PlaneRCNN: 3D Plane Detection and Reconstruction from a Single Image

Chen Liu[1,2*]   Kihwan Kim[1]   Jinwei Gu[1,3*]   Yasutaka Furukawa[4]   Jan Kautz[1]

[1]NVIDIA   [2]Washington University in St. Louis

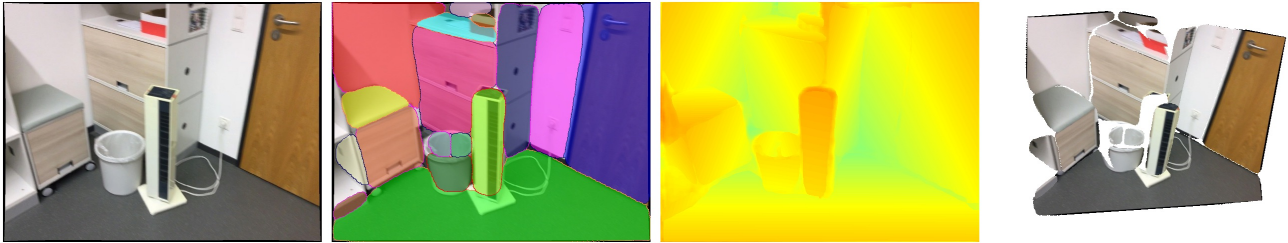[3]SenseTime   [4]Simon Fraser University

Figure 1. This paper proposes a deep neural architecture, PlaneRCNN, that detects planar regions and reconstructs a piecewise planar depthmap from a single RGB image. From left to right, an input image, segmented planar regions, estimated depthmap, and reconstructed planar surfaces.

Abstract

This paper proposes a deep neural architecture, PlaneR-CNN, that detects and reconstructs piecewise planar surfaces from a single RGB image. PlaneRCNN employs a variant of Mask R-CNN to detect planes with their plane parameters and segmentation masks. PlaneRCNN then jointly refines all the segmentation masks with a novel loss enforcing the consistency with a nearby view during training. The paper also presents a new benchmark with more fine-grained plane segmentations in the ground-truth, in which, PlaneRCNN outperforms existing state-of-the-art methods with significant margins in the plane detection, segmentation, and reconstruction metrics. PlaneR-CNN makes an important step towards robust plane extraction, which would have an immediate impact on a wide range of applications including Robotics, Augmented Reality, and Virtual Reality. Code and data are available at https://research.nvidia.com/publication/2019-06_PlaneRCNN.

1. Introduction

Planar regions in 3D scenes offer important geometric cues in a variety of 3D perception tasks such as scene understanding [42], scene reconstruction [3], and robot navigation [18, 56]. Accordingly, piecewise planar scene reconstruction has been a focus of computer vision research

for many years, for example, plausible recovery of planar structures from a single image [16], volumetric piecewise planar reconstruction from point clouds [3], and Manhattan depthmap reconstruction from multiple images [11].

A difficult yet fundamental task is the inference of a piecewise planar structure from a single RGB image, posing two key challenges. First, 3D plane reconstruction from a single image is an ill-posed problem, requiring rich scene priors. Second, planar structures abundant in man-made environments often lack textures, requiring global image understanding as opposed to local texture analysis. Recently, PlaneNet [27] and PlaneRecover [49] made a breakthrough by introducing the use of Convolutional Neural Networks (CNNs) and formulating the problem as a plane segmentation task. While generating promising results, they suffer from three major limitations: 1) Missing small surfaces; 2) Requiring the maximum number of planes in a single image a priori; and 3) Poor generalization across domains (e.g., trained for indoors images and tested outdoors).

This paper proposes a novel deep neural architecture, PlaneRCNN, that addresses these issues and more effectively infers piecewise planar structure from a single RGB image (Fig. 1). PlaneRCNN consists of three components.

The first component is a plane detection network built upon Mask R-CNN [14]. Besides an instance mask for each planar region, we also estimate the plane normal and per-pixel depth values. With known camera intrinsics, we can further reconstruct the 3D planes from the detected planar regions. This detection framework is more flexible and can

---

handle an arbitrary number of planar regions in an image. To the best of our knowledge, this paper is the first to introduce a detection network, common in object recognition, to the depthmap reconstruction task. The second component is a segmentation refinement network that jointly optimizes extracted segmentation masks to more coherently explain a scene as a whole. The refinement network is designed to handle an arbitrary number of regions via a simple yet effective neural module. The third component, the warping-loss module, enforces the consistency of reconstructions with another view observing the same scene *during training* and improves the plane parameter and depthmap accuracy in the detection network via end-to-end training.

The paper also presents a new benchmark for the piecewise planar depthmap reconstruction task. We collected 100,000 images from ScanNet [6] and generated the corresponding ground-truth by utilizing the associated 3D scans. The new benchmark offers 14.7 plane instances per image on the average, in contrast to roughly 6 instances per image in the existing benchmark [27].

The performance is evaluated via plane detection, segmentation, and reconstruction metrics, in which PlaneR-CNN outperforms the current state-of-the-art with significant margins. Especially, PlaneRCNN is able to detect small planar surfaces and generalize well to new scene types.

The contributions of the paper are two-fold:

Technical Contribution: The paper proposes a novel neural architecture PlaneRCNN, where 1) a detection network extracts an arbitrary number of planar regions; 2) a refinement network jointly improves all the segmentation masks; and 3) a warping loss improves plane-parameter and depthmap accuracy via end-to-end training.

System Contribution: The paper provides a new benchmark for the piecewise planar depthmap reconstruction task with much more fine-grained annotations than before, in which PlaneRCNN makes significant improvements over the current state-of-the-art.

2. Related Work

For 3D plane detection and reconstruction, most traditional approaches [10, 12, 37, 38, 52] require multiple views or depth information as input. They generate plane proposals by fitting planes to 3D points, then assign a proposal to each pixel via a global inference. Deng *et al*. [7] proposed a learning-based approach to recover planar regions, while still requiring depth information as input.

Recently, PlaneNet [27] revisited the piecewise planar depthmap reconstruction problem with an end-to-end learning framework from a single indoor RGB image. PlaneRecover [49] later proposed an un-supervised learning approach for outdoor scenes. Both PlaneNet and PlaneRecover formulated the task as a pixel-wise segmentation

problem with a fixed number of planar regions (i.e., 10 in PlaneNet and 5 in PlaneRecover), which severely limits the expressiveness of their reconstructions and generalization capabilities to different scene types. We address these limitations by utilizing a detection network, commonly used for object recognition.

Detection-based framework has been successfully applied to many 3D understanding tasks for objects, for example, predicting object shapes in the form of bounding boxes [5, 9, 32], wire-frames [22, 47, 57], or template-based shape compositions [2, 21, 31, 48]. However, the coarse representation employed in these methods lack the ability to accurately model complex and cluttered indoor scenes.

In addition to the detection, joint refinement of segmentation masks is also a key to many applications that require precise plane parameters or boundaries. In recent semantic segmentation techniques, fully connected conditional random field (CRF) is proven to be effective for localizing segmentation boundaries [4, 20]. CRFasRNN [55] further makes it differentiable for end-to-end training. CRF only utilizes low-level information, and global context is further exploited via RNNs [1, 23, 36], more general graphical models [30, 24], or novel neural architectural designs [53, 54, 51]. These segmentation refinement techniques are NOT instance-aware, merely inferring a semantic label at each pixel and cannot distinguish multiple instances belonging to the same semantic category.

Instance-aware joint segmentation refinement poses more challenges. Traditional methods [39, 40, 41, 43, 50] model the scene as a graph and use graphical model inference techniques to jointly optimize all instance masks. With a sequence of heuristics, these methods are often not robust. To this end, we will propose a segmentation refinement network that jointly optimizes an arbitrary number of segmentation masks on top of a detection network.

3. Approach

PlaneRCNN consists of three main components (See Fig. 2): a plane detection network, a segmentation refinement network, and a warping loss module. Built upon Mask R-CNN [14], the plane proposal network (Sec. 3.1) detects planar regions given a single RGB image and predicts 3D plane parameters together with a segmentation mask for each planar region. The refinement network (Sec. 3.2) takes all detected planar regions and jointly optimizes their masks. The warping loss module (Sec. 3.3) enforces the consistency of reconstructed planes with another view observing the same scene to further improve the accuracy of plane parameters and depthmap during training.

3.1. Plane Detection Network

Mask R-CNN was originally designed for semantic segmentation, where images contain instances of varying cat-
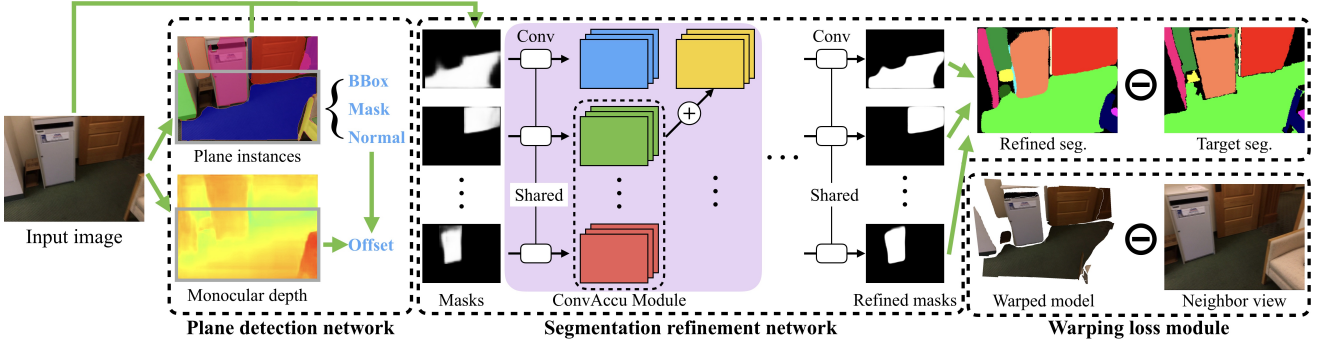
Figure 2. Our framework consists of three building blocks: 1) a plane detection network based on Mask R-CNN [14], 2) a segmentation refinement network that jointly optimizes extracted segmentation masks, and 3) a warping loss module that enforces the consistency of reconstructions with a nearby view during training.

egories (e.g., person, car, train, bicycle and more). Our problem has only two categories "planar" or "non-planar", defined in a geometric sense. Nonetheless, Mask R-CNN works surprisingly well in detecting planes in our experiments. It also enables us to handle an arbitrary number of planes, where existing approaches need the maximum number of planes in an image a priori (i.e., 10 for PlaneNet [27] and 5 for PlaneRecover [49]).

We treat each planar region as an object instance and let Mask R-CNN detect such instances and estimate their segmentation masks. The remaining task is to infer 3D plane parameters, which consists of the normal n and the offset information d (See Fig. 3). While CNNs have been successful for depthmap [28] and surface normal [45] estimation, direct regression of plane offset turns out to be a challenge (even with the use of CoordConv [29]). Instead of direct regression, we solve it in three steps: (1) predict a normal per planar instance, (2) estimate a depthmap for an entire image, and (3) use a simple algebraic formula (Eq. 1) to calculate the plane offset (which is differentiable for end-to-end training). We now explain how we modify Mask-RCNN to perform these three steps.

Plane normal estimation: Directly attaching a parameter regression module after the ROI pooling produces reasonable results, but we borrow the idea of 2D anchor boxes for bounding box regression [14] to further improve accuracy. More precisely, we consider *anchor normals* and estimate a plane normal in the local camera coordinate frame by 1) picking an anchor normal, 2) regressing the residual 3D vector, and 3) normalizing the sum to a unit-length vector.

Anchor normals are defined by running the K-means clustering algorithm on the plane normals in $10,000$ randomly sampled training images. We use $k = 7$ and the cluster centers become anchor normals, which are up-facing, down-facing, and horizontal vectors roughly separated by $45°$ in our experiments (See Fig. 3).

We replace the object category prediction in the original Mask R-CNN with the anchor ID prediction, and append

one separate fully-connected layer to regress the 3D residual vector for each anchor normal (i.e., $21 = 3 \times 7$ output values). To generate supervision for each ground-truth plane normal, we find the closest anchor normal and compute the residual vector. We use the cross-entropy loss for the anchor normal selection, and the smooth L1 loss for the residual vector regression as in the bounding box regression of Mask R-CNN.
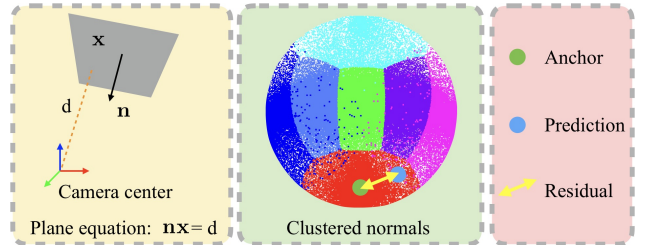

Figure 3. A 3D point $\mathbf{x}$ on the plane follows the equation $\mathbf{n}\mathbf{x} = d$. We estimate a plane normal $\mathbf{n}$ by first picking one of the 7 anchor normals and then regressing the residual 3D vector. Anchor normals are defined by running the K-means clustering algorithm on the ground-truth plane normal vectors.

Depthmap estimation: While local image analysis per region suffices for surface normal prediction, global image analysis is crucial for depthmap inference. We add a decoder after the feature pyramid network (FPN) [25] in Mask R-CNN to estimate the depthmap D for the entire image with the same resolution. Details of the decoder network can be found in the supplementary document.

Plane offset estimation: Given a plane normal $\mathbf{n}$, it is straightforward to estimate the plane offset d:

$$d = \frac{\sum_i m_i(\mathbf{n}^\mathsf{T}(z_i K^{-1}\mathbf{x_i}))}{\sum_i m_i} \qquad (1)$$

where K is the 3×3 camera intrinsic matrix, $\mathbf{x_i}$ is the i_{th} pixel coordinate in a homogeneous representation, z_i is its predicted depth value, and m_i is an indicator variable,

which becomes 1 if the pixel belongs to the plane. The summation is over all the pixels in the image. Note that we do not have a loss on the plane offset parameter, which did not make differences in the results. However, the plane offset influences the warping loss module below.

3.2. Segmentation Refinement Network

The plane detection network predicts segmentation masks independently. The segmentation refinement network jointly optimizes all the masks, where the major challenge is in the varying number of detected planes. One solution is to assume the maximum number of planes in an image, concatenate all the masks, and pad zero in the missing entries. However, this does not scale to large number of planes, and is prone to missing small planes.

Instead, we propose a simple yet effective module, ConvAccu, by combining a U-Net [33] and the idea of non-local module [46]. ConvAccu processes each plane segmentation mask represented in the entire image window with a convolution layer. We then calculate and concatenate the mean feature volumes over all the other planes at the same layer before passing to the next layer (See Fig. 2). As its name indicates, ConvAccu combines a convolution layer and an accumulation scheme, which resembles the non-local module and can effectively aggregate information from all the masks.

Refined plane masks are concatenated at the end and compared against target masks with a cross-entropy loss. Each of the target mask is generated by finding the ground truth mask which has the maximum overlap with the predicted mask. The target mask is set to be empty if the overlap is smaller than half of the predicted mask. Note that besides the plane mask, the refinement network also takes the original image, the union of all the other plane masks, the depthmap derived from plane detection results, the pixel-wise depthmap, and a 3D coordinate map for the specific plane (i.e., a three-channel image representing the corresponding 3D coordinates computed using the plane equation) as input. We refer to the supplementary document for the specification of all the network parameters.

3.3. Warping Loss Module

The warping loss module enforces the consistency of reconstructed 3D planes with a nearby view during training. Specifically, our training samples come from RGB-D videos in ScanNet [6], and the nearby view is defined to be the one 20 frames ahead from the current. The module first builds a depthmap D^c of the current view by 1) computing depth values from the plane equations of existing planar regions and 2) using pixel-wise depth values predicted inside the plane detection network for the remaining pixels. The reconstructed depth D^c is then warped from the current view to the nearby view D^w which is compared against the

ground-truth depthmap for the nearby view $\hat{D}^n$.

To compute D^w, we warp every pixel in the nearby view to the current view given camera intrinsics K, rotation R, translation t, and ground-truth depthmap for the nearby view $\hat{D}^n$,

$$(u^w, v^w) = \text{warp}((u^n, v^n)|K, R, t, \hat{D}^n) \qquad (2)$$

where (u^n, v^n) is a pixel in the nearby view and (u^w, v^w) is the warped pixel in the current view. The details of the warp function can be found in the supplementary document. (u^w, v^w) is then used to retrieve a depth value from D^c via bilinear interpolation, and unprojected to 3D space based on the retrieved depth value. The unprojected 3D point is transformed back to the nearby view, and its depth is assigned to D^w at pixel (u^n, v^n). The final warping loss is defined as,

$$\text{Loss}^{\text{warp}} = ||D^w - \hat{D}^n|| \qquad (3)$$

The projection, un-projection, and coordinate frame transformation are all simple algebraic operations, whose gradients can be passed for training. Note that the warping loss module and the nearby view is utilized only during training to boost geometric reconstruction accuracy, and the system runs on a single image at test time.

4. Benchmark construction

Following steps described in PlaneNet [27], we build a new benchmark from RGB-D videos in ScanNet [6]. We add the following three modifications to recover more fine-grained planar regions, yielding 14.7 plane instances per image on the average, which is more than double the PlaneNet dataset containing 6.0 plane instances per image.

• First, we keep more small planar regions by reducing the plane area threshold from 1% of the image size to 0.16% (i.e., 500 pixels) and not dropping small planes when the total number is larger than 10.

• Second, PlaneNet merges co-planar planes into a single region as they share the same plane label. The merging of two co-planar planes from different objects causes loss of semantics. We skip the merging process and keep all instance segmentation masks.

• Third, the camera pose quality in ScanNet degrades in facing 3D tracking failures, which causes misalignment between image and the projected ground-truth planes. Since we use camera poses and aligned 3D models to generate ground-truth planes, we detect such failures by the discrepancy between our ground-truth 3D planes and the raw depthmap from a sensor. More precisely, we do not use images if the average depth discrepancy over planar regions is larger than 0.1m. This simple strategy removes approximately 10% of the images.
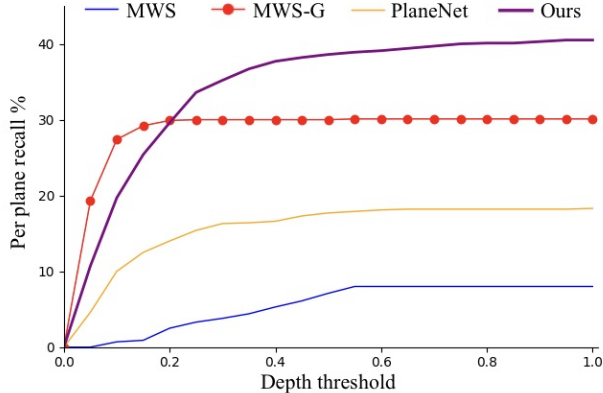
Figure 4. Plane-wise accuracy against baselines. PlaneRCNN outperforms all the competing methods except when the depth threshold is very small and MWS-G can fit 3D planes extremely accurately by utilizing the ground-truth depth values.

5. Experimental results

We have implemented our network in PyTorch. We use pre-trained Mask R-CNN [14] and initialize the segmentation refinement network with the existing model [15]. We train the network end-to-end on an NVIDIA V100 GPU for 10 epochs with 100,000 randomly sampled images from training scenes in ScanNet. We use the same scale factor for all losses. For the detection network, we scale the image to 640×480 and pad zero values to get a 640×640 input image. For the refinement network, we scale the image to 256×192 and align the detected instance masks with the image based on the predicted bounding boxes.

5.1. Qualitative evaluations

Fig. 6 demonstrates our reconstructions results for Scan-Net testing scenes. PlaneRCNN is able to recover planar surfaces even for small objects. We include more examples in the supplementary document.

Fig. 7 compares PlaneRCNN against two competing methods, PlaneNet [27] and PlaneRecover [49], on a variety of scene types from unseen datasets (except the SYNTHIA dataset is used for training by PlaneRecover). Note that PlaneRCNN and PlaneNet are trained on the ScanNet which contains indoor scenes, while PlaneRecover is trained on the SYNTHIA dataset (i.e., the 7th and 8th rows in the figure) which consist of synthetic outdoor scenes. The figure shows that PlaneRCNN is able to reconstruct most planes in varying scene types from unseen datasets regardless of their sizes, shapes, and textures. In particular, our results on the KITTI dataset are surprisingly better than PlaneRecover for planes close to the camera. In indoor scenes, our results are consistently better than both PlaneNet and PlaneRecover.

While the detection network is able to robustly extract planar regions for images from unseen datasets, the gener-

alization for 3D geometries is harder than the generalization of region masks. In Fig. 5, we show depth visualization for four examples from unseen datasets.
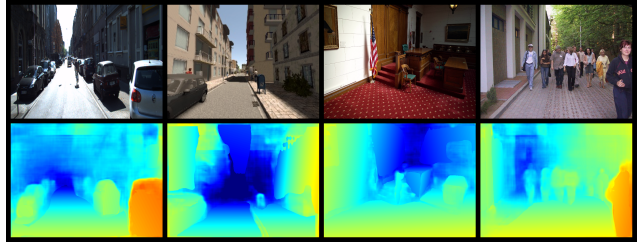

Figure 5. We show input images and depth reconstruction results on unseen datasets without fine-tuning. From left to right: we show one examples from each dataset in the order of KITTI [13], SYNTHIA [34], Tank and Temple [19], and PhotoPopup [17].

5.2. Plane reconstruction accuracy

Following PlaneNet [27], we evaluate plane detection accuracy by measuring the plane recall with a fixed Intersection over Union (IOU) threshold 0.5 and a varying depth error threshold (from 0 to $1m$ with an increment of $0.05m$). The accuracy is measured inside the overlapping regions between the ground-truth and inferred planes. Besides PlaneNet, we compare against Manhattan World Stereo (MWS) [10], which is the most competitive traditional MRF-based approach as demonstrated in prior evaluations [27]. MWS requires a 3D point cloud as an input, and we either use the point cloud from the ground-truth 3D planes (MWS-G) or the point cloud inferred by our depthmap estimation module in the plane detection network (MWS). PlaneRecover [49] was originally trained with the assumption of at most 5 planes in an image. We find it difficult to train PlaneRecover successfully for cluttered indoor scenes by simply increasing the threshold. We believe that PlaneNet, which is explicitly trained on ScanNet, serves as a stronger competitor for the evaluation.

We randomly sample 100 images from ScanNet testing scenes for the evaluation. As demonstrated in Fig. 4, PlaneRCNN significantly outperforms all other methods, except when the depth threshold is small and MWS-G can fit planes extremely accurately with the ground-truth depth values. Nonetheless, even with ground-truth depth information, MWS-G fails in extracting planar regions robustly, leading to lower recalls in general. Our results are superior also qualitatively as shown in Fig. 8.

5.3. Geometric accuracy

We propose a new metric in evaluating the quality of piecewise planar surface reconstruction by mixing the inferred depthmaps and the ground-truth plane segmentations. More precisely, we first generate a depthmap from our reconstruction by following the process in the warping loss

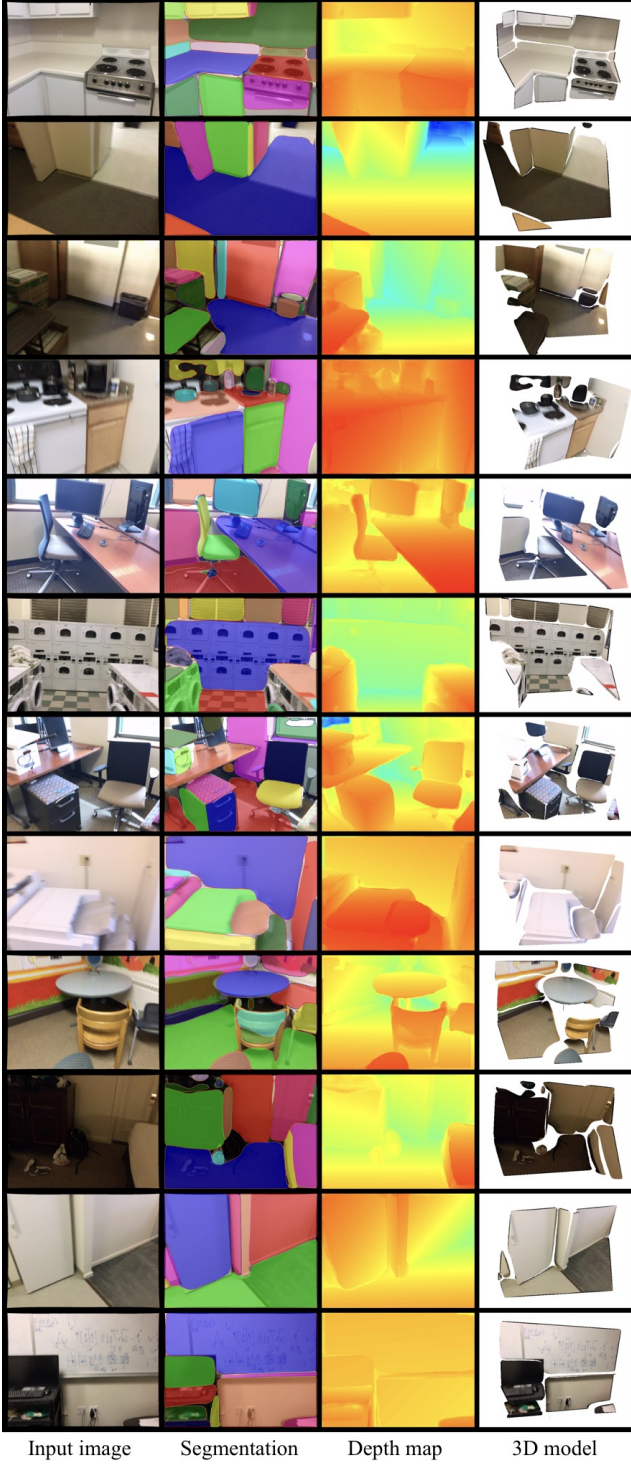| Input image | Segmentation | Depth map | 3D model |

Figure 6. piecewise planar reconstruction results by PlaneRCNN. From left to right: input image, plane segmentation, depthmap reconstruction, and 3D rendering of our depthmap (rendered from a new view with -0.4m and 0.3m translation along x-axis and z-axis respectively and $10°$ rotation along both x-axis and z-axis).

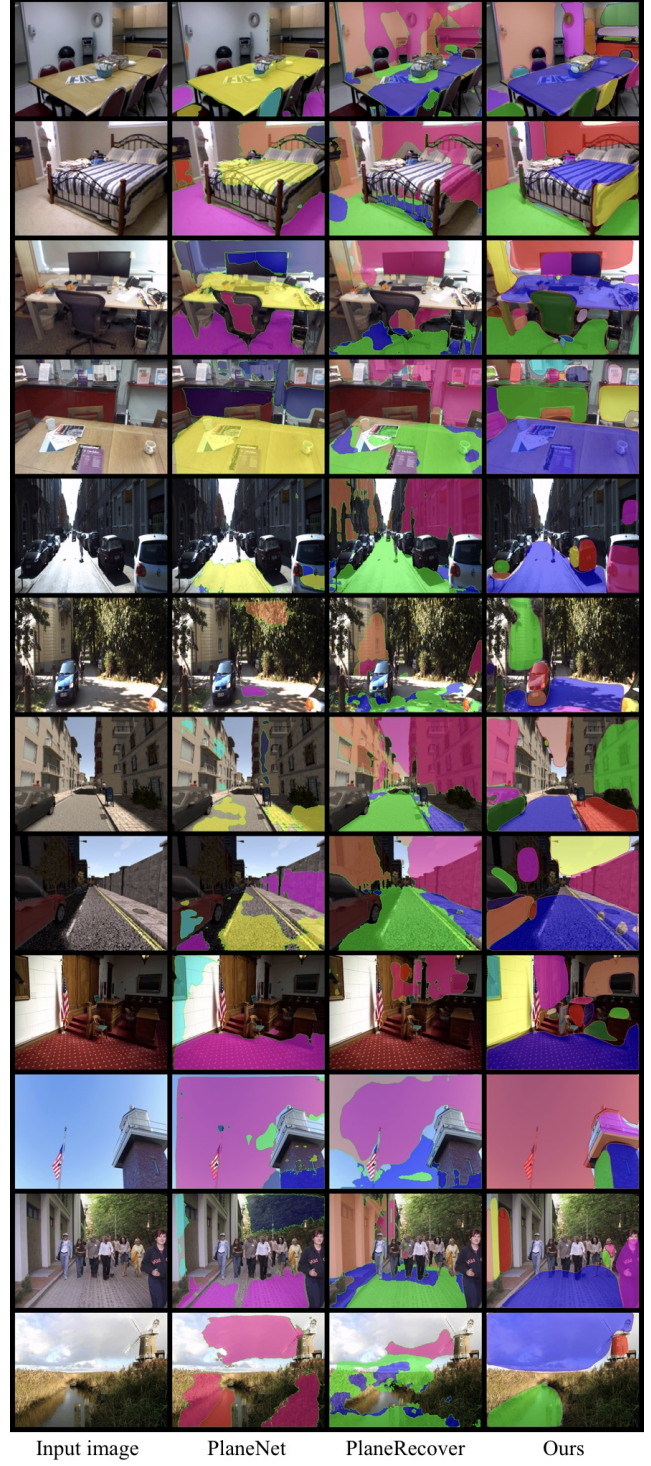| Input image | PlaneNet | PlaneRecover | Ours |

Figure 7. Plane segmentation results on unseen datasets without fine-tuning. From left to right: input image, PlaneNet [27] results, PlaneRecover [49] results, and ours. From top to the bottom, we show two examples from each dataset in the order of NYUv2 [37], 7-scenes [35], KITTI [13], SYNTHIA [34], Tank and Temple [19], and PhotoPopup [17].

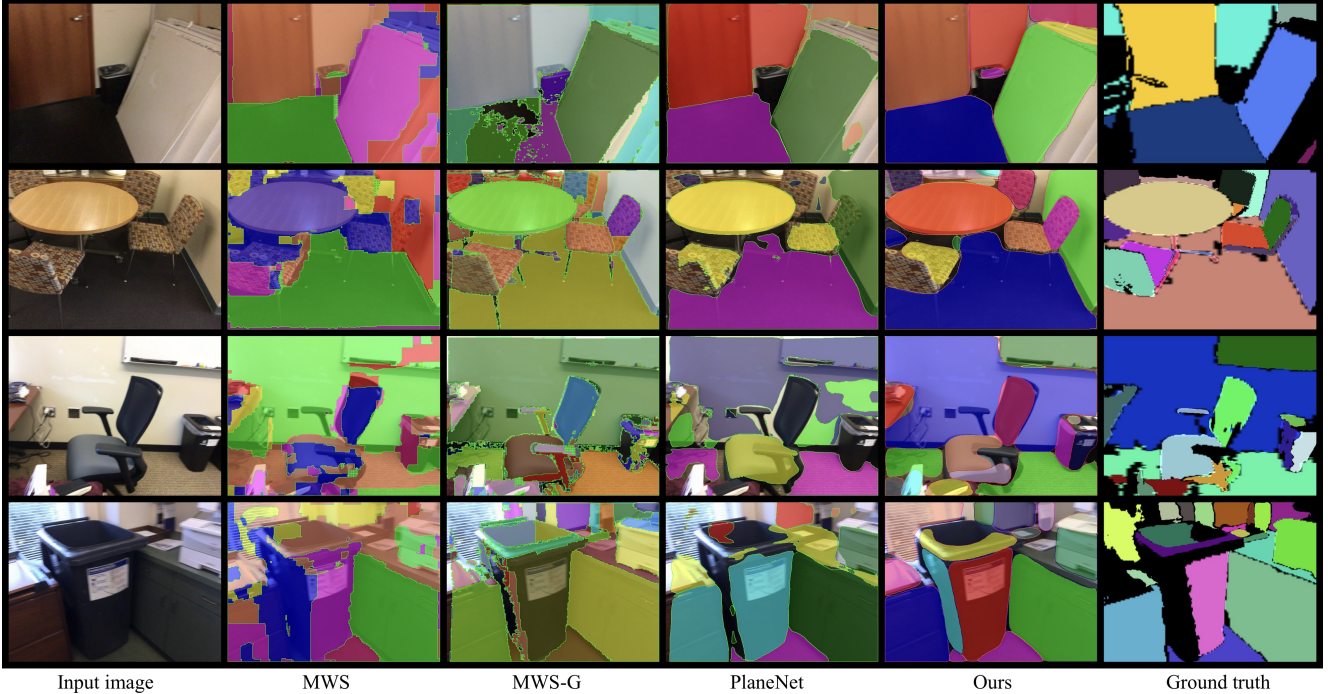| Input image | MWS | MWS-G | PlaneNet | Ours | Ground truth |

Figure 8. Plane segmentation comparisons. From left to right: 1) input image, 2) MWS with inferred depths, 3) MWS with ground-truth depths, 4) PlaneNet, 5) Ours, and 6) ground-truth.

evaluation (Sec. 3.3). Next, for every ground-truth planar segment, we convert depth values in the reconstructed depthmap to 3D points, fit a 3D plane by SVD, and normalize the plane coefficients to make the normal component into a unit vector. Finally, we compute the mean and the area-weighted mean of the parameter differences to serve as the plane evaluation metrics. Besides the plane parameter metrics, we also consider depthmap metrics commonly used in the literature [8]. We evaluate over the NYU dataset [37] for a fair comparison. Table 1 shows that, with more flexible detection network, PlaneRCNN generalizes much better without fine-tuning. PlaneRCNN also outperforms PlaneNet [27] in every metric after fine-tuning using the ground-truth depths from the NYU dataset.

5.4. Ablation studies

PlaneRCNN adds the following components to the Mask R-CNN [14] backbone: 1) the pixel-wise depth estimation network; 2) the anchor-based plane normal regression; 3) the warping loss module; and 4) the segmentation refinement network. To evaluate the contribution of each component, we measure performance changes while adding the components one by one. Following [49], we evaluate the plane segmentation quality by three clustering metrics: variation of information (VOI), Rand index (RI), and segmentation covering (SC). To further assess the geometric accuracy, we compute the average precision (AP) with IOU threshold 0.5 and three different depth error thresholds

Table 1. Geometric accuracy comparison over the NYUv2 dataset.

Method	PlaneNet [27]	Ours
w/o fine-tuning		
Rel	0.220	**0.164**
log_{10}	0.114	**0.077**
$RMSE$	0.858	**0.644**
Param.	0.939	**0.776**
Param. (weighted)	0.771	**0.641**
w/ fine-tuning		
Rel	0.129	**0.124**
log_{10}	0.079	**0.073**
$RMSE$	0.397	**0.395**
Param.	0.713	**0.642**
Param. (weighted)	0.532	**0.505**

$[0.3m, 0.6m, 0.9m]$. Larger value means higher quality for all the metrics except for VOI.

Table 2 shows that all the components have positive contribution to the final performance. Fig. 9 further highlights the contributions of the warping loss module and the segmentation refinement network qualitatively. The first example shows that the segmentation refinement network fills in gaps between adjacent planar regions, while the second example shows that the warping loss module improves reconstruction accuracy with the help from the second view.

Table 2. Ablation studies on the contributions of the four components in PlaneRCNN. Plane segmentation and detection metrics are calculated over the ScanNet dataset. PlaneNet represents the competing state-of-the-art.

Method	Plane segmentation metrics			Plane detection metrics		
	VOI↓	RI	SC	$AP^{0.3m}$	$AP^{0.6m}$	$AP^{0.9m}$
PlaneNet	2.142	0.797	0.692	0.156	0.178	0.182
Ours (basic)	2.113	0.851	0.719	0.269	0.329	0.355
Ours (depth)	2.041	0.856	0.752	0.352	0.376	0.386
Ours (depth + anch.)	2.021	0.855	0.761	0.352	0.378	0.392
Ours (depth + anch. + warp.)	1.990	0.855	0.766	**0.365**	0.384	0.401
Ours (depth + anch. + warp. + refine.)	**1.809**	**0.880**	**0.810**	**0.365**	**0.386**	**0.405**

Input image   w/o refinement   w/ refinement

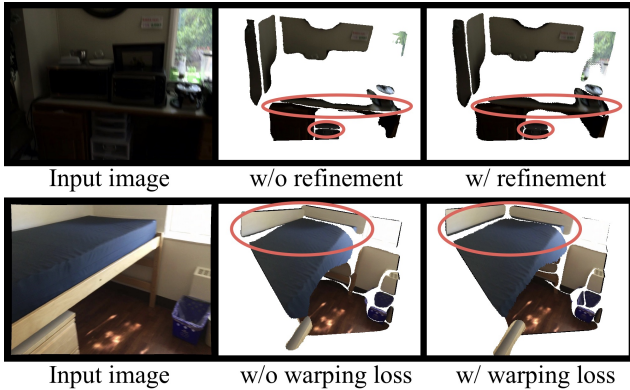Input image   w/o warping loss   w/ warping loss

Figure 9. Effects of the surface refinement network and the warping loss module. Top: the segmentation refinement network narrows the gap between adjacent planes. Bottom: the warping loss helps to correct erroneous plane geometries from the second view.

5.5. Occlusion reasoning

A simple modification allows PlaneRCNN to infer occluded/invisible surfaces and reconstruct layered depthmap models. First, ground-truth layered depthmaps are constructed as follows. In our original process, we fit planes to aligned 3D scans to obtain ground-truth 3D planar surfaces, then rasterize the planes to an image with a depth testing. We simply remove the depth testing and generate a "complete-mask" for each plane. Second, we add one more mask prediction module to PlaneRCNN to infer the complete-mask for each plane instance. Please refer to the supplementary document for the details of this experiment.

Fig. 10 shows the new view synthesis examples, in which the modified PlaneRCNN successfully infers occluded surfaces, for example, floor surfaces behind tables and chairs. Note that a depthmap is rendered as a depth mesh model (i.e., a collection of small triangles) in the figure. The layered depthmap representation enables new applications such as artifacts-free view synthesis, better scene completion, and object removal [26, 44]. This experiment demonstrates yet another flexibility and potential of the proposed PlaneRCNN architecture.

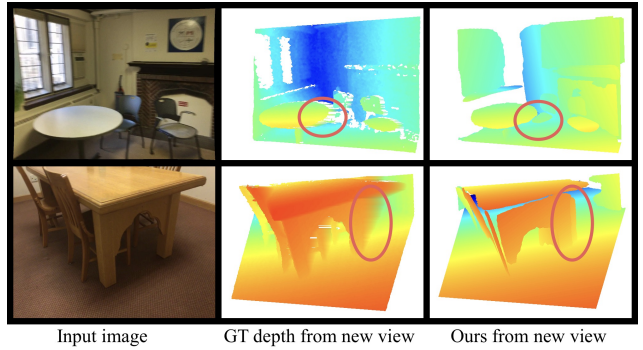Input image   GT depth from new view   Ours from new view

Figure 10. New view synthesis results with the layered depthmap models. A simple modification allows PlaneRCNN to also infer occluded surfaces and reconstruct layered depthmap models.

6. Conclusion and future work

This paper proposes PlaneRCNN, the first detection-based neural network for piecewise planar reconstruction from a single RGB image. PlaneRCNN learns to detect planar regions, regress plane parameters and instance masks, globally refine segmentation masks, and utilize a neighboring view during training for a performance boost. PlaneRCNN outperforms competing methods by a large margin based on our new benchmark with fine-grained plane annotations.

An interesting future direction is to process an image sequence during inference which requires learning correspondences between plane detections. Another design choice worth exploring is to estimate pixel-wise normals and depth within the same module to share features.

7. Acknowledgement

References

[1] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3547–3555, 2015. 2

[2] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2040–2049, 2017. 2

[3] A. Chauve, P. Labatut, and J. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1261–1268, 2010. 1

[4] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 2

[5] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 2

[6] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 4

[7] Z. Deng, S. Todorovic, and L. J. Latecki. Unsupervised object region proposals for rgb-d indoor scenes. *Computer Vision and Image Understanding*, 154:127–136, 2017. 2

[8] D. Eigen and R. 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. 7

[9] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Advances in neural information processing systems*, pages 611–619, 2012. 2

[10] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *Computer V@inproceedingsfidler20123d, title=3d object detection and viewpoint estimation with a deformable 3d cuboid model, author=Fidler, Sanja and Dickinson, Sven and Urtasun, Raquel, booktitle=Advances in neural information processing systems, pages=611–619, year=2012 ision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1422–1429. IEEE, 2009. 2, 5

[11] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 00, pages 1422–1429, 2018. 1

[12] D. Gallup, J.-M. Frahm, and M. Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. 2010. 2

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 5, 6

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 2, 3, 5, 7

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 5

[16] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, July 2005. 1

[17] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. In *ACM transactions on graphics (TOG)*, volume 24, pages 577–584. ACM, 2005. 5, 6

[18] M. Kaess. Simultaneous localization and mapping with infinite planes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4605–4611, 2015. 1

[19] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):78, 2017. 5, 6

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 2

[21] A. Kundu, Y. Li, and J. M. Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *CVPR*, 2018. 2

[22] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. *arXiv preprint arXiv:1612.02699*, 2016. 2

[23] X. Liang, X. Shen, D. Xiang, J. Feng, L. Lin, and S. Yan. Semantic object parsing with local-global long short-term memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3185–3193, 2016. 2

[24] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3194–3203, 2016. 2

[25] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 3

[26] C. Liu, P. Kohli, and Y. Furukawa. Layered scene decomposition via the occlusion-crf. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–173, 2016. 8

[27] C. Liu, J. Yang, D. Ceylan, E. Yumer, and Y. 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, 3, 4, 5, 6, 7

[28] F. Liu, C. Shen, G. Lin, and I. D. Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(10):2024–2039, 2016. 3

[29] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *arXiv preprint arXiv:1807.03247*, 2018. 3

[30] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015. 2

[31] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 418–426, 2015. 2

[32] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017. 2

[33] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 4

[34] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 5, 6

[35] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013. 6

[36] B. Shuai, Z. Zuo, B. Wang, and G. Wang. Dag-recurrent neural networks for scene labeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3620–3629, 2016. 2

[37] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 2, 6, 7

[38] S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *Proceedings of the IEEE International Conference on Computer Vision*, 2009. 2

[39] M. Sun, B.-s. Kim, P. Kohli, and S. Savarese. Relating things and stuff via objectproperty interactions. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1370–1383, 2014. 2

[40] J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3001–3008, 2013. 2

[41] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3748–3755, 2014. 2

[42] G. Tsai, C. Xu, J. Liu, and B. Kuipers. Real-time indoor scene understanding using bayesian filtering with motion cues. In *2011 International Conference on Computer Vision*, pages 121–128, 2011. 1

[43] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005. 2

[44] S. Tulsiani, R. Tucker, and N. Snavely. Layer-structured 3d scene inference via view synthesis. *arXiv preprint arXiv:1807.10264*, 2018. 8

[45] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015. 3

[46] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 4

[47] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016. 2

[48] Y. Xiang, W. Choi, Y. Lin, and S. Savarese. Data-driven 3d voxel patterns for object category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1903–1911, 2015. 2

[49] F. Yang and Z. Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 1, 2, 3, 5, 6, 7

[50] J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 702–709. IEEE, 2012. 2

[51] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2

[52] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature-and area-based information for urban buildings modeling from aerial imagery. In *European conference on computer vision*, pages 873–886. Springer, 2008. 2

[53] R. Zhang, S. Tang, M. Lin, J. Li, and S. Yan. Global-residual and local-boundary refinement networks for rectifying scene parsing predictions. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3427–3433. AAAI Press, 2017. 2

[54] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017. 2

[55] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537, 2015. 2

[56] J. Zhou and B. Li. Homography-based ground detection for a mobile robot platform using a single camera. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 4100–4105, 2006. 1

[57] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2608–2623, 2013. 2