This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Geometry-guided Feature Learning and Fusion for Indoor Scene Reconstruction

Ruihong Yin[1], Sezer Karaoglu[1,2], Theo Gevers[1,2]
[1]University of Amsterdam, Amsterdam, The Netherlands
[2]3DUniversum, Amsterdam, The Netherlands
r.yin@uva.nl, s.karaoglu@3duniversum.com, Th.Gevers@uva.nl

Abstract

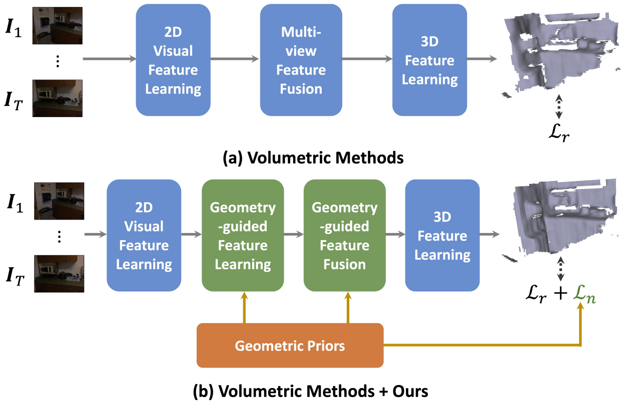
In addition to color and textural information, geometry provides important cues for 3D scene reconstruction. However, current reconstruction methods only include geometry at the feature level thus not fully exploiting the geometric information.

In contrast, this paper proposes a novel geometry integration mechanism for 3D scene reconstruction. Our approach incorporates 3D geometry at three levels, i.e. feature learning, feature fusion, and network supervision. First, geometry-guided feature learning encodes geometric priors to contain view-dependent information. Second, a geometry-guided adaptive feature fusion is introduced which utilizes the geometric priors as a guidance to adaptively generate weights for multiple views. Third, at the supervision level, taking the consistency between 2D and 3D normals into account, a consistent 3D normal loss is designed to add local constraints.

Large-scale experiments are conducted on the ScanNet dataset, showing that volumetric methods with our geometry integration mechanism outperform state-of-the-art methods quantitatively as well as qualitatively. Volumetric methods with ours also show good generalization on the 7-Scenes and TUM RGB-D datasets.

1. Introduction

3D scene reconstruction is an important topic in 3D computer vision, with many applications such as mixed/augmented reality, autonomous navigation, and robotics. It is also considered one of the fundamental tasks in 3D scene understanding including 3D segmentation [22, 33, 36] and object detection [37, 40]. Although nowadays cameras equipped with depth sensors (*e.g.* Lidar and Kinect) can reconstruct scenes using perspective projection and depth fusion [18], these *RGB-D* cameras are still expensive, and not yet widely used in consumer cameras. Therefore, they are limited in their applicability. In contrast, scene reconstruction from *RGB* images (multi-view or video) is much more accessible.


I_1
I_T
2D Visual Feature Learning
Multi-view Feature Fusion
3D Feature Learning
$\mathcal{L}_r$
I_1
I_T
2D Visual Feature Learning
Geometry-guided Feature Learning
Geometry-guided Feature Fusion
3D Feature Learning
$\mathcal{L}_r + \mathcal{L}_n$
Geometric Priors
(a) Volumetric Methods

(b) Volumetric Methods + Ours

Figure 1. **Pipeline of existing volumetric methods compared to our proposed geometry-guided feature learning and fusion for 3D scene reconstruction.** Our approach (green parts) integrates view-dependent and local geometry into (1) feature learning, (2) multi-view feature fusion, and (3) network supervision.

A standard approach to 3D scene reconstruction is to compute the Truncated Signed Distance Function (TSDF) volume and then apply the marching cubes algorithm [15] to capture the surface. To generate the TSDF volume, traditional reconstruction methods [11, 42, 43, 23, 38, 44] first generate depth maps for each RGB image and then apply depth fusion [5]. Due to pixel-level prediction, depth-based methods can generate dense 3D points but may suffer from scale ambiguity and depth inconsistency between overlapping regions in different views. Recently, volumetric (direct) methods [17, 13, 2, 28] are proposed to predict the TSDF directly, without reliance on depth estimation. 3D scenes are modeled using volumetric methods that employ 3D CNNs, allowing for the filling of unobserved gaps and resulting in enhanced predictions. However, both depth-based and volumetric methods still capture texture and color features based on RGB information.

For multi-view tasks, geometric information (*e.g.* surface normal and viewing direction) provides rich view-dependent cues of 3D scenes. For example, the best viewing direction is perpendicular to the viewing position. This viewpoint (or one close to it) is preferred over other views. Also, voxels derived from the same plane should have similar surface normals. Hence, extracting important cues from these geometries can be beneficial for feature learning and scene representation. In addition, multi-view feature fusion plays a vital role in volumetric reconstruction methods. Due to changing imaging conditions (*e.g.* illumination, camera orientation, and occlusion), instead of simply averaging views, some views may be preferred over others in terms of their positioning (*i.e.* more useful geometry information). Furthermore, volumetric methods usually supervise the predicted TSDF in a voxel-to-voxel manner ignoring local information, and hence may deviate from the actual surfaces.

To address the aforementioned issues, in this paper, a geometry integration mechanism is proposed for 3D scene reconstruction. To this end, geometric information is exploited by our method at three different stages (see Figure 1b): (1) feature learning, (2) feature fusion, and (3) network supervision. Firstly, to exploit discriminative information for 3D reconstruction, a geometry-guided feature learning (G2FL) is introduced to encode and integrate geometric priors (*e.g.* surface normal, projected depth, and viewing direction) into the multi-view features. Transformers and multilayer perceptron (MLP) are utilized to exploit the geometric information. Secondly, during multi-view feature fusion, the occluded views and views away from others may be assigned different attention levels. Therefore, the occlusion prior and relative pose distance are adopted to construct the multi-view attention function, forming a geometry-guided adaptive feature fusion (G2AFF). Thirdly, at the supervision level, the 3D surface normal is calculated from the TSDF, which at the same time maintains local information. To enhance the local constraints and improve the reconstruction quality, a consistent 3D normal loss (C3NL) is proposed, considering the consistency between 2D and 3D normal, discarding boundaries and thin objects.

Our main contributions are summarized as follows:

- A novel geometry integration mechanism is proposed for 3D scene reconstruction, encoding geometric priors at three levels, *i.e.* feature learning, feature fusion, and network supervision.

- A geometry-guided feature learning scheme encodes 3D geometry into multi-view features. A geometry-guided adaptive feature fusion method uses geometric priors as a guidance to learn a multi-view weight function adaptively.

- The consistency between 2D and 3D normal is exploited. A consistent 3D normal loss is introduced to constrain local planar regions in the prediction.

- Volumetric methods enhanced with our method show state-of-the-art performance on the ScanNet dataset and demonstrates convincing generalization on the 7-Scenes and TUM RGB-D datasets.

2. Related work

2.1. 3D scene reconstruction

Depth-based reconstruction. Depth-based methods typically follow a similar approach, *i.e.* first building a plane sweep cost volume [4, 8] at the image or feature level, and then using convolutional layers to extract and fuse features from neighbouring views, finally predicting the depth maps. Cost volume aims to capture information from source images, as complementary features for the reference image. Different cost metrics are used, *e.g.* concatenation, dot product, and per-channel variance. For example, MVS-Net [42] proposes a variance-based cost in each channel. In DPSNet [11], the cost is calculated by concatenating reference features and the warped features. MVDepthNet [38] and GP-MVS [10] adopt absolute differences between input images to measure the similarity of different views, while Neural RGBD [13] uses the same metric at the feature level. DeepVideoMVS [7] and SimpleRecon [23] compute the dot product between reference and warped features.

Volumetric (direct) reconstruction. Atlas [17] is the first work to regress the TSDF directly, without the depth map as an intermediate product. Compared to depth-based methods, Atlas learns to fill in unobserved regions. Based on Atlas, NeuralRecon [30] designs a learning-based TSDF fusion to transfer features from previous to current fragments. TransformerFusion [2] proposes a learned multi-view fusion module using a Transformer and predicts the occupancy similar to [19]. VoRTX [28] adopts a Transformer to extract features and proposes an occlusion-aware fusion module.

2.2. Geometric priors in 3D scene reconstruction

A number of methods use geometric information for 3D scene reconstruction. For example, GP-MVS [13] applies a relative pose distance to the Gaussian kernel [25], which guides the learning in latent space. NeuralRecon [30] concatenates the projected depth to 3D features after multi-view fusion. TransformerFusion [2] integrates pixel validity, viewing direction, and projected depth into the features. Viewing direction and occlusion prior are exploited in VoRTX [28]. SimpleRecon [23] introduces the use of geometric metadata for scene reconstruction, *e.g.* ray angle and depth validity mask. However, they only use a limited number of geometric priors and exploit 3D geometry at the feature level. In contrast, our method proposes to exploit geometric priors at different stages of the 3D scene reconstruction pipeline.

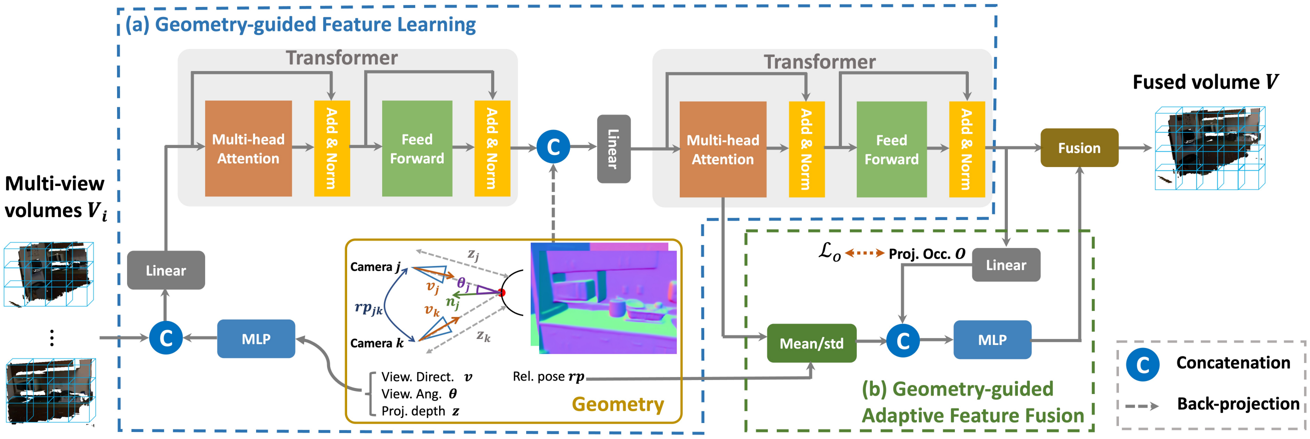


Figure 2. **Details of our proposed geometry-guided feature learning and geometry-guided adaptive feature fusion.** (a) Geometry-guided feature learning: After 2D visual feature learning, view-dependent geometric priors (*e.g.* surface normal and viewing direction) are encoded and fused into the visual features of the multi-view volume using a MLP, linear layers, and Transformers. (b) Geometry-guided adaptive feature fusion: Fusion weighting is adaptively learned by a MLP with the guidance of features, relative pose distances, and occlusion priors.

2.3. Multi-view feature fusion

The standard way of fusing multi-view features, *i.e.* computing the average, considers each view in the same way. In contrast, attention-based fusion gives attention according to the information in each view. For instance, the AttnSets module [41] is proposed to aggregate multi-view features. DeepVideoMVS [7] includes a ConvLSTM [26] to integrate past information into the current view. In particular, the use of Transformers [35, 3, 39, 45] shows their effectiveness in feature awareness. Also, other methods are proposed, designing their fusion module based on Transformers. For example, TransformerFusion [2] adopts Transformers to learn weights for each view and to select views during inference. VoRTX [28] constructs an occlusion-aware fusion using Transformers. In contrast to existing methods, in this paper, an adaptive feature fusion is proposed to model the attention by the guidance of multi-view features and geometries.

3. Method

T images $\mathbf{I}_i \in \mathbb{R}^{3\times H\times W}$ with camera intrinsics $\mathbf{K}_i \in \mathbb{R}^{3\times 3}$ and camera pose $\mathbf{P}_i = (\mathbf{R}_i, \mathbf{t}_i) \in \mathbb{R}^{4\times 4}$ are taken as an input, where i is the view index. As shown in Figure 1a, volumetric (direct) methods generally consist of three core components, *i.e.* 2D visual feature learning, multi-view feature fusion, and 3D feature learning. 2D visual feature learning exploits 2D convolutional neural networks (CNNs) [9, 31] to extract 2D features, after which they are back-projected to a 3D space. Next, multi-view feature fusion combines these features into one volume. Finally, 3D feature learning adopts a 3D CNN [32] to regress the TSDF value. Our geometry integration mechanism aims to combine 3D geometry into general volumetric methods, see Figure 1b. The key differences are: **(1)** After 2D visual feature learning, geometry-guided feature learning incorporates view-dependent geometric information (*e.g.* surface normal and viewing direction) into the 3D volume, which is processed by Transformers and a MLP to exploit useful cues. Details are given in Section 3.1. **(2)** In the multi-view fusion stage, a geometry-guided adaptive feature fusion is proposed. Features, occlusion approximation, and relative pose distances are used to guide view-attention learning. The multi-view volumes are integrated into one volume by learned weights. Section 3.2 outlines this approach. **(3)** In the loss function, the 3D normal calculated from the TSDF contains local information of the TSDF. To encourage the network to generate consistent scenes, a 3D normal loss is added to the output. The normal loss keeps consistency between 2D and 3D normals and ignores boundaries and thin parts. The normal loss is only computed during training. Section 3.3 provides more details about this stage.

3.1. Geometry-guided feature learning

Planar structures are common in indoor scenes, *e.g.* walls and tables. Hence, surface normals provide vital information to determine the relationship between planes. Due to back-projection, voxels along the camera ray correspond to the same 2D features. Thus, depth can add discriminative cues, *e.g.* voxels close to the camera provide more details, while distant voxels contain richer contextual information. Furthermore, the viewing direction corresponds to the orientation of voxels in a camera coordinate frame, which is also related to the amount of camera distortion. Therefore, projected surface normal (back-projected from 2D normal), viewing angle, projected depth (calculated from the voxel

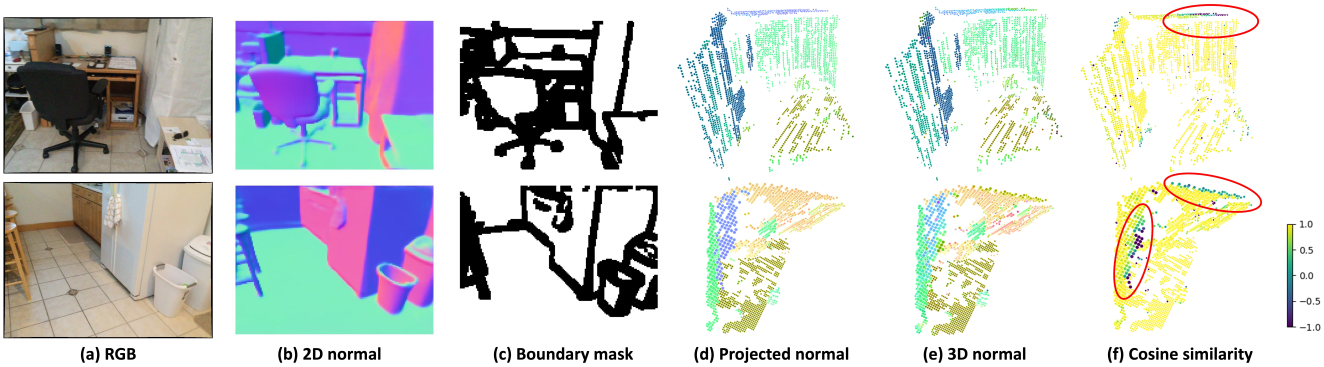


Figure 3. **Boundary and consistency analysis of our proposed 3D normal loss.** (a) RGB images. (b) 2D surface normals predicted by a pre-trained normal network [1]. (c) 2D boundary masks. White regions are planes, which are retained for normal loss computation. (d) Projected normal $\widetilde{\mathbf{N}}$ is the 3D normal back-projected from the 2D normal. (e) 3D normal $\mathbf{N}$ is generated from the ground truth of the TSDF, showing noise near the boundaries. (f) Cosine similarity between (d) and (e). Blue points in the red circle mean that angles between (d) and (e) are greater than 90°.

coordinate by perspective projection), and viewing direction are all informative for indoor scenes. (More details about geometry calculation can be found in the supplementary material). In Figure 2a, these geometric priors are explicitly integrated in the feature learning process.

In our approach, the use of normal and other geometry information is exploited by two separate modules, *i.e.* $\mathcal{T}_1$ and $\mathcal{T}_2$. In $\mathcal{T}_1$, to provide high-frequency information, the viewing direction $\mathbf{v}_i \in \mathbb{R}^{3\times N_v}$ (N_v is the number of voxels) and projected depth $\mathbf{z}_i \in \mathbb{R}^{1\times N_v}$ are encoded similar to NeRF [16]. Then, the encoded priors $\gamma(\mathbf{v}_i) \in \mathbb{R}^{(6L)\times N_v}$ and $\gamma(\mathbf{z}_i) \in \mathbb{R}^{(2L)\times N_v}$ are concatenated using the original viewing direction $\mathbf{v}_i$, projected depth $\mathbf{z}_i$, and viewing angle $\boldsymbol{\theta}_i$. After this, they are processed by a MLP. Next, the processed geometry is concatenated to the 3D volume $\mathbf{V}_i \in \mathbb{R}^{C_v\times N_v}$. Then, a linear layer is applied to reduce the channel dimension generating $\mathbf{g}_i \in \mathbb{R}^{C_v\times N_v}$. Finally, $\mathbf{g}_i$ is used as input of a Transformer to further combine visual and geometric features.

$$\begin{aligned} \mathcal{T}_1 : \mathbf{g}_i &= \text{Linear}([\text{MLP}([\gamma(\mathbf{v}_i), \gamma(\mathbf{z}_i), \mathbf{v}_i, \mathbf{z}_i, \boldsymbol{\theta}_i]), \mathbf{V}_i]) \\ \varphi_i &= \text{Transformer}(\mathbf{g}_i) \end{aligned} \tag{1}$$

where $[\cdot]$ denotes channel-wise concatenation.

In $\mathcal{T}_2$, the projected surface normal $\mathbf{n}_i \in \mathbb{R}^{3\times N_v}$ is concatenated with feature $\varphi_i \in \mathbb{R}^{C_v\times N_v}$. Then, a linear layer is applied to reduce the dimension and to combine normal with previous features. Another Transformer is adopted to integrate the geometry and visual features. In particular, Transformers in $\mathcal{T}_1$ and $\mathcal{T}_2$ are applied in a temporal manner, also exploiting information between multiple views.

$$\mathcal{T}_2 : \phi_i = \text{Transformer}(\text{FCN}([\mathbf{n}_i, \varphi_i])) \tag{2}$$

3.2. Geometry-guided adaptive feature fusion

In back-projection, the occluded 3D voxel may be mapped to an irrelevant pixel (*i.e.* 2D features) adding noise to the feature fusion module. Moreover, although voxels between the camera and the surface are given, empty space regions are useless for the reconstruction task. As a result, projective occupancy is adopted as an approximation of occlusion, which also allows features to include relative depth information. After $\mathcal{T}_2$, the projective occupancy probabilities are predicted by a linear layer and sigmoid function as follows,

$$\mathbf{O}_i = \text{Sigmoid}(\text{Linear}(\phi_i)) \tag{3}$$

Binary cross-entropy loss is applied on $\mathbf{O}_i \in \mathbb{R}^{1\times N_v}$ as an occupancy loss $\mathcal{L}_o$ to supervise the prediction.

The Transformer in $\mathcal{T}_2$ computes an attention matrix $\mathbf{A} \in \mathbb{R}^{T\times T}$ for each voxel, in which each row $\mathbf{A}_i$ represents the relationship between the ith and other views. If the weights in the same row are similar (the row-wise standard deviation is close to 0), this means that the view has the same features as the other views. Conversely, if the row-wise weights are different from each other (the row-wise standard deviation is large), this implies that the view may carry important features. Additionally, if the camera location of the view is distant from the others, the voxel in this view may contain distinctive features, or the voxel may be occluded. Therefore, the occlusion prior, relative pose distance, and the row-wise statistics information (*i.e.* mean and standard deviation) of the attention matrix can provide vital information for determining the best views. To this end, a geometry-guided adaptive feature fusion is designed to generate weights for multiple views. The mean and standard deviation for the attention matrix and relative pose distance are first computed. Then, they are concatenated based on projective occupancy probabilities. A MLP is applied to filter the noise and adaptively learn the weight from the visual features and geometric priors. Finally, a softmax is used to generate the weight:

$$\begin{aligned}\mathbf{w} &= \text{Softmax}(\text{MLP}([\boldsymbol{\mu}^A, \boldsymbol{\sigma}^A, \boldsymbol{\mu}^{rp}, \boldsymbol{\sigma}^{rp}, \mathbf{O}])) \\ \mathbf{V} &= \sum_i \mathbf{w}_i \mathbf{V}_i \end{aligned} \tag{4}$$

where $\boldsymbol{\mu}^A \in \mathbb{R}^{T \times N_v}$ and $\boldsymbol{\sigma}^A \in \mathbb{R}^{T \times N_v}$ represent the row-wise mean and standard deviation of the attention matrix. $\boldsymbol{\mu}^{rp} \in \mathbb{R}^{(3T) \times N_v}$ and $\boldsymbol{\sigma}^{rp} \in \mathbb{R}^{(3T) \times N_v}$ denote the mean and standard deviation of the relative pose distance.

3.3. Consistent 3D normal loss

When the voxel $\mathbf{p} \in \mathbb{R}^3$ is on or close to the surface (TSDF S is 0), the numerical derivative of the TSDF is the 3D surface normal $\mathbf{N}(\mathbf{p}) \in \mathbb{R}^{3 \times 1}$ of the voxel [18]:

$$\mathbf{N}(\mathbf{p}) = \nu[\nabla S(\mathbf{p})], \nabla S(\mathbf{p}) = [\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial z}]^T. \tag{5}$$

where $\nu[\mathbf{x}] = \mathbf{x}/\|\mathbf{x}\|_2$. In our approach, the numerical derivative is implemented by a derivative operator [20]. In this way, the ground truth of the 3D normal $\mathbf{N}_{gt}$ is generated from the ground truth of TSDF by Eq. 5, while the prediction of the 3D normal $\mathbf{N}_{pred}$ corresponds to the predicted TSDF.

Because normals near the boundaries or thin/small objects are usually inaccurate, a boundary mask $\mathbf{M}_{2d} \in \mathbb{R}^{1 \times H_f \times W_f}$ is introduced to filter out these parts and to ensure that the normal loss is only calculated for planar regions. $\mathbf{M}_{2d}$, shown in Figure 3c, is calculated as follows: a 2D edge detector [12] is used to compute gradients at 2D normals. Then pixels with gradient values greater than a threshold are regarded as boundary pixels. Finally, an 8 neighbor of boundary pixels is also regarded as boundary pixels (implemented by a max-pooling layer with kernel size 3 and stride 1). $\mathbf{M}_{2d}$ is back-projected to 3D space, forming the 3D boundary mask $\mathbf{M}_{3d} \in \mathbb{R}^{1 \times N_v}$. After masking the voxels by $\mathbf{M}_{3d}$, other noise sources in the computation of 3D normal may exist, as shown in Figure 3e and 3f. Hence, this paper introduces a consistency between projected normal $\widetilde{\mathbf{N}}(\mathbf{p})$ and 3D normal $\mathbf{N}(\mathbf{p})$ to suppress noise. The projected surface normal $\mathbf{n}_i$, see Section 3.1, is in camera coordinates while $\mathbf{N}(\mathbf{p})$ is in world coordinates. Therefore, $\mathbf{n}_i$ is transformed to world coordinates by $\widetilde{\mathbf{N}}_i(\mathbf{p}) = \mathbf{R}_i \mathbf{n}_i$ and then averaged between views by $\widetilde{\mathbf{N}}(\mathbf{p}) = \frac{1}{T}\sum_{i=1}^{T} \widetilde{\mathbf{N}}_i(\mathbf{p})$. s_{2d3d} in Eq. 6 computes the cosine similarity between $\mathbf{N}(\mathbf{p})$ and $\widetilde{\mathbf{N}}(\mathbf{p})$ to measure consistency.

$$s_{2d3d}(\mathbf{p}) = \frac{\mathbf{N} \cdot \widetilde{\mathbf{N}}}{\|\mathbf{N}\|_2 \|\widetilde{\mathbf{N}}\|_2} \tag{6}$$

The indicator function $[s_{2d3d}(\mathbf{p}) > 0]$ is applied to generate the consistency measure, *i.e.* if the 3D surface normal $\mathbf{N}(\mathbf{p})$ is similar to the projected 3D surface normal $\widetilde{\mathbf{N}}(\mathbf{p})$, the normal loss is computed. Otherwise, the 3D normal is considered as noise. The weight $W_{2d3d}(\mathbf{p})$ is given by

$$W_{2d3d}(\mathbf{p}) = [s_{2d3d}(\mathbf{p}) > 0] \equiv \begin{cases} 1, & s_{2d3d}(\mathbf{p}) > 0 \\ 0, & s_{2d3d}(\mathbf{p}) \leq 0 \end{cases} \tag{7}$$

By excluding boundary voxels and considering consistency between projected and 3D normals, the 3D normal loss is defined by:

$$\mathcal{L}_n = 1 - \frac{1}{N}\sum_{m=1}^{N} M_{3d}^m W_{2d3d}^m \frac{\mathbf{N}_{gt}^m \cdot \mathbf{N}_{pred}^m}{\|\mathbf{N}_{gt}^m\|_2 \|\mathbf{N}_{pred}^m\|_2} \tag{8}$$

where N is the number of voxels used in the normal loss.

Method	Comp↓	Acc↓	Recall↑	Prec↑	F-score↑
COLMAP [24]	0.069	0.135	0.634	0.505	0.558
MVDepthNet[38]	0.040	0.240	0.831	0.208	0.329
GPMVS [10]	**0.031**	0.879	**0.871**	0.188	0.304
DPSNet [11]	0.045	0.284	0.793	0.223	0.344
SimpleRecon [23]	0.078	0.065	0.641	0.581	0.608
Atlas [17]	0.084	0.102	0.598	0.565	0.578
TransformerFusion [2]	0.099	0.078	0.648	0.547	0.591
3DVNet [21]	0.077	0.221	0.506	0.545	0.520
NeuralRecon [30]	0.138	0.051	0.478	0.683	0.560
NeuralRecon + ours	0.099	**0.048**	0.545	**0.722**	0.619
VoRTX [28]	0.108	0.062	0.545	0.666	0.598
VoRTX + Ours	0.098	0.059	0.585	0.687	**0.630**

Table 1. 3D reconstruction mesh evaluation following Atlas [17] for ScanNet. The best results are **bold**, and the second best ones are underlined.

7-Scenes					
Method	Comp↓	Acc↓	Prec↑	Recall↑	F-score↑
DeepV2D [34]	0.180	0.518	0.087	0.175	0.115
CNMNet[14]	**0.150**	0.398	0.111	0.246	0.149
NeuralRecon [30]	0.228	0.100	0.389	0.227	0.282
NeuralRecon + ours	0.289	**0.086**	**0.476**	0.294	**0.359**
VoRTX [28]	0.286	0.103	0.364	0.267	0.304
VoRTX + ours	0.231	0.100	0.381	**0.299**	0.332
TUM RGB-D					
Method	Comp↓	Acc↓	Prec↑	Recall↑	F-score↑
Atlas [17]	2.344	0.208	0.360	0.089	0.132
NeuralRecon [30]	1.341	0.092	**0.564**	0.155	0.232
NeuralRecon + ours	0.851	**0.087**	0.517	0.175	0.256
VoRTX [28]	0.911	0.136	0.434	0.203	0.268
VoRTX + ours	**0.722**	0.128	0.445	**0.217**	**0.284**

Table 2. 3D reconstruction mesh evaluation following Atlas [17] on the 7-Scenes and TUM RGB-D datasets.

4. Experiments

4.1. Datasets and metrics

Our method is evaluated on three challenging indoor *RGB-D* datasets, *i.e.* ScanNet(V2) [6], 7-Scenes [27], and TUM RGB-D [29] datasets. ScanNet consists of 807 unique indoor scenes, which is composed of 1613 scans (1201 for training, 312 for validation, and 100 for testing). Our method is trained on the training set of ScanNet. To validate the generalization, the method is also tested on 7-Scenes and TUM RGB-D datasets without fine-tuning. The ground-truth meshes for 7-Scenes and TUM RGB-D are produced by TSDF fusion with a voxel size of 4cm.

For quantitative comparison, 3D geometry metrics defined by [17] are adopted to measure the quality of 3D reconstruction, including accuracy (acc), completeness (comp), precision (prec), recall, and F-score. F-score is considered the most reliable metric. The computation of each metric is detailed in the supplementary material.

4.2. Implementation details

The online method NeuralRecon and the offline method VoRTX are chosen as our baselines. The number of heads in the Transformer is 2. The weights for occupancy loss, TSDF loss, projective occupancy loss, and 3D normal loss are $\{1.5, 1.0, 0.5, 0.1\}$. At the start of training, predicted TSDF may be inaccurate, causing a high 3D normal loss. Hence, the 3D normal loss is added after 5 epochs. The batch size per GPU is 4. Other settings (*e.g.* view selection and voxel size) are similar to baselines. The network is trained on three NVIDIA RTX A6000 GPUs. The 2D surface normal is predicted by the pre-trained model in [1].

4.3. Evaluation results

ScanNet [6]. Comparison between our version and other SOTA methods is shown in Table 1. When contrasted with NeuralRecon and VoRTX, both enhanced with our approach, NeuralRecon + our method and VoRTX + our method exhibit better performance across all 3D metrics. For example, F-score, precision, recall of NeuralRecon + ours are 5.9%, 3.9%, 6.7% higher than NeuralRecon. This is because our geometry integration mechanism adds more information to the voxels. VoRTX + ours and NeuralRecon + ours outperform SOTA methods in precision and F-score. In particular, compared to the depth-based method SimpleRecon, NeuralRecon + ours shows strong accuracy (26.2% decrease) and precision (14.1% increase) performances. VoRTX + ours outperforms the volumetric method TransformerFusion by 14.0% in precision, 24.4% in accuracy, and 3.9% in F-score. NeuralRecon + ours also outperforms VoRTX on almost all metrics. Qualitative results are presented in Figure 4. It is shown that NeuralRecon falls short in a number of regions (*e.g.* floors). VoRTX generates over-smoothed and inaccurate surfaces and has problems yielding the correct geometry for planar surfaces (*e.g.* walls). Due to pixel prediction, SimpleRecon is able to produce more voxels than volumetric methods. However, some meshes generated by SimpleRecon are uneven, caused by depth inconsistency. In contrast, our geometry integration mechanism (the 4th column) shows an improvement in reconstruction quality, *i.e*. recovering more surfaces (*e.g.* textureless regions), yielding smoother and more accurate meshes, and providing proper geometry relationships (*e.g.* perpendicular connections between adjacent walls). More qualitative results can be found in the supplementary material.

		Prec ↑	Recall ↑	F-score↑
a	NeuralRecon	0.683	0.478	0.560
b	+ Trans.	0.678	0.488	0.566
c	+ Trans. + norm.	0.691	0.513	0.587
d	+ Trans. + norm. + view. (same)	0.686	0.520	0.590
e	+ Trans. + norm. + view.	**0.709**	0.521	0.598
f	+ Trans. + norm. + view. + depth	0.701	**0.530**	**0.602**
g	+ geo. (SimpleRecon)	0.678	0.496	0.571

Table 3. Ablation study for geometry-guided feature learning. *Trans.* and *norm.* denote the Transformer and projected surface normals. *view.* is the viewing direction and viewing angle. *geo.(SimpleRecon)* refers to the geometry used in SimpleRecon.

7-Scenes [27] and TUM RGB-D [29]. Table 2 shows the results on 7-Scenes and TUM RGB-D datasets. Although no fine-tuning is applied to these two datasets, the method using our geometry integration mechanism demonstrates an improvement in performance. On 7-Scenes, the F-score of NeuralRecon + ours is better than the other methods, For instance, there are improvements of 7.7% and 5.3% when compared to NeuralRecon and VoRTX, respectively. In the case of TUM RGB-D dataset, within the VoRTX framework, incorporating our method results in a 1.6% increase in F-score and a 1.4% increase in recall. Qualitative comparisons on the 7-Scenes and TUM RGB-D datasets are provided in the supplementary material.

Efficiency. The average running time during forward propagation is shown in Table 6. Depth-based methods focus on a single key-frame, while volumetric methods run on several key-frames at the same time. Thus, for fairness, only volumetric methods are compared in Table 6.

Like [30], the reconstruction time of a local fragment is divided by the number of keyframes. The methods in Table 6 are tested on an NVIDIA RTX A6000 GPU and use the same number of key-frames, *i.e*. 9. It can be derived that our geometry integration mechanism increases the reconstruction performance at the cost of speed. However, the inference costs are comparable. Additionally, NeuralRecon + ours is faster than VoRTX.

4.4. Ablation study

In this section, based on NeuralRecon, an ablation study is conducted to assess the effectiveness of our geometry-guided feature learning, geometry-guided adaptive feature fusion, and consistent 3D normal loss on ScanNet.

Geometry-guided feature learning. Table 3 provides the

		Prec↑	Recall↑	F-score↑
a	NeuralRecon + G2FL	0.701	0.530	0.602
b	+ weight μ/σ	0.703	0.534	0.605
c	+ weight μ/σ + rp μ/σ	0.704	0.541	0.610
d	+ weight μ/σ + rp μ/σ + proj. tsdf.	0.707	0.539	0.609
e	+ weight μ/σ + rp μ/σ + vis.	0.711	0.541	0.612
f	+ weight μ/σ + rp μ/σ + proj. occ.	**0.713**	**0.542**	**0.614**

Table 4. Ablation study for geometry-guided adaptive feature fusion. *weight* μ */* σ and *rp* μ */* σ are the mean and standard deviation of attention weight and relative pose distance. *proj. tsdf*, *vis.*, *proj. occ.* are projective TSDF, visibility, projective occupancy.

ablation study for geometry-guided feature learning. In rows a and b, it's evident that the Transformer blocks enhance both recall and F-score. Moving to row c, the incorporation of projected surface normals leads to a 2.1% increase in F-score. A comparison between rows *c* and *e* highlights the influence of viewing angle and direction, contributing to heightened precision, recall, and F-score. Furthermore, row *d* presents outcomes from combining normal and viewing priors within the same Transformer. In contrast to row *e* where priors are distributed across different modules, row *d* indicates an insufficient exploration of geometric information. Projected depth in row f results in a 0.9% improvement in recall and a 0.4% improvement in F-score. Row g shows the results with geometry used in SimpleRecon, which is worse than ours in row *f*. Finally, compared to NeuralRecon in row a, our geometry-guided feature learning in row f improves the performance, with an increase in recall by 5.2% and F-score by 4.2%. This is attributed to not only the Transformer blocks, but also the geometric priors.

Geometry-guided adaptive feature fusion. Ablation experiments for geometry-guided adaptive feature fusion are presented in Table 4, which are built on Table 3f, *i.e.*, $NeuralRecon + G2FL$. In Table 4, rows *b-f* with adaptively learned weights all outperform $NeuralRecon + G2FL$. Although the Transformer in G2FL can learn attention for multiple views, only using the attention weight $\mathbf{A}$ as a weight guidance in row b provides 0.3% improvement in F-score. Furthermore, the relative pose distance in row c is able to increase the F-score by another 0.5%. The *d-f* rows explore different representations (*i.e.* projective TSDF, visibility, and projective occupancy) as approximations to occlusion. In row d, F-score decreases slightly. The reason is that the prediction of projective TSDF is a regression task, and the network has difficulty optimizing it. Projective occupancy reaches a better performance than other approximations, not only because it can be used to measure the occlusion but also because the representation is close to the reconstruction task. Compared to $NeuralRecon + G2FL$, the performance of our geometry-guided feature fusion in the last row is increased by 1.2% in precision, recall, and F-score, which shows the effectiveness of our fusion module.

		Prec↑	Recall↑	F-score↑
a	NeuralRecon + G2FL + G2AFF	0.713	0.542	0.614
b	+ normal loss (w/o weight)	0.699	0.542	0.609
c	+ normal loss (w/o consist. weight)	0.708	0.543	0.613
d	+ normal loss (w/o boundary mask)	0.698	0.547	0.611
e	+ normal loss (Gaussian weight)	0.705	**0.549**	0.615
f	Ours	**0.722**	0.545	**0.619**

Table 5. Ablation study for consistent 3D normal loss.

Method	Time ↓	F-score ↑
NeuralRecon	**27**	0.560
NeuralRecon + ours	35	0.619
VoRTX	37	0.598
VoRTX + Ours	41	**0.630**

Table 6. Comparison of average running time in milliseconds per keyframe.

Consistent 3D normal loss. The results in Table 5 validate the effectiveness of our consistent 3D normal loss, which is based on Table 4f, *i.e.* $NeuralRecon + G2FL + G2AFF$. Experiments of rows *b-d* are conducted to show the importance of consistency weighting and boundary masking. Row *b* shows the results for 3D normal loss without consistency weights and boundary masks, which is worse than row a. This means that the normal loss should not be applied to all voxels. Compared to rows *b* and *d*, rows *c* and *f* present an increase in F-score. This indicates that the boundary mask is useful in our 3D normal loss. The recall and F-score in row *d* are 0.5% and 0.2% higher than in row *b*. The same trends can also be observed for rows *c* and *f*, which shows that consistent weights play an important role in our 3D normal loss. Row *e* replaces the indicator function in Eq. 7 by a Gaussian function. the results of our indicator function are better than row *e*. The comparison between rows f and a demonstrates that our consistent normal loss gives a better reconstruction performance.

Qualitative comparison. Visualization results of the ablation study are presented in Figure 5. Unlike NeuralRecon, our suggested integration of geometry (refer to columns 2-4) plays a significant role in recovering regions, establishing coherent planes and accurate interrelationships among walls. This underscores the significance of incorporating 3D geometry at various stages.

5. Conclusion

In this paper, a novel geometry integration mechanism is presented to explore 3D geometry in indoor scene reconstruction. The key contribution is to encode geometric priors at three levels, *i.e.* feature learning, feature fusion, and network supervision. Geometry-guided feature learning is proposed to integrate view-dependent geometry into the multi-view visual features, enhancing the reconstruction features. The proposed geometry-guided adaptive feature

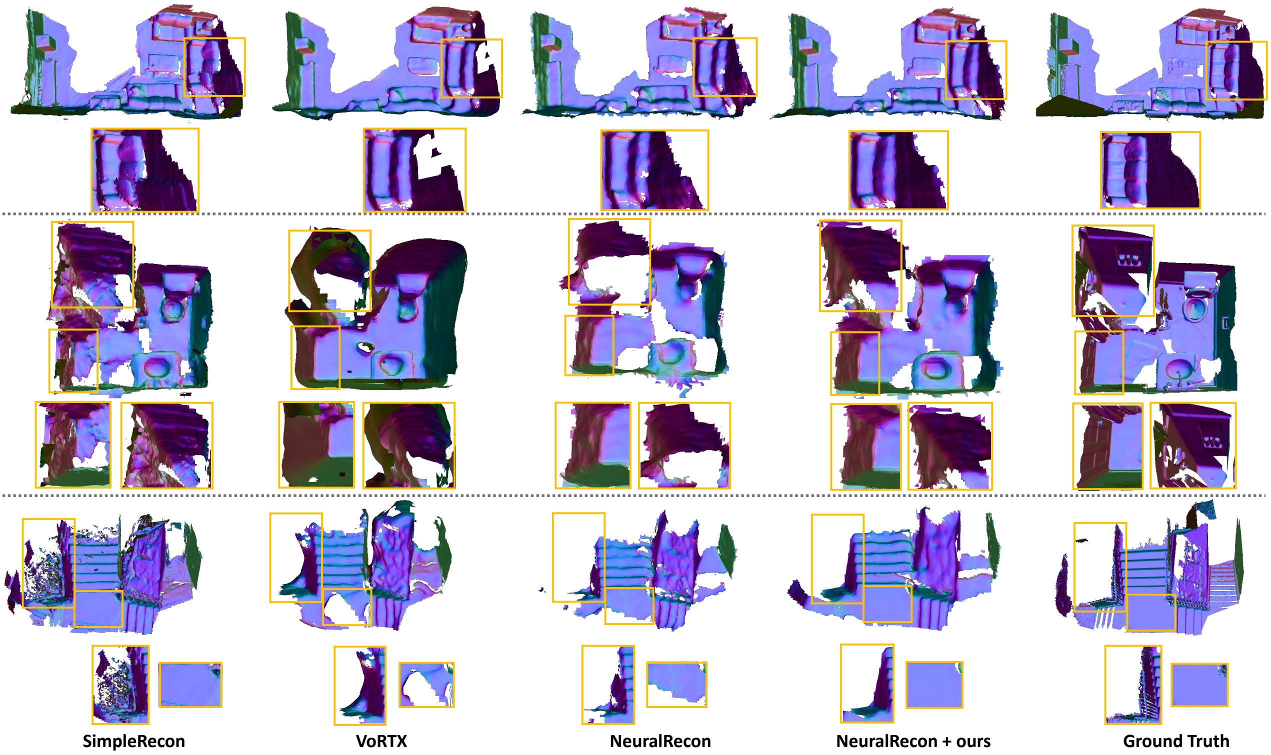


Figure 4. **Qualitative results on ScanNet.** Colors on the meshes are related to surface normals. Compared to other methods, NeuralRecon + ours is able to generate more regions, smoother planes, and more accurate geometry relationships.

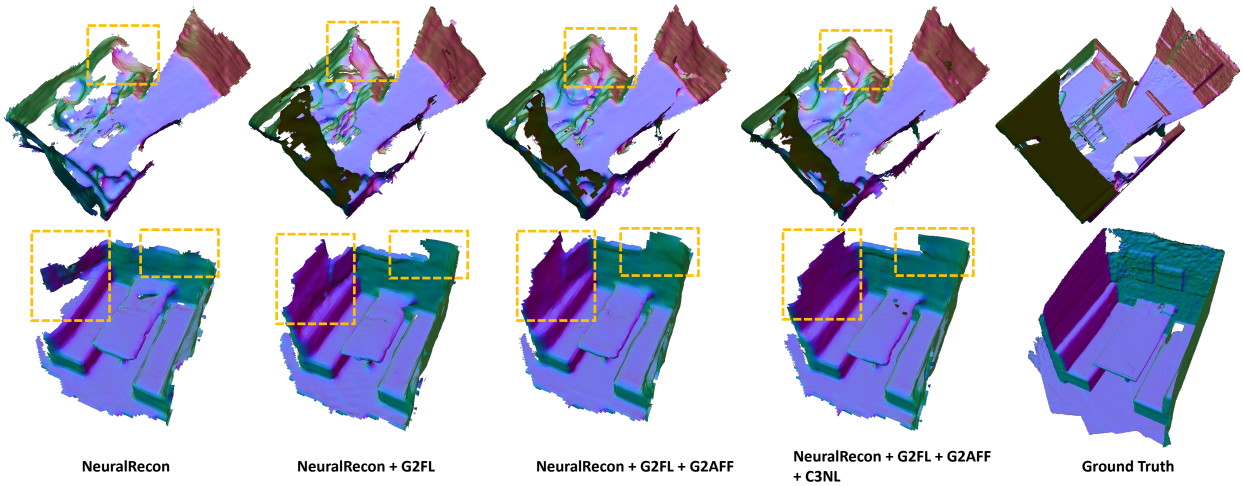


Figure 5. **Visualization comparison of the ablation study.** Our proposed geometry-guided feature learning (G2FL), geometry-guided adaptive feature fusion (G2AFF), and consistent 3D normal loss (C3NL) all contribute to an improved reconstruction quality.

fusion adopts geometry as guidance to model the weight function for multiple views. A consistent 3D normal loss is designed to add local supervision, considering only planar regions and consistency between 2D and 3D normals.

Large-scale experiments are conducted on the ScanNet dataset, showing that our method outperforms state-of-the-art methods quantitatively as well as qualitatively. Volumetric methods with our geometry integration mechanism also show good generalization on 7-Scenes and TUM RGB-D.

Acknowledgment. The authors deeply appreciate the helpful suggestions on the manuscript from Haochen Wang, Qi Bi, Yongtuo Liu, and Yunlu Chen. The authors also would like to thank all reviewers for their constructive feedback.

References

[1] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13137–13146, 2021.

[2] Aljaz Bozic, Pablo Palafox, Justus Thies, Angela Dai, and Matthias Nießner. Transformerfusion: Monocular rgb scene reconstruction using transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[4] Robert T Collins. A space-sweep approach to true multi-image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 358–363, 1996.

[5] 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*, pages 303–312, 1996.

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[7] Arda Duzceker, Silvano Galliani, Christoph Vogel, Pablo Speciale, Mihai Dusmanu, and Marc Pollefeys. Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15324–15333, 2021.

[8] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] Yuxin Hou, Juho Kannala, and Arno Solin. Multi-view stereo by temporal nonparametric fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2651–2660, 2019.

[11] Sunghoon Im, Hae-Gon Jeon, Stephen Lin, and In So Kweon. Dpsnet: End-to-end deep plane sweep stereo. *arXiv preprint arXiv:1905.00538*, 2019.

[12] FG Irwin et al. An isotropic 3x3 image gradient operator. *Presentation at Stanford AI Project*, 2014(02), 1968.

[13] Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10986–10995, 2019.

[14] Xiaoxiao Long, Lingjie Liu, Christian Theobalt, and Wenping Wang. Occlusion-aware depth estimation with adaptive normal constraints. In *Proceedings of European Conference on Computer Vision*, pages 640–657, 2020.

[15] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4):163–169, 1987.

[16] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proceedings of European Conference on Computer Vision*, pages 405–421, 2020.

[17] 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 *Proceedings of European Conference on Computer Vision*, pages 414–431, 2020.

[18] 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 *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011.

[19] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proceedings of European Conference on Computer Vision*, pages 523–540, 2020.

[20] Judith MS Prewitt et al. Object enhancement and extraction. *Picture Processing and Psychopictorics*, 10(1):15–19, 1970.

[21] Alexander Rich, Noah Stier, Pradeep Sen, and Tobias Höllerer. 3dvnet: Multi-view depth prediction and volumetric refinement. In *2021 International Conference on 3D Vision*, pages 700–709, 2021.

[22] Damien Robert, Bruno Vallet, and Loic Landrieu. Learning multi-view aggregation in the wild for large-scale 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5575–5584, 2022.

[23] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. Simplerecon: 3d reconstruction without 3d convolutions. In *Proceedings of European Conference on Computer Vision*, pages 1–19, 2022.

[24] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of European Conference on Computer Vision*, pages 501–518, 2016.

[25] Matthias Seeger. Gaussian processes for machine learning. *International Journal of Neural Systems*, 14(02):69–106, 2004.

[26] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, 28, 2015.

[27] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[28] Noah Stier, Alexander Rich, Pradeep Sen, and Tobias Höllerer. Vortx: Volumetric 3d reconstruction with transformers for voxelwise view selection and fusion. In *2021 International Conference on 3D Vision*, pages 320–330, 2021.

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the International Conference on Intelligent Robot Systems*, pages 573–580, 2012.

[30] Jiaming Sun, Yiming Xie, Linghao Chen, Xiaowei Zhou, and Hujun Bao. Neuralrecon: Real-time coherent 3d reconstruction from monocular video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15598–15607, 2021.

[31] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019.

[32] 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 *Proceedings of European Conference on Computer Vision*, pages 685–702, 2020.

[33] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8489–8499, 2022.

[34] Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. *arXiv preprint arXiv:1812.04605*, 2018.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[36] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022.

[37] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *arXiv preprint arXiv:2210.04264*, 2022.

[38] Kaixuan Wang and Shaojie Shen. Mvdepthnet: Real-time multiview depth estimation neural network. In *2018 International Conference on 3D Vision*, pages 248–257, 2018.

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 568–578, 2021.

[40] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12186–12195, 2022.

[41] Bo Yang, Sen Wang, Andrew Markham, and Niki Trigoni. Robust attentional aggregation of deep feature sets for multi-view 3d reconstruction. *International Journal of Computer Vision*, 128(1):53–73, 2020.

[42] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision*, pages 767–783, 2018.

[43] Yao Yao, Zixin Luo, Shiwei Li, Tianwei Shen, Tian Fang, and Long Quan. Recurrent mvsnet for high-resolution multi-view stereo depth inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5525–5534, 2019.

[44] Hongwei Yi, Zizhuang Wei, Mingyu Ding, Runze Zhang, Yisong Chen, Guoping Wang, and Yu-Wing Tai. Pyramid multi-view stereo net with self-adaptive view aggregation. In *Proceedings of European Conference on Computer Vision*, pages 766–782, 2020.

[45] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13577–13587, 2021.