PlaneMVS: 3D Plane Reconstruction from Multi-View Stereo

Jiachen Liu[1,2*] Pan Ji[2†] Nitin Bansal[2] Changjiang Cai[2] Qingan Yan[2] Xiaolei Huang[1] Yi Xu[2]
[1]The Pennsylvania State University [2]OPPO US Research Center, InnoPeak Technology, Inc.

Abstract

We present a novel framework named PlaneMVS for 3D plane reconstruction from multiple input views with known camera poses. Most previous learning-based plane reconstruction methods reconstruct 3D planes from single images, which highly rely on single-view regression and suffer from depth scale ambiguity. In contrast, we reconstruct 3D planes with a multi-view-stereo (MVS) pipeline that takes advantage of multi-view geometry. We decouple plane reconstruction into a semantic plane detection branch and a plane MVS branch. The semantic plane detection branch is based on a single-view plane detection framework but with differences. The plane MVS branch adopts a set of slanted plane hypotheses to replace conventional depth hypotheses to perform plane sweeping strategy and finally learns pixel-level plane parameters and its planar depth map. We present how the two branches are learned in a balanced way, and propose a soft-pooling loss to associate the outputs of the two branches and make them benefit from each other. Extensive experiments on various indoor datasets show that PlaneMVS significantly outperforms state-of-the-art (SOTA) single-view plane reconstruction methods on both plane detection and 3D geometry metrics. Our method even outperforms a set of SOTA learning-based MVS methods thanks to the learned plane priors. To the best of our knowledge, this is the first work on 3D plane reconstruction within an end-to-end MVS framework. Source code: https://github.com/oppo-us-research/PlaneMVS.

1. Introduction

3D planar structure reconstruction from RGB images has been an important yet challenging problem in computer vision for decades. It aims to detect piece-wise planar regions and predict the corresponding 3D plane parameters from RGB images. The recovered 3D planes can be used in various applications such as robotics [46], Augmented Reality (AR) [4], and indoor scene understanding [51].

*Work primarily done while interning at OPPO US Research Center
†Corresponding author (peterji530@gmail.com)

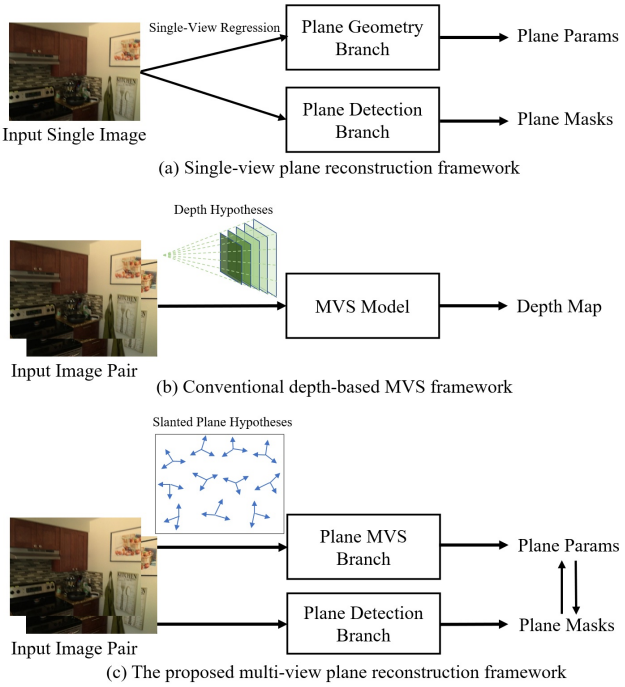

Figure 1. Comparison among: (a) single-view plane reconstruction framework, (b) conventional depth-based MVS framework, (c) the proposed multi-view plane reconstruction framework. Our system employs slanted plane hypotheses for plane-sweeping to build the plane MVS branch, which interacts with the plane detection branch. The two branches can benefit from each other.

Traditional methods [9, 13, 43] work well in certain cases but usually highly rely on some assumptions (*e.g.*, Manhattan-world assumption [9]) of the target scene and are thus not always robust in complicated real-world cases. Recently, some methods [31, 32, 47, 56, 61] have been proposed to recover planes from single-view images based on Convolutional Neural Networks (CNNs). These methods could reconstruct 3D planes better in terms of completeness and robustness compared with traditional methods. However, all of them, albeit achieving reasonable results on 2D plane segmentation, attempt to recover 3D plane geometry from a single image, which is an ill-posed problem as it only relies on single-view regression for plane parameters and has ambiguity in depth scale recovery. Thus the recovered 3D planes from those methods are far from being accurate.

The limitations of these methods motivate us to consider the possibility of reconstructing 3D planes from multiple views with CNNs in an end-to-end framework.

In contrast to reconstructing 3D geometry from single images, multi-view-stereo (MVS) [10] takes multiple images as input with known relative camera poses. MVS methods achieve superior performance on 3D reconstruction compared with single-view methods since the scale of a scene can be resolved by triangulating matched feature points on calibrated images [17]. Recently, a few learning-based MVS methods [16, 19, 57, 58] have been proposed and have achieved promising improvements for a wide range of scenes. While effective in reconstructing areas with rich textures, their pipelines would suffer from ambiguity in finding feature matches in the textureless area, which often belongs to planar regions. Besides, the generated depth map usually lacks smoothness as planar structures are not explicitly parsed. Some recent MVS approaches [26, 34, 63] propose to jointly learn the geometric relationship between depth and normal to capture local planarity. However, these methods usually estimate depth and normal separately, and only learn pixel-level planarity by enforcing the constraints by additional losses. Piece-wise planar structures, *e.g.*, walls and floors, which usually indicate strong global geometric smoothness, are not well captured in these approaches.

In this work, as shown in Fig. 1, we take advantage of both sets of methods and propose to reconstruct planar structures in an MVS framework. Our framework consists of dual branches: a plane detection branch and a plane MVS branch. The plane detection branch predicts a set of 2D plane masks with their corresponding semantic labels of the target image. The plane MVS branch, which is our key contribution, takes posed target and source images as input. Inspired by the frontal plane sweeping formulation that is widely used in MVS pipelines, we propose a *slanted* plane sweeping strategy to learn the plane parameters without ambiguity. Specifically, instead of using a set of frontal plane hypotheses (*i.e.*, depth hypotheses) for plane sweeping as in conventional MVS methods, we perform plane sweeping with a group of slanted plane hypotheses to build a plane cost volume for per-pixel plane parameter regression.

To associate the two branches, we present a soft-pooling strategy to get piece-wise plane parameters and propose a loss objective based on it to make the two branches benefit from each other. We apply learned uncertainties [25] on different loss terms to train the multi-task learning system in a balanced way. Moreover, our system can generalize well in new environments with different data distributions. The results can be further improved with a simple but effective finetuning strategy without groundtruth plane annotations.

To the best of our knowledge, this is the first work that reconstructs planar structures in an end-to-end MVS framework. The reconstructed depth map takes advantage of multi-view geometry to resolve the scale ambiguity issue. It is much smoother geometrically compared with depth-based MVS schemes by parsing planar structures. Experimental results across different indoor datasets demonstrate that our proposed PlaneMVS not only significantly outperforms single-view plane reconstruction methods, but is also better than several SOTA learning-based MVS approaches.

2. Related Work

Piece-wise planar reconstruction. Traditional plane reconstruction methods [9,13,43] usually take a single or multiple images as input and detect the primitives such as vanishing points and lines as geometric cues to recover planar structures. Such methods make strong assumptions about the environment and often do not generalize well into various scenarios. Recent learning-based approaches [31, 32, 39,47,56,61] handle the plane reconstruction problem from a single image with Deep Neural Networks (DNNs) and achieve promising results. PlaneNet [32] proposes a multi-branch network to learn plane masks and parameters jointly. PlaneRecover [56] proposes to segment piece-wise planes with only groundtruth depth supervision but without any plane groundtruth. PlaneAE [61] and PlaneTR [31] learn to cluster image pixels into piece-wise planes with bottom-up frameworks. Alternatively, PlaneRCNN [31] takes advantage of a two-stage detection framework [18] to estimate plane segmentation and plane geometry in several parallel branches. Qian and Furukawa [39] model the inter-plane relationships to further refine the initial planar reconstruction. However, although being possible to learn 2D plane segmentation with a single image, it is still challenging to learn accurate 3D plane geometry only with single-view regression. Most recently, Jin *et al.* [23] proposes a framework to jointly reconstruct planes and estimate camera poses from sparse views. In our work, we assume the camera poses are obtained from some SLAM systems, and design our plane detection branch based on PlaneRCNN [31], but learn plane geometry in a separate multi-view-stereo (MVS) branch.

Multi-view stereo. Different from single-view depth estimation [7, 15, 22, 49, 64], multi-view stereo transforms the depth estimation problem into triangulating corresponding points from a pair of posed images. Thus, it could solve the scale ambiguity issue in the single-view case. Traditional MVS approaches can be roughly categorized as voxel-based methods [27, 41], point-cloud-based methods [10, 28] and depth-map-based methods [3, 11, 50].

In recent years, some learning-based methods have been proposed and have shown superior robustness and generalizability. Volumetric methods such as [21, 24] aggregate multi-view information to learn a voxel representation of the scene. However, they can only be applied to small-sized

scenes due to high memory consumption of volumetric representation. For depth-based methods, MVSNet [58] utilizes an end-to-end framework to reconstruct the depth map of the reference image from multi-view input based on the plane-sweeping strategy. Some follow-up methods aim to achieve better accuracy-speed trade-off [52, 59, 60] or refine the depth map in a cascaded framework [16, 57], or incorporate visibility as well as uncertainty into the framework [35, 55, 62]. These depth map-based MVS approaches usually apply the fronto-parallel plane hypothesis for plane sweeping, aiming to learn pixel-level feature correspondences at correct depths. However, for textureless areas or repetitive patterns, it is challenging for the network to accurately match pixel-level features, thus making the inferred depth less accurate. Different from depth-map based MVS, Atlas [37] and NeuralRecon [45] propose to learn a TSDF [5] representation from posed images for 3D surface reconstruction which avoids multi-view depth fusion.

Due to the matching ambiguity in textureless areas, some MVS works [1, 2, 12] aim to model local planarity since textureless areas are usually planar. Traditionally, Birchfield and Tomasi [1] introduce slanted-plane with Markov Random Fields for stereo matching. Gallup *et al.* [12] first estimate dominant plane directions and warp along those planes based on plane sweeping. A few methods [2, 40, 54] perform stereo patch matching in textureless regions based on iterative optimizations or probabilistic frameworks. For learning-based methods, derived from the idea of patchmatch stereo, a line of works [26, 34, 63] incorporate the geometric relationship between depth and surface normal into MVS framework. Although sharing high-level ideas, our work differs from these methods in several aspects. First, some work [34] segments piece-wise planes as an offline pre-processing step to generate smooth and consistent normals. However, we jointly learn plane segmentation and plane geometry within the proposed framework. Second, they usually learn depth and normal separately and apply loss objectives as extra constraints based on local planarity. In contrast, we directly learn to regress pixel-level plane parameters with a set of slanted plane hypotheses with the plane-sweeping strategy in one MVS pipeline, so the joint relationship between depth and normal is learned implicitly. Third, while those works aim to employ planar priors to assist multi-view stereo, our goal is to reconstruct piece-wise planar structures with an MVS framework.

3. Method

This section is organized as follows: we first introduce our semantic plane detection branch in Sec. 3.1, and present our plane MVS branch in Sec. 3.2. Then we describe the piece-wise plane reconstruction process in Sec. 3.3. Finally, we introduce our loss objectives in Sec. 3.4.

3.1. Plane detection

An overview of PlaneRCNN. PlaneRCNN [31] is one of the state-of-the-art single-view plane reconstruction approaches, which builds upon Mask-RCNN [18]. It designs several separate branches for estimating 2D plane masks and 3D plane geometry. It first applies FPN [29] to extract a feature map, then adopts a two-stage detection framework to predict 2D plane masks $\mathcal{M}$. An encoder-decoder architecture processes the feature map to get a per-pixel depth map $\mathbf{D}$. Instance-level plane features from ROI-Align [18] are passed into a plane normal branch to predict plane normals $\mathcal{N}$. They also design a refinement network to refine initial plane masks and a reprojection loss between neighboring views to enforce multi-view geometry consistency during training. With predicted $\mathcal{M}, \mathcal{N}$ and $\mathbf{D}$, the piecewise planar depth map $\mathbf{D}_p$ can be reconstructed.

Our semantic plane detection. Our detection head is based on PlaneRCNN [31] but with several modifications. First, we remove all the geometry estimation modules including the plane normal prediction module and the monocular depth estimation module since 3D plane geometry is estimated by our MVS branch. Second, we also remove the plane refinement module and the multi-view reprojection loss used in PlaneRCNN to conserve memory. Additionally, since semantic information is helpful for scene understanding, as in Mask-RCNN [18], we add semantic label prediction for each plane instance to get its semantic class. We will introduce the details on how we define and generate the semantic plane annotations in Sec. 4.2. To summarize, for an input image with resolution $H \times W$, our plane detection head predicts a set of plane bounding boxes $\mathcal{B} = \{b_1, b_2, ..., b_k\}$, their confidence scores $\mathcal{S} = \{s_1, s_2, ..., s_k\}$ where $s_i \in (0, 1)$, the binary plane masks $\mathcal{M} = \{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_k\}$ where $\mathbf{m}_i \in \mathbb{R}^{H \times W}$, and their corresponding semantic labels $\mathcal{C} = \{c_1, c_2, ..., c_k\}$.

3.2. Planar MVS

Next, we introduce our plane MVS head, which is our key contribution in this work. Fig. 2 shows the architecture of this branch, and we will present each part sequentially.

Feature extraction. The 2D image feature extraction for the MVS head is shared with the plane detection head. Specifically, we obtain multi-scale 2D feature maps with $L = 5$ levels from the FPN feature backbone. Here we only utilize the finest level feature $f_0 \in \mathbb{R}^{\frac{1}{4}H \times \frac{1}{4}W \times C}$. To further balance the memory consumption and accuracy, we pass f_0 into a dimension-deduction layer and an average pooling layer to get reduced feature representation $f_0' \in \mathbb{R}^{\frac{1}{8}H \times \frac{1}{8}W \times \frac{1}{2}C}$. f_0' serves as the feature map input of the MVS network. It is worth exploring whether using multiple levels of features would bring benefits, but that is not our current focus, and we leave it to future work.

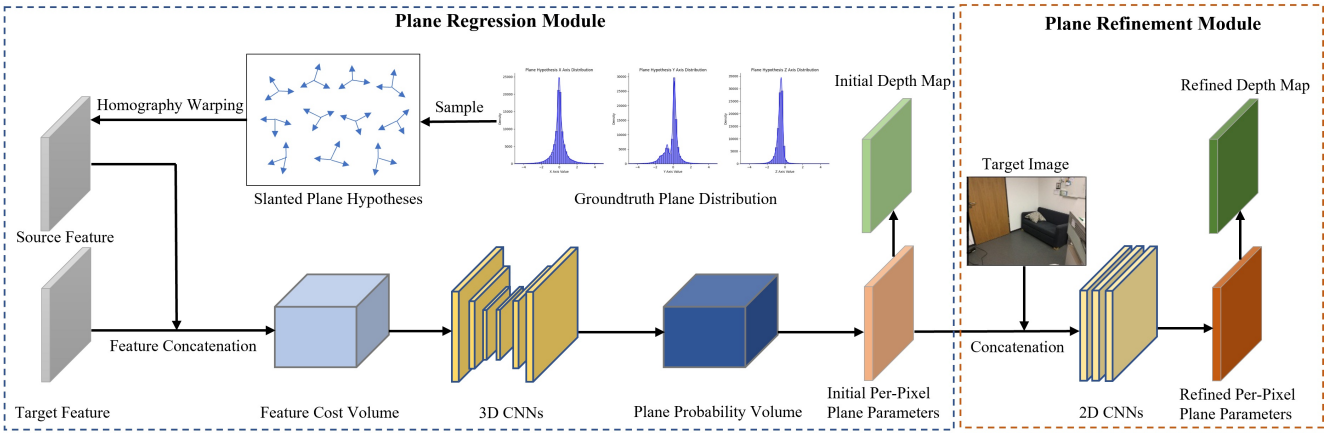

Figure 2. The architecture of our proposed plane MVS head. It consists of a plane regression module to regress initial pixel-level plane parameters and a plane refinement module to refine the initial prediction. The key difference of our work from conventional MVS methods is that we apply slanted plane hypotheses for homography warping. The loss objectives during training are omitted for simplicity.

Differentiable planar homography. Previous MVS methods [19, 58] propose to warp the source feature with fronto-parallel planes, *i.e.*, depth hypotheses, to the target view. This is effective in associating the features from multiple views at the correct depth values of the target view. In our setup, the objective is to learn per-pixel plane parameters instead of depth. To this end, we propose to leverage slanted plane hypotheses for performing plane sweeping to learn per-pixel plane parameters with the MVS framework. The representation of differentiable homography using slanted plane hypotheses is the same as using depth hypotheses. The homography between two views at plane $(\boldsymbol{n_i})^T\mathbf{x} + e_i = 0$, where $\boldsymbol{n_i}$ is the plane normal and e_i is the offset at pixel i of the target view, can be represented as:

$$H_i(\boldsymbol{n_i}, e_i) \sim \boldsymbol{K}(\boldsymbol{R} - \frac{\boldsymbol{t}\boldsymbol{n_i}^T}{e_i})\boldsymbol{K^{-1}} \ , \tag{1}$$

where symbol $\sim$ means "equality up to a scale". $\boldsymbol{K}$ is the intrinsic matrix. $\boldsymbol{R}$ and $\boldsymbol{t}$ are the relative camera rotation and translation matrices between two views, respectively. Therefore it can be concluded that, without considering occlusion and object motion, the homography at pixel i between two views is only determined by the plane $\boldsymbol{p_i} = \boldsymbol{n_i}^T/e_i$ with known camera poses. This perfectly aligns with our goal to learn 3D plane parameters with MVS. We can learn pixel-level plane parameters $\boldsymbol{p_i} = \boldsymbol{n_i}^T/e_i$, which is a non-ambiguous representation for a plane by employing slanted plane-sweeping in an MVS framework.

Slanted plane hypothesis generation. One of the main differences of our framework from conventional MVS methods lies in the hypothesis representation. In depth-based MVS pipelines, their plane hypotheses are fronto-parallel w.r.t. the camera. Therefore, a set of one-dimensional depth hypotheses which cover the depth range of the target 3D space are sufficient for depth regression. However, in our work, we need a set of three-dimensional slanted plane hypotheses $\boldsymbol{n}^T/e$. Finding slanted plane hypotheses is a non-trivial task since the number of candidate planes that pass through a 3D point is infinite. We need to determine the appropriate hypothesis range for each dimension of $\boldsymbol{n}^T/e$. To this end, we randomly sample $10,000$ training images and plot the distribution for every axis of groundtruth plane $\boldsymbol{n}^T/e$, which reflects the general distribution for plane parameters in various scenes. Then we select the upper and lower bounds for each axis by ensuring most groundtruth values lie within the selected range. We sample the hypotheses uniformly between the bounds along every axis. Please see details of the hypothesis range and number we choose in our supplementary material.

Cost volume construction. After determining plane hypotheses, we warp the source feature map into the target view by Eq. (1). For every slanted plane hypothesis, we concatenate the warped source feature and target feature to associate them, which can better keep the original single-view feature representation than applying distance metrics [58]. Then we stack the features along the hypothesis dimension to build a feature cost volume C. Following [58], we utilize an encoder-decoder architecture with 3D CNN layers to regularize the feature cost volume. Finally, we use a single 3D CNN layer with softmax activation to transform the cost volume C into a plane probability volume U.

Per-pixel plane parameters inference and refinement. To make the whole system differentiable, following [58], soft-argmax is applied to get the initial pixel-level plane parameters. Given the plane hypothesis set $\mathcal{P}_h = \{\boldsymbol{p_0}, \boldsymbol{p_1}, ..., \boldsymbol{p_{N-1}}\}$, 3D plane parameter p_i at pixel i can be inferred as:

$$\boldsymbol{p_i} = \sum_{j=0}^{N-1} \boldsymbol{p_j} \cdot U(\boldsymbol{p_j}) \ , \tag{2}$$

where $U(\boldsymbol{p_i})$ is the probability of hypothesis $\boldsymbol{p_i}$ at pixel i.

With soft-argmax, we can get an initial pixel-level plane parameter tensor $\mathbf{P} \in \mathbb{R}^{\frac{1}{8}H \times \frac{1}{8}W \times 3}$. We need to upsample it back to the original image resolution. We find that

directly applying bilinear upsampling will lead to the over-smoothness issue. Here we adopt the upsampling method proposed by RAFT [48]. Specifically, for each pixel of $\mathbf{P}$, we learn a convex combination by first predicting an $8 \times 8 \times 3 \times 3$ grid, then applying weighted combination over the learned weights of its 3×3 coarse neighbors to get the upsampled plane parameters $\mathbf{P}^{'} \in \mathbb{R}^{H \times W \times 3}$. This upsampling approach better preserves the boundaries of planes and other details in the reconstructed planar depth map.

Following [58], we apply a refinement module, which aims to learn the residual of the initial plane parameters w.r.t. groundtruth. The upsampled initial plane parameters $\mathbf{P}^{'}$ is concatenated with the normalized original image I_t as input to preserve image details, then passed into several 2D CNN layers to predict its residual $\delta\mathbf{P}^{'}$. Then we get the refined pixel-level plane parameters $\mathbf{P}_r = \mathbf{P}^{'} + \delta\mathbf{P}^{'}$, which is our final per-pixel plane parameters prediction.

3.3. Planar depth map reconstruction

In this subsection, we present how we associate the above two branches to make them benefit from each other. We also demonstrate how to get the piece-wise planar depth map as the final reconstructed plane representation.

Plane instance-aware soft pooling. After getting per-pixel plane parameters and plane masks from the two branches, the natural question is, can we associate the two heads and make them benefit from each other? To this end, inspired by [56, 61], we design a soft-pooling operation and propose a loss supervision on the depth map. For a detected plane, we output its soft mask $\mathbf{m}_s \in \mathbb{R}^{H \times W}$, where each element σ_i at each pixel i of $\mathbf{m}_s$ is the predicted foreground probability instead of a binary value $\in \{0, 1\}$ for differentiability. Then the instance plane parameter $\boldsymbol{p_t}$ can be computed by a soft-pooling operation with weighted averaging:

$$\boldsymbol{p_t} = \frac{\sum_{i=1}^{N} \sigma_i \cdot \boldsymbol{p_i}}{\sum_{i=1}^{N} \sigma_i} \ . \tag{3}$$

Then the instance-level planar depth map can be reconstructed:

$$\mathbf{D}_i = -\frac{\mathbb{1}_i}{\boldsymbol{p_t}^T \boldsymbol{K^{-1}} \mathbf{x}_i} \ , \tag{4}$$

where $\mathbb{1}_i$ is an indicator variable to identify foreground pixels. A threshold of 0.5 is applied on σ_i to determine whether pixel i is identified as foreground. $\boldsymbol{K^{-1}}$ is the inverse intrinsic matrix and $\mathbf{x}_i$ is the homogeneous coordinate of pixel i.

Depth map representation and loss supervision. We can obtain a stitched depth map $\mathbf{D} \in \mathbb{R}^{H \times W}$ for the image by filling the planar pixels with instance planar depth maps from Eq. (4). Since the learned pixel-wise plane parameters capture local planarity, we fill the non-planar pixels with the reconstructed pixel-wise planar depth map.

Then we can design a soft-pooling loss L_{sp} between the reconstructed depth map $\mathbf{D}$ and groundtruth depth map $\mathbf{D}^*$, with L_1 loss supervision as our soft-pooling loss L_{sp}:

$$L_{sp} = \left\| \mathbf{D} - \mathbf{D}^* \right\|_1 \ . \tag{5}$$

By supervising the model with L_{sp}, because of Eq. (3), the planar depth map is not only determined by the plane MVS head but also the plane detection head. In other words, the model is supposed to make the learned 2D plane segmentation and 3D plane parameters consistent with each other. During training, one module is able to get constraints from the other one's output. Note that although this loss shares similarity with [56, 61], there are differences between their work and ours. PlaneRecover [56] applies a similar loss to assign pixels to different plane instances. PlaneAE [61] builds the loss on plane parameter instead of depth map and targets to improve instance-level parameters. In contrast, our soft-pooling loss is mainly designed for making possible interactions between 2D plane segmentation and 3D parameter predictions.

3.4. Supervision with loss term uncertainty

Our supervisions have three parts: the plane detection losses L_D, the plane MVS losses L_M, and the soft pooling loss L_{sp}. L_D includes two-stage classification and bounding box regression losses, and the mask loss in the 2^{nd} stage. L_M includes the loss built on initial per-pixel plane parameters and its reconstructed depth map, and the refined ones. For each term of L_M, we adopt masked L_1 loss which is only applied on the pixels with valid groundtruth. Since the goals of plane detection and plane MVS branch are distinct, we weight each loss term by its learned uncertainty as introduced in [25]. This is effective in our experiments and can outperform the results without applying uncertainty by a large margin. Our final loss objective can be written as:

$$L = \sum_{i}^{N_D} \omega_{D_i} L_{D_i} + \sum_{j}^{N_M} \omega_{M_j} L_{M_j} + \omega_{sp} L_{sp} \ , \tag{6}$$

where w is the learned uncertainty for each loss term.

4. Experiments

4.1. Implementation details

We implement our framework in Pytorch [38]. The SGD optimizer is applied with an initial learning rate of 3×10^{-3} and a weight decay of 5×10^{-4}. The batch size is set to be 6, and the model is trained end-to-end on 3 NVIDIA 2080Ti GPUs for 10 epochs on the ScanNet [6] benchmark. The learning rate decays to 3×10^{-4} and 3×10^{-5} at 7^{th} and 9^{th} epoch respectively. We re-implement plane detection module following [31] but with a publicly released implementation [36] of Mask-RCNN [18]. Following [31], we initialize the weights with a detection model pretrained on COCO [30]. The input image size is set to be 640×480

Method	Depth Metrics							Detection Metrics					
	AbsRel↓	SqRel↓	RMSE↓	RMSE_log↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑	$AP^{0.2m}$↑	$AP^{0.4m}$↑	$AP^{0.6m}$↑	$AP^{0.9m}$↑	AP↑	mAP↑
PlaneRCNN [31]	0.164	0.068	0.284	0.186	0.780	0.953	0.989	0.310	0.475	0.526	0.546	0.554	0.452
MVSNet [58]	0.105	0.040	0.232	0.145	0.882	0.972	0.993	-	-	-	-	-	-
DPSNet [19]	0.100	0.035	0.215	0.135	0.896	0.977	0.994	-	-	-	-	-	-
NAS [26]	0.098	0.035	0.213	0.134	0.905	0.979	0.994	-	-	-	-	-	-
ESTDepth [33]	0.113	0.037	0.219	0.147	0.879	0.976	0.995	-	-	-	-	-	-
PlaneMVS-pixel (Ours)	0.091	0.029	0.194	0.120	0.920	0.987	0.997	0.448	0.535	0.556	0.560	**0.564**	**0.466**
PlaneMVS-final (Ours)	**0.088**	**0.026**	**0.186**	**0.116**	**0.926**	**0.988**	**0.998**	**0.456**	**0.540**	**0.559**	**0.562**	**0.564**	**0.466**

Table 1. Evaluation results for plane geometry and detection on ScanNet [6] among different methods.

during training and testing. Since our batch size is relatively small, we freeze all the batch normalization [20] layers of the plane detection head during training. We apply group normalization [53] as the normalization function in our plane MVS head.

4.2. Training data generation

Semantic plane groundtruth generation. To build our plane dataset with semantic labels, we first obtain and preprocess the raw rendered plane masks from [31], and get the 2D raw semantic maps from ScanNet [6]. Then we map the semantic labels from ScanNet to NYU40 [42]. We merge some semantically similar labels in NYU40 and finally pick 11 labels that are likely to contain planar structures. We obtain the semantic label for each plane instance by projecting its mask onto the semantic map then performing a majority vote. If the voted result does not belong to any of the labels we select, we simply label the raw mask as non-planar and treat it as a negative sample during training and evaluation.

View selection for MVS. We have to sample stereo pairs from ScanNet [6] monocular sequences and a stereo pair is considered appropriate if the images in the pair have a large enough camera baseline and sufficient overlap. In this work, we choose those stereo pairs as qualified ones if their relative translations lie between $0.05m$ and $0.15m$. We select 2 views (a target and a source view) during training and testing. We believe adding more views could further improve performance, but that is outside the scope of this work.

4.3. Datasets

In our experiments, we use ScanNet [6] for training and evaluation. We further generalize our model to two other RGB-D indoor datasets, *i.e.*, 7-Scenes [14] and TUM-RGBD [44], by testing with and without fine-tuning, to demonstrate the generalizability. Since the two datasets do not contain any plane groundtruths, we only evaluate the planar geometry metrics and show some qualitative results for plane detection on them. Due to space limit, here we only introduce how we use ScanNet. Please refer to our supplementary material for information on other datasets.

ScanNet. ScanNet [6] is a large indoor benchmark containing hundreds of scenes. We sample the training and testing stereo pairs from its official training and validation split, respectively. After pre-processing and filtering out the unqualified data following the steps in PlaneRCNN [31], we randomly subsample $20,000$ pairs for training. However, since the raw 3D meshes of ScanNet are not always complete, the rendered plane masks from meshes are noisy and inaccurate in quite a few images. This results in unconvincing plane detection evaluation if we directly test on those images. To this end, we manually pick 950 stereo pairs whose plane mask annotations are visually clean and complete from the original testing set for our evaluation.

4.4. Evaluation metrics

Following previous plane reconstruction methods [31, 32], we mainly evaluate the plane reconstruction quality on average precision (AP) of plane detection with varying depth error thresholds $[0.2m, 0.4m, 0.6m, 0.9m]$, and the widely-used depth metrics [8]. Since we introduce plane semantics in our framework, we also evaluate the mean average precision (mAP) [30] which couples semantic segmentation and detection as used in object detection papers.

4.5. Comparison with state-of-the-arts

Single-view plane reconstruction methods. We first compare our PlaneMVS with a SOTA single-view plane reconstruction method PlaneRCNN [31], which also serves as the baseline of our model. We test it on our re-implemented version with plane semantic predictions with the same training and testing data as ours. Tab. 1 shows that our method outperforms PlaneRCNN in terms of both plane geometry and 3D plane detection by a large margin. As shown in Fig. 3, PlaneRCNN does well in obtaining geometrically smooth planar depth maps, but their plane parameters are far from accurate (*e.g.*, the 2^{nd} and 4^{th} row of Fig. 3), which rely on single-view regression and suffer from the depth scale ambiguity issue (*e.g.*, the 1^{st} and 3^{rd} row of Fig. 3). For AP without considering depth, we also get considerable improvements benefiting from multi-task learning and the proposed soft-pooling loss. Although PlaneRCNN is a strong baseline, Fig. 4 clearly shows that our method better perceives plane boundaries, and our segmentation aligns better with 3D plane geometry. For mAP evaluation which considers plane semantic accuracy with detection, our method also outperforms PlaneRCNN by a non-trivial margin.

Learning-based MVS methods. We also compare our method against several representative MVS methods. We select two representative depth-based MVS methods, MVS-

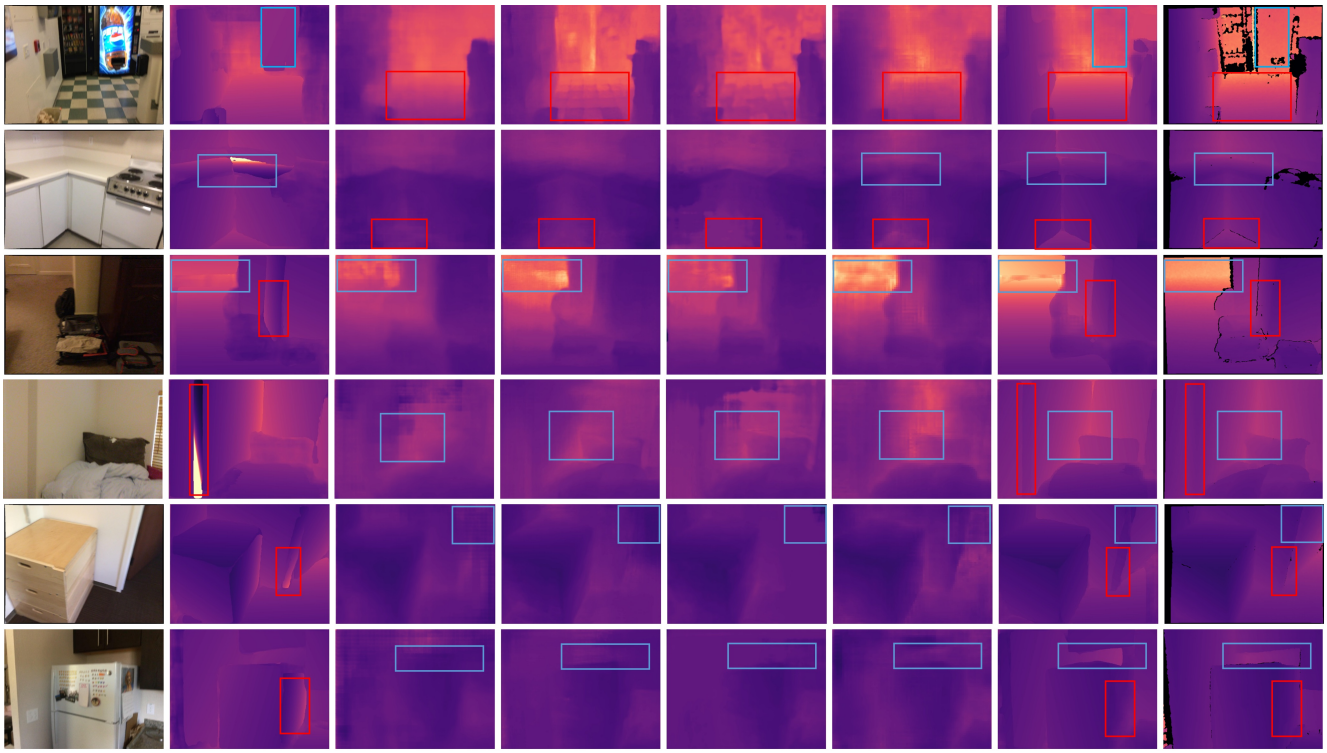

Figure 3. Qualitative results of reconstructed depth maps on ScanNet [6]. "PlaneMVS-pixel" denotes the depth reconstructed from pixel-level plane parameters. "PlaneMVS-final" denotes the final depth from the instance plane parameters after plane soft-pooling. Regions with salient differences between our results and others are highlighted with blue and red boxes. Best viewed on screen with zoom-in.

Net [58] and DPSNet [19] since our MVS module shares similar network architecture with them. Besides, we also compare with NAS [26] which aims to enforce depth-normal geometric consistency in MVS. We train and test these methods on our ScanNet data split with their released code for fair comparisons. We also compare with one of the state-of-the-art multi-view depth estimation methods ESTDepth [33]. From Tab. 1, our method clearly outperforms those MVS methods. Note that [33] is designed for temporally longer frames which may explain its possible performance drop when testing on two-views. The qualitative results in Fig. 3 clearly show that compared with conventional depth-based MVS methods, our "PlaneMVS-pixel" results reconstructed from pixel-level plane parameters have shown more accurate depth, especially over textureless areas, which can be accredited to the proposed slanted plane hypothesis that learns planar geometry. By applying soft-pooling with detected plane masks (*i.e.*, our "PlaneMVS-final"), global geometric smoothness and sharper boundaries can be achieved over planar regions. The texture-copy issue in some cases (*e.g.*, the 1^{st} row of Fig. 3) of other methods can also be effectively avoided in ours.

Method	AbsRel↓	$\delta < 1.25$↑
PlaneRCNN [31]	0.221	0.640
MVSNet [58]	0.162	0.766
DPSNet [19]	0.159	0.788
NAS [26]	0.154	0.784
ESTDepth [34]	**0.153**	0.786
Ours	0.158	**0.793**
Ours-FT	0.113	0.890

Table 2. Reconstructed depth on 7-Scenes [14] dataset by different methods. "Ours" means directly testing with ScanNet-pretrained model. "Ours-FT" means testing with 7-Scenes-finetuned model.

4.6. Results on 7-Scenes

We evaluate our approach on the 7-Scenes [14] dataset to check its generalizability. Tab. 2 shows that our method also significantly outperforms PlaneRCNN [31], and is better than or comparable with other MVS methods [19,26,33,58]. Since PlaneRCNN [31] learns plane geometry from single views, the ability to generalize beyond the domain of training scenes is limited. However, our method benefits from multi-view geometry to learn multi-view feature correspondences and thus has superior generalizability on unseen data. We leave how we perform finetuning on 7-Scenes with only groundtruth depth to the supplementary material.

Method	Depth Metrics							Detection Metrics				
	AbsRel↓	SqRel↓	RMSE↓	RMSE_log↓	$\delta < 1.25$↑	$\delta < 1.25^2$↑	$\delta < 1.25^3$↑	$AP^{0.2m}$↑	$AP^{0.4m}$↑	$AP^{0.6m}$↑	$AP^{0.9m}$↑	AP↑
Baseline	0.170	0.074	0.305	0.200	0.746	0.944	0.990	0.288	0.458	0.519	0.545	0.551
+ Soft-pooling loss	0.119	0.042	0.234	0.148	0.871	0.979	0.995	0.380	0.520	0.549	0.557	0.561
+ Loss term uncertainty	0.089	0.027	0.190	0.119	0.922	0.987	0.997	0.449	0.535	0.556	0.560	0.562
+ Convex upsampling	**0.088**	**0.026**	**0.186**	**0.116**	**0.926**	**0.988**	**0.998**	**0.456**	**0.540**	**0.559**	**0.562**	**0.564**

Table 3. Ablation study on the components of our proposed method.

Figure 4. Qualitative results of plane detection on ScanNet [6]. The regions with salient differences between our method and PlaneRCNN are highlighted with red boxes.

4.7. Ablation study

In this subsection, we evaluate the effectiveness of each proposed component (*i.e.*, soft-pooling loss, loss term uncertainty, convex upsampling, and slanted plane hypothesis). We leave some comparisons on hyper-parameters and settings to the supplementary material.

Method	AbsRel↓	SqRel↓	$\delta < 1.25$↑	$AP^{0.2m}$ ↑	AP↑
Fronto-MVS	0.094	0.033	0.917	0.433	0.548
Ours	**0.088**	**0.026**	**0.926**	**0.456**	**0.564**

Table 4. Ablation study: slanted v.s. fronto-parallel plane.

Soft-pooling loss. The soft-pooling loss is designed for coupling plane detection and plane geometry. From Tab. 3, it significantly improves both plane depth and 3D detection on all metrics. It helps the pixels within the same plane learns consistent plane parameters which benefits the plane geometry. For plane detection, as shown in Fig. 4, our detected planes also align better with 3D plane geometry, especially over plane boundaries.

Training with loss term uncertainty. Tab. 3 shows that by weighting each loss term with learned uncertainty, our model can get further improvement on plane geometry and 3D detection. There are two possible reasons. First, our model has different branches with multiple losses for 2D or 3D objectives. Setting each loss with the same weight may not make them converge smoothly. Second, as introduced in Sec. 4.1, the plane detection head is initialized with a COCO-pretrained model. But for the MVS head, we train it from scratch. Thus the learning procedures for the two branches may be imbalanced if we do not adaptively change the weights for each term. After adopting the learnable uncertainty, the weights of different terms are automatically tuned during training, which has brought great benefits.

Convex upsampling. We analyze the effect of applying the convex combination upsampling to replace bilinear upsampling. As shown in Tab. 3, we get considerable improvement. The learned convex upsampling better keeps fine-grained details than bilinear upsampling.

Slanted plane hypothesis. We conduct an additional ablation study, *i.e.*, replacing our slanted plane hypothesis with the frontal-parallel plane hypothesis, using the same network architecture. We also apply convex upsampling and loss-term uncertainty for fair comparison. We employ the least-squares algorithm to fit planes with the predicted per-pixel depth map and plane masks, and then transform the plane parameters to planar depth maps. As shown in Tab. 4, our proposed method outperforms the 'Fronto-MVS' baseline on both 3D plane detection and depth metrics. Besides, our model learns plane parameters in an end-to-end manner instead of fitting planes as a post-processing step. This verifies the effectiveness of the proposed slanted plane hypothesis w.r.t. fronto-parallel hypothesis.

5. Conclusion and Future Work

In this work, we propose PlaneMVS, a deep MVS framework for multi-view plane reconstruction. Based on our proposed slanted plane hypothesis for plane-sweeping, 3D plane parameters can be learned by deep MVS in an end-to-end manner. We also couple the plane detection branch and the plane MVS branch with the proposed soft pooling loss. Compared with single-view methods, our system can reconstruct 3D planes with significantly better accuracy, robustness, and generalizability. Without sophisticated designs, our system even outperforms several state-of-the-art MVS approaches. Please refer to our supplementary material for more results, discussions and potential limitations.

There are a few directions worth exploring in the future. First, recent advanced designs of deep MVS sys-

tems [16, 52, 57, 62] could be incorporated to further improve MVS reconstruction. Second, temporal information from videos (beyond two frames as we are currently using) can be exploited to achieve temporally coherent plane reconstruction, such that consistent single-view predictions could be fused into a global 3D model of the entire scene.

Acknowledgement. The research of J. Liu and X. Huang was supported in part by the NSF award #1815491.

6. Supplementary Material

6.1. Hypothesis selection for slanted planes

Fig. 5 shows the distribution of the three axes of plane $\boldsymbol{n}^T/e$ sampled from $10,000$ training images. Based on the distribution, we select $(-2, 2)$, $(-2, 2)$, $(-2, 0.5)$ as the range of x, y, z axis for $\boldsymbol{n}^T/e$, respectively, to ensure at least 95% of the groundtruth planes lie within the ranges. Since our plane hypothesis is a three-dimensional vector, the computational cost of the cost volume is cubic w.r.t. the number of hypothesis per axis. To reach a balance between accuracy and memory consumption, we sample 8 hypotheses uniformly along every axis and finally have $N = 8^3 = 512$ plane hypotheses in total.

6.2. Semantic classes on ScanNet

After merging the semantically-similar categories in NYU40 [42] labels, we pick 11 classes: wall, floor, door, chair, window, picture, desk & table, bed & sofa, monitor & screen, cabinet & counter, box & bin, which are likely to contain planar structures in indoor scenes. Please refer to Fig. 6 for some visualization examples of the generated planar instance and semantic groundtruth from ScanNet [6].

6.3. Benchmark setup

7-Scenes. 7-Scenes [14] collects posed RGB-D camera frames of seven indoor scenes. We sample stereo pairs in the same manner as in ScanNet [6] and follow the official split to get finetuning and evaluation data. We finally have $26,358$ pairs for finetuning and $15,508$ pairs for evaluation.

TUM-RGBD. TUM-RGBD [44] is an indoor RGB-D monocular SLAM dataset with calibrated cameras. We randomly select 4 scenes (*i.e.*, fr1-desk, fr1-room, fr1-desk2, fr3-long-office-household) with $5,013$ pairs for finetuning and 2 scenes (*i.e.*, fr2-desk, fr3-long-office-household-validation) containing $4,817$ pairs for evaluation.

6.4. Results on 7-Scenes and TUM-RGBD

We have discussed how we deal with 7-Scenes and have demonstrated its quantitative results in the main paper. Here we introduce our simple but effective strategy to perform finetuning with only groundtruth depth. We first generate pseudo groundtruths of plane masks by getting the predictions with the ScanNet-pretrained model on the testing images. Then we train our model without plane parameter losses but maintain other losses. We simply set each loss weight to 1 instead of adopting the loss term uncertainty during finetuning since we find it cannot bring much improvement. We finetune the model for 5 epochs. The planar depth gets much improved and we find that the plane detection results also tend to be visually better, which may be accredited to multi-task training and our soft-pooling loss

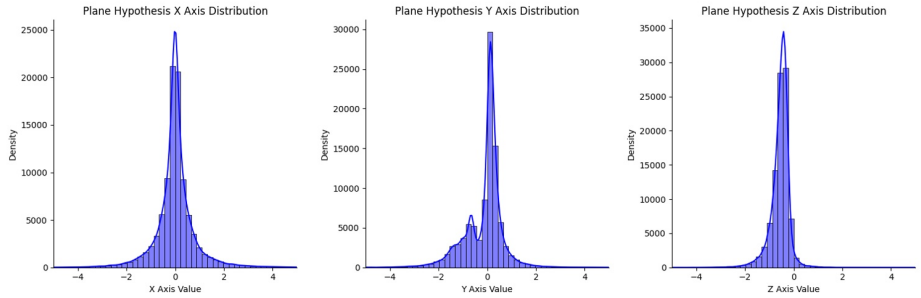

Figure 5. Plane hypothesis distribution of the three axes.

Figure 6. Examples of planar semantic and instance groudtruths on ScanNet [6]. Different colors represent different plane instances (2^{nd} column) or semantic categories (3^{rd} column).

to associate 2D with 3D. The same applies to the TUM-RGBD [44] dataset. Some qualitative examples of 7-Scenes are shown in Fig. 7.

As shown in Tab. 5 and Fig. 8, similar to 7-Scenes, our approach generalizes much better on TUM-RGBD compared with PlaneRCNN [31], thanks to the learned multi-view geometric relationship. By performing the proposed finetuning strategy, the results get further improved on both 3D planar geometry and 2D planar detection.

6.5. More Ablation studies

In this section, we discuss the impact of applying different hyper-parameters or settings in our experiments. Then we show qualitative examples on the two components of our proposed method to intuitively demonstrate their effects.

Method	AbsRel↓	SqRel↓	$\delta < 1.25$↑
PlaneRCNN [31]	0.243	0.105	0.655
Ours	0.143	0.07	0.795
Ours-FT	**0.120**	**0.054**	**0.851**

Table 5. Reconstructed depth on TUM-RGBD dataset [44] of different methods. "Ours" means directly testing with the ScanNet-pretrained model. "Ours-FT" means testing with the TUM-RGBD-finetuned model.

6.5.1 Hyper-parameters and settings

Plane hypothesis range. We first study the effect of the plane hypothesis range we set. We compare the results of different hypothesis ranges while keeping the hypothesis number N unchanged: (i) use the same range of $(-2, 2)$ for the x, y, z axes; (ii) broaden the range to $(-2.5, 2.5)$; (iii) shorten the range to $(-1.75, 1.75)$; (iv) employ the same range of $(-2, 2)$ for the x, y axes and a different range of $(-2, 0.5)$ for the z axis. As shown in Tab. 6, setting (iv), which serves as our default setting, achieves the best result. The performance drops when using the same range for all axes as (i), since z values mainly distribute between $(-2, 0.5)$. Using a broader range, *e.g.* (i) and (ii), covers some marginal values but decreases the density of the plane hypothesis, thus leading to less accurate results. In setting (iv), although shortening ranges can increase the hypothesis density, some non-negligible groundtruth values are not well covered, thus also leading to worse results.

Hypos range	AbsRel↓	$\delta < 1.25$↑
(-2, 2) for x,y,z	0.093	0.920
(-1.75, 1.75) for x,y,z	0.094	0.921
(-2.5, 2.5) for x,y,z	0.096	0.919
(-2, 2) for x,y; (-2, 0.5) for z	**0.088**	**0.926**

Table 6. Ablation study on the range of slanted plane hypothesis.

Plane hypothesis number. When keeping the plane hypothesis range constant, varying hypothesis number N

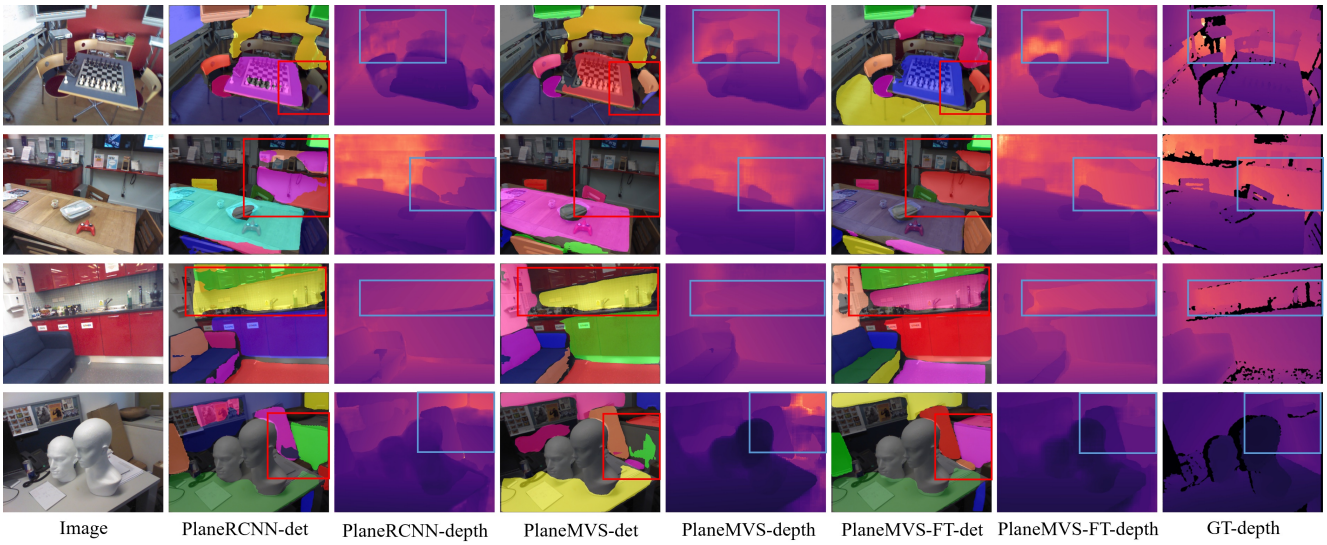

Figure 7. The plane reconstruction results on 7-scenes [14] among different methods. "FT" denotes "finetuned" and "det" is short for "detection". Regions with salient differences are highlighted with blue and red boxes.

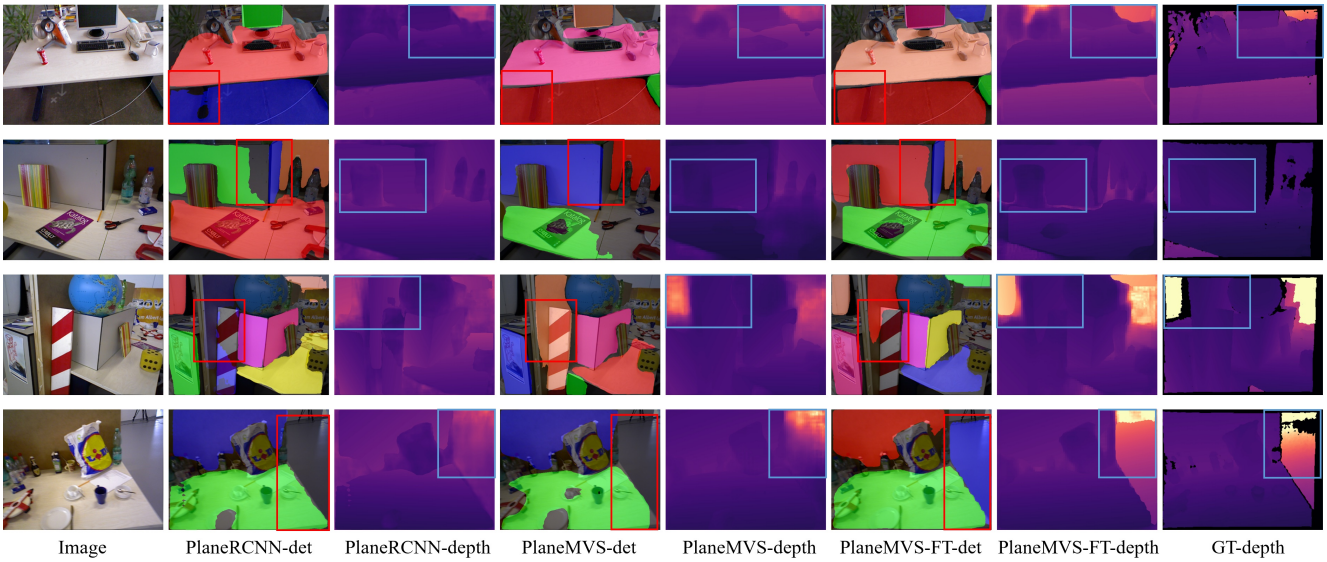

Figure 8. The plane reconstruction results on TUM-RGBD [44] among different methods. Regions with salient differences are highlighted with blue and red boxes.

changes the hypothesis density. We test our model using $6, 8, 10$ hypotheses per axis, *i.e.*, $N = 216, 512$ and $1,000$ respectively. The results are listed in Tab. 7. As expected, in general, the higher density we set, the better geometry performance we achieve. The performance gaps among different numbers are small, which demonstrates that our model is robust to these hyper-parameters to some extent. Note that using $N = 1,000$ will substantially increase the memory consumption. So we choose $N = 512$ in our default setting.

Hypos number per axis	AbsRel↓	$\delta < 1.25$↑
6 hypos (216 in total)	0.091	0.924
8 hypos (512 in total)	**0.088**	0.926
10 hypos (1,000 in total)	**0.088**	**0.927**

Table 7. Ablation study on plane hypothesis number.

Plane instance-aware soft pooling. We now evaluate the recovered depths among different pooling strategies reflecting the efficacy of plane detection on the learned 3D planar geometry. As shown in Tab. 8, when evaluat-

Method	AbsRel↓	$\delta < 1.25$↑
Pixel-planar w/o pooling	0.091	0.920
Pooling with predicted masks	0.088	0.925
Soft-pooling with predicted masks	0.088	0.926
Pooling with groundtruth masks	**0.087**	**0.932**

Table 8. Ablation study on plane instance pooling with plane masks during testing.

ing the depth reconstructed from pixel-level plane parameters, it underperforms the results with plane instance pooling since the generated depth maps cannot capture piecewise planarity. The result improves when we apply hard-pooling with predicted plane masks over the pixel-level plane parameters. Applying soft-pooling weighted with pixel-level probability further brings a minor improvement since the probability reflects the confidence of a pixel belonging to a plane instance. Finally, we use groundtruth plane masks to perform pooling, which represents the upper bound of the impact of plane detection on geometry. As expected, it achieves the best result among the settings. Since groundtruth plane masks are not available during testing, we always apply the soft-pooling with predicted masks in other experiments.

Depth on planar region. We further compare the reconstructed depth over only planar regions *v.s.* the whole image. Specifically, we conduct experiments only evaluating depth on the pixels that belong to any of the groundtruth planes. As shown in Tab. 9, compared with the depth over the whole image, the quantitative result over planar regions is better, no matter whether plane-instance-pooling is applied or not. This demonstrates that our proposed method's geometry improvement mainly comes from the pixels of planar regions, which conforms to our initial motivation and objective.

Method	AbsRel↓	$\delta < 1.25$↑
Depth over whole image w/o pooling	0.091	0.920
Depth over planar region w/o pooling	0.086	0.929
Depth over whole image	0.088	0.926
Depth over planar region	**0.081**	**0.938**

Table 9. Ablation study on the evaluations over planar region.

Training dataset scale. In our default setting, we only sample $20,000$ stereo pairs for training. To analyze the impact of the scale of training data, we sample a larger training set with $66,000$ stereo pairs from the same scene split but keep the evaluation split unchanged. As shown in Tab. 10, our performance can be further improved with more training data on both plane detection and geometry metrics.

Dataset Scale	AbsRel↓	$\delta < 1.25$↑	$AP^{0.2m}$ ↑	AP↑
20,000 training pairs	0.088	0.926	0.456	0.564
66,000 training pairs	**0.082**	**0.934**	**0.470**	**0.570**

Table 10. Ablation study on the scale of training dataset.

6.5.2 Qualitative ablation analysis

This section gives some qualitative ablation analysis on the two components (*i.e.*, convex upsampling and the soft-pooling loss) used in our method. Fig. 9 shows the efficacy of convex upsampling. We show the depth map recovered from pixel-level parameters to eliminate the effect of plane instance pooling. It is clear that the results upsampled by convex combination have sharper boundaries and fewer artifacts than using bilinear upsampling.

Fig. 10 shows the effectiveness of the proposed soft-pooling loss. The detected planes from the model trained with the soft-pooling loss are much more complete and align better with their boundaries.

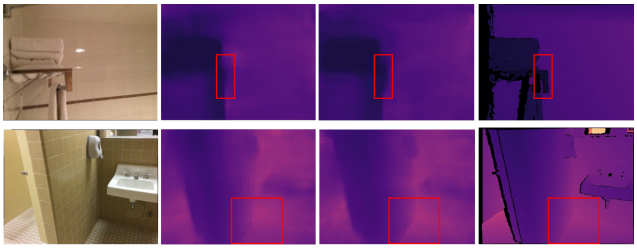

Figure 9. Effects of the convex upsampling on the depth map from pixel-level plane parameters. Regions with salient differences are highlighted with red boxes. Best viewed on screen with zoom-in.

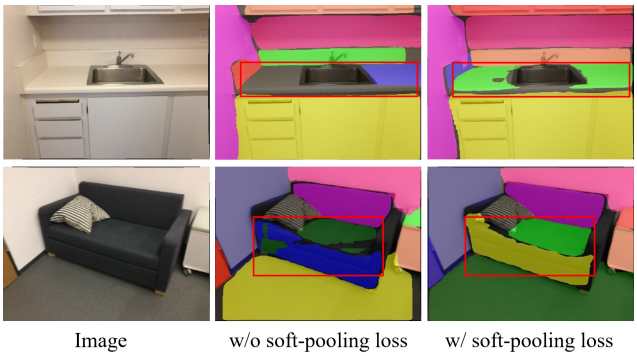

Figure 10. Effects of the soft-pooling loss on plane detection. Regions with salient differences are highlighted with red boxes.

6.6. Additional visualizations

We provide additional visualizations on predicted instance plane detection, planar semantic map, reconstructed planar depth map and 3D point cloud in Fig. 11, from our testing set on ScanNet [6].

6.7. Discussions and limitations

Our method *v.s.* patchmatch stereo. Our method shares high-level ideas with traditional patchmatch stereo

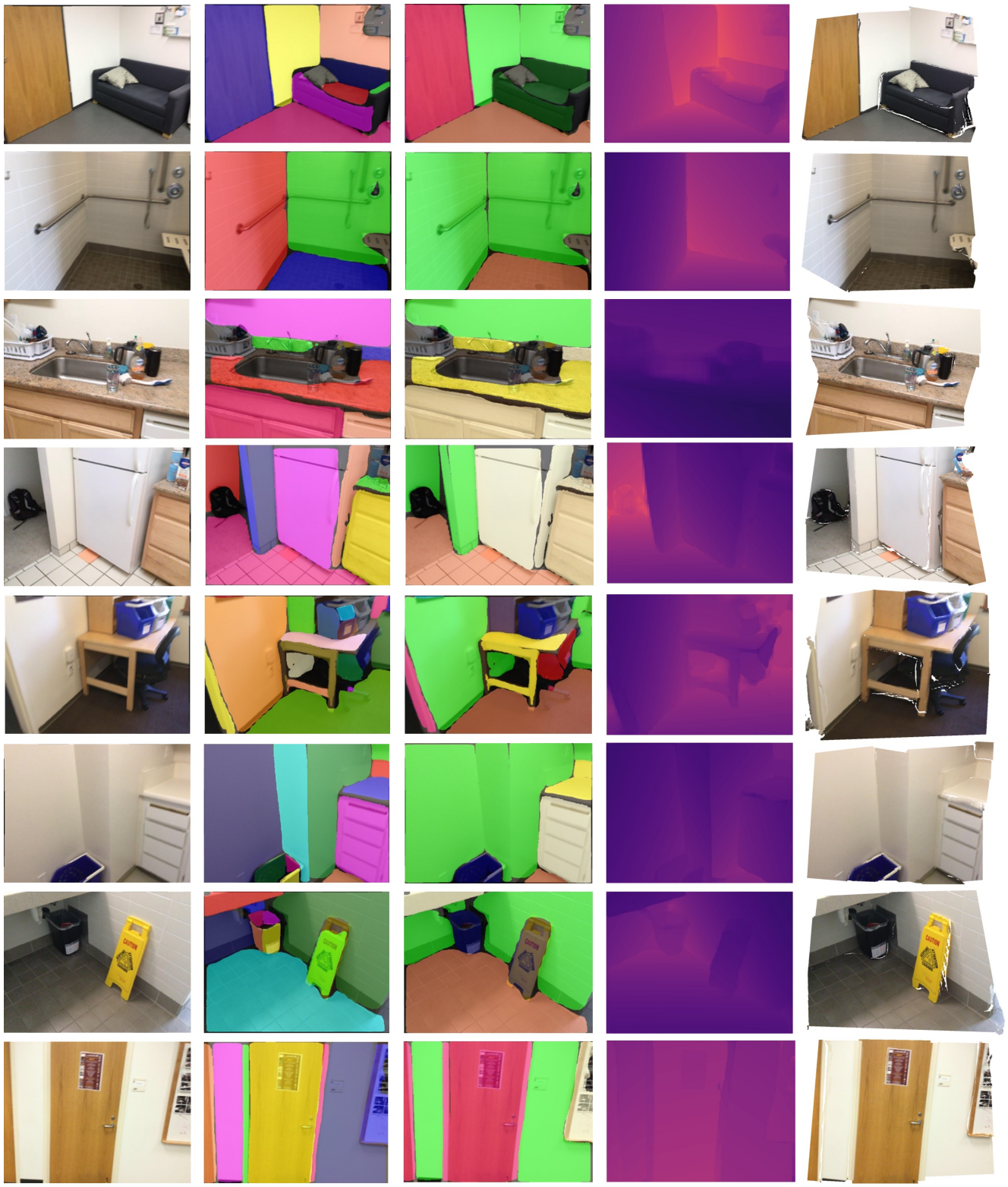

Figure 11. More qualitative results on ScanNet [6], including the instance planar masks, planar semantic map, planar depth map and the reconstructed 3D point cloud.

works [2, 11] which aim to estimate a slanted plane for each pixel on the stereo reconstruction problem. However, our method differs from them in several aspects. (i) They perform patch matching around a pixel within a squared support window, where the patch size requires to be carefully set, thus not flexible and adaptive across various real-world cases. Instead of explicitly defining a patch, we associate and match the multi-view deep features. This is based on the observation that a pixel's receptive field on the feature map is far beyond itself because of stacked CNNs. The model can automatically learn the appropriate field for matching local features with end-to-end training. (ii) These methods usually first initialize pixels with random slanted plane hypotheses, then undergo sophisticated, multi-stage schemes with iterative optimizations. In contrast, we generate more reliable slanted plane hypotheses based on a data-driven approach (*i.e.*, analyzing the groundtruth plane distribution), and learn the pixel-wise plane parameters in an end-to-end manner, which is much easier to optimize. (iii) They usually adopt the photometric pixel dissimilarity as the matching cost function, which is sensitive to illumination changes and motion blurs across views. In contrast, we apply a feature-metric matching strategy, which is more robust to potential noises compared with applying photometric distance.

Potential limitations. Although we have achieved superior performance in most images, our system generates some failure cases as well. Firstly, as shown in Fig. 12, because of the large temporal gap, there exist areas in the target image which are invisible in the source image and thus do not follow the planar homography relationship. This issue may be mitigated by introducing a network to learn the pixel-wise visibility or uncertainty [62]. Secondly, as in Fig. 13, there exist holes on some adjacent planes reconstructed from our method. An existing work [39] proposes to infer and enforce the inter-plane relationship from single images. This approach may solve the second issue and could be incorporated to further improve the final plane reconstruction. We also leave it into future work to explore.

Figure 12. Failure case I: large temporal gap between two views. Problematic regions are highlighted with blue and red boxes.

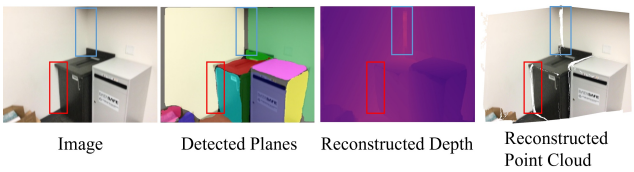

Figure 13. Failure case II: holes between adjacent planes. Problematic regions are highlighted with blue and red boxes.

References

[1] Stan Birchfield and Carlo Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 1, pages 489–495. IEEE, 1999. 3

[2] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011. 3, 14

[3] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conference on Computer Vision*, pages 766–779. Springer, 2008. 2

[4] Denis Chekhlov, Andrew P Gee, Andrew Calway, and Walterio Mayol-Cuevas. Ninja on a plane: Automatic discovery of physical planes for augmented reality using visual slam. In *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 153–156. IEEE, 2007. 1

[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. 3

[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. 5, 6, 7, 8, 9, 10, 12, 13

[7] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 2

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *arXiv preprint arXiv:1406.2283*, 2014. 6

[9] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429. IEEE, 2009. 1, 2

[10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 2

[11] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 873–881, 2015. 2, 14

[12] David Gallup, Jan-Michael Frahm, Philippos Mordohai, Qingxiong Yang, and Marc Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *2007*

IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2007. 3

[13] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1418–1425. IEEE, 2010. 1, 2

[14] Ben Glocker, Shahram Izadi, Jamie Shotton, and Antonio Criminisi. Real-time rgb-d camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179. IEEE, 2013. 6, 7, 9, 11

[15] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 2

[16] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504, 2020. 2, 3, 9

[17] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003. 2

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 3, 5

[19] 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. 2, 4, 6, 7

[20] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 6

[21] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. SurfaceNet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017. 2

[22] Pan Ji, Runze Li, Bir Bhanu, and Yi Xu. Monoindoor: Towards good practice of self-supervised monocular depth estimation for indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12787–12796, 2021. 2

[23] Linyi Jin, Shengyi Qian, Andrew Owens, and David F Fouhey. Planar surface reconstruction from sparse views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12991–13000, 2021. 2

[24] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *arXiv preprint arXiv:1708.05375*, 2017. 2

[25] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017. 2, 5

[26] Uday Kusupati, Shuo Cheng, Rui Chen, and Hao Su. Normal assisted stereo depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2189–2199, 2020. 2, 3, 6, 7

[27] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38(3):199–218, 2000. 2

[28] Maxime Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005. 2

[29] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 6

[31] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. PlaneRCNN: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 1, 2, 3, 5, 6, 7, 10

[32] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. PlaneNet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018. 1, 2, 6

[33] Xiaoxiao Long, Lingjie Liu, Wei Li, Christian Theobalt, and Wenping Wang. Multi-view depth estimation using epipolar spatio-temporal networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8258–8267, 2021. 6, 7

[34] Xiaoxiao Long, Lingjie Liu, Christian Theobalt, and Wenping Wang. Occlusion-aware depth estimation with adaptive normal constraints. In *European Conference on Computer Vision*, pages 640–657. Springer, 2020. 2, 3, 7

[35] Keyang Luo, Tao Guan, Lili Ju, Haipeng Huang, and Yawei Luo. P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10452–10461, 2019. 3

[36] Francisco Massa and Ross Girshick. MaskRCNN-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 5

[37] 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 *European conference on computer vision*, pages 414–431. Springer, 2020. 3

[38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[39] Yiming Qian and Yasutaka Furukawa. Learning pairwise inter-plane relations for piecewise planar reconstruction. In

European Conference on Computer Vision, pages 330–345. Springer, 2020. 2, 14

[40] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10413–10422, 2019. 3

[41] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999. 2

[42] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 6, 9

[43] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. In *International Conference on Computer Vision*, pages 1881–1888, 2009. 1, 2

[44] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 6, 9, 10, 11

[45] 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. 3

[46] Yuichi Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *2013 IEEE international conference on robotics and automation*, pages 5182–5189. IEEE, 2013. 1

[47] Bin Tan, Nan Xue, Song Bai, Tianfu Wu, and Gui-Song Xia. PlaneTR: Structure-guided transformers for 3d plane recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4186–4195, 2021. 1, 2

[48] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 5

[49] Lokender Tiwari, Pan Ji, Quoc-Huy Tran, Bingbing Zhuang, Saket Anand, and Manmohan Chandraker. Pseudo rgb-d for self-improving monocular slam and depth prediction. In *European conference on computer vision*, pages 437–455, 2020. 2

[50] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012. 2

[51] Grace Tsai, Changhai Xu, Jingen Liu, and Benjamin Kuipers. Real-time indoor scene understanding using bayesian filtering with motion cues. In *2011 International Conference on Computer Vision*, pages 121–128. IEEE, 2011. 1

[52] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. PatchmatchNet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14194–14203, 2021. 3, 9

[53] Yuxin Wu and Kaiming He. Group Normalization. In *Proceedings of the European conference on computer vision*, pages 3–19, 2018. 6

[54] Qingshan Xu and Wenbing Tao. Planar prior assisted patchmatch multi-view stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12516–12523, 2020. 3

[55] Qingshan Xu and Wenbing Tao. PVSNet: Pixelwise visibility-aware multi-view stereo network. *arXiv preprint arXiv:2007.07714*, 2020. 3

[56] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pages 85–100, 2018. 1, 2, 5

[57] Jiayu Yang, Wei Mao, Jose M Alvarez, and Miaomiao Liu. Cost volume pyramid based depth inference for multi-view stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4877–4886, 2020. 2, 3, 9

[58] 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. 2, 3, 4, 5, 6, 7

[59] 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. 3

[60] Zehao Yu and Shenghua Gao. Fast-mvsnet: Sparse-to-dense multi-view stereo with learned propagation and gauss-newton refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1949–1958, 2020. 3

[61] Zehao Yu, Jia Zheng, Dongze Lian, Zihan Zhou, and Shenghua Gao. Single-image piece-wise planar 3d reconstruction via associative embedding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1029–1037, 2019. 1, 2, 5

[62] Jingyang Zhang, Yao Yao, Shiwei Li, Zixin Luo, and Tian Fang. Visibility-aware multi-view stereo network. *British Machine Vision Conference (BMVC)*, 2020. 3, 9, 14

[63] Wang Zhao, Shaohui Liu, Yi Wei, Hengkai Guo, and Yong-Jin Liu. A confidence-based iterative solver of depths and surface normals for deep multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6168–6177, 2021. 2, 3

[64] Yuliang Zou, Pan Ji, Quoc-Huy Tran, Jia-Bin Huang, and Manmohan Chandraker. Learning monocular visual odometry via self-supervised long-term modeling. *arXiv preprint arXiv:2007.10983*, 2020. 2