are reconstructed with better accuracy.
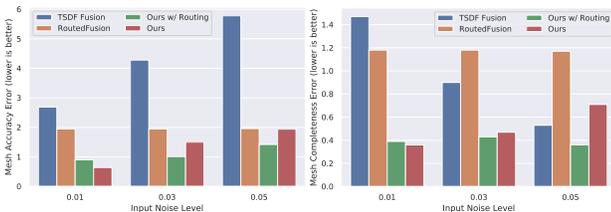

Figure 6. **Reconstruction from Noisy Depth Maps.** Our method outperforms existing depth fusion methods for various input noise levels. The performance can be further boosted by preprocessing the depth maps with the routing network from [62] leading to better robustness to high input noise levels, while it over-smoothes in the absence of noise.

Higher Input Noise Levels. We also assess our method in fusing depth maps corrupted with higher noise levels on the ModelNet dataset [64] in Figure 6. For this experiment, we augment the input depth maps with three different noise levels. We fuse the corrupted depth maps using standard TSDF Fusion [9] and RoutedFusion [62]. Since Weder *et al.* [62] showed that their proposed routing network significantly improved the robustness to higher input noise levels, we also tested our method with depth maps pre-processed by a routing network. For these experiments, we use the pre-trained routing network provided by [62].

Outlier Handling. A main drawback of [62] is its limitation in handling outliers. To this end, we run an experiment, where we augment the input depth maps with random outlier blobs. We create this data by sampling an outlier map from a fixed distribution scaled by a fixed outlier scale. Additionally, we sample three masks with a given probability (outlier fraction) and dilate it once, twice, and three times, respectively. Then, these masks are used to select the outliers from the outlier map. We report the results of this experiment in Figure 7. Note that the results might be better with even higher outlier fractions since we only evaluate on updated grid locations. The consistency in outlier filtering and increase in updated grid locations improves the metrics.

N	MSE↓ [e-05]	MAD↓ [e-02]	Acc.↑ [%]	IoU↑ [0,1]
1	-	-	-	-
2	9.45	0.64	94.67	0.717
4	4.03	0.30	97.51	0.863
8	3.99	0.29	97.46	0.862
16	3.91	0.29	97.50	0.863

Table 1. **Ablation Study.** We assess our method for different numbers of feature dimensions N. The performance saturates around $N = 8$. Note that $N = 1$ did not converge.

4.2. Ablation Study

In a series of ablation studies, we discuss several benefits of our pipeline and justify design choices.

Iterative Fusion. Ideally, fusion algorithms should be independent from the number of integrated frames and steadily improve the reconstruction as new information becomes available. In Figure 8, we show that our method is not only better than competing algorithms from the start, but also continuously improves the reconstruction as more data is fused. The metrics are averaged at every fusion step over all scenes in the test set used for all experiments on ShapeNet [4].

Frame Order Permutation. Our method does not leverage any temporal information from the camera trajectory apart from the previous fusion result. This design choice allows to apply the method also to a broader class of scenarios (*e.g.* Multi-View Stereo). Ideally, an online fusion method should be invariant to permutations of the fusion frame order. To verify this property, we evaluated the performance of our method in fusing the same set of frames in ten different random frame orders. Figure 9 shows that our method converges to the same result for any frame order and thus seems to be invariant to frame order permutations.

Feature Dimension. An important hyperparameter of our method is the feature dimension N. Therefore, we show quantitative results for the reconstruction from noisy and outlier contaminated depth maps using varying N in Table 1. We observe that a larger N clearly improves the results, but the performance eventually saturates, which justifies our choice of $N = 8$ features throughout the paper.

Latent Space Visualization. In order to verify our hypothesis that the translator network mostly filters outliers, since the fusion network can hardly distinguish between first entries and outliers, we visualize the fused latent space in Figure 10. While the translated end result is outlier free, the latent space clearly shows that the fusion network keeps track of most measurements.

4.3. Real-World Data

We also evaluate on real-world data and large-scale scenes to demonstrate scalability and generalization.

Scene3D Dataset. For real-world data evaluation, we use the lounge and stonewall scenes from the Scene3D dataset [71]. For comparability to [62], we only fuse every 10th frame from the trajectory using a model solely trained on synthetic ModelNet [64] data augmented with artificial noise and outliers.

Outlier Fraction	Method	MSE↓ [e-05]	MAD↓ [e-02]	Acc.↑ [%]	IoU↑ [0,1]
0.01	TSDF Fusion	34.51	1.17	85.17	0.645
	Routed-Fusion	5.43	0.57	95.21	0.837
	Ours	**2.27**	**0.29**	**97.57**	**0.884**
0.05	TSDF Fusion	80.72	2.02	73.86	0.432
	Routed-Fusion	9.84	0.68	94.46	0.803
	Ours	**4.91**	**0.22**	**98.05**	**0.851**
0.1	TSDF Fusion	102.50	2.43	67.47	0.341
	Routed-Fusion	14.25	0.77	92.95	0.764
	Ours	**3.35**	**0.22**	**98.48**	**0.865**

Figure 7. **Reconstruction from Outlier-Contaminated Data.** The left table states performance measures for various outlier fractions. The figures on the right show corresponding reconstruction results and errors projected on the xy-plane for an exemplary ModelNet [64] model. Our method outperforms state-of-the-art depth fusion methods regardless of the outlier fraction, but in particular with larger outlier amounts. Note that high outlier rates are common in multi-view stereo as shown in the supp. material of [29].
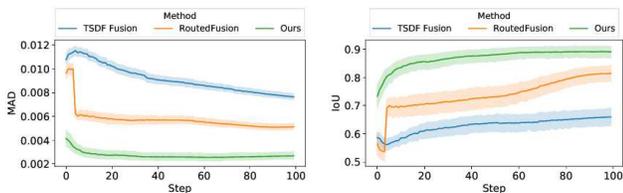

Figure 8. **Performance of iterative fusion over time.** Our method consistently outperforms both baselines [9, 62] at every step of the fusion procedure.
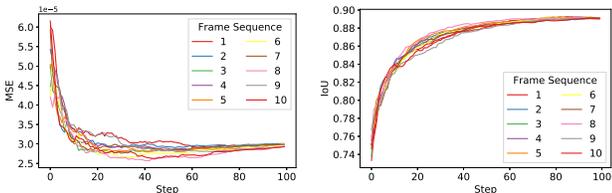

Figure 9. **Random frame order permutations.** The proposed method seems to be largely invariant to the frame integration order, since it always converges to the same result.
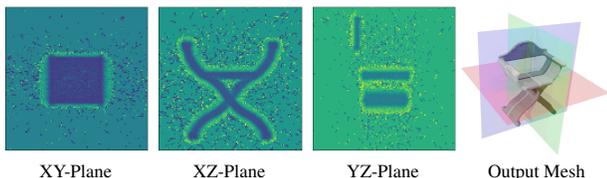

Figure 10. **Visualization of our learned latent space encoding.** Our asynchronous fusion network integrates all measurements including outliers, but the translator effectively filters the outliers to generate a clean output mesh.

	Burghers		Stonewall		Lounge		Copyroom		Cactusgarden	
Method	M.A.	M.C.	M.A.	M.C.	M.A.	M.C.	M.A.	M.C.	M.A.	M.C.
TSDF Fusion [9]	21.01	22.58	17.67	21.16	21.88	26.31	39.56	42.57	18.91	18.63
RoutedFusion [62]	20.50	41.32	19.44	80.54	22.63	53.45	38.07	57.35	19.20	41.41
Ours	**18.19**	**18.88**	**17.01**	**20.27**	**16.45**	**17.96**	**19.06**	**20.25**	**15.87**	**16.96**

Table 2. **Quantitative evaluation on Scene3D [71].** Evaluated are mesh accuracy (M.A.) [mm] & mesh completeness (M.C.) [mm].

In Figure 11, we present qualitative results of reconstructions from real-world depth maps compared to RoutedFusion [62] and TSDF Fusion [9]. We note that our method reconstructs the scene with significantly higher completeness than [62]. This is due to our learned translation from feature to TSDF space, which allows to better handle noise artifacts and outliers without the need for hand-tuned, heuristic post-filtering. Further, we show improved outlier and noise artefact removal compared to TSDF Fusion [9] while being on par with respect to completeness. These results are also quantitatively shown in Table 2.

Tanks and Temples Dataset. In order to demonstrate our methods outlier handling capability, we also run experiments on the Tanks and Temples dataset [29]. We computed stereo depth maps using COLMAP [54, 55] and fused this data using our method, PSR [26], TSDF Fusion [9], and Routed-Fusion [62]. To demonstrate the easy applicability to new datasets in scenarios with limited ground-truth, we train our method on one single scene (Ignatius) from the Tanks and Temples training set. We reconstructed a dense mesh using Poisson Surface Reconstruction (PSR) [26], rendered artificial depth maps, and used TSDF Fusion to generate a ground-truth SDF. Then, we used this ground-truth to train
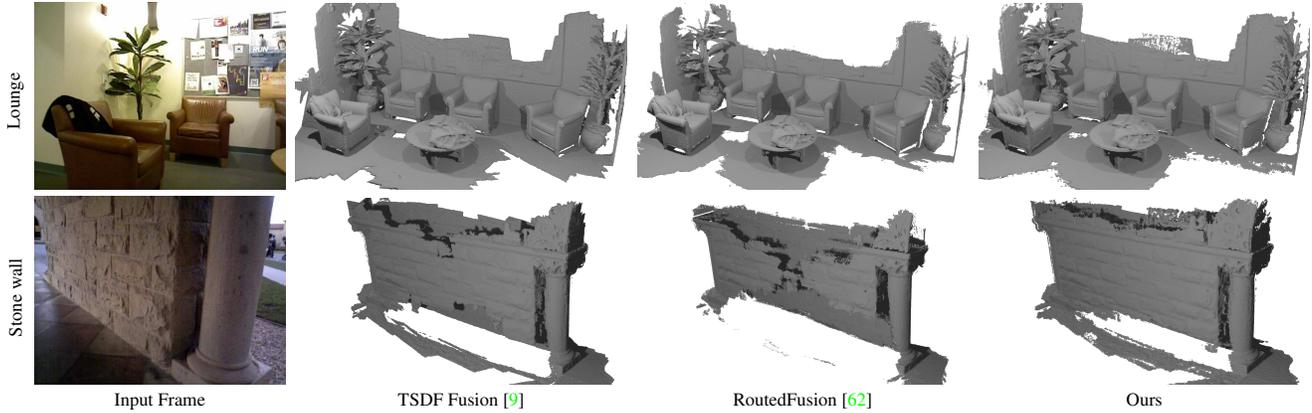
Figure 11. **Depth map fusion results on Scene3D [71]**. Our method yields significantly better completeness than RoutedFusion [62] (see stonewall) and is on par with TSDF Fusion [9]. However, our method better removes noise and outlier artifacts (see lounge reconstruction).
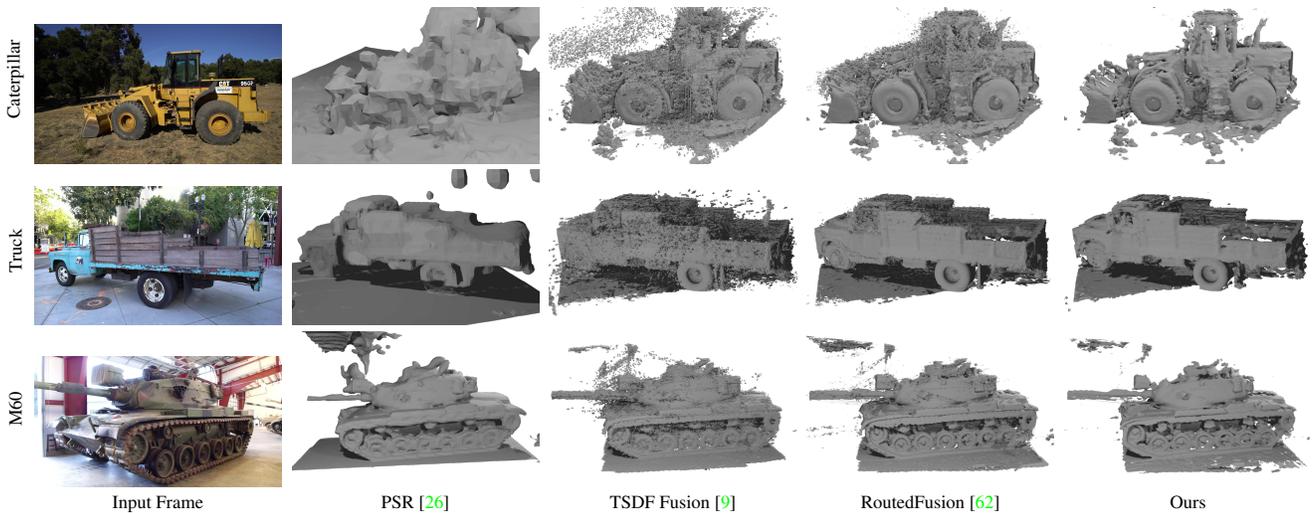

Figure 12. **Results on Tanks and Temples [29]**. Our method significantly reduces the number of outliers compared to the other methods without using any outlier filtering heuristic and solely learning it from data. We especially highlight the results on the caterpillar scene, where our proposed method filters most outliers while the reconstructions of competing methods are heavily cluttered with outliers.

the fusion of stereo depth maps. Figure 12 shows the reconstructions of the unseen Caterpillar, Truck, and M60 scene from [29]. Our proposed method significantly reduces the amount of outliers in the scene across all models. While [62] shows comparable results on some scenes, it is heavily dependent on its outlier post-filter, which fails as soon as there are too many outliers in the scene (see also Figure 1). Further, they also pre-process the depth maps using a 2D denoising network while our network uses the raw depth maps.

Limitations. While our pipeline shows excellent generalization capabilities (*e.g.* generalizing from a single MVS training scene), it is biased to the number of observations integrated during training leading to less complete results on some test scene parts with very few observations. However, this issue can be overcome by a more diverse set of training sequences with different number of observations.

5. Conclusion

We presented a novel approach to online depth map fusion with real-time capability. The key idea is to perform the fusion operation in a learned latent space that allows to encode additional information about undesired outliers and super-resolution complex shape information. The separation of scene representations for fusion and final output allows for an end-to-end trainable post-filtering as a translator network, which takes the latent scene encoding and decodes it into a standard TSDF representation. Our experiments on various synthetic and real-world datasets demonstrate superior reconstruction results, especially in the presence of large amounts of noise and outliers.

References

[1] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Meshlet priors for 3d mesh reconstruction. *CoRR*, abs/2001.01744, 2020.

[2] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. Codeslam - learning a compact, optimisable representation for dense visual SLAM. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2560–2568, 2018.

[3] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard A. Newcombe. Deep local shapes: Learning local SDF priors for detailed 3d reconstruction. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proc. European Conference on Computer Vision (ECCV)*, volume 12374 of *Lecture Notes in Computer Science*, pages 608–625. Springer, 2020.

[4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing 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. *CoRR*, abs/1512.03012, 2015.

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5939–5948, 2019.

[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.

[7] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020.

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[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.

[10] J Czarnowski, T Laidlow, R Clark, and AJ Davison. Deepfactors: Real-time probabilistic dense monocular slam. *IEEE Robotics and Automation Letters*, 5:721–728, 2020.

[11] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X*, pages 458–474, 2018.

[12] 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.

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

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

[15] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A. Funkhouser. Deep structured implicit functions. *CoRR*, abs/1912.06126, 2019.

[16] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas A. Funkhouser. Learning shape templates with structured implicit functions. *CoRR*, abs/1904.06447, 2019.

[17] 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.

[18] 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.

[19] Zeng Huang, Tianye Li, Weikai Chen, Yajie Zhao, Jun Xing, Chloe LeGendre, Linjie Luo, Chongyang Ma, and Hao Li. Deep volumetric video from very sparse multi-view performance capture. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Proc. European Conference on Computer Vision (ECCV)*, volume 11220 of *Lecture Notes in Computer Science*, pages 351–369. Springer, 2018.

[20] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 2326–2334, 2017.

[21] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6000–6009. IEEE, 2020.

[22] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. Local implicit grid representations for 3d scenes. *CoRR*, abs/2003.08981, 2020.

[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.

[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.

[25] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017,*

4-9 December 2017, Long Beach, CA, USA, pages 365–376, 2017.

[26] Michael M. Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3):29:1–29:13, 2013.

[27] 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.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[29] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017.

[30] 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.

[31] Lubor Ladicky, Olivier Saurer, SoHyeon Jeong, Fabio Maninchedda, and Marc Pollefeys. From point clouds to mesh using regression. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 3913–3922, 2017.

[32] 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.

[33] Shichen Liu, Shunsuke Saito, Weikai Chen, and Hao Li. Learning to infer implicit surfaces without 3d supervision. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 8293–8304, 2019.

[34] Shaohui Liu, Yinda Zhang, Songyou Peng, Boxin Shi, Marc Pollefeys, and Zhaopeng Cui. DIST: rendering deep implicit signed distance function with differentiable sphere tracing. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2016–2025. IEEE, 2020.

[35] 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.

[36] John McCormac, Ankur Handa, Andrew J. Davison, and Stefan Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, pages 4628–4635. IEEE, 2017.

[37] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093 – 1113, 2014.

[38] Lars M. 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)*, pages 4460–4470, 2019.

[39] Zak Murez, Tarrence van As, James Bartolozzi, Ayan Sinha, Vijay Badrinarayanan, and Andrew Rabinovich. Atlas: End-to-end 3d scene reconstruction from posed images. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[40] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136, 2011.

[41] Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3501–3512. IEEE, 2020.

[42] 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.

[43] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019.

[44] Despoina Paschalidou, Ali Osman Ulusoy, Carolin Schmitt, Luc Van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3897–3906, 2018.

[45] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020.

[46] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip H. S. Torr, and David William Murray. Infinitam v3: A framework for large-scale 3d reconstruction with loop closure. *CoRR*, abs/1708.00783, 2017.

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

[48] 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.

[49] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6620–6629, 2017.

[50] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. PIFu: Pixel-aligned

implicit function for high-resolution clothed human digitization. In *Proc. International Conference on Computer Vision (ICCV)*, pages 2304–2314. IEEE, 2019.

[51] Shunsuke Saito, Tomas Simon, Jason M. Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 81–90. IEEE, 2020.

[52] 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.

[53] 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.

[54] J. L. Schönberger and J. Frahm. Structure-from-motion revisited. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016.

[55] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Proc. European Conference on Computer Vision (ECCV)*, volume 9907 of *Lecture Notes in Computer Science*, pages 501–518. Springer, 2016.

[56] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019.

[57] Vincent Sitzmann, Justus Thies, Felix Heide, Matthias Nießner, Gordon Wetzstein, and Michael Zollhöfer. Deepvoxels: Learning persistent 3d feature embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2437–2446, 2019.

[58] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 1119–1130, 2019.

[59] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *ICML*, 2015.

[60] 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.

[61] 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.

[62] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. RoutedFusion: Learning real-time depth map fusion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[63] 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.

[64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1912–1920. IEEE Computer Society, 2015.

[65] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. DISN: deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 490–500, 2019.

[66] Long Yang, Qingan Yan, Yanping Fu, and Chunxia Xiao. Surface reconstruction via fusing sparse-sequence of depth images. *IEEE Trans. Vis. Comput. Graph.*, 24(2):1190–1203, 2018.

[67] 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.

[68] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

[69] Shuaifeng Zhi, Michael Bloesch, Stefan Leutenegger, and Andrew J. Davison. Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 11776–11785, 2019.

[70] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping with convolutional neural networks. *Int. J. Comput. Vis.*, 128(3):756–769, 2020.

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

[72] Michael Zollhöfer, Angela Dai, Matthias Innmann, Chenglei Wu, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Shading-based refinement on volumetric signed distance functions. *ACM Trans. Graph.*, 34(4):96:1–96:14, 2015.

[73] 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.