CHUNKFUSION: A LEARNING-BASED RGB-D 3D RECONSTRUCTION FRAMEWORK VIA CHUNK-WISE INTEGRATION

Chaozheng Guo[1], Lin Zhang[1], Ying Shen[1], Yicong Zhou[2]*

[1]School of Software Engineering, Tongji University, Shanghai, China
[2]Department of Computer and Information Science, University of Macau, Macau, China

ABSTRACT

Recent years have witnessed a growing interest in online RGB-D 3D reconstruction. On the premise of ensuring the reconstruction accuracy with noisy depth scans, making the system scalable to various environments is still challenging. In this paper, we devote our efforts to try to fill in this research gap by proposing a scalable and robust RGB-D 3D reconstruction framework, namely ChunkFusion. In ChunkFusion, sparse voxel management is exploited to improve the scalability of online reconstruction. Besides, a chunk-wise TSDF (truncated signed distance function) fusion network is designed to perform a robust integration of the noisy depth measurements on the sparsely allocated voxel chunks. The proposed chunk-wise TSDF integration scheme can accurately restore surfaces with superior visual consistency from noisy depth maps and can guarantee the scalability of online reconstruction simultaneously, making our reconstruction framework widely applicable to scenes with various scales and depth scans with strong noises and outliers. The outstanding scalability and efficacy of our ChunkFusion have been corroborated by extensive experiments. To make our results reproducible, the source code is made online available at https://cslinzhang.github.io/ChunkFusion/.

***Index Terms*—** 3D Reconstruction, RGB-D Sensors, TSDF, Deep Learning

1. INTRODUCTION

Accurate online 3D reconstruction is a fundamental technology in robotic navigation and augmented reality. Recently, a large number of studies [1, 2] have been explored to attempt to leverage RGB-D camera to achieve online 3D reconstruction owing to its portability, popularity, and ability to capture visual and geometric information concurrently.

The key of online RGB-D 3D reconstruction is to encode the depth measurements into a 3D model incrementally. One kind of compact and effective model representation to support the reconstruction is the voxel grid. The seminal work presented by Curless and Levoy [3] first leverages the signed distance function (SDF) to represent depth maps. SDF uses the distance to the nearest surface to encode the geometry information into the discretized grid of voxels. In an advanced version of SDF, truncated signed distance function (TSDF) [4] further reduces the computational overhead by truncating the SDF values with a certain threshold and only storing a truncated region around the actual surface. TSDF has been widely used in many RGB-D 3D reconstruction schemes [4, 5] due to its simplicity and efficiency, yet it still has limitations mainly in two aspects: 1) TSDF relies on the memory-inefficient voxel grid with a fixed size, which restricts the scale of reconstruction; 2) The linear fusion adopted in TSDF fails to handle the depth-related noises and outliers, which may result in poor reconstruction quality.

*Corresponding author: cslinzhang@tongji.edu.cn.

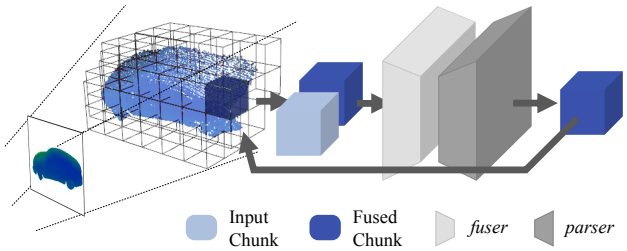


Fig. 1. The overall pipeline of ChunkFusion. Based on the point cloud projected from the scanned depth map, the corresponding chunks will be updated by fusing the newly measured TSDF via a two-stage 3D convolutional network.

To improve the scalability of TSDF, some scalable volumetric representations like MovingVolume [6, 7] and VoxelHashing [8] have been proposed. The former maintains a volume that moves with the camera and streams out data outside the volume. The latter allocates voxels sparsely where measurements are observed, enabling scalable management of the voxel volume. Besides, hierarchical data structures, like octree [9], optimize the memory management by subdividing the space, allowing for large-scale reconstruction.

In terms of enhancing the surface quality of the TSDF-based 3D reconstruction, some deep learning based schemes emerged have shown the potential to fulfill this goal. Among them, a few focus on the model representation. For instance, the designed networks in [10, 11, 12] can elegantly parameterize the representation of various 3D models. Another branch concentrates on learning-based TSDF integration. For example, RoutedFusion [13] resorts to the convolutional network to predict the TSDF update for volumetric integration. DI-Fusion [14] and NeuralFusion [15] perform geometric integration on the domain of the latent vector and achieve compelling results on noise suppression and outlier removal. Meanwhile, OctNetFusion [16, 17] manages to handle hierarchical model representation based on the octree using a 3D neural network. However, these schemes are all based on predefined volumes with fixed sizes. While in most cases, the specific required sizes of volumes are unknown before the accomplishment of the reconstruction, which limits their performance in large-scale scenes.

The aforementioned reconstruction solutions can only focus on either the scalability or the surface quality, while a framework that can balance both of the two aspects is still lacking. As an attempt to fill in the research gap to some extent, we propose ChunkFusion, a scalable learning-based RGB-D 3D reconstruction framework. Our contributions can be summarized as follows:

- ChunkFusion manages to employ the scalable voxel hashing scheme to the learning-based TSDF integration. Such a novel strategy eliminates the restriction of previous learning-based schemes and makes it possible to adapt the learning-based TSDF fusion to scenes with various scales.
- A two-stage fusion network is designed to perform the TSDF integration in an end-to-end manner. It is demonstrated that our fusion network can accurately restore the actual surfaces from noisy depth maps, yielding satisfactory reconstruction results both qualitatively and quantitatively.
- The proposed method fully exploits the sparsity of voxel representation by utilizing the chunk-wise fusion strategy and the sparsity-aware 3D convolutional network. Such an implementation scheme can further improve the computational efficiency and the surface quality of reconstruction.

2. METHOD

In this part, we will present the proposed ChunkFusion in detail, the schematic architecture of which is shown in Fig. 1. As illustrated, for a frame scanned by an RGB-D camera with a known pose, ChunkFusion first allocates chunks according to the distribution of the projected point cloud. Then the newly allocated chunks storing the depth information will be fused individually with the historical chunks by a two-stage fusion network. Subsequently, the standard iso-surface mesh extraction will be conducted on the fused chunks. As a result, the global consistent 3D model of the scanned object can be yielded.

2.1. Chunk Management

To support a scalable reconstruction, we divide the reconstructed scene evenly into small chunks. For a chunk $\mathbf{C}_i$ with an edge length of k, it is represented as $\mathbf{C}_i = (\boldsymbol{x}_i, \boldsymbol{v}_i^{t-1}, \mathcal{M}_i)$, where $\boldsymbol{x}_i \in \mathbb{Z}^3$ is the coordinate of the chunk, $\boldsymbol{v}_i^{t-1} \in \mathbb{R}^{k\times k\times k}$ is the cumulative TSDF value at timestamp $t-1$, and $\mathcal{M}_i$ is the triangle mesh extracted from $\boldsymbol{v}_i^{t-1}$.

When a new frame comes in, all the existing chunks which are occupied by the projected point cloud will be integrated individually, and the corresponding new chunks will be allocated and assigned to store those unfused points. To reduce the memory overhead and ensure a superior scalability, all the allocated chunks are organized sparsely in a hash map, with the coordinate $\boldsymbol{x}_i$ of each chunk as the key. Moreover, to enable a real-time visualization, the mesh $\mathcal{M}_i$ of each updated chunk will be re-computed with the marching-cube iso-surface extraction from the current fused TSDF value $\boldsymbol{v}_i^t$.

With the chunk-wise integration scheme, the learning-based TSDF fusion can be conducted on partial regions instead of on the model level or the scene level, which means that our fusion network only needs to learn to integrate the geometry information on surface units. Besides, these surface units are commonly shared across different models, explaining why the chunk-wise integration can better generalize to various data. Such a local implicit learning strategy has demonstrated its superiority in related 3D learning-based tasks [18, 19, 14].

However, since each chunk is integrated separately in a nonlinear manner, the discontinuity may exist on the chunk boundaries, which will lead to defects on the reconstructed surface. To solve this problem, as shown in Fig. 2 (a), we pad each chunk with voxels from its neighbor chunks before updating it. Thus, the fusion module is expected to generate a smooth transition across the boundaries with adjacent geometric information and ensure a better continuity among chunks.

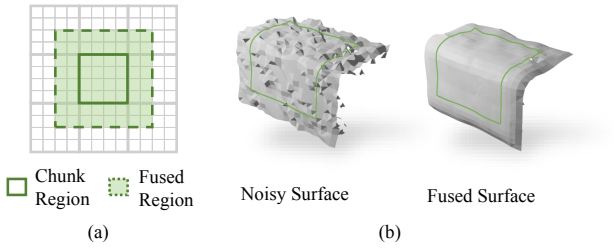


Fig. 2. Chunk-wise surface reconstruction. (a) A 2D example of chunk padding. (b) An example of chunk surface reconstruction.

2.2. Fusion Network

The proposed chunk-wise TSDF fusion network is composed of two modules, a *fuser* and a *parser*. The *fuser* integrates TSDF value $\hat{\boldsymbol{v}}_i^t$ computed from the current depth map to the historical state $\boldsymbol{v}_i^{t-1}$ in an end-to-end manner, rather than utilizing a hand-crafted weight. The *parser* subsequently refines the fused TSDF $\boldsymbol{v}_i^t$ by suppressing noises and outliers. An example of the fused chunk is illustrated in Fig. 2 (b).

The *fuser* and the *parser* are both implemented with 3D convolutional neural networks. Since most of the voxels within a chunk are unoccupied and do not carry valid geometric information, the standard 3D convolution layer that traverses all input voxels is redundant and will affect such sparsity. Therefore, we resort to sparse submanifold convolutional networks (**SSCNs**) [20] [21], which can perform convolution sparsely on occupied voxels solely. With SSCNs, substantial computational time can be saved and the sparsity of the TSDF voxels can be maintained. Thanks to the sparsity-aware convolution and the chunk-wise fusion scheme, an 8-layer convolutional network is qualified for our implicit TSDF integration. Our fusion network can efficiently extract the geometric features from input depth measurements with fewer network parameters. Such a lightweight fusion network can reduce the computational cost for integration and ensure the online capability of the reconstruction pipeline.

To train the network in a supervised manner, the loss function is defined as,

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_{grad} + \mathcal{L}_{sign}, \tag{1}$$

where $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_{grad}$ and $\mathcal{L}_{sign}$ are the L1 loss, the L2 loss, the gradient loss and the sign loss, respectively. Among them, L1 loss and L2 loss are the corresponding l_1 and l_2 distances between the predicted TSDF values and the ground-truth values.

To ensure the smoothness of the reconstructed surface, the gradient loss $\mathcal{L}_{grad}$ is introduced to restrict the 3D gradient of TSDF values, which is defined as,

$$\mathcal{L}_{grad} = \sum_{j=x,y,z} \frac{1}{k^3} \left\| \nabla_j(\mathcal{F}(\hat{\boldsymbol{v}}_i^t, \boldsymbol{v}_i^{t-1})) - \nabla_j(\boldsymbol{v}_i^*) \right\|_1, \tag{2}$$

where $\mathcal{F}(\cdot,\cdot)$ represents the TSDF fusion network, $\nabla_x(\cdot)$, $\nabla_y(\cdot)$, $\nabla_z(\cdot)$ return the 3D Sobel gradients along the axis x, y and z, and $\boldsymbol{v}_i^*$ is the corresponding ground-truth TSDF value.

The signs of the TSDF values encode whether voxels are interior or exterior to the surface, which can significantly influence the accuracy and quality of the reconstructed surface. For such a reason,

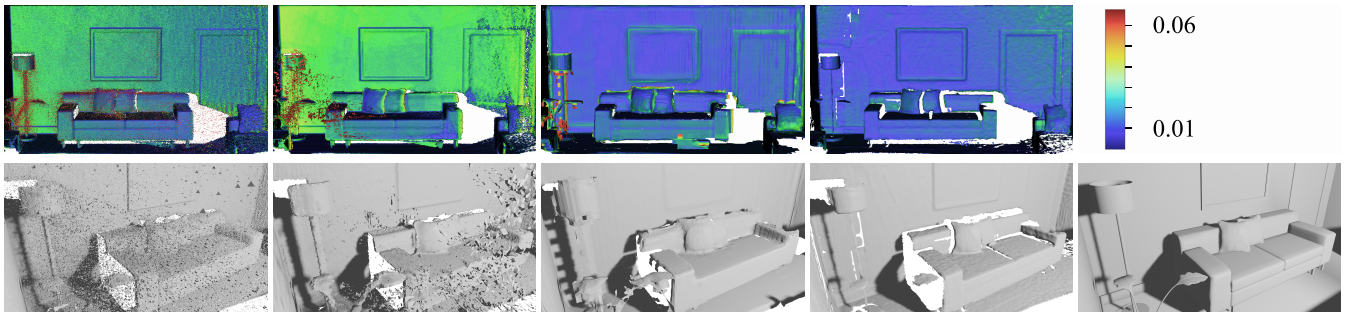


Fig. 3. Reconstruction results on "lr kt0" sequence of the ICL-NUIM dataset. The top row is the heatmaps of surface errors. The bottom row is the closed-up view of the reconstruction results. As presented, ChunkFusion can reconstruct more preferable surfaces with high accuracy and visual consistency.

we exploit the binary cross entropy to guarantee the correctness of the sign of each voxel. Specifically, the sign loss $\mathcal{L}_{sign}$ is formed as,

$$\mathcal{L}_{sign} = BCE(\frac{\mathcal{F}(\hat{\boldsymbol{v}}_i^t, \boldsymbol{v}_i^{t-1}) + 1}{2}, sign(\boldsymbol{v}_i^*)), \tag{3}$$

where $BCE(\cdot,\cdot)$ is the binary cross entropy and $sign(\cdot)$ returns the signs of the given values. The signs of the ground-truth TSDF values are treated as the classification targets in BCE.

2.3. Network Training

The *fuser* and the *parser* are trained on the synthetic dataset, ModelNet [22], from which we sample 60K depth frames and the associated ground-truth TSDF voxels from 300 different mesh models. To narrow the gap between these synthesized depth maps and the real measurements from RGB-D sensors, we add depth-related noises to the depth maps as suggested in [23].

Since the TSDF integration is conducted incrementally, the frame-based TSDF fusion networks [13, 15] are also trained incrementally with batch-size one. Instead, we train the networks with the shuffled chunk data extracted from different frames, supporting a larger batch-size configuration. Moreover, we train the *parser* first and consider the output of the pre-trained *parser* as historical states to further train the *fuser*. By doing so, a faster convergence and a better generalization ability can be guaranteed.

3. EXPERIMENT

We conducted thorough experiments to validate the performance of ChunkFusion both qualitatively and quantitatively. The experiments were conducted on two synthetic datasets, ModelNet [22] and ICL-NUIM [23], as well as self-collected real-world data. ChunkFusion was compared with three state-of-the-art competitors, including the standard TSDF [4], RoutedFusion [13], and DI-Fusion [14]. Besides, ablation studies were also performed to evaluate the efficacy of each module in ChunkFusion.

3.1. Setup

ChunkFusion was implemented with PyTorch and trained on a workstation with an Intel Xeon E5-2630 v3 @ 2.40GHz CPU and an NVIDIA GeForce Titan X GPU. The real-world data was collected using an Orbbec Astra pro RGB-D camera, and the trajectory was restored by ORB-SLAM2 [24]. The voxel resolution and truncation distance of TSDF were set to $0.01m$ and $0.04m$, respectively.

Table 1. Quantitative results on the ModelNet dataset.

Methods	MSE ↓	MAD ↓	IoU [%]↑
Standard TSDF [4]	0.0706	0.1992	0.7750
RoutedFusion [13]	0.0664	0.1879	0.7561
Ours (ChunkFusion)	**0.0409**	**0.1491**	**0.7779**

Table 2. Quantitative results on the ICL-NUIM dataset.

Methods	lr kt0	lr kt1	lr kt2	lr kt3
Standard TSDF [4]	0.0567	0.0667	0.0486	0.0441
RoutedFusion [13]	0.0491	0.0414	0.0327	0.0391
DI-Fusion [14]	0.0104	0.0120	0.0172	0.0113
Ours (ChunkFusion)	**0.0063**	**0.0060**	**0.0086**	**0.0044**

3.2. Evaluation Metrics

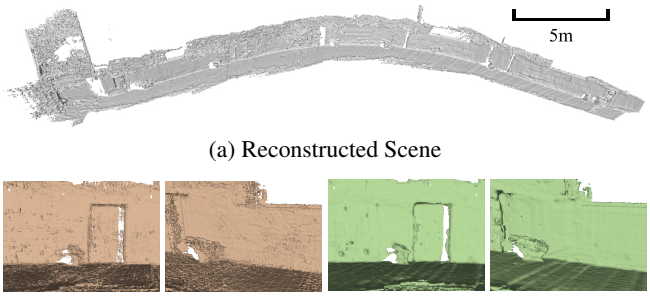
Three criteria at voxel level were considered for quantitative evaluation on the results of the ModelNet dataset. Mean square error (**MSE**) and mean absolute distance (**MAD**) were measured between the reconstructed TSDF values and the ground-truth TSDF values over all valid voxels within the truncated region. MSE and MAD could evaluate the reconstruction performance with the deviation on the TSDF field. Intersection over union (**IoU**) was also computed over the occupied voxels of reconstructed volume and ground-truth volume, which could measure the correctness of voxel occupancy.

For the results on the ICL-NUIM dataset, the **cloud/mesh distance** [23] was used to measure the performance at surface level. The reconstructed models were first finely aligned to the ground-truth mesh model. Then, the perpendicular distance to the closest triangle mesh in the ground-truth model was recorded for each vertex in the reconstruction results. The average distance of all vertices could quantify the accuracy of reconstructed surfaces.

3.3. Performance on Synthetic Dataset

The results on the ModelNet dataset are summarized in Table 1. In this experiment, we reconstructed the voxel volumes of 50 models from the corresponding synthesized depth map sequences. We can see from the results that ChunkFusion outperforms both the standard TSDF and RoutedFusion on all three metrics. It implies that our end-to-end TSDF integration network can effectively extract the geometry features from noisy depth maps and thus can obtain the precise TSDF values.

Experiments were also conducted on the ICL-NUIM dataset. We fused every 30th frame to reconstruct the scene and measured the reconstruction quality via surface error. According to the results



(a) Reconstructed Scene

TSDF Ours (ChunkFusion)

(b) Surface Detail

Fig. 4. Reconstruction results of a large indoor scene. (a) The complete view of the reconstructed scene. (b) The reconstructed details of standard TSDF and ChunkFusion. As illustrated, ChunkFusion can support the reconstruction of a large scene as well as improve the surface quality.

shown in Table 2 and Fig. 3, it can be found that ChunkFusion can achieve better surface accuracy than its counterparts. As shown in the top row of Fig. 3, the standard TSDF and RoutedFusion can not achieve satisfactory surface accuracy due to the outliers on the left side of the scene, which are in red. Such defects are shown more clearly on the bottom row of Fig. 3, in which a large number of fragments corrupt the reconstructed results. By contrast, DI-Fusion and ChunkFusion manage to suppress the influence of these outliers and restore the actual surface, thus resulting in lower surface errors. For the aspect of visual consistency, DI-Fusion tends to smooth out the details and fails to guarantee accuracy on surface boundaries. However, our method is able to balance the smoothness and accuracy of the reconstructed surfaces.

3.4. Performance on Real-world Data

The qualitative results on real-world data are illustrated in Fig. 4. As shown, thanks to the chunk-wise integration scheme, our ChunkFusion has successfully reconstructed an indoor scene of approximately $250m^2$ while its counterparts fail to support the reconstruction of such a large-scale scene.

We further demonstrate the generalization ability of ChunkFusion with surface details shown in Fig. 4 (b). As shown, the model reconstructed using standard TSDF is with a rough surface and obvious outliers, while a smooth and clean surface can be reconstructed accurately by ChunkFusion. The compelling results of ChunkFusion demonstrate that it can generalize well to real depth maps captured by RGB-D sensors, though trained on a synthetic dataset.

3.5. Ablation Analysis

Detailed ablation studies were conducted on the ModelNet dataset to validate the contribution of each network module and loss term in the proposed ChunkFusion. The four baselines for the ablation study are elaborated as follows,

- **Ours w/o *parser***: The *parser* is removed from the fusion network and the network is trained with the *fuser* solely;
- **Ours w/o $\mathcal{L}_{grad}$ & $\mathcal{L}_{sign}$**: The network is trained without $\mathcal{L}_{grad}$ and $\mathcal{L}_{sign}$, using only $\mathcal{L}_1$ and $\mathcal{L}_2$;
- **Ours w/o $\mathcal{L}_{grad}$**: The network is trained without $\mathcal{L}_{grad}$;
- **Ours w/o $\mathcal{L}_{sign}$**: The network is trained without $\mathcal{L}_{sign}$.

Table 3. Ablation study.

Methods	MSE ↓	MAD ↓	IoU [%]↑
Ours w/o *parser*	0.1152	0.2754	0.7779
Ours w/o $\mathcal{L}_{grad}$ & $\mathcal{L}_{sign}$	0.1186	0.2693	0.7779
Ours w/o $\mathcal{L}_{grad}$	0.0948	0.2446	0.7779
Ours w/o $\mathcal{L}_{sign}$	0.0500	0.1584	0.7779
Ours Full	**0.0409**	**0.1491**	0.7779

Table 4. Efficiency analysis

Methods	RoutedFusion [13]	DI-Fusion [14]	Ours (ChunkFusion)
Time (ms)	4744	334	540

The results of above baselines are summarized in Table 3.

It's worth mentioning that the IoU results remain unchanged across all ablation baselines. The underlying reason is that the fully-SSCNs implementation of ChunkFusion does not change the distribution of occupied voxels throughout the whole fusion stage.

As shown in Table 3, MSE and MAD increase with the absence of the *parser*, showing that it can help improving the accuracy of reconstructed TSDF values. We also demonstrate how different loss terms affect the reconstruction results. It can be seen that the network fails to generate satisfactory results with $\mathcal{L}_1$ and $\mathcal{L}_2$ solely, whereas MSE and MAD are reduced by nearly 50% when sign loss is introduced, demonstrating the considerable contribution of the sign loss for ensuring surface accuracy. The error is further reduced when we combine both the sign loss and the grad loss. Such a progressive improvement implies the efficacy of the proposed loss terms.

3.6. Efficiency Analysis

We also provide the comparison on time consumption of fusing a single keyframe with two baselines [13, 14] and results can be found in Table 4. ChunkFusion achieves satisfied time efficiency compared with other learning-based methods. Although ChunkFusion performs slightly slower than DI-Fusion [14], it can generate more accurate reconstruction results.

4. CONCLUSION

In this paper, we proposed a scalable learning-based RGB-D 3D reconstruction framework, ChunkFusion. The key idea of ChunkFusion is to combine the sparse voxel management with a chunk-wise fusion network to achieve improved memory efficiency and reconstruction quality. The proposed reconstruction framework is widely applicable to scenes with various scales and depth scans with strong noises and outliers. Experiments on various datasets showed the superiority of our method on both reconstruction scalability and geometric consistency over the state-of-the-art competitors.

5. ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grants 61973235, 61936014, and 61972285, in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Dawn Program of Shanghai Municipal Education Commission under Grant 21SG23, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities.

6. REFERENCES

[1] Thomas Schops, Torsten Sattler, and Marc Pollefeys, "BAD SALM: Bundle adjusted direct RGB-D SLAM," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 134–144.

[2] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface reintegration," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1, 2017.

[3] Brian Curless and Marc Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 303–312.

[4] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[5] Victor Adrian Prisacariu, Olaf Kähler, Stuart Golodetz, Michael Sapienza, Tommaso Cavallari, Philip HS Torr, and David W Murray, "InfiniTAM v3: A framework for large-scale 3D reconstruction with loop closure," *arXiv preprint arXiv:1708.00783*, 2017.

[6] Henry Roth and Marsette Vona, "Moving volume KinectFusion," in *Proceedings of the British Machine Vision Conference*, 2012, pp. 1–11.

[7] Thomas Whelan, Michael Kaess, Maurice H Fallon, Hordur Johannsson, and John B Mcdonald, "Kintinuous: Spatially extended KinectFusion," in *Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, 2012.

[8] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger, "Real-time 3D reconstruction at scale using voxel hashing," *ACM Transactions on Graphics*, vol. 32, no. 6, pp. 1–11, 2013.

[9] Ming Zeng, Fukai Zhao, Jiaxiang Zheng, and Xinguo Liu, "Octree-based fusion for realtime 3D reconstruction," *Graphical Models*, vol. 75, no. 3, pp. 126–136, 2013.

[10] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 165–174.

[11] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[12] Zhiqin Chen and Hao Zhang, "Learning implicit fields for generative shape modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5939–5948.

[13] Silvan Weder, Johannes Schonberger, Marc Pollefeys, and Martin R Oswald, "RoutedFusion: Learning real-time depth map fusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4887–4897.

[14] Jiahui Huang, Shisheng Huang, Haoxuan Song, and Shimin Hu, "DI-Fusion: Online implicit 3D reconstruction with deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8932–8941.

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

[16] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger, "OctNet: Learning deep 3D representations at high resolutions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3577–3586.

[17] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger, "OctNetFusion: Learning depth fusion from data," in *Proceedings of the International Conference on 3D Vision*, 2017, pp. 57–66.

[18] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al., "Local implicit grid representations for 3D scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6001–6010.

[19] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe, "Deep local shapes: Learning local SDF priors for detailed 3D reconstruction," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 608–625.

[20] Benjamin Graham and Laurens van der Maaten, "Submanifold sparse convolutional networks," *arXiv preprint arXiv:1706.01307*, 2017.

[21] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten, "3D semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9224–9232.

[22] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1912–1920.

[23] Ankur Handa, Thomas Whelan, John McDonald, and Andrew J Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2014, pp. 1524–1531.

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.