

# Occlusion-Aware Depth Estimation with Adaptive Normal Constraints

Xiaoxiao Long<sup>1</sup>, Lingjie Liu<sup>2</sup>, Christian Theobalt<sup>2</sup>, and Wenping Wang<sup>1</sup>

<sup>1</sup> The University of Hong Kong  
{xxlong,wenping}@cs.hku.hk

<sup>2</sup> Max Planck Institute for Informatics  
{lliu,theobalt}@mpi-inf.mpg.de

**Abstract.** We present a new learning-based method for multi-frame depth estimation from a color video, which is a fundamental problem in scene understanding, robot navigation or handheld 3D reconstruction. While recent learning-based methods estimate depth at high accuracy, 3D point clouds exported from their depth maps often fail to preserve important geometric feature (e.g., corners, edges, planes) of man-made scenes. Widely-used pixel-wise depth errors do not specifically penalize inconsistency on these features. These inaccuracies are particularly severe when subsequent depth reconstructions are accumulated in an attempt to scan a full environment with man-made objects with this kind of features. Our depth estimation algorithm therefore introduces a *Combined Normal Map (CNM)* constraint, which is designed to better preserve high-curvature features and global planar regions. In order to further improve the depth estimation accuracy, we introduce a new occlusion-aware strategy that aggregates initial depth predictions from multiple adjacent views into one final depth map and one occlusion probability map for the current reference view. Our method outperforms the state-of-the-art in terms of depth estimation accuracy, and preserves essential geometric features of man-made indoor scenes much better than other algorithms.

**Keywords:** Multi-view Depth Estimation, Normal Constraint, Occlusion-aware Strategy, Deep Learning

## 1 Introduction

Dense multi-view stereo is one of the fundamental problems in computer vision with decades of research, e.g. [31, 14, 4, 40, 13, 43]. Algorithms vastly differ in their assumptions on input (pairs of images, multi-view images etc.) or employed scene representations (depth maps, point clouds, meshes, patches, volumetric scalar fields etc.) [11].

In this paper, we focus on the specific problem of multi-view depth estimation from a color video from a single moving RGB camera [11]. Algorithms to solve this problem have many important applications in computer vision, graphics and robotics. They empower robots to avoid collisions and plan paths using

only one onboard color camera [3]. Such algorithms thus enable on-board depth estimation from mobile phones to properly combine virtual and real scenes in augmented reality in an occlusion-aware way. Purely color-based solutions bear many advantages over RGB-D based approaches [47] since color cameras are ubiquitous, cheap, consume little energy, and work nearly in all scene conditions.

Depth estimation from a video sequence is a challenging problem. Traditional methods [14, 4, 40, 13, 43] achieve impressive results but struggle on important scene aspects: texture-less regions, thin structures, shape edges and features, and non-Lambertian surfaces. Recently there are some attempts to employ learning techniques to this problem [20, 16, 36, 41, 18]. These methods train an end-to-end network typically with a pixel-wise depth loss function. They lead to significant accuracy improvement in depth estimation compared to non-learning-based approaches. However, most of these works fail to preserve prominent features of 3D shapes, such as corners, sharp edges and planes, because they use only depth for supervision and their loss functions are therefore not built to preserve these structures. This problem is particularly detrimental when reconstructing indoor scenes with man-made objects or regular shapes, as shown in Fig. 4. Another problem is the performance degradation caused by the depth ambiguity in the occluded region, which has been ignored by most of the existing works.

We present a new method for depth estimation with a single moving color camera that is designed to preserve important local features (edges, corners, high curvature features) and planar regions. It takes one video frame as *reference image* and uses some other frames as *source images* to estimate depth in the reference frame. Pairing the reference image with each source image, we first build an initial 3D cost volume from each pair via plane-sweeping warping. Subsequent cost aggregation for each initial cost volume yields an initial depth maps for each image pair (e.g. the reference image and a source image). Then we employ a new occlusion-aware strategy to combine these initial depth maps into one final reference-view depth map, along with an occlusion probability map.

Our first main contribution is a new structure preserving constraint enforced during training. It is inspired by learning-based monocular depth estimation methods using normal constraints for structure preservation. GeoNet [29] enforces a surface normal constraint, but their results have artifacts due to noise in the ground truth depth. Yin et al. [42] propose a global geometric constraint, called *virtual normal*. However, they cannot preserve intrinsic geometric features of real surfaces and local high-curvature features. We therefore propose a new *Combined Normal Map (CNM)* constraint, attached to local features for both local high-curvature regions and global planar regions. For training our network, we use a differentiable least squares module to compute normals directly from the estimated depth and use the *CNM* as ground truth in addition to the standard depth loss. Experiments in Section 5.2 show that the use of this novel *CNM* constraint significantly improves the depth estimation accuracy and outperforms those approaches that use only local or global constraints.

Our second contribution is a new neural network that combines depth maps predicted with individual source images into one final reference-view depth map,

together with an occlusion probability map. It uses a novel occlusion-aware loss function which assigns higher weights to the non-occluded regions. Importantly, this network is trained without any occlusion ground truth.

We experimentally show that our method significantly outperforms the state-of-the-art multi-view stereo from monocular video, both quantitatively and qualitatively. Furthermore, we show that our depth estimation algorithm, when integrated into a fusion-based handheld 3D scene scanning approach, enables interactive scanning of man-made scenes and objects in much higher shape quality (See Fig. 6).

## 2 Related Work

**Multi-view stereo.** MVS algorithms [14, 4, 40, 13, 43] are able to reconstruct 3D models from images under the assumptions of known materials, viewpoints, and lighting conditions. The MVS methods vary significantly according to different scene representations, but the typical workflow is to use hand-crafted photo-consistency metrics to build a cost volume and do the cost aggregation and estimate the 3D geometry from the aggregated cost volume. These methods tend to fail for thin objects, non-diffuse surfaces, or the objects with insufficient features.

**Learning-based depth estimation.** Recently some learning-based methods achieve compelling results in depth estimation. They can be categorized into four groups: i) single-view depth estimation [9, 25, 22, 10]; ii) two-view stereo depth estimation [44, 26, 34, 5, 20]; iii) multi-view stereo depth estimation [16, 41, 18]; iv) depth estimation from a video sequence [36, 23]. Single-view depth estimation is an ill-posed problem due to inherent depth ambiguity. On the other hand, two-view and multi-view depth estimation [11] is also challenging due to the difficulties in dense correspondence matching and depth prediction in featureless or specular regions. Some two-view [44, 26, 5, 20] and multi-view [16, 36, 41, 18] depth estimation algorithms have produced promising results in accuracy and speed by integrating cost volume generation, cost volume aggregation, disparity optimization and disparity refinement in an end-to-end network.

Depth estimation from video frames is becoming popular with the high demand in emerging areas, such as AR/VR, robot navigation, autonomous driving and view-dependent realistic rendering. Wang et al. [36] design a real-time multi-view depth estimation network. Liu et al. [23] propose a Bayesian filtering framework to use frame coherence to improve depth estimation. Since these methods enforce depth constraint only, they often fail to preserve important geometric feature (e.g., corners, edges, planes) of man-made scenes.

**Surface normal constraint.** Depth-normal consistency has been explored before for the depth estimation task [8, 46, 29, 42]. Eigen et al. [8] propose a neural network with three decoders to separately predict depth, surface normal and segmentation. Zhang et al. [46] enforce the surface normal constraint for depth completion. Qi et al. [29] incorporate geometric relations between depth and surface normal by introducing the depth-to-normal and normal-to-depth networks. The

performance of these methods suffers from the noise in surface normal stemming from ground truth depth. Yin et al. [42] propose a global geometric constraint, called *virtual normal*, which is defined as the normal of a virtual surface plane formed by three randomly sampled non-collinear points with a large distance. The *virtual normal* is unable to faithfully capture the geometric features of real surfaces and the local high-curvature features. Kusupati et al. [21] use the sobel operator to calculate the spatial gradient of depth and enforce the consistency of the spatial gradient and the normal in the pixel coordinate space. However the gradient calculated by the sobel operator is very sensitive to local noise, and causes obvious artifacts in the surface normal calculated from estimated depth (See Fig. 5 for the visual result).

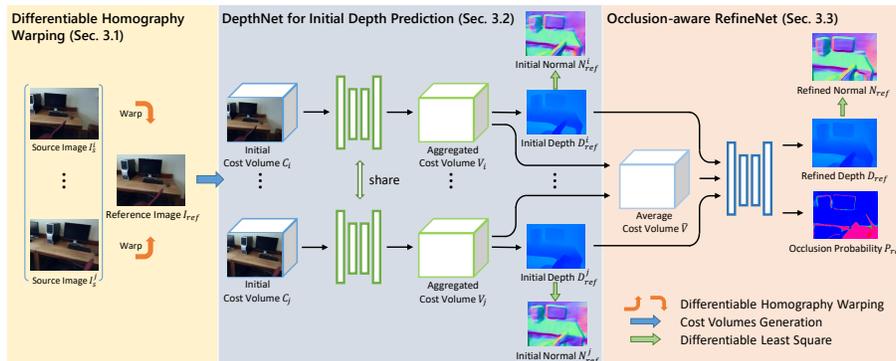
**Occlusion handling for depth estimation.** Non-learning methods [19, 7, 39, 37, 27, 1] use post-processing, such as left-right consistency check [7], to handle the occlusion issue. There are some learning-based methods for handling occlusion as well. Ilg et al. [17] and Wang et al. [35] predict occlusions by training a neural network on a synthetic dataset in a supervised manner. Qiu et al. [30] directly learn an intermediate occlusion mask for a single image; it has no epipolar geometry guarantee and just drives network to learn experienced patterns. We propose an occlusion-aware strategy to jointly refine depth prediction and estimate an occlusion probability map for multi-view depth estimation without any ground truth occlusion labels.

### 3 Method

In this section we describe the proposed network, as outlined in Fig. 1. Our pipeline takes the frames in a local time window in the video as input. Note that in our setting the video frame rate is 30 fps and we sample video frames with the interval of 10.

For the sake of simplicity in exposition, we suppose that the time window size is 3. We take the middle frame as the reference image  $I_{ref}$ , and the two adjacent images as the two source images  $\{I_s^1, I_s^2\}$  which are supposed to have sufficient overlap with the reference image. Our goal is to compute an accurate depth map from the view of the reference image.

We first outline our method. With a differentiable homography warping operation [41], all the source images are first warped into a stack of different fronto-parallel planes of the reference camera to form an initial 3D cost volume (see Section 3.1). Next, cost aggregation is applied to the initial cost volume to rectify any incorrect cost values and then an initial depth map is extracted from the aggregated cost volume. Besides pixel-wise depth supervision over the initial depth map, we also enforce a novel local and global geometric constraint, namely *Combined Normal Map (CNM)*, for training the network to produce superior results (see Section 3.2). We then further improve the accuracy of depth estimation by applying a new occlusion-aware strategy to aggregate the depth predictions from different adjacent views into one depth map for the reference



**Fig. 1.** Overview of our method. The network input consists of one reference image and  $n$  ( $n = 2, 4, \dots$ ) source images. With homography warping, each source image is combined with the reference image as a pair to generate a cost volume. Then the cost volume is fed into the *DepthNet* to generate an initial depth map, with  $\ell_1$  depth supervision and the constraint of *Combined Normal Map*. Finally, with the  $n$  initial depth maps and the average cost volume of the aggregated cost volumes as input, the *RefineNet* employs an occlusion-aware loss to produce the final reference-view depth map and an occlusion probability map, with again the supervision by the depth and the *CNM* constraints.

view, with an occlusion probability map (see Section 3.3). The details of each step will be elaborated in the subsequent sections.

### 3.1 Differentiable Homography Warping

Consider the reference image  $I_{ref}$  and one of its source images, denoted  $I_s^i$ . We warp  $I_s^i$  into fronto-parallel virtual planes of  $I_{ref}$  to form an initial cost volume. Similar to [16, 41], we first uniformly sample  $D$  depth planes at depth  $d_n$  in a range  $[d_{min}, d_{max}]$ ,  $n = 1, 2, \dots, D$ . The mapping from the warped  $I_s^i$  to the  $n^{th}$  virtual plane of  $I_{ref}$  at depth  $d_n$  is determined by a planar homography transformation  $H_n$ , following the classical plane-sweeping stereo [12].

$$u'_n \sim H_n u, u'_n \sim \mathbf{K}[\mathbf{R}_{s,i} | \mathbf{t}_{s,i}] \begin{bmatrix} (\mathbf{K}^{-1}u)d_n \\ 1 \end{bmatrix}, \quad (1)$$

where  $u$  is the homogeneous coordinate of a pixel in reference image,  $u'_n$  is projected coordinate of  $u$  in source image  $I_s^i$  with virtual plane  $d_n$ .  $K$  denotes the intrinsic parameters of the camera,  $\{\mathbf{R}_{s,i}, \mathbf{t}_{s,i}\}$  are the rotation and the translation of the source image  $I_s^i$  relative to the reference image  $I_{ref}$ .

Next, we measure the visual consistency of the warped  $I_s^i$  and  $I_{ref}$  at depth  $d_n$  and build a cost volume  $C_i$  with the size of  $W \times H \times D$ , where  $W, H, D$  are the image width, image height and the number of depth planes, respectively. Unlike previous works [41, 23] that use extracted feature maps of an image pair for warping and building a 4D cost volume, here we use the image pair directly

to avoid the memory-heavy and time-consuming 3D convolution operation on a 4D cost volume.

### 3.2 DepthNet for Initial Depth Prediction

Similar to recent learning-based stereo [20] and MVS [41, 18] methods, after getting cost volumes  $\{C_i\}_{i=1,2}$  from image pairs  $\{I_{ref}, I_s^i\}_{i=1,2}$ , we first use the neural network *DepthNet* to do cost aggregation for each  $C_i$ , which rectifies the incorrect values by aggregating the neighbouring pixel values. Note that we feed  $C_i$  stacked with  $I_s$  together into the *DepthNet* in order to make use of more detailed context information as [36] does. We denote the aggregated cost volume as  $V_i$ . Next, we retrieve the initial depth map  $D_{ref}^i$  from the aggregated cost volume  $V_i$  using a 2D convolution layer. Note that two initial depth maps,  $D_{ref}^1$  and  $D_{ref}^2$ , are generated for the reference view  $I_{ref}$ .

We train the network with depth supervision. With only depth supervision, the point cloud converted from the estimated depth does not preserve regular features, such as sharp edges and planar regions. We therefore propose to also enforce the normal constraint for further improvement. Note that, instead of just using local surface normal [29] or just a global virtual normal [42], we use the so called *Combine Normal Map* (CNM) that combines the local surface normal and the global planar structural feature in an adaptive manner.

**Pixel-wise Depth Loss** We use a standard pixel-wise depth map loss as follows,

$$l_{id} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \left\| \hat{D}(q) - D_i(q) \right\|_1 \quad (2)$$

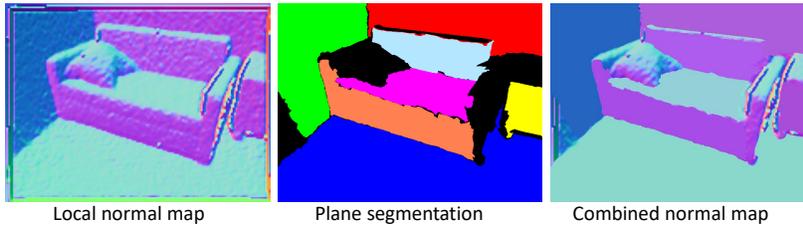
where  $\mathcal{Q}$  is the set of all pixels that are valid in ground truth depth,  $\hat{D}(q)$  is the ground truth depth value of pixel  $q$ , and  $D_i(q)$  is the initial estimated depth value of pixel  $q$ .

**Combined Normal Map** In order to preserve both local and global structures of scenes, we introduce the *Combined Normal Map* (CNM) as ground truth for the supervision with the normal constraint. To obtain this normal map, we first use PlaneCNN [24] to extract planar regions, such as walls, tables, and floors. Then we apply the local surface normals to non-planar regions, and use the mean of the surface normals in a planar region as the assigned to the region. The visual comparison of the local normal map and the CNM can be seen in Fig. 2.

The key insight here is to use local surface normal to capture rich local geometric features in the high-curvature regions and to employ the average normal to filter out noise in the surface normals for the planar regions to preserve global structures. In this way, the CNM significantly improves the depth prediction and the recovery of good 3D structures of the scene, compared to using only local or global normal supervision (see the ablation study in Section 5.4).

**Combined Normal Loss** We define the loss on the CNM as follows:

$$l_{in} = -\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \hat{N}(q) \cdot N_i(q), \quad (3)$$



**Fig. 2.** Visual comparison of local normal map and combined normal map.

where  $\mathcal{Q}$  is the set of valid ground truth pixels,  $\hat{N}(q)$  is the combined normal of pixel  $q$ , and  $N_i(q)$  is the surface tangent normal of the 3D point corresponding to the pixel  $q$ , both normalized to unit vectors.

To obtain an accurate depth map and preserve geometric features, we combine the Pixel-wise Depth Loss and the Combined Normal Loss together to supervise the network output. The overall loss is:

$$l_i = l_{id} + \lambda l_{in}, \quad (4)$$

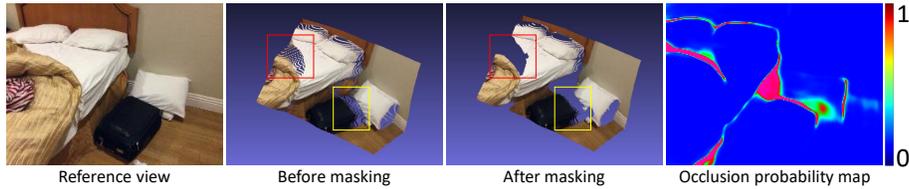
where  $\lambda$  is a trade-off parameter, which is set to 1 in all experiments.

### 3.3 Occlusion-aware RefineNet

The next step is to combine the initial depth maps  $D_{ref}^1$  and  $D_{ref}^2$  of the reference image predicted from different image pairs  $\{I_{ref}, I_s^i\}_{i=1,2}$  into one final depth map, which is denoted as  $D_{fin}$ . We design an occlusion-aware network, namely *RefineNet*, based on an occlusion probability map. Note that the occlusion refers to the region in  $I_{ref}$  where it cannot be observed in either  $I_s^1$  or  $I_s^2$ . In contrast to treating all pixels equally, when calculating the loss we assign the lower weights to the occluded region and the higher weights to the non-occluded region, which shifts the focus of the network to the non-occluded regions, since the depth prediction in the non-occluded regions is more reliable. Furthermore, the occlusion probability map predicted by the network can be used to filter out unreliable depth samples, as shown in Fig. 3, which is useful when the depth maps are fused for 3D reconstruction (see more details in Section 5.4).

Here we describe more technical details of this step. We design the *RefineNet* to predict the final depth map and the occlusion probability map from the average cost volume  $\bar{V}$  of the two cost volumes  $\{V_i\}_{i=1,2}$  and two initial depth maps  $\{D_{ref}^i\}_{i=1,2}$ . The *RefineNet* has one encoder and two decoders.

The first decoder estimates the occlusion probabilities based on the occlusion information encoded in the average cost volume  $\bar{V}$  and the initial depth maps. Intuitively, for a pixel in the non-occluded region, it has the strongest response (peak) at  $n^{th}$  layer with depth  $d_n$  of the average cost volume  $\bar{V}$ , and  $D_{ref}^1$  and  $D_{ref}^2$  at this pixel have similar depth values. However, for a pixel in the occluded region, it has scattered responses at the depth layers of  $\bar{V}$  and has very



**Fig. 3.** Efficiency of occlusion probability map. Masked by the occlusion map, the point cloud converted from estimated depth has fewer outlying points.

different values on the initial depth maps at this pixel. The other decoder predicts the refined depth map with the depth constraint and the CNM constraint, as described in Section 3.2

To train the *RefineNet*, we design a novel occlusion-aware loss function as follows,

$$L_{refine} = (l_{rd} + \beta \cdot l_{rn}) - \alpha \cdot \frac{1}{|\mathcal{Q}|} \cdot \sum_{q \in \mathcal{Q}} (1 - P(q)), \quad (5)$$

where

$$l_{rd} = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} (1 - P(q)) \left\| \hat{D}(q) - D_r(q) \right\|_1 \quad (6)$$

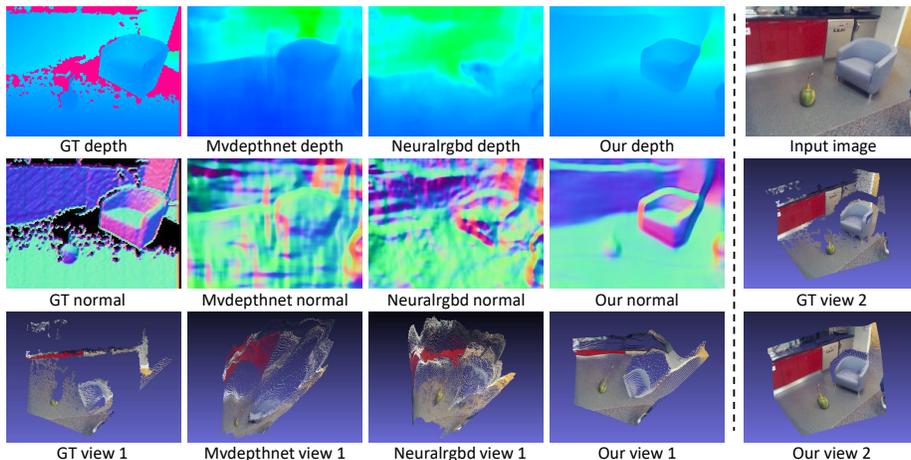
$$l_{rn} = -\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} (1 - P(q)) \hat{N}(q) \cdot N_r(q). \quad (7)$$

Here  $D_r(q)$  denotes the refined estimated depth at pixel  $q$ ,  $N_r(q)$  denotes the surface normal of the 3D point corresponding to  $q$ ,  $\hat{D}(q)$  is the ground truth depth of  $q$ ,  $\hat{N}(q)$  is the combined normal of  $q$ , and  $P(q)$  denotes the occlusion probability value of  $q$  ( $P(q)$  is high when  $q$  is occluded). The weight  $\alpha$  is set to 0.2 and  $\beta$  is set to 1 in all experiments.

## 4 Datasets and Implementation Details

**Dataset** We train our network with the ScanNet dataset [6], and evaluate our method on the 7scenes dataset [32] and the SUN3D dataset [38]. ScanNet consists of 1600 different indoor scenes, which are divided into 1000 for training and 600 for testing. ScanNet provides RGB images, ground truth depth maps and camera poses. We generate the *CNM* as described in Section 3.2.

**Implementation details** Our training process consists of three stages. First, we train the *DepthNet* using the loss function defined in Equation 4. Then, we fix the parameters of the *DepthNet* and train the *RefineNet* with the loss function Equation 5. Next, we finetune the parameters of the *DepthNet* and the *RefineNet* together with the loss terms in Equation 4 and Equation 5. For all the training stages, the ground truth depth map and the *CNM* are used as supervision. We use Adam optimizer ( $lr = 0.0001$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $weight\_decay = 0.00001$ )



**Fig. 4.** Visual depth comparison with mvdepthnet [36] and neuralrgb [23]. Our estimated depth maps preserve shape regularity better than by mvdepthnet and neuralrgb, for example, in the regions of sofa, cabinet and wall. The result by mvdepthnet (column 2) shows strong discontinuity of depth value at the lower end of the red wall, which is incorrect. We also show the visualization of normal maps computed from these depth maps to further show the superior quality of our depth estimation in terms of shape regularity. Note that the red region of GT depth (black region of GT normal) is invalid due to scanning failure of the Kinect sensor. More comparisons can be found in the supplementary materials

and run for 6 epochs for each stage. We implemented our model in Pytorch [28]. Training the network with two GeForce RTX 2080 Ti GPUs, which takes two days for all stages.

## 5 Experiments

To evaluate our method, we compare our method with state-of-the-arts in three aspects: accuracy of depth estimation, geometric consistency, and video-based 3D reconstruction with TSDF fusion.

### 5.1 Evaluation Metrics

For depth accuracy, we compute the widely-used statistical metrics defined in [9]: i) accuracy under threshold ( $\sigma < 1.25^i$  where  $i \in \{1, 2, 3\}$ ); ii) scale-invariant error (scale.inv); iii) absolute relative error (abs.rel); iv) square relative error (sq.rel); v) root mean square error (rmse); vi) rmse in log space (rmse log).

We evaluate the surface normal accuracy with the following metrics used in the prior works [8, 29]: i) the mean of angle error (mean); ii) the median of the angle error (median); iii) rmse; iv) the percentage of pixels with angle error below threshold  $t$  where  $t \in [11.25^\circ, 22.5^\circ, 30^\circ]$ .

**Table 1.** Comparison of depth estimation over 7-Scenes dataset [32] with the metrics defined in [9].

	$\sigma < 1.25$	$\sigma < 1.25^2$	$\sigma < 1.25^3$	abs.rel	sq.rel	rmse	rmse log	scale.inv
DORN [10]	60.05	87.76	96.33	0.2000	0.1153	0.4591	0.2813	0.2207
GeoNet [29]	55.10	84.46	94.76	0.2574	0.1762	0.5253	0.3952	0.3318
Yin et al. [42]	60.62	88.81	97.44	0.2154	0.1245	0.4500	0.2597	0.1660
DeMoN [34]	31.88	61.02	82.52	0.3888	0.4198	0.8549	0.4771	0.4473
MVSNet-retrain <sup>†</sup> [41]	64.09	87.73	95.88	0.2339	0.1904	0.5078	0.2611	0.1783
neuralrgbd <sup>†</sup> [23]	69.26	91.77	96.82	0.1758	0.1123	0.4408	0.2500	0.1899
mvdepthnet [36]	71.97	92.00	97.31	0.1865	0.1163	0.4124	0.2256	0.1691
mvdepthnet-retrain <sup>†</sup> [36]	71.79	92.56	97.83	0.1925	0.2350	0.4585	0.2301	0.1697
DPSNet [18]	63.65	85.73	94.33	0.2710	0.2752	0.5632	0.2759	0.1856
DPSNet-retrain <sup>†</sup> [18]	70.96	91.42	97.17	0.1991	0.1420	0.4382	0.2284	0.1685
Kusupati et al. <sup>†</sup> [21]	73.12	92.33	97.78	0.1801	0.1179	0.4120	0.2184	/
ours <sup>†</sup>	<b>76.64</b>	<b>94.46</b>	<b>98.56</b>	<b>0.1612</b>	<b>0.0832</b>	<b>0.3614</b>	<b>0.2049</b>	<b>0.1603</b>

<sup>†</sup> Trained on ScanNet dataset.

## 5.2 Comparisons

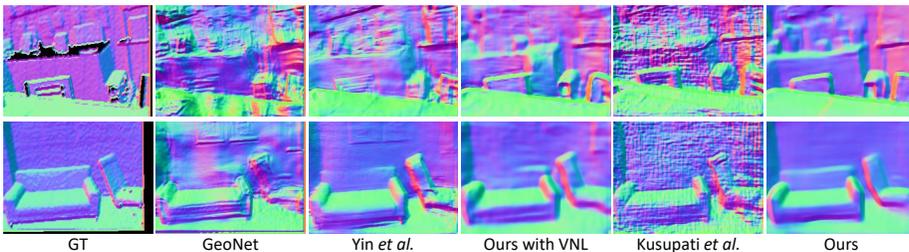
**Depth prediction** We compare our method with several other depth estimation methods. We categorize them according to their input formats: i) one single image: DORN [10], GeoNet [29] and VNL [42]; ii) two images from monocular camera: DeMoN [34]; iii) multiple unordered images: MVSNet [41], DPSNet [18], kusupati et al. [21]; iv) a video sequence: mvdepthnet [36] and neuralrgbd [23]. The models provided by these methods were trained on different datasets and are evaluated on a separate dataset, 7Scenes dataset [32]. For the multi-view depth estimation methods, neuralrgbd [23] and kusupati et al. [21] are trained on ScanNet, MVSNet [41] is trained on DTU dataset [2], mvdepthnet [36] and DPSNet [18] are trained on mixed datasets (SUN3D [38], TUM RGB-D [33], MVS datasets [34], SceneNN [15] and synthetic dataset Scenes11 [34]). For fair comparison, we further retrained MVSNet [41], mvdepthnet [36] and DPSNet [18] on ScanNet. Table 1 shows that our method outperforms other methods in terms of all evaluation metrics.

**Visual comparison** In Fig. 4, compared to mvdepthnet [36] and neuralrgbd [23], our estimated depth map has less noise, sharper boundaries and spatially consistent depth values, which can be also seen in the surface normal visualization. Furthermore, the 3D point cloud exported from the estimated depth preserves global planar features and local features in the high-curvature regions. More comparison examples are included in the supplementary materials.

**Surface normal accuracy** To evaluate the accuracy of normal calculated from estimated depth, we choose two single-view depth estimation methods: GeoNet [29] and Yin et al. [42], and one multi-view depth estimation method: Kusupati et al. [21]. GeoNet [29] incorporates surface normal constraint to depth estimation, while Yin et al. [42] propose a global normal constraint, namely *Virtual Normal*, which is defined by three randomly sampled non-collinear points.

**Table 2.** Evaluation of the calculated surface normal from estimated depth on 7scenes[32] and SUN3D [38]. “Ours with VNL” denotes our model retrained using virtual normal loss [42].

Dataset	7scenes						SUN3D					
	error			accuracy			error			accuracy		
Method	mean	median	rmse	11.25°	22.5°	30°	mean	median	rmse	11.25°	22.5°	30°
GeoNet [29]	57.04	52.43	65.50	4.396	15.16	23.83	47.39	40.25	57.11	9.311	26.63	37.49
Yin et al. [42]	45.17	37.77	54.77	11.05	29.76	40.78	38.50	30.03	48.90	19.06	40.32	51.17
Ours with VNL	43.08	34.98	53.15	13.84	33.37	43.90	31.68	22.10	42.74	29.50	52.97	62.81
Kusupati et al. [21]	55.54	50.46	64.28	5.774	18.03	26.81	52.86	47.03	62.30	7.507	22.56	32.22
Ours	<b>36.34</b>	<b>27.21</b>	<b>46.46</b>	<b>20.24</b>	<b>43.36</b>	<b>54.11</b>	<b>29.21</b>	<b>20.89</b>	<b>38.99</b>	<b>30.45</b>	<b>55.05</b>	<b>65.25</b>



**Fig. 5.** Visual comparison of surface normal calculated from 3D point cloud exported from estimated depth with GeoNet [29], Yin et al. [42] and Kusupati et al. [21].

Kusupati et al. [21] use the sobel operator to calculate depth gradient and enforce the consistency of the depth gradient and the normal in the pixel coordinate space. To demonstrate that the improved performance of our method indeed benefits from the new CNM constraint (rather than entirely due to our multi-view input), we also retrained our network by replacing the Combined Normal Loss with the Virtual Normal loss (denoted as “Ours with VNL” in Table 2) on ScanNet dataset.

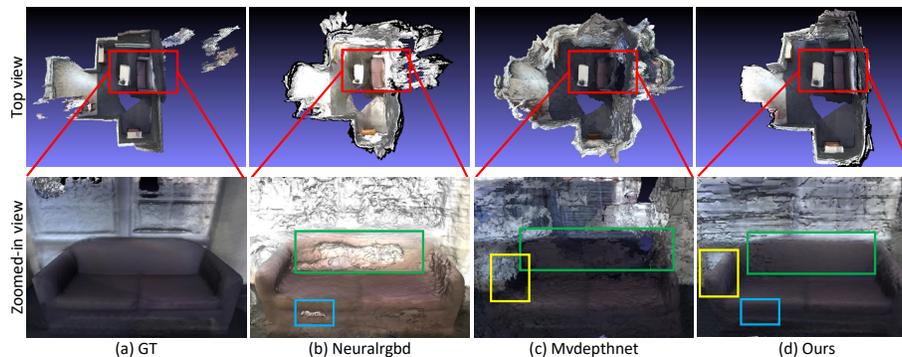
As shown in Table 2 and Fig. 5, our method outperforms GeoNet [29], Yin et al. [42] and Kusupati et al. [21] both quantitatively and qualitatively. Compared with our model retrained using VNL, our model with the *CNM* constraint works much better, which preserves local and global features.

### 5.3 Video Reconstruction

With the high-fidelity depth map and the occlusion probability map obtained by our method, high-quality reconstruction of video even in texture-less environments can be achieved by applying TSDF fusion method [45]. We compare our method with mvdepthnet [36] and neuralrgbd [23]. Note that these three methods are all trained on the ScanNet dataset. As shown in Fig. 6, even for white walls, feature-less sofa and table, our reconstructed result is much better than the other two methods in the aspects of local and global structural recovery, and completion. Also, the color of our reconstruction is closest to the color of

**Table 3.** Evaluation of the usefulness of *CNM* and the occlusion-aware refinement module. The first four rows show the results without any constraint, with only local normal constraint, with only global normal constraint, and with the *CNM* constraint respectively. The last two rows show the results without/with the occlusion-aware loss.

Components				Estimated depth evaluation					Calculated normal evaluation					
local	global	refine	occlu.	1.25	1.25 <sup>2</sup>	abs.rel	sq.rel	rmse	scale.inv	11.25°	22.5°	mean	median	rmse
×	×	×	×	71.79	92.56	0.1925	0.2350	0.4585	0.1697	9.877	27.20	46.62	39.91	55.96
✓	×	×	×	71.95	92.45	0.1899	0.1188	0.4060	0.1589	16.97	39.49	37.92	29.60	47.53
×	✓	×	×	66.85	90.17	0.2096	0.1462	0.4570	0.1752	15.39	37.09	39.15	31.33	48.45
✓	✓	×	×	73.17	92.75	0.1812	0.1076	0.3952	0.1654	17.67	40.23	37.65	29.18	47.36
✓	✓	✓	×	74.77	93.22	0.1726	0.0999	0.3877	0.1758	18.61	41.13	37.67	28.77	47.74
✓	✓	✓	✓	<b>75.80</b>	<b>93.79</b>	<b>0.1669</b>	<b>0.0909</b>	<b>0.3731</b>	<b>0.1638</b>	<b>20.12</b>	<b>42.76</b>	<b>36.82</b>	<b>27.67</b>	<b>47.03</b>



**Fig. 6.** Comparison with neuralrgbD [23] and mvdepthnet [36] for 3D reconstruction on a scene from ScanNet. (a) With ground truth depth; (b) With estimated depth and confidence map from neuralrgbD; (c) With estimated depth from mvdepthnet; (d) With our estimated depth and occlusion probability map. All reconstructions are done with TSDF fusion [45] using 110 frames uniformly selected from a video of 1100 frames.

the ground truth. This confirms that the individual depth maps have less noise and high-consistency so the colors would not be mixed up in the fusion process.

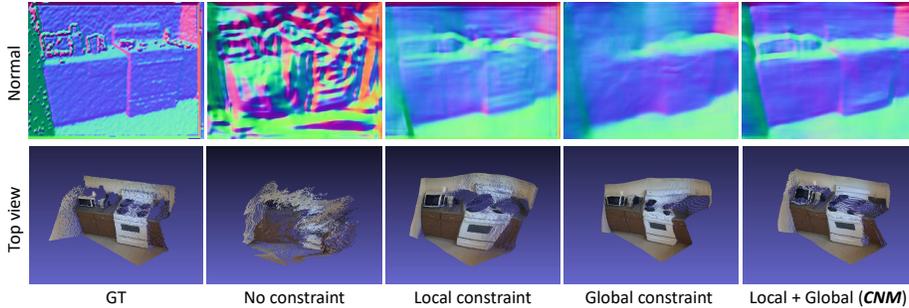
#### 5.4 Ablation studies

Compared with prior works, we explicitly drive the depth estimation network to adaptively learn local surface directions and global planar structures. Taking multiple views as inputs, our refinement module jointly refines depth and predicts occlusion probability. In this section, we evaluate the usefulness of each component.

**Combined normal map** As shown in Table 3, enforced by our *CNM* constraint (local + global), our model achieves better performance in terms of estimated depth and calculated normal from estimated depth, compared to that without geometric constraint, only with local normal constraint, and only with global

**Table 4.** Effect of the number of source views  $N$  and the choice of reference view. We use the model trained with  $N = 2$ , and test with different number of views over 7-Scenes dataset [32]. The first row shows the result of using the last frame in the local window as the reference frame. The other rows are the results of using the middle frame as the reference frame.

view num	Estimated depth evaluation					Calculated normal evaluation				
	1.25	1.25 <sup>2</sup>	abs.rel	rmse	scale.inv	11.25°	22.5°	mean	median	rmse
2 views (last)	72.88	92.70	0.1750	0.3910	0.1686	19.14	41.83	37.17	28.27	47.20
2 views	75.80	93.79	0.1669	0.3731	0.1638	20.12	42.76	36.82	27.67	47.03
4 views	76.29	94.28	0.1647	0.3652	0.1608	20.30	43.17	36.57	27.37	46.78
6 views	76.64	94.46	0.1612	0.3614	0.1602	20.37	43.14	36.53	27.37	46.72

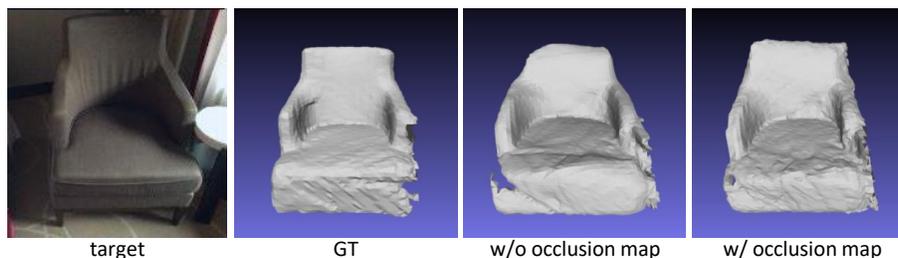


**Fig. 7.** Effects of using local surface normal and *CNM* as constraints for supervision.

normal constraint (the plane normal constraint). Fig. 7 also shows that with *CNM*, the calculated normal is more consistent in planar regions and the point cloud converted from estimated depth keeps better shape.

**The number of source views for refinement** Our refinement module can allow any arbitrary even number of source views as input, and generate a refined depth map and an occlusion probability map. In Table 3, with refinement module, the performance of our model has been significantly improved. As shown in Table 4, the quality of refined depth will be improved gradually with the increase of source views. A forward pass with 2/4/6 source views all takes nearly 0.02s on a GeForce RTX 2080 Ti GPU. Furthermore, we evaluate how the performance is affected by the choice of the reference view. We find that the performance of our model will degrade significantly if we use the last frame as the reference view rather than the middle frame in the local time window. This is because the middle frame shares more overlapping areas than the last frame with the other frames in the time window.

**The usefulness of occlusion probability map** Unlike left-right check as a post-processing operation, our *RefineNet* jointly refines depth and generates an occlusion probability map. As shown in Table. 3, our model with the occlusion-aware refinement achieves better results than that with a naive refinement (w/o



**Fig. 8.** Effect of occlusion probability map for video reconstruction. The occlusion probability map provides weighting parameters into TSDF fusion system for fusing estimated depth maps, and enables reconstructed model to keep sharp boundaries.

occlusion map) and without any refinement. The naive refinement treats every pixel equally regardless whether it is on the occluded region or non-occluded region. In contrast, our occlusion-aware refinement penalizes occluded pixels less and pays more attention to the non-occluded region. In Fig. 3, the point cloud converted from estimated depth becomes cleaner and has sharp boundaries by applying the occlusion probability map to the multi-view depth integration process. Moreover unlike binary mask, our occlusion map can be easily used in TSDF fusion system [45] as weights. In Fig. 8, the fused mesh using estimated depth and the occlusion probability map has less artifact than that without occlusion map.

## 6 Conclusion and Limitations

In this paper, we propose a new method for estimating depth from a video sequence. There are two main contributions. We propose a novel normal-based constraint *CNM*, which is designed to preserve local and global geometric features for depth estimation, and a new occlusion-aware strategy to aggregate multiple depth predictions into one single depth map. Experiments demonstrate that our method outperforms than the state-of-the-art in terms of the accuracy of depth prediction and the recovery of geometric features. Furthermore, the high-fidelity depth prediction and the occlusion detection make the highly-detailed reconstruction with only a commercial RGB camera possible.

Now we discuss the limitations of our work and possible future directions. First, the performance of our method relies on the quality of the *CNM*, which is based on the segmentation of global planar regions. It has been observed that existing plane segmentation methods are not robust for all the scenes. One possible solution is to jointly learn the segmentation labels and the depth prediction. Second, for the task of video-based 3D reconstruction, our work can be extended by designing an end-to-end network to directly generate 3D reconstruction from video, rather than having to invoke the explicit step of using TSDF fusion to integrate the estimated depth maps.

## References

1. Learning to find occlusion regions. In: CVPR 2011. pp. 2161–2168. IEEE (2011)
2. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision* **120**(2), 153–168 (2016)
3. Alvarez, H., Paz, L.M., Sturm, J., Cremers, D.: Collision avoidance for quadrotors with a monocular camera. In: *Experimental Robotics*. pp. 195–209. Springer (2016)
4. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo-stereo matching with slanted support windows. In: *Bmvc*. vol. 11, pp. 1–11 (2011)
5. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5410–5418 (2018)
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 5828–5839 (2017)
7. Egnal, G., Wildes, R.P.: Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transactions on pattern analysis and machine intelligence* **24**(8), 1127–1133 (2002)
8. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2650–2658 (2015)
9. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: *Advances in neural information processing systems*. pp. 2366–2374 (2014)
10. Fu, H., Gong, M., Wang, C., Batmanghelich, K., Tao, D.: Deep ordinal regression network for monocular depth estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2002–2011 (2018)
11. Furukawa, Y., Hernández, C.: Multi-view stereo: A tutorial. *Found. Trends. Comput. Graph. Vis.* **9**(1-2), 1–148 (Jun 2015). <https://doi.org/10.1561/06000000052>, <http://dx.doi.org/10.1561/06000000052>
12. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1–8. IEEE (2007)
13. Hosni, A., Bleyer, M., Gelautz, M., Rhemann, C.: Local stereo matching using geodesic support weights. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. pp. 2093–2096. IEEE (2009)
14. Hosni, A., Rhemann, C., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(2), 504–511 (2012)
15. Hua, B.S., Pham, Q.H., Nguyen, D.T., Tran, M.K., Yu, L.F., Yeung, S.K.: Scennn: A scene meshes dataset with annotations. In: *2016 Fourth International Conference on 3D Vision (3DV)*. pp. 92–101. IEEE (2016)
16. Huang, P.H., Matzen, K., Kopf, J., Ahuja, N., Huang, J.B.: Deepmvs: Learning multi-view stereopsis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2821–2830 (2018)
17. Ilg, E., Saikia, T., Keuper, M., Brox, T.: Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 614–630 (2018)

18. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: end-to-end deep plane sweep stereo. arXiv preprint arXiv:1905.00538 (2019)
19. Kang, S.B., Szeliski, R., Chai, J.: Handling occlusions in dense multi-view stereo. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. vol. 1, pp. I–I. IEEE (2001)
20. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: The IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
21. Kusupati, U., Cheng, S., Chen, R., Su, H.: Normal assisted stereo depth estimation. arXiv preprint arXiv:1911.10444 (2019)
22. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2041–2050 (2018)
23. Liu, C., Gu, J., Kim, K., Narasimhan, S.G., Kautz, J.: Neural rgb (r) d sensing: Depth and uncertainty from a video camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10986–10995 (2019)
24. Liu, C., Kim, K., Gu, J., Furukawa, Y., Kautz, J.: Planercnn: 3d plane detection and reconstruction from a single image. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
25. Liu, F., Shen, C., Lin, G.: Deep convolutional neural fields for depth estimation from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5162–5170 (2015)
26. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5695–5703 (2016)
27. Min, D., Sohn, K.: Cost aggregation and occlusion handling with wls in stereo matching. *IEEE Transactions on Image Processing* **17**(8), 1431–1442 (2008)
28. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch (2017)
29. Qi, X., Liao, R., Liu, Z., Urtasun, R., Jia, J.: Geonet: Geometric neural network for joint depth and surface normal estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 283–291 (2018)
30. Qiu, J., Cui, Z., Zhang, Y., Zhang, X., Liu, S., Zeng, B., Pollefeys, M.: Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3313–3322 (2019)
31. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1. pp. 519–528. CVPR '06, IEEE Computer Society, Washington, DC, USA (2006). <https://doi.org/10.1109/CVPR.2006.19>, <http://dx.doi.org/10.1109/CVPR.2006.19>
32. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2930–2937 (2013)
33. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 573–580. IEEE (2012)

34. Ummenhofer, B., Zhou, H., Uhrig, J., Mayer, N., Ilg, E., Dosovitskiy, A., Brox, T.: Demon: Depth and motion network for learning monocular stereo. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5038–5047 (2017)
35. Wang, J., Zickler, T.: Local detection of stereo occlusion boundaries. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3818–3827 (2019)
36. Wang, K., Shen, S.: Mvdepthnet: real-time multiview depth estimation neural network. In: 2018 International Conference on 3D Vision (3DV). pp. 248–257. IEEE (2018)
37. Xiao, J., Shah, M.: Motion layer extraction in the presence of occlusion using graph cuts. *IEEE transactions on pattern analysis and machine intelligence* **27**(10), 1644–1659 (2005)
38. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1625–1632 (2013)
39. Xu, L., Jia, J.: Stereo matching: An outlier confidence approach. In: European Conference on Computer Vision. pp. 775–787. Springer (2008)
40. Yang, Q.: A non-local cost aggregation method for stereo matching. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1402–1409. IEEE (2012)
41. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 767–783 (2018)
42. Yin, W., Liu, Y., Shen, C., Yan, Y.: Enforcing geometric constraints of virtual normal for depth prediction. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
43. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. *IEEE transactions on pattern analysis & machine intelligence* (4), 650–656 (2006)
44. Zbontar, J., LeCun, Y.: Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1592–1599 (2015)
45. Zeng, A., Song, S., Nießner, M., Fisher, M., Xiao, J., Funkhouser, T.: 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In: CVPR (2017)
46. Zhang, Y., Funkhouser, T.: Deep depth completion of a single rgb-d image. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
47. Zollhöfer, M., Stotko, P., Görnitz, A., Theobalt, C., Nießner, M., Klein, R., Kolb, A.: State of the Art on 3D Reconstruction with RGB-D Cameras. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)* **37**(2) (2018)