NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video

Jiaming Sun[1,2*] Yiming Xie[1*] Linghao Chen[1] Xiaowei Zhou[1] Hujun Bao[1†]
[1]Zhejiang University [2]SenseTime Research

Abstract

We present a novel framework named NeuralRecon for real-time 3D scene reconstruction from a monocular video. Unlike previous methods that estimate single-view depth maps separately on each key-frame and fuse them later, we propose to directly reconstruct local surfaces represented as sparse TSDF volumes for each video fragment sequentially by a neural network. A learning-based TSDF fusion module based on gated recurrent units is used to guide the network to fuse features from previous fragments. This design allows the network to capture local smoothness prior and global shape prior of 3D surfaces when sequentially reconstructing the surfaces, resulting in accurate, coherent, and real-time surface reconstruction. The experiments on ScanNet and 7-Scenes datasets show that our system outperforms state-of-the-art methods in terms of both accuracy and speed. To the best of our knowledge, this is the first learning-based system that is able to reconstruct dense coherent 3D geometry in real-time. Code is available at the project page: https://zju3dv.github.io/neuralrecon/.

1. Introduction

3D scene reconstruction is one of the central tasks in 3D computer vision with many applications. In augmented reality (AR) for example, to enable realistic and immersive interactions between AR effects and the surrounding physical scene, 3D reconstruction needs to be accurate, coherent and performed in real-time. While camera motion can be tracked accurately with state-of-the-art visual-inertial SLAM systems [3, 35, 1], real-time image-based dense reconstruction remains to be a challenging problem due to low reconstruction quality and high computation demands.

Most image-based real-time 3D reconstruction pipelines [38, 52] adopt the depth map fusion approach, which resemble RGB-D reconstruction methods like KinectFusion [31]. Single-view depth maps from each key frame are first estimated with real-time multi-view depth estimation methods like [48, 24, 13, 46]. The estimated depth maps are later filtered with criteria like multi-view consistency and temporal smoothness, and fused into a Truncated Signed Distance Function (TSDF) volume. The reconstructed mesh can be extracted from the fused TSDF volume with the Marching Cubes algorithm [27]. This depth-based pipeline has two major drawbacks. First, since single-view depth maps are estimated individually on each key frame, each depth estimation is from scratch instead of conditioned on the previous estimations even the view-overlapping is substantial. As a result, the scale-factor may vary even with the correct camera ego-motion. Due to depth inconsistencies between different views, the reconstruction result is prone to be either layered or scattered. One example is shown in the red boxes in Fig. 1, where the depth-based method struggles to produce coherent depth estimations on the chairs and wall. Second, since key-frame depth maps need to be estimated separately in overlapped local windows, geometry of the same 3D surface is estimated multiple times in different key

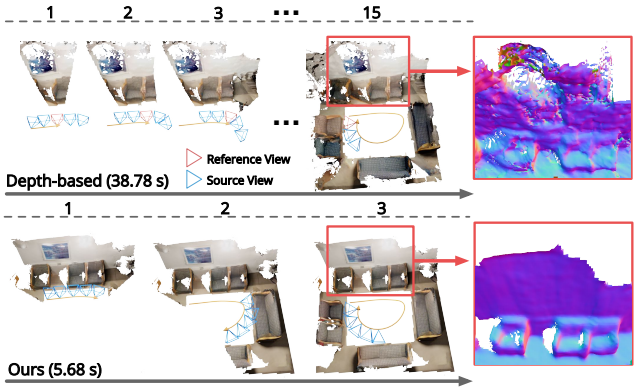

Figure 1. **Comparison between depth-based 3D reconstruction methods and the proposed method.** In depth-based methods, key-frame depths are estimated separately from each key frame, and later fused into a TSDF volume. In the proposed method, the TSDF volume is directly predicted with all the key frames in a local window. This design leads to a much more coherent reconstruction and real-time speed.

*The first two authors contributed equally. The authors are affiliated with the State Key Lab of CAD&CG and ZJU-SenseTime Joint Lab of 3D Vision. †Corresponding author: Hujun Bao.

frames, causing redundant computation.

In this paper, we propose a novel framework for real-time monocular reconstruction named NeuralRecon that jointly reconstructs and fuses the 3D geometry directly in the volumetric TSDF representation. Given a sequence of monocular images and their corresponding camera poses estimated by a SLAM system, NeuralRecon incrementally reconstructs local geometry in a view-independent 3D volume instead of view-dependent depth maps. Specifically, it unprojects the image features to form a 3D feature volume and then uses sparse convolutions to process the feature volume to output a sparse TSDF volume. With a coarse-to-fine design, the predicted TSDF is gradually refined at each level. By directly reconstructing the implicit surface (TSDF), the network is able to learn the local smoothness and global shape prior of natural 3D surfaces. Different from depth-based methods that predict depth maps for each key frame separately, the surface geometry within a local fragment window is jointly predicted in NeuralRecon, and thus *locally* coherent geometry estimation can be produced. To make the current-fragment reconstruction to be *globally* consistent with the previously reconstructed fragments, a learning-based TSDF fusion module using the Gated Recurrent Unit (GRU) is proposed. The GRU fusion makes the current-fragment reconstruction conditioned on the previously reconstructed global volume, yielding a joint reconstruction and fusion approach. As a result, the reconstructed mesh is dense, accurate and globally coherent in scale. Furthermore, predicting the volumetric representation also removes the redundant computation in depth-based methods, which allows us to use a larger 3D CNN while maintaining the real-time performance.

We validate our system on the ScanNet and 7-Scenes datasets. The experimental results show that NeuralRecon outperforms multiple state-of-the-art multi-view depth estimation methods and the volume-based reconstruction method Atlas [30] by a large margin, while achieving a real-time performance at 33 key frames per second, $\sim 10\times$ faster compared to Atlas. As shown in the supplementary video, our method is able to reconstruct large-scale 3D scenes from a video stream on a laptop GPU in real-time. To the best of our knowledge, this is the first learning-based system that is able to reconstruct dense and coherent 3D scene geometry in real-time.

2. Related Work

Multi-view Depth Estimation. The most related line of research is *real-time methods* for multi-view depth estimation. Before the age of deep learning, many renowned works in monocular 3D reconstruction [47, 21, 38, 34] have achieved good performance with plane-sweeping stereo and depth filters under the assumption of photo-consistency. [46, 51] optimize this line of research towards low power consumption on mobile platforms. Learning-based methods on real-time multi-view depth estimation try to alleviate the photo-consistency assumption with a data-driven approach. Notably, MVDepthNet [48] and Neural RGB->D [24] use 2D CNNs to process the 2D depth cost volume constructed from multi-view image features. CNMNet [26] further leverages the planar structure in indoor scenes to constrain the surface normals calculated from the predicted depth maps to obtain smooth depth estimation. These learning-based methods use 2D CNNs to process the depth cost volume to maintain a low computational cost for near real-time performance.

When the input images are high-resolution and offline computation is allowed, multi-view depth estimation is also known as the *Multiple View Stereo (MVS)* problem. PatchMatch-based methods [56, 37] have achieved impressive accuracy and are still the most popular methods applicable to high-resolution images. Learning-based approaches in MVS have recently dominated several benchmarks [2, 20] in terms of accuracy, but are only limited to processing mid-resolution images due to the GPU memory constraint. Different from the real-time methods, 3D cost volumes are constructed and 3D CNNs are used to process the cost volume as proposed in MVSNet [53]. Some recent works [12, 4] improve this pipeline with a coarse-to-fine approach. Similar design can also be found in many learning-based SLAM systems [45, 57, 42, 44].

All the above-mentioned works adopt single-view depth maps as intermediate representations. SurfaceNet [15, 16] takes a different approach and uses a unified *volumetric representation* to predict the volume occupancy. Recently, Atlas [30] also proposes a volumetric design and direct predicts TSDF and semantic labels with 3D CNN. As an offline method, Atlas aggregates the image features of the entire sequence and then predicts the global TSDF volume only once with a decoder module. We further elaborate the relationship between the proposed method and Atlas in the supplementary material. The proposed method is also related to [5, 18] in terms of using recurrent networks for multi-view feature fusion. However, their recurrent fusion is applied to only the global features and their focus is to reconstruct single objects.

3D Surface Reconstruction. After depth maps are estimated and converted to point clouds, the remaining task for 3D reconstruction is to estimate the 3D surface position and produce the reconstructed mesh. In an offline MVS pipeline [37], Poisson reconstruction [19] and Delaunay triagulation [22] are often used to fulfill this purpose. Proposed by the seminal work KinectFusion [31], incremental volumetric TSDF fusion [7] gets widely adopted in real-time reconstruction scenarios due to its simplicity and parallelization capability. [32, 10] improve KinectFusion by making it

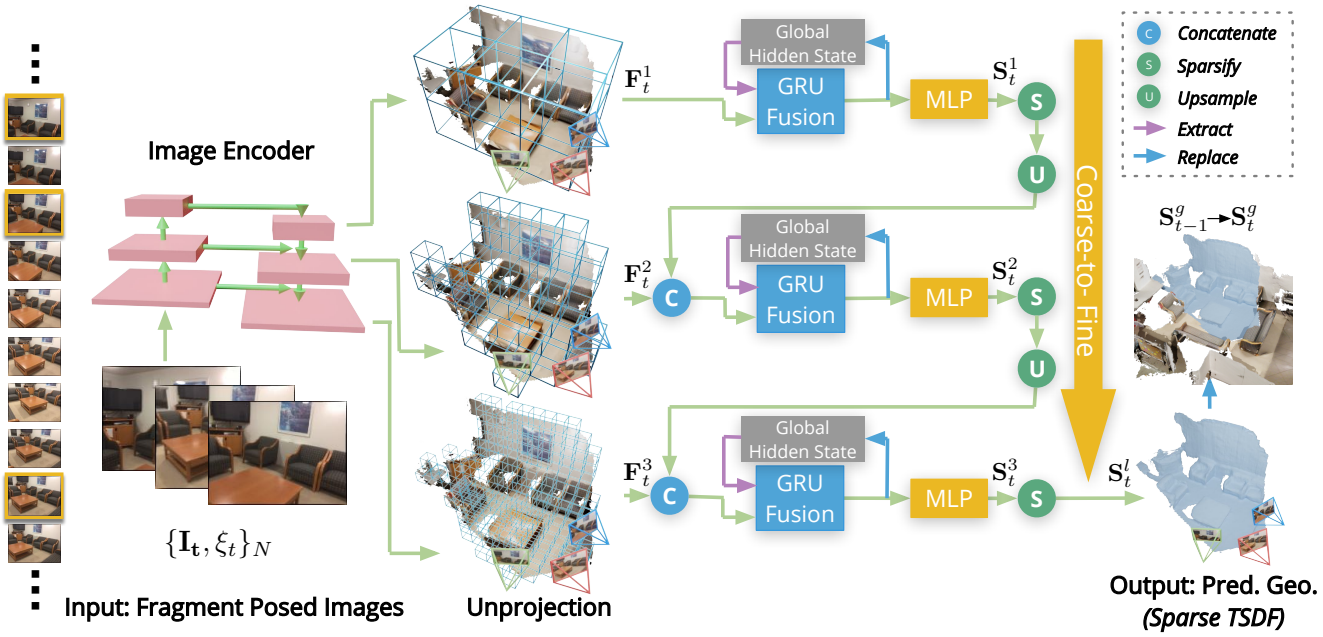

Figure 2. **NeuralRecon architecture.** NeuralRecon predicts TSDF with a three-level coarse-to-fine approach that gradually increases the density of sparse voxels. Key-frame images in the local fragment are first passed through the image backbone to extract the multi-level features. These image features are later back-projected along each ray and aggregated into a 3D feature volume $\mathbf{F}_t^l$, where l represents the level index. At the first level ($l = 1$), a dense TSDF volume $\mathbf{S}_t^1$ is predicted. At the second and third levels, the upsampled $\mathbf{S}_t^{l-1}$ from the last level is concatenated with $\mathbf{F}_t^l$ and used as the input for the **GRU Fusion** and **MLP** modules. A feature volume defined in the world frame is maintained at each level as the global hidden state of the GRU. At the last level, the output $\mathbf{S}_t^l$ is used to replace corresponding voxels in the global TSDF volume $\mathbf{S}_t^g$, yielding the final reconstruction at time t.

more scalable and robust. RoutedFusion [49, 50] changes the fusion operation from a simple linear addition into a data-dependent process.

Neural Implicit Representations. Recently, neural implicit representations [29, 33, 36, 17, 54, 25] have gained significant advances. Our work also learns a neural implicit representation by predicting SDF with the neural network from the encoded image features similar to PIFu [36]. The key difference is that we are using sparse 3D convolution to predict a discrete TSDF volume, instead of querying the MLP network with image features and 3D coordinates.

3. Methods

Given a sequence of monocular images $\{\mathbf{I}_t\}$ and camera pose trajectory $\{\xi_t\} \in \mathbb{SE}(3)$ provided by a SLAM system, the goal is to reconstruct dense 3D scene geometry accurately in real-time. We denote the global TSDF volume to reconstruct as $\mathbf{S}_t^g$, where t represents the current time step. The system architecture is illustrated in Fig. 2.

3.1. Key Frame Selection

To achieve real-time 3D reconstruction that is suitable for interactive applications, the reconstruction process needs to be incremental and the input images should be processed sequentially in local fragments [40]. We seek to find a set of suitable key frames from the incoming image stream as input for the networks. To provide enough motion parallax while keeping multi-view co-visibility for reconstruction, the selected key frames should be neither too close nor far from each other. Following [13], a new incoming frame is selected as a key frame if its relative translation is greater than t_{max} and the relative rotation angle is greater R_{max}. A window with N key frames is defined as a local fragment. After key frames are selected, a cubic-shaped fragment bounding volume (FBV) that encloses all the key frame view-frustums is computed with a fixed max depth range d_{max} in each view. Only the region within the FBV is considered during the reconstruction of each fragment.

3.2. Joint Fragment Reconstruction and Fusion

We propose to simultaneously reconstruct the TSDF volume of a local fragment $\mathbf{S}_t^l$ and fuse it with global TSDF volume $\mathbf{S}_t^g$ with a learning-based approach. The joint reconstruction and fusion is carried out in the local coordinate system. The definition of the local and global coordinate systems as well as the construction of FBV are illustrated in Fig. 1 of the supplementary material.

Image Feature Volume Construction. The N images in the local fragment are first passed through the image backbone to extract the multi-level features. Similar to previous works on volumetric reconstruction [18, 15, 30], the extracted features are back-projected along each ray into the 3D feature volume. The image feature volume $\mathbf{F}_t^l$ is

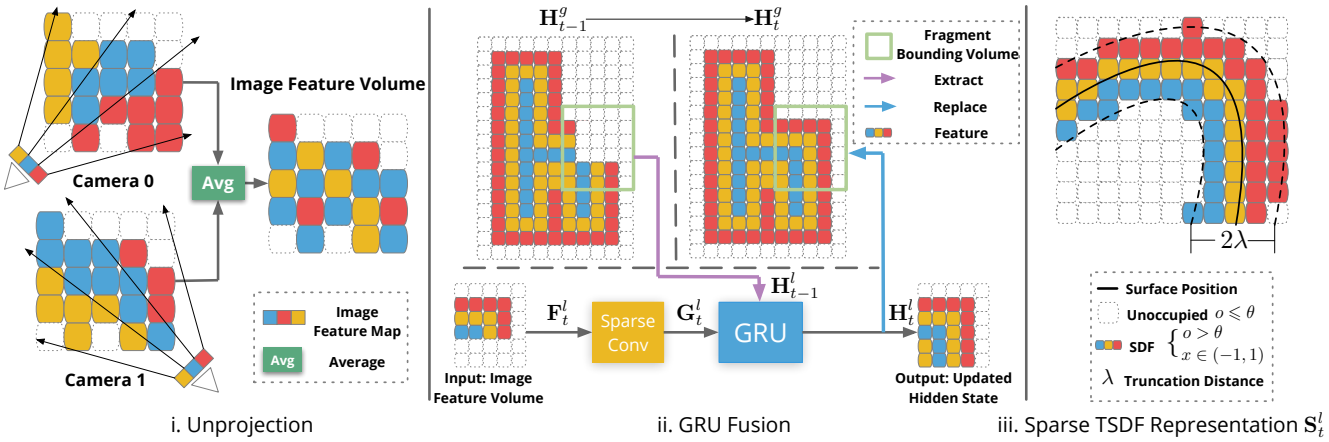

Figure 3. **2D toy examples to illustrate the unprojection, GRU fusion and sparse TSDF representation.** In figure i and ii, the colored grids mean different features. In figure iii, the colored grids mean different TSDF values. Best viewed in color.

obtained by averaging the features from different views according to the visibility weight of each voxel. The visibility weight is defined as the number of views from which a voxel can be observed in the local fragment. A visualization of this unprojection process can be found in Fig.3 i.

Coarse-to-fine TSDF Reconstruction. We adopt a coarse-to-fine approach to gradually refine the predicted TSDF volume at each level. We use 3D sparse convolution to efficiently process the feature volume $\mathbf{F}_t^l$. The sparse volumetric representation also naturally integrates with the coarse-to-fine design. Specifically, each voxel in the TSDF volume $\mathbf{S}_t^l$ contains two values, the occupancy score o and the SDF value x. At each level, both o and x are predicted by the MLP. The occupancy score represents the confidence of a voxel being within the TSDF truncation distance λ. The voxel whose occupancy score is lower than the sparsification threshold θ is defined as void space and will be sparsified. This representation of sparse TSDF volume is visually illustrated in Fig.3 iii. After the sparsification, $\mathbf{S}_t^l$ is upsampled by $2\times$ and concatenated with the $\mathbf{F}_t^{l+1}$ as the input for the GRU Fusion module (introduced later) in the next level.

Instead of estimating single-view depth maps for each key frame, NeuralRecon jointly reconstructs the implicit surface within the bounding volume of the local fragment window. This design guides the network to learn the natural surface prior directly from the training data. As a result, the reconstructed surface is locally smooth and coherent in scale. Notably, this design also leads to less redundant computation compared to depth-based methods since each area on the 3D surface is estimated only once during the fragment reconstruction.

GRU Fusion. To make the reconstruction consistent between fragments, we propose to make the current-fragment reconstruction to be conditioned on the reconstructions in previous fragments. We use a 3D convolutional variant of Gated Recurrent Unit (GRU) [6] module for this purpose. As illustrated in Fig.3 ii, at each level the image feature volume $\mathbf{F}_t^l$ is first passed through the 3D sparse convolution layers to extract 3D geometric features $\mathbf{G}_t^l$. The hidden state $\mathbf{H}_{t-1}^l$ is extracted from the global hidden state $\mathbf{H}_{t-1}^g$ within the fragment bounding volume. GRU fuses $\mathbf{G}_t^l$ with hidden state $\mathbf{H}_{t-1}^l$ and produces the updated hidden state $\mathbf{H}_t^l$, which will be passed through the MLP layers to predict the TSDF volume $\mathbf{S}_t^l$ at this level. The hidden state $\mathbf{H}_t^l$ will also be updated to global hidden state $\mathbf{H}_t^g$ by directly replacing the corresponding voxels. Formally, denoting z_t as the update gate, r_t as the reset gate, σ as the sigmoid function and W_* as the weight for sparse convolution, GRU fuses $\mathbf{G}_t^l$ with hidden state $\mathbf{H}_{t-1}^l$ with the following operations:

$$
\begin{aligned}
z_t &= \sigma(\text{SparseConv}([\mathbf{H}_{t-1}^l, \mathbf{G}_t^l], W_z)) \\
r_t &= \sigma(\text{SparseConv}([\mathbf{H}_{t-1}^l, \mathbf{G}_t^l], W_r)) \\
\tilde{\mathbf{H}}_t^l &= \tanh(\text{SparseConv}([r_t \odot \mathbf{H}_{t-1}^l, \mathbf{G}_t^l], W_h)) \\
\mathbf{H}_t^l &= (1 - z_t) \odot \mathbf{H}_{t-1}^l + z_t \odot \tilde{\mathbf{H}}_t^l
\end{aligned}
$$

Intuitively, in the context of joint reconstruction and fusion of TSDF, the update gate z_t and forget gate r_t in the GRU determine how much information from the previous reconstructions (i.e. hidden state $\mathbf{H}_{t-1}^l$) is fused to the current-fragment geometric feature $\mathbf{G}_t^l$, as well as how much information from the current-fragment will be fused into the hidden state $\mathbf{H}_t^l$. As a data-driven approach, the GRU serves as a selective attention mechanism that replaces the linear running-average operation in conventional TSDF fusion [31]. By predicting $\mathbf{S}_t^l$ after the GRU, the MLP network can leverage the context information accumulated from history fragments to produce consistent surface geometry across local fragments. This is also conceptually analogous to the depth filter in a non-learning-based 3D reconstruction pipeline [38, 34], where the current observation and the temporally-fused depths are fused with the Bayesian filter. The effectiveness of joint reconstruction and fusion is validated in the ablation study.

Integration to the Global TSDF Volume. At the last coarse-to-fine level, $\mathbf{S}_t^3$ is predicted and further sparsified

to $\mathbf{S}_t^l$. Since the fusion between $\mathbf{S}_t^l$ and $\mathbf{S}_t^g$ has been done in GRU Fusion, $\mathbf{S}_t^l$ is integrated into $\mathbf{S}_t^g$ by directly replacing the corresponding voxels after being transformed into the global coordinate. At each time step t, Marching Cubes is performed on $\mathbf{S}_t^g$ to reconstruct the mesh.

Supervision. Following [9], two loss functions are used to supervise the network. The occupancy loss is defined as the binary cross-entropy (BCE) between the predicted occupancy values and the ground-truth occupancy values. The SDF loss is defined as the ℓ_1 distance between the predicted SDF values and the ground-truth SDF values. We log-transform the SDF values of predictions and ground-truth before applying the ℓ_1 loss. The supervision is applied to all the coarse-to-fine levels.

3.3. Implementation Details

We use `torchsparse` [43] as the implementation of 3D sparse convolution. The image backbone is a variant of MnasNet [41] and is initialized with the weights pretrained from ImageNet. Feature Pyramid Network [23] is used in the backbone to extract more representative multi-level features. The entire network is trained end-to-end with randomly initialized weights except for the image backbone. The occupancy score o is predicted with a Sigmoid layer. The voxel size of the last level is $4cm$ and the TSDF truncation distance λ is set to $12cm$. d_{max} is set to $3m$. R_{max} and t_{max} are set to 15°and $0.1m$ respectively. θ is set to 0.5. Nearest-neighbor interpolation is used in the upsampling between coarse-to-fine levels.

4. Experiments

In this section, we conduct a series of experiments to evaluate the reconstruction quality and different design considerations of NeuralRecon.

4.1. Datasets, Metrics, Baselines and Protocols.

Datasets. We perform the experiments on two indoor datasets, ScanNet (V2) [8] and 7-Scenes [39]. The ScanNet dataset contains 1613 indoor scenes with ground-truth camera poses, surface reconstructions, and semantic segmentation labels. There are two training/validation splits commonly used in previous works (defined in [30] and [42]) for the ScanNet dataset. We use the same training and validation data with the corresponding baseline methods to make a fair comparison. The 7-Scenes dataset is another challenging RGB-D dataset captured in indoor scenes. Following the baseline method [26], we use the model trained on ScanNet to perform the validation on 7-Scenes.

Metrics. The 3D reconstruction quality is evaluated using 3D geometry metrics presented in [30], as well as standard 2D depth metrics defined in [11]. The definitions of these metrics are detailed in the supplementary material. Among these 3D and 2D metrics, we consider *F-score* as the most suitable metrics to measure 3D reconstruction quality since both the accuracy and completeness of the reconstruction are considered.

Baselines. We compare our method with the following baseline methods in three categories: 1) *Real-time methods* for multi-view depth estimation [48, 13, 24, 26]. Due to the efficiency constraints, the estimated depth accuracy by these methods is rather limited. We compare with these methods to demonstrate the better reconstruction accuracy of NeuralRecon given the same efficiency. 2) *Multiple View Stereo* methods [37, 14, 53, 30, 28]. These offline methods have much higher accuracy compared to real-time methods. These baselines are used to demonstrate that NeuralRecon achieves a reconstruction quality on-par with offline methods but runs in real-time. 3) *Learning-based SLAM* methods [45, 42, 44]. These monocular SLAM methods estimate camera poses and perform reconstruction simultaneously, thus the scale factor of pose and depth is usually not accurately estimated. For a fair comparison, we use ground-truth camera poses for these methods and apply a scaling factor to the predicted depth map using ground-truth depth. Among all these baseline methods, *GPMVS* [13] and *Atlas* [30] are the most relevant real-time and offline methods, respectively.

Evaluation Protocols. Since our method does not estimate depth maps explicitly, we render the reconstructed mesh to the image plane and obtain depth map estimations [30]. Key frames used for evaluation are sampled from the video sequence with an interval of 10 frames for both depth-based methods and Atlas. Following [30, 26], [53, 48, 14, 13] are fine-tuned on ScanNet. To evaluate depth-based methods [37, 48, 13, 14] in 3D, we use the point cloud fusion to obtain the 3D reconstruction following Atlas. For other depth-based methods, we use the standard TSDF fusion proposed in [31, 7]. For the reasons we detailed in the supplementary material, in order to make a fair comparison with Atlas, we also report the evaluation results using the double-layered mesh (same as Atlas). The evaluation of 3D geometry on 7-Scenes uses the single-layered mesh. We also evaluate the depth filtering operation with multi-view consistency check, which will be elaborated in the supplementary material.

4.2. Evaluation Results

ScanNet. 2D depth metrics and 3D geometry metrics are used on the ScanNet dataset. The 3D geometry evaluation results are shown in Tab. 1. Our method produces much better performance than recent learning-based methods and achieves slightly better results than COLMAP. We believe that the improvements come from the joint reconstruction and fusion design achieved by the GRU Fusion module. Compared to depth-based methods, NeuralRecon

Method	Layer	Comp ↓	Acc ↓	Recall ↑	Prec ↑	**F-score** ↑	Time (ms) ↓
MVDepthNet [48]	single	0.040	0.240	0.831	0.208	0.329	48
GPMVS [13]	single	**0.031**	0.879	**0.871**	0.188	0.304	51
DPSNet [14]	single	0.045	0.284	0.793	0.223	0.344	322
COLMAP [37]	single	0.069	0.135	0.634	0.505	0.558	2076
Ours	single	0.128	**0.054**	0.479	**0.684**	**0.562**	**30**
Atlas [30]	double	**0.062**	0.128	**0.732**	0.382	0.499	292
Ours	double	0.106	**0.073**	0.609	**0.450**	**0.516**	**30**
DeepV2D [44]	single	**0.057**	0.239	**0.646**	0.329	0.431	347
Consistent Depth [28]	single	0.091	0.344	0.461	0.266	0.331	2321
Ours	single	0.120	**0.062**	0.428	**0.592**	**0.494**	**30**

Table 1. **3D geometry metrics on ScanNet.** We use two different training/validation splits following Atlas [30] (top block) and BA-Net [42] (bottom block). We elaborate the meaning of the single and double layer in the supplementary material.

Method	Abs Rel ↓	Abs Diff ↓	Sq Rel ↓	RMSE ↓	$\delta < 1.25$ ↑	Comp ↑
COLMAP [37]	0.137	0.264	0.138	0.502	83.4	0.871
MVDepthNet [48]	0.098	0.191	0.061	0.293	89.6	0.928
GPMVS [13]	0.130	0.239	0.339	0.472	90.6	0.928
DPSNet [14]	0.087	0.158	0.035	0.232	92.5	0.928
Atlas [30]	0.065	0.123	0.045	0.251	93.6	**0.999**
Ours	**0.065**	**0.106**	**0.031**	**0.195**	**94.8**	0.909
Method	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	Sc Inv ↓	-
DeMoN [45]	0.231	0.520	0.761	0.289	0.284	-
BA-Net [42]	0.161	0.092	0.346	0.214	0.184	-
DeepV2D [44]	0.057	**0.010**	0.168	**0.080**	**0.077**	-
Consistent Depth [28]	0.073	0.037	0.217	0.105	0.103	-
Ours	**0.047**	0.024	**0.164**	0.093	0.092	-

Table 2. **2D depth metrics on ScanNet.** We use two different training/validation splits following Atlas [30] (top block) and BA-Net [42] (bottom block).

can produce coherent reconstructions both locally and globally. Our method also surpasses the volumetric baseline method Atlas [30] on the accuracy, precision, and F-score. The improvements potentially come from the design of local fragment separation in our method, which can act as a view-selection mechanism that avoids irrelevant image features to be fused into the 3D volume. In terms of completeness and recall, the proposed method has an inferior performance compared to both depth-based methods and Atlas. Since depth-based methods predict pixel-wise depth maps on each view, the coverage of their predictions is high by nature, but with the cost of accuracy. Being an offline approach, Atlas has the advantage of having a global context from the entire sequence before predicting the geometry. As a result, Atlas sometimes achieves even better completeness compared to the ground-truth due to its TSDF completion capability. However, Atlas tends to predict over-smoothed geometries, and the completed regions may be inaccurate. As for 2D depth metrics, NeuralRecon also outperforms previous state-of-the-art methods for almost all 2D depth metrics, as shown in Tab. 2.

7-Scenes. 2D depth metrics and 3D geometry metrics are evaluated on the 7-Scenes dataset. As shown in Tab. 3, our method achieves comparable performance to the state-of-the-art method CNMNet [26] and outperforms all other methods. We believe that the accuracy of the proposed method can be further improved by leveraging the planar structure information as in CNMNet. Since the model used here is only trained on ScanNet, the results also demonstrate that NeuralRecon can generalize well beyond the domain of the training data.

Efficiency. We also report the average running time of the baselines and our method in Tab. 1. Only the inference time on key frames is computed. A detailed timing analysis for each module of NeuralRecon is presented in Table 4. For volumetric methods (Atlas and ours), the running time is obtained by dividing the time of reconstructing the TSDF volume of a local fragment by the number of key frames in the local fragment. Notice that the time for TSDF fusion is not included for depth-based methods. The running time for [44, 28, 24, 26, 45] and NeuralRecon is measured on an NVIDIA RTX 2080Ti GPU. We use running time reported in [30] and [55] for [48, 14, 37, 13, 30] and [53], respectively.

As shown in Tab. 1, our time cost is 30ms per key frame, achieving real-time speed at 33 key frames per second and outperforming all previous methods. Specifically, our method runs $\sim$**10**$\times$ **faster** than Atlas, and **77**$\times$ **faster** than Consistent Depth. Predicting the volumetric representation removes the redundant computation in depth-based methods, which contributes to the fast running speed of our method. Compared to Atlas, incrementally reconstructing geometry in local fragment avoids processing a huge 3D volume, leading to a faster speed than Atlas. The use

Method	Comp ↓	Acc ↓	Recall ↑	Prec ↑	**F-score** ↑
DeepV2D [44]	0.180	0.518	0.175	0.087	0.115
CNMNet [26]	**0.150**	0.398	**0.246**	0.111	0.149
Ours	0.228	**0.100**	0.227	**0.389**	**0.282**
Method	$\delta < 1.25$ ↑	Abs Rel ↓	Sq Rel ↓	RMSE ↓	Time ↓
DeMoN [45]	31.88	0.3888	0.4198	0.8549	110
MVSNet [53]	64.09	0.2339	0.1904	0.5078	1050
N-RGBD [24]	69.26	0.1758	0.1123	0.4408	202
MVDNet [48]	71.79	0.1925	0.2350	0.4585	48
DPSNet [14]	70.96	0.1991	0.1420	0.4382	322
DeepV2D [44]	42.80	0.4370	0.5530	0.8690	347
CNMNet [26]	76.64	0.1612	**0.0832**	0.3614	80
Ours	**82.00**	**0.1550**	0.1040	**0.3470**	**30**

Table 3. **3D geometry metrics (top block) and 2D depth metrics (bottom block) on 7-Scenes.** Time is measured in milliseconds.

of sparse convolution also contributes to the superior efficiency of NeuralRecon.

4.3. Ablation Study

In this section, we conduct several ablation experiments on the ScanNet dataset to discuss the effectiveness of components in our method.

GRU Fusion. We validate the GRU Fusion design by comparing rows from (i) to (iv) in Tab. 5.

To validate the benefit of feature fusion, we compare row (i) and row (ii) in Tab. 5. Using feature fusion with the average operation obtains nearly 5% improvement for the precision metric than conventional linear TSDF fusion. Visualization in Fig. 5 shows that feature fusion with the average operation can reconstruct smoother geometry. These results demonstrate that feature fusion can be more effective than TSDF fusion using the same average operation.

Comparing row (ii) and row (iii) in Tab. 5 shows that replacing average operation with GRU gives 4% improvement in terms of recall. The mesh in Fig. 5 (iii) is also more complete than that in Fig. 5 (ii). These results demonstrate that the GRU is more effective to selectively integrate *only* the consistent information from the current-fragment to the hidden state.

The recalls in row (iii) and row (iv) in Tab. 5 show that fusion in the fragment bounding volume can produce much more complete results. Visualization results in Fig. 5 (iii) and (iv) show that, with fusion in the fragment bounding volume, our method produces fewer artifacts on the ground. Fusion in the fragment bounding volume can leverage the context information in boundaries and produce more consistent and complete surface estimation.

Number of views. We set 5, 7, 9 and 11 views as the length of a fragment respectively. As shown in row (v) in Tab. 5, the F-score has over 2% improvement when 9 views are used as a fragment. As shown in visualization results in Fig. 5 (v), with more views in a fragment, the geometry can be reconstructed more accurately compared to Fig. 5 (iv).

Img. Enc.		Unproj.	Sparse Conv.	GRU	Total
4.03	Level 1	1.27	3.70	2.18	29.56
	Level 2	1.21	3.84	2.24	
	Level 3	2.18	5.11	3.80	

Table 4. **Timing analysis** of NeuralRecon measured in milliseconds per key frame. The level number indicates the different coarse-to-fine level. Img. Enc. stands for image encoder, Unproj. stands for unprojection.

	#views	Fusion		3D Geometry Metrics		
		Area	Method	Recall	Prec	**F-score**
i	5	OCC	Linear	0.576	0.386	0.462
ii	5	OCC	Avg	0.535	0.432	0.478
iii	5	OCC	GRU	0.572	0.426	0.488
iv	5	FBV	GRU	**0.613**	0.421	0.494
-	7	FBV	GRU	0.607	0.435	0.507
v	9	FBV	GRU	0.609	**0.450**	**0.516**
-	11	FBV	GRU	0.593	0.398	0.474

Table 5. **Ablation study.** We report 3D geometry metrics on ScanNet. *OCC*: fuse 3D geometric features $\mathbf{G}_t^l$ within the occupied area where occupancy score $o > \theta$. *FBV*: fuse 3D geometric features $\mathbf{G}_t^l$ within the fragment bounding volume. *Linear*: remove GRU-Fusion and use the conventional running-average-based linear TSDF fusion to update the global TSDF volume. *Avg*: fuse 3D geometric features $\mathbf{G}_t^l$ with the average operation. *GRU*: fuse 3D geometric features $\mathbf{G}_t^l$ with GRU. We use row (v) in all other experiments. More details about ablation experiments can be found in the supplementary material.

Qualitative Results. We provide the qualitative results and the corresponding analysis in Fig. 4.

5. Conclusion

In this paper, we introduced a novel system NeuralRecon for real-time 3D reconstruction with monocular video. The key idea is to jointly reconstruct and fuse sparse TSDF volumes for each video fragment incrementally by 3D sparse convolutions and GRU. This design enables NeuralRecon to output accurate and coherent reconstruction in real-time. Experiments show that NeuralRecon outperforms state-of-the-art methods in both reconstruction quality and running speed. The sparse TSDF volume reconstructed by NeuralRecon can be directly used in downstream tasks like 3D object detection, 3D semantic segmentation and neural rendering. We believe that, by jointly training with the downstream tasks end-to-end, NeuralRecon enables new possibilities in learning-based multi-view perception and recognition systems.

Acknowledgement. The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901), NSFC (No. 61806176), and ZJU-SenseTime Joint Lab of 3D Vision.

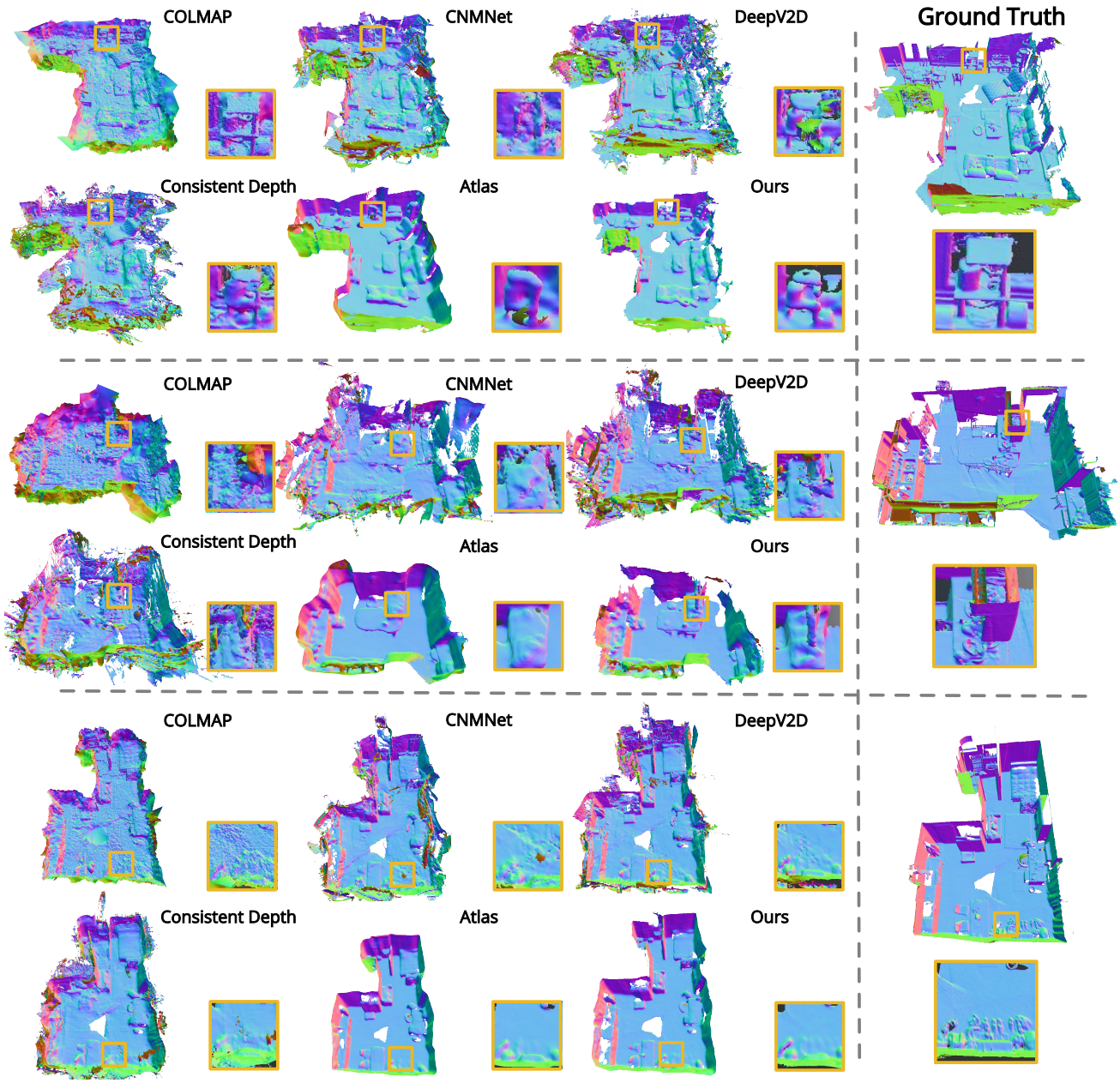

Figure 4. **Qualitative results on ScanNet.** Compared to depth-based methods, NeuralRecon can produce much more coherent reconstruction results. Notice that our method also recovers sharper geometry compared to Atlas [30], which illustrates the effectiveness of the local fragment design in our method. Reconstructing only within the local fragment window avoids irrelevant image features from far-away camera views to be fused into the 3D volume. The color indicates surface normal. More qualitative results can be found in the supplementary material and the project webpage. Zoom in for details.

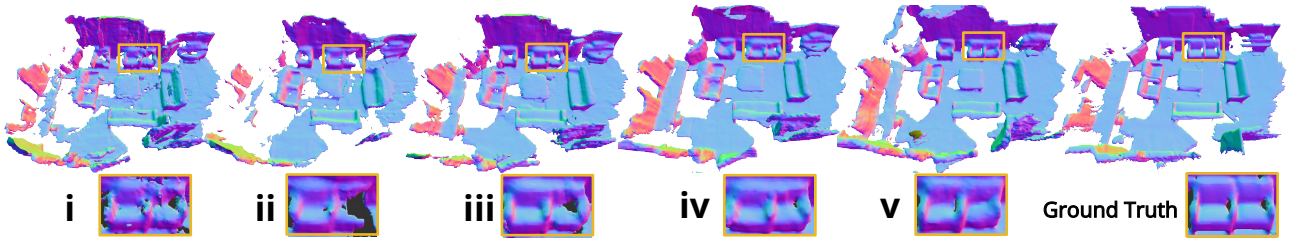

Figure 5. **Ablation study.** The indications of Roman numerals are in Tab. 5. The analysis is presented in Sec. 4.3.

References

[1] Augmented Reality with ARKit- Apple Developer. 1

[2] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-Scale Data for Multiple-View Stereopsis. *IJCV*, 2016. 2

[3] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM. *ArXiv*, 2020. 1

[4] Shuo Cheng, Zexiang Xu, Shilin Zhu, Zhuwen Li, Li Erran Li, Ravi Ramamoorthi, and Hao Su. Deep stereo using adaptive thin volume representation with uncertainty awareness. In *CVPR*, 2020. 2

[5] 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 *ECCV*, 2016. 2

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NeurIPS 2014 Workshop on Deep Learning*, 2014. 3.2

[7] Brian Curless and Marc Levoy. A Volumetric Method for Building Complex Models from Range Images. In *SIGGRAPH*, 1996. 2, 4.1

[8] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes. In *CVPR*, 2017. 4.1

[9] Angela Dai, Christian Diller, and Matthias Nießner. SG-NN: Sparse Generative Neural Networks for Self-Supervised Scene Completion of RGB-D Scans. In *CVPR*, 2020. 3.2

[10] 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 TOG*, 2017. 2

[11] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 4.1

[12] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020. 2

[13] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *ICCV*, 2019. 1, 3.1, 4.1, 4.1, 4.2

[14] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. DPSNet: End-to-end Deep Plane Sweep Stereo. In *ICLR*, 2019. 4.1, 4.1, 4.2

[15] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An end-to-end 3D neural network for multiview stereopsis. In *ICCV*, 2017. 2, 3.2

[16] Mengqi Ji, Jinzhi Zhang, Qionghai Dai, and Lu Fang. SurfaceNet+: An End-to-End 3D Neural Network for Very Sparse Multi-View Stereopsis. *IEEE TPAMI*, 2020. 2

[17] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *CVPR*, 2020. 2

[18] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a Multi-View Stereo Machine. In *NeurIPS*, 2017. 2, 3.2

[19] Michael Kazhdan and Hugues Hoppe. Screened Poisson Surface Reconstruction. *ACM TOG*, 2013. 2

[20] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM TOG*, 2017. 2

[21] Kalin Kolev, Petri Tanskanen, Pablo Speciale, and Marc Pollefeys. Turning Mobile Phones into 3D Scanners. In *CVPR*, 2014. 2

[22] P. Labatut, J.-P. Pons, and R. Keriven. Robust and Efficient Surface Reconstruction From Range Data. *Computer Graphics Forum*, 2009. 2

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 3.3

[24] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural RGB-> D Sensing: Depth and uncertainty from a video camera. In *CVPR*, 2019. 1, 2, 4.1, 4.2

[25] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural Sparse Voxel Fields. In *NeurIPS*, 2020. 2

[26] Xiaoxiao Long, Lingjie Liu, Christian Theobalt, and Wenping Wang. Occlusion-Aware Depth Estimation with Adaptive Normal Constraints. In *ECCV*, 2020. 2, 4.1, 4.2

[27] William E. Lorensen and Harvey E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *SIGGRAPH*, 1987. 1

[28] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent Video Depth Estimation. *ACM TOG*, 2020. 4.1, 4.1, 4.2

[29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy Networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2

[30] 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 *ECCV*, 2020. 1, 2, 3.2, 4.1, 1, 4.1, 2, 4.2, 4

[31] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 1, 2, 3.2, 4.1

[32] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-Time 3D Reconstruction at Scale Using Voxel Hashing. *ACM TOG*, 2013. 2

[33] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*, 2019. 2

[34] Matia Pizzoli, Christian Forster, and Davide Scaramuzza. REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time. In *ICRA*, 2014. 2, 3.2

[35] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A General Optimization-Based Framework for Local Odometry Estimation with Multiple Sensors. *ArXiv*, 2019. 1

[36] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. PIFu: Pixel-Aligned Implicit Function for High-Resolution Clothed Human Digitization. In *ICCV*, 2019. 2

[37] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise View Selection for Unstructured Multi-View Stereo. In *ECCV*. 2016. 2, 2, 4.1, 4.1, 4.2

[38] Thomas Schops, Torsten Sattler, Christian Hane, and Marc Pollefeys. 3D Modeling on the Go: Interactive 3D Reconstruction of Large-Scale Scenes on Mobile Devices. In *3DV*, 2015. 1, 2, 3.2

[39] 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 *CVPR*, 2013. 4.1

[40] Sungjoon Choi, Q. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *CVPR*, 2015. 3.1

[41] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. MnasNet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 3.3

[42] Chengzhou Tang and Ping Tan. BA-Net: Dense Bundle Adjustment Networks. In *ICLR*, 2019. 2, 4.1, 1, 4.1, 2

[43] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *ECCV*, 2020. 3.3

[44] Zachary Teed and Jia Deng. DeepV2D: Video to Depth with Differentiable Structure from Motion. In *ICLR*, 2020. 2, 4.1, 4.1, 4.2

[45] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. DeMoN: Depth and Motion Network for Learning Monocular Stereo. In *CVPR*, 2017. 2, 4.1, 4.1, 4.2

[46] Julien Valentin, Adarsh Kowdle, Jonathan T. Barron, Neal Wadhwa, Max Dzitsiuk, Michael Schoenberg, Vivek Verma, Ambrus Csaszar, Eric Turner, Ivan Dryanovski, Joao Afonso, Jose Pascoal, Konstantine Tsotsos, Mira Leung, Mirko Schmidt, Onur Guleryuz, Sameh Khamis, Vladimir Tankovitch, Sean Fanello, Shahram Izadi, and Christoph Rhemann. Depth from Motion for Smartphone AR. *ACM TOG*, 2019. 1, 2

[47] George Vogiatzis and Carlos Hernández. Video-Based, Real-Time Multi-View Stereo. *Image and Vision Computing*, 2011. 2

[48] Kaixuan Wang and Shaojie Shen. MVDepthNet: Real-Time Multiview Depth Estimation Neural Network. In *3DV*, 2018. 1, 2, 4.1, 4.1, 4.2

[49] Silvan Weder, Johannes Schönberger, Marc Pollefeys, and Martin R. Oswald. RoutedFusion: Learning Real-Time Depth Map Fusion. In *CVPR*, 2020. 2

[50] Silvan Weder, Johannes L. Schönberger, Marc Pollefeys, and Martin R. Oswald. NeuralFusion: Online Depth Fusion in Latent Space, 2020. 2

[51] Xingbin Yang, L. Zhou, Hanqing Jiang, Z. Tang, Yuanbo Wang, H. Bao, and Guofeng Zhang. Mobile3DRecon: Real-time Monocular 3D Reconstruction on a Mobile Phone. *IEEE TVCG*, 2020. 2

[52] Zhenfei Yang, Fei Gao, and Shaojie Shen. Real-Time Monocular Dense Mapping on Aerial Robots Using Visual-Inertial Fusion. In *ICRA*, 2017. 1

[53] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth Inference for Unstructured Multi-View Stereo. In *ECCV*, 2018. 2, 4.1, 4.2

[54] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview Neural Surface Reconstruction by Disentangling Geometry and Appearance. In *NeurIPS*, 2020. 2

[55] Zehao Yu and Shenghua Gao. Fast-MVSNet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *CVPR*, 2020. 4.2

[56] Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm. PatchMatch Based Joint View Selection and Depthmap Estimation. In *CVPR*, 2014. 2

[57] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. DeepTAM: Deep Tracking and Mapping. In *ECCV*, 2018. 2