Peek-a-Boo: Occlusion Reasoning in Indoor Scenes with Plane Representations

Ziyu Jiang[1,3*] Buyu Liu[1] Samuel Schulter[1] Zhangyang Wang[3] Manmohan Chandraker[1,2]
[1]NEC Laboratories America [2]UC San Diego [3]Texas A&M Universaity

Abstract

We address the challenging task of occlusion-aware indoor 3D scene understanding. We represent scenes by a set of planes, where each one is defined by its normal, offset and two masks outlining (i) the extent of the visible part and (ii) the full region that consists of both visible and occluded parts of the plane. We infer these planes from a single input image with a novel neural network architecture. It consists of a two-branch category-specific module that aims to predict layout and objects of the scene separately so that different types of planes can be handled better. We also introduce a novel loss function based on plane warping that can leverage multiple views at training time for improved occlusion-aware reasoning. In order to train and evaluate our occlusion-reasoning model, we use the ScanNet dataset [1] and propose (i) a strategy to automatically extract ground truth for both visible and hidden regions and (ii) a new evaluation metric that specifically focuses on the prediction in hidden regions. We empirically demonstrate that our proposed approach can achieve higher accuracy for occlusion reasoning compared to competitive baselines on the ScanNet dataset, e.g. 42.65% *relative improvement on hidden regions.*

1. Introduction

Reasoning about occlusions occurring in the 3D world is an ability at which human visual perception excels. While we develop an understanding for the concept of object permanence already as toddlers, for instance by playing peek-a-boo, it is a very challenging skill for machine intelligence to acquire, since it requires strong contextual and prior knowledge about objects and scenes. This is particularly true for indoor scenes where the composition of objects and scenes is highly complex and leads to numerous and strong occlusions. And while several works exist that investigate this problem for outdoor scenes [5, 13, 24], there has been comparatively little work for indoor scenes. But indoor applications that can potentially benefit from occlusion reasoning are ample, like robot navigation or augmented reality.

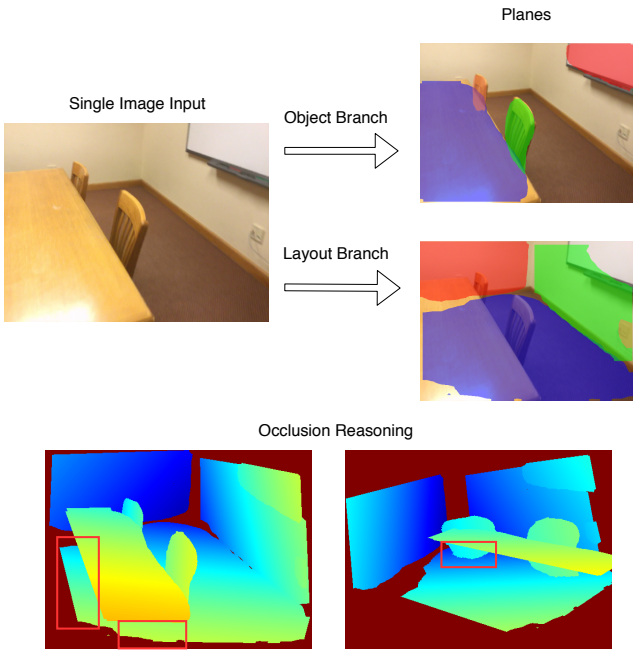


Figure 1: Given a single image as input, our model predicts planes to describe both visible and occluded areas of the scene with separate branches for objects and layout (top). This model can be used for occlusion reasoning and novel view synthesis (bottom).

To address the problem of occlusion reasoning, the first step is to find a suitable representation to describe scenes. Prior work has explored several representations like 3D bounding boxes for objects [4, 28] or voxels [19]. However, the former representation is rather coarse for general objects and is not suitable for scene layouts, while the latter has a significant memory footprint. Most recently, layered depth images (LDI) [18] have been leveraged for predicting scene geometry in occluded areas [3, 22], which was also extended to the object level [2]. We rely on planes as another promising representation that shows several benefits over the aforementioned ones. Planes can compactly describe scenes in a semi-parametric way: each plane is defined by a normal vector, an offset and a (non-parametric) mask to

*Part of this work was conducted during a summer internship at NEC Laboratories America.

outline its extent. Liu *et al*. [10] have recently demonstrated these benefits with their Plane-RCNN model, but it focuses mainly on the visible part of the scene.

In this work, we extend Plane-RCNN [10] to infer a full scene representation that also reasons about hidden areas of the input scene. Starting with a Plane-RCNN model that predicts both the visible and occluded extent of each plane, we propose a novel network architecture that separates the prediction of planes based on semantics, see Sec. 3.2. Given the often stark difference in size and shape of planes on "foreground" objects, like chairs or tables, and planes on "background" stuff, like walls, separating the predictions lowers output space variations and thus eases network optimization. With our proposed merging strategy, this novel architecture design significantly boosts performance, particularly in hidden areas. Finally, we also present a novel objective that is based on warping planes from one view to another to obtain a training signal if multi-view input is available, which additionally improves performance. Fig. 1 gives an overview of our approach.

In order to train such a model, however, training data is required that contains ground truth about the geometry and semantics in occluded areas. While several synthetic datasets exist where full 3D information can be extracted [27, 20], no dataset with real images exists that provides such ground truth. In this work, we demonstrate how to process the ScanNet dataset [1] such that approximate but reliable ground truth for the problem of occlusion reasoning can be generated, which we will make publicly available.[1]

While we follow the standard evaluation proposed in Plane-RCNN [10], average precision (AP), for the visible part of planes, we found that it does not well capture the impact of predictions in the hidden part. For this reason, we propose a novel metric, average precision hidden (APH), which is specifically designed for occluded areas and described in Sec. 3.3.

Our results on the ScanNet dataset with our newly generated annotations show that each component of our approach aids in better occlusion reasoning. Our model is competitive with Plane-RCNN [10] for foreground areas and outperforms strong baselines in occluded areas, see Sec. 4.

To summarize, our contributions are:

- Extend Plane-RCNN [10] to predict the occluded part of planes in indoor scenes
- A novel network architecture (DualRPN) and training objective (plane-warp) specifically designed for the task of occlusion reasoning.
- Approximate ground truth of semantics and geometry in occluded areas generated automatically from the ScanNet dataset.
- An evaluation metric designed to analyze the prediction quality of occlusions.

[1]Available at: www.nec-labs.com/~mas/peekaboo/

2. Related Work

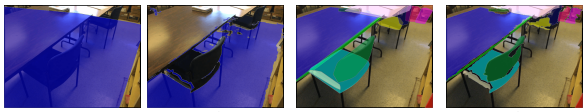
We focus our discussion of related work specifically on occlusion reasoning in scene understanding.

Depth-ordering: Representing objects in an amodal fashion (*e.g*., 3d boxes) and assigning them an ordering based on depth is a simple way to reason about occlusions. For instance, Yang *et al*. [26] explore layered representations to express relative order. Specifically, an image will be decomposed into multiple regions and each region is associated with its semantic class and relative order [21]. The main problem lies in how to generate high quality semantically complete and meaningful regions. An exemplar-based detector is utilized in [21] and other instance-level detection methods can also be applied when occlusion reasoning happens among foreground objects. In contrast, our model is more generic since we work on both object and background/stuff classes and can be extended to unseen classes. Moreover, our representation is more complete since we provide absolute depth rather than just relative order.

Layout estimation: Many prior works have tried to estimate the layout of a scene, both for outdoor [5, 9, 24] as well as indoors [7, 8, 23]. The layout is typically described by a parametric model, *e.g*., a cube for indoor rooms [8] or by more complex attributes for outdoor scenes [5, 24]. All these works naturally address occlusion reasoning since most scenes also contain foreground objects that occlude the scene layout. In contrast, our work addresses both foreground objects and the scene layout simultaneously.

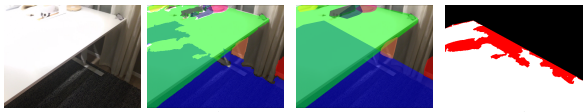
Explicit representations for occlusion reasoning: Another direction that addresses the occlusion reasoning problem more directly is via predicting multiple depth and/or semantic segmentation maps. Many works have been proposed for completing depth with RGB-D input, including ones that employ optimization of the Mumford-Shah functional [12], background surface extrapolation [14] or deep-learning based inpainting [17]. Similar works [3, 22] rely on Layer Depth Images (LDI) by Shade *et al*. [18]. While these methods separate foreground and background with two distinct depth maps (and segmentation maps), multiple occlusions of foreground objects are not representable. Most recently, Dhamo *et al*. [2], successfully addressed this issue by combining LDIs with the concept of object detection frameworks [16]. One limitation of [2] is that the definition of foreground objects is bound by the availability of object detection datasets, whereas our proposed approach is more generic and can generalize to arbitrary objects in a scene.

Plane-based representations: Planes are an alternative representation of scenes, which have shown to be promising [10, 11, 25] due to their compactness and flexibility. Essentially, each plane in the scene can be described by a normal vector and an offset along with a mask that outlines the extent of the plane. Both [11] and [25] reconstruct



(a) (b) (c) (d)

Figure 2: An illustration of our plane representation: (a) complete layout masks (b) visible layout masks (c) complete object masks (d) visible object masks. Separately showing layout and object planes is for better visualization.



(a) (b) (c) (d)

Figure 3: Complete masks for ground truth generation before (b) and after (c) applying the proposed filling method for image in (a). We can observe that after the proposed filling method, we are able to have a better complete mask for floor (blue area). (d) highlights the filled regions. White area is the original complete mask without filling. Red area shows the filled area, where pixel value is set to -1.

scenes as piecewise planar depthmap from a single RGB image but are limited by the need to define a maximum number of planes. Very recently, Liu *et al*. [10] have proposed Plane-RCNN, which is a deep-learning based model to predict a full scene description based on planes. Similar to [2], Plane-RCNN leverages an object detection framework [16] to predict an arbitrary number of planes. In this work, we extend Plane-RCNN to explicitly do occlusion reasoning with a plane representation, which are very suitable since planes naturally extend to occluded areas.

3. Our Approach

In this section, we introduce our proposed approach for occlusion reasoning and discuss four main aspects. First, we introduce in Sec. 3.1 our procedure for generating ground truth from the ScanNet dataset [1] that is required to train our occlusion-reasoning model. We also discuss in that section the plane-based representation that we build our model upon. Given the plane representation, we then introduce our main contributions in Sec. 3.2: A two-branch category-specific module for predicting planes, the corresponding fusion scheme for objects and scene layout, and a novel training objective based on plane warping. Finally, in Sec. 3.3, we propose a novel metric to evaluate occlusion-reasoning specifically for occluded areas.

3.1. Data generation and plane representation

Our dataset is built based on the RGB-D videos in ScanNet [1]. We first convert the mesh of the room into a set of multiple planes following the steps described in PlaneNet [11]. In our work, we describe each plane with a normal vector, an offset and *two masks*: one for the visible part of the plane and one to outline the full extent of a plane even if occluded by other objects. We denote these two types of masks as "visible mask" and "complete mask". The normal vector indicates the direction of the plane, the offset defines the closest distance from the camera to the plane, and the masks indicate the size and shape of the plane (visible and complete, respectively). Fig. 2 illustrates this plane representation. For a full representation of the scene, we also use (and predict) a depthmap for areas that are not covered by any plane. To finally generate the data for every single view of the scene, we utilize the camera parameters to compute the parameters and masks (both visible and complete) of our plane representation.

However, due to the camera views as well as the noisy meshes [2], there can be holes in the complete masks for occluded areas. This would lead to a wrong training signal for our prediction network since the holes are merely artifacts of the data generation process. Observing that complete planes like the wall, the floor or the top of a table are, in most cases, of convex shape while holes generally occur inside the full planes, we propose to fill the complete masks to be the convex closure. However, since we cannot be sure that each hole is actually surrounded by a mask that corresponds to the same plane, assignment to masks becomes ambiguous and can potentially lead to wrong training signals. We thus choose to flag those filled areas as "ignore" (-1), meaning they have no influence during training our model. Fig. 3 shows comparisons of full masks before and after filling. Note that the filled floor is set as "ignore" (red area) in Fig. 3d. We also report quantitative results of with and without the filling step later in our experiments (see Sec. 4 for more details.)

3.2. Occlusion-aware plane detection network

Our occlusion-aware plane detection network follows the design of PlaneRCNN [10], which we briefly review in the following paragraph before introducing our proposed model.

Plane-RCNN [10]: The main architecture of PlaneRCNN [10] follows the design of Mask R-CNN [6]. Specifically, it detects planes by first depicting their enclosing bounding boxes (bbox). Then it predicts the normal and binary mask for each individual plane, indicating the corresponding orientation as well as the visible region of this plane. Meanwhile, there is a depth branch that uses the global feature map to predict per-pixel depth values. Given the per-pixel depth and the plane (normal and visible mask) predictions, it further estimates the offset of each plane. Note that in the original paper of [10], the authors also propose to add one more mask prediction module for inferring the

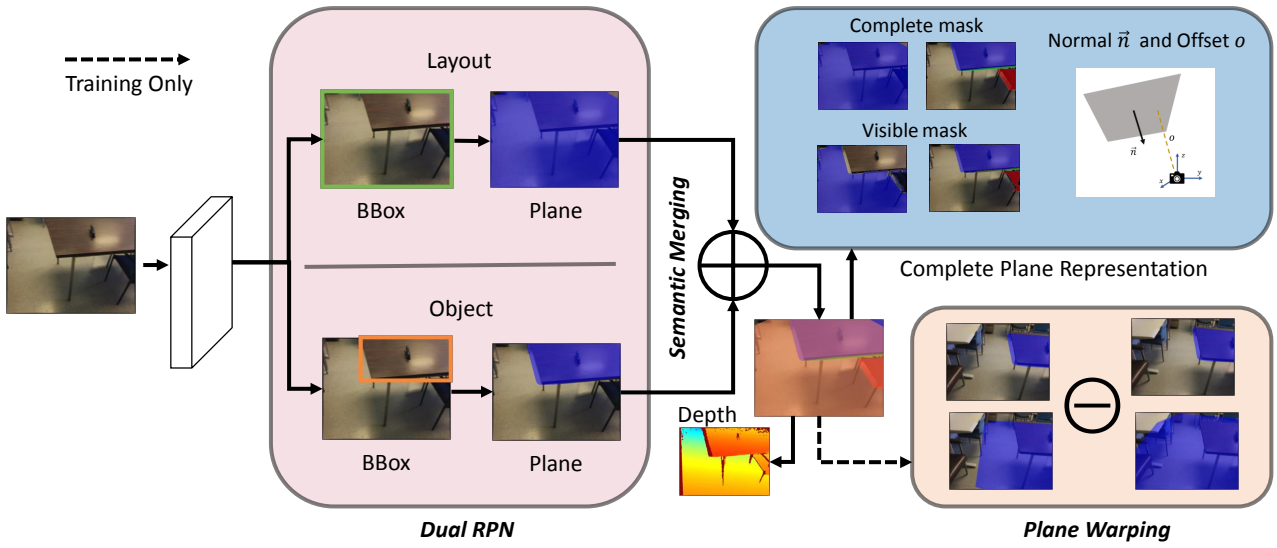


Figure 4: Our framework consists of three parts: (1) A dual RPN model which splits the head of PlaneRCNN-OR [10] into two, for predicting layout masks and object masks separately. (2) A plane merging module for fusing predictions from the two branches. (3) An occlusion-aware warping loss module for measuring the consistency with neighbor frames during training when multiple views are available.

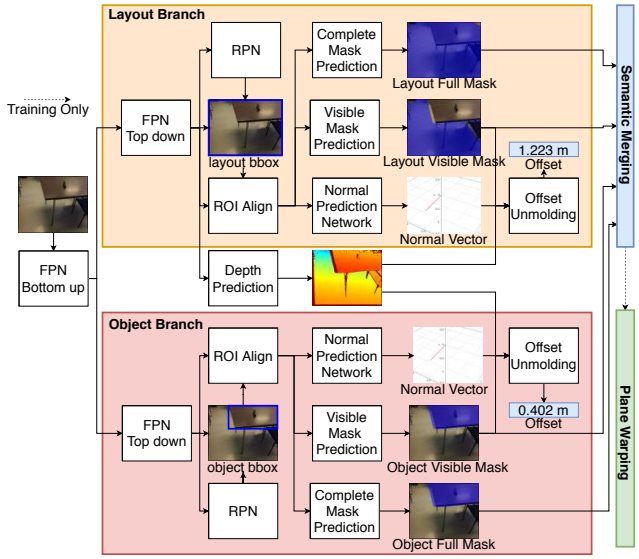


Figure 5: The detailed flowchart of our DualRPN module.

complete mask to handle the occlusion reasoning. Although no quantitative results are provided, we implement their proposal and take this as one of our baselines, which we denote as PlaneRCNN-OR.

DualRPN: With the introduction of both visible and complete masks, the variation in shape, size and distribution of planes belonging to different categories becomes larger than only looking at visible masks as in PlaneRCNN [10]. A good example for this are categories like floor and wall, where large differences can be observed compared to typical foreground categories, but also between visible and complete masks of the same plane. To this end, we propose to handle background and foreground separately. Specifically, we divide our classes into two, foreground and background, and propose a category-specific network, which is referred to as DualRPN in this paper. Compared to the baseline (PlaneRCNN-OR), we aim to learn different priors for different categories by employing separate RPNs with independent mask prediction modules. As can be viewed in Fig. 4, we have one object and one layout branch, for foreground and background categories, respectively. The object branch is trained with the object plane ground truth and vice versa. In this way, we are able to handle and learn different priors for both categories without adding too many parameters. Ideally, given a single image, the layout branch predicts only the masks for background classes, e.g. walls and floors, while the object branch focuses only on foreground classes and ignores others. The detailed flowchart is shown in Fig. 5. Note that we also tried splitting at later stages in the network architecture, however, it did not provide comparably good results.

Semantic merging: The results of DualRPN are two sets of predicted planes from object and layout branches, respectively. In order to obtain the final representation for the entire image, we should fuse the predictions from both. To achieve that, one naive way is to take the union of the two sets and use the full predictions as the final results. However, this would lead to duplicated results. For instance, one region can get predictions both from object and layout branch by mistake since both of them believe this region should be represented by their own category. To address this problem,

one can apply non-maxima suppression (NMS) over the full predictions. However, this simple NMS-based fusion can lead to over-suppressing of planes. Fig. 9 gives an example. If we directly apply NMS between predictions from object and layout branch, we will suppress the plane for the table. This is due to the heavy occlusion between table and floor boxes and NMS would only keep one of them in this case. This type of problem is even more severe for our task since heavy occlusions happen frequently.

To this end, we propose a novel fusion method and denote it as Semantic Merging. We firstly apply NMS separately on each branch's output planes. Then, we feed the suppressed results from the two branches to our semantic merging module. This module then utilizes semantic segmentation results as reference to effectively fuse the results from both branches (see Fig. 4). Specifically, we can check the overlap between visible masks from object and layout branches. For those pairs whose overlapping score is greater than the pre-defined threshold θ, we further turn to the semantic segmentation results to help determine which of the planes to keep. For these paired visible masks, we compute their confidence scores based on their overlapping score w.r.t. semantic segmentation and leave the one that has higher confidence score in our final predictions. The overlapping score of the layout class can be computed by counting the percentage of pixels that are inside the layout visible mask and belong to a layout class in the segmentation map and vice versa. In practice, we use an off-the-shelf semantic segmentation network [15] to predict the per-pixel semantic segmentation map and set the threshold θ to 0.3. More detailed algorithm can be found in supplementary material.

Plane warping module: We now introduce a novel training objective specifically designed for our plane representation that leverages the availability of multiple views of the same scene in the training dataset. The loss tries to encourage consistency between planes across different views, which is useful since many planes are likely only occluded in one view, but visible from another one. Different from the warping loss introduced in [10], our warping loss can enforce consistency also in hidden regions.

Given the camera transformation between two views, we warp each predicted plane P_i. First, the plane normals and offsets are projected by the camera rotation and translation. With this information, we can then further project the mask of the predicted plane P_i to the other view via bilinear interpolation. We denote the warped plane as P_{w_i}. Then, we match each warped prediction P_{w_i} with ground truth planes P_{g_j}, which can be formalized as

$$\max_{i,j} \text{IoU}(P_{g_i}, P_{w_j}) \ , \tag{1}$$

subject to

$$D_n(P_{g_i}, P_{w_j}) \leq \eta_{\text{depth}} \wedge \text{IoU}(P_{g_i}, P_{w_j}) \geq \eta_{\text{iou}} \ , \tag{2}$$

with

$$D_n(P_{g_i}, P_{w_j}) = \left\| N_{P_{g_i}} \cdot o_{g_i} - N_{P_{w_j}} \cdot o_{w_j} \right\|^2 \ . \tag{3}$$

where $\text{IoU}(\cdot)$ calculates the intersection-over-union overlap between two planes and N_P and o indicate the normal and offset of a plane. The two thresholds η_{depth} and η_{iou} are hyper-parameters which are set to 0.5 and 0.3, respectively for all experiments. Finally, the loss is calculated as the cross entropy between the warped mask prediction and the matched neighbor ground truth mask, which provides an additional training signal and improves our results as we empirically demonstrate in Sec. 4. A more detailed algorithm can be found in the supplementary material.

3.3. Metric for occluded regions

To measure the performance of plane predictions, Liu *et al*. [10] propose to utilize *Average Precision* (AP), as in instance segmentation [6], but with an additional depth constraint. The metric AP0.4 only counts predicted masks as correct if the intersection-over-union (IoU) between predicted and ground-truth masks is greater than 0.5 and the average pixel-level depth difference for these two planes is within 0.4 meter. However, directly applying this evaluation metric for our task is not desired. The main reason is that this evaluation metric never explicitly measures the performance on hidden regions, but only considers the complete region as a whole. Since it often happens that the area of a visible region is far larger than that of the corresponding hidden/invisible region, a model can still have a very high AP0.4 value even if it predicts the visible region well and sacrifices the hidden one. To this end, we propose a novel evaluation metric, termed *Average Precision Hidden* (APH), which complements AP0.4 specifically for occlusion-reasoning. For calculating APH, the **F**ully **V**isible planes $\{P_g^{\text{FV}}\}$ and their corresponding estimations $\{P_e^{\text{FV}}\}$ need first to be removed. For the jth plane P_{g_j}, it belongs to $\{P_g^{\text{FV}}\}$ as long as its hidden mask $\text{Area}(G_{H_j}) < \kappa_{\text{area}}$. Here G_{H_j} is visible mask of P_{g_j}. For the ith plane P_{e_i}, it belongs to $\{P_e^{\text{FV}}\}$ as long as the output j of Equ. 4 satisfies $P_{g_j} \in \{P_g^{\text{FV}}\}$.

$$\text{argmax}_j \text{IoU}(M_i, G_j) \tag{4}$$

where M_i is the complete mask of the i-th plane estimation P_{e_i}, G_j is the complete mask of the j-th ground truth P_{g_j}. A predicted plane must satisfy the following three criteria to be considered as true positive:

$$\text{IoU}(M_i \cap \overline{G_{V_j}}, G_{H_j}) \geq \kappa_{\text{iou}} \tag{5a}$$

$$D(P_{e_i}, P_{g_j}) \leq \kappa_{\text{depth}} \ , \tag{5b}$$

$$P_{g_j} \notin \{P_g^{\text{FV}}\}, P_{e_i} \notin \{P_e^{\text{FV}}\} \ , \tag{5c}$$

Figure 6: Data filling: The first column are input images. Columns two and three show the Origin and Refined complete mask of layout class, respectively. Columns four and five show the Origin and Refined complete mask of object.

where G_{V_j} is visible part of the complete mask G_j. The function $D(\cdot)$ calculates the depth difference. κ_{area}, κ_{iou} and κ_{depth} are thresholds that we set as $\kappa_{\text{area}} = 100$ pixel, $\kappa_{\text{iou}} = 0.5$ for all experiments and $\kappa_{\text{depth}} = [0.4\text{m}, 0.6\text{m}, 0.9\text{m}]$ for the three instantiation of the metric: [AP0.4, AP0.6, AP0.9]. By excluding visible region from the ground truth, this metric focuses only on predictions in hidden regions and it cannot be cheated by improving predicting the visible planes. Jointly with AP on the complete masks, we now have a better and more comprehensive understanding of the performance of our occlusion reasoning method.

4. Experiments

4.1. Experimental setup

Dataset: Our dataset is built on the large-scale RGB-D video dataset ScanNet[1]. It consists of 2.5 million views in more than 1500 room scans. The annotation includes 3D camera poses, 3D room reconstructions and semantic information. We follow [10] for assigning each scene to training, validation and testing. Then we uniformly sample each split and finally obtain 100k, 1k and 1k images for training, evaluation and testing. To obtain the ground-truth for object and layout categories, we first project the semantic label of each plane to the NYU40 classes. Afterwards, 'wall', 'ceiling', 'floor' and 'window' are categorized as layout. Others belong to the 'object' category.

Implementation details: We implement our network with PyTorch. The pre-trained weights of PlaneRCNN [10] are employed for initializing our model. Our models are trained on an NVIDIA 1080 Ti GPU for 100k iterations with batch size set to 1. Following the setting of PlaneRCNN, input images are scaled to 640×480 and then padded with zero values to be 640×640. The learning rate is set as 1e-4 when we start the training process and it decays to [5e-5, 2e-5, 1e-5] at [5k, 10k, 15k] iterations.

Baselines: We choose the modification of PlaneRCNN [10] for occlusion reasoning as the main baseline for validating the effectiveness of the proposed method. And we denote it as PlaneRCNN-OR (We refer the readers to Sec. 3.2 for more details). Please note that the original PlaneRCNN-OR is proposed in [10] to address the occlusion problem for indoor scenes. Since code for this version is not released, we implement ourselves based on [10]. Moreover, to have a fair comparison, we re-train this model on our dataset where visible and complete masks are generated and report the numbers.

Metric: We evaluate both the plane prediction and depth prediction tasks. As for complete plane prediction, we follow [11] and use AP with different depth constrains to measure the performance. As for performance measurement on hidden regions, we employ the APH introduced in Sec. 3.3. We further employ the following two metrics for depth measurement:

- Root Mean Squared Error(RMSE): $\sqrt{\frac{1}{T}\sum (d_i - g_i)^2}$
- Threshold accuracy: Percentage of d_i, such that $\max(\frac{d_i}{g_i}, \frac{g_i}{d_i}) < \lambda$

T stands for the number of pixels in the image, d_i and g_i indicate the depth value of pixel i in depth and ground truth image. We adopt threshold accuracy with $\lambda = [1.25, 1.25^2, 1.25^3]$ for Acc1, Acc2 and Acc3 respectively.

4.2. Evaluations on data generation

As can be seen in Tab. 2, evaluating model on the "Refine" dataset always gives better results, especially for hidden regions. This is mainly due to the fact that without hole filling, the networks are required to learn to predict those arbitrary holes in the masks (see the highlighted region in Fig. 6), which can be very challenging. Another observation is that models trained with the "Refine" dataset gives a better performance w.r.t. that trained on "Origin". Again, this is mainly because that with the "Refine" dataset, the network is more likely to learn something from meaningful complete masks, especially for hidden regions. While if the network is trained on "Origin", it is harder for this network to learn to predict masks due to these noises/holes.

4.3. DualRPN with semantic merging

DualRPN helps hidden mask prediction Employing DualRPN enables our model to learn category-specific prior knowledge. As shown in Fig. 9b, the baseline method is only able to extend the visible region of floor by a small margin. After introducing DualRPN, we can observe that a larger hidden region of the floor can be predicted in Fig. 9c. We demonstrate quantitative results in Tab. 1, where we can observe the proposed DualRPN greatly improving APH whiel not harming AP. As we described in Sec. 3.2, this behaviour is largely due to the over-suppressing of NMS. We report below the effectiveness of our proposed fusion module.

Semantic merging surpasses NMS We now demonstrate the superiority of the proposed semantic merging module over NMS. Qualitative results are shown in Fig. 9c, where the table is suppressed by NMS due to its heavy overlapping w.r.t. the floor. Our proposed module is able to solve this

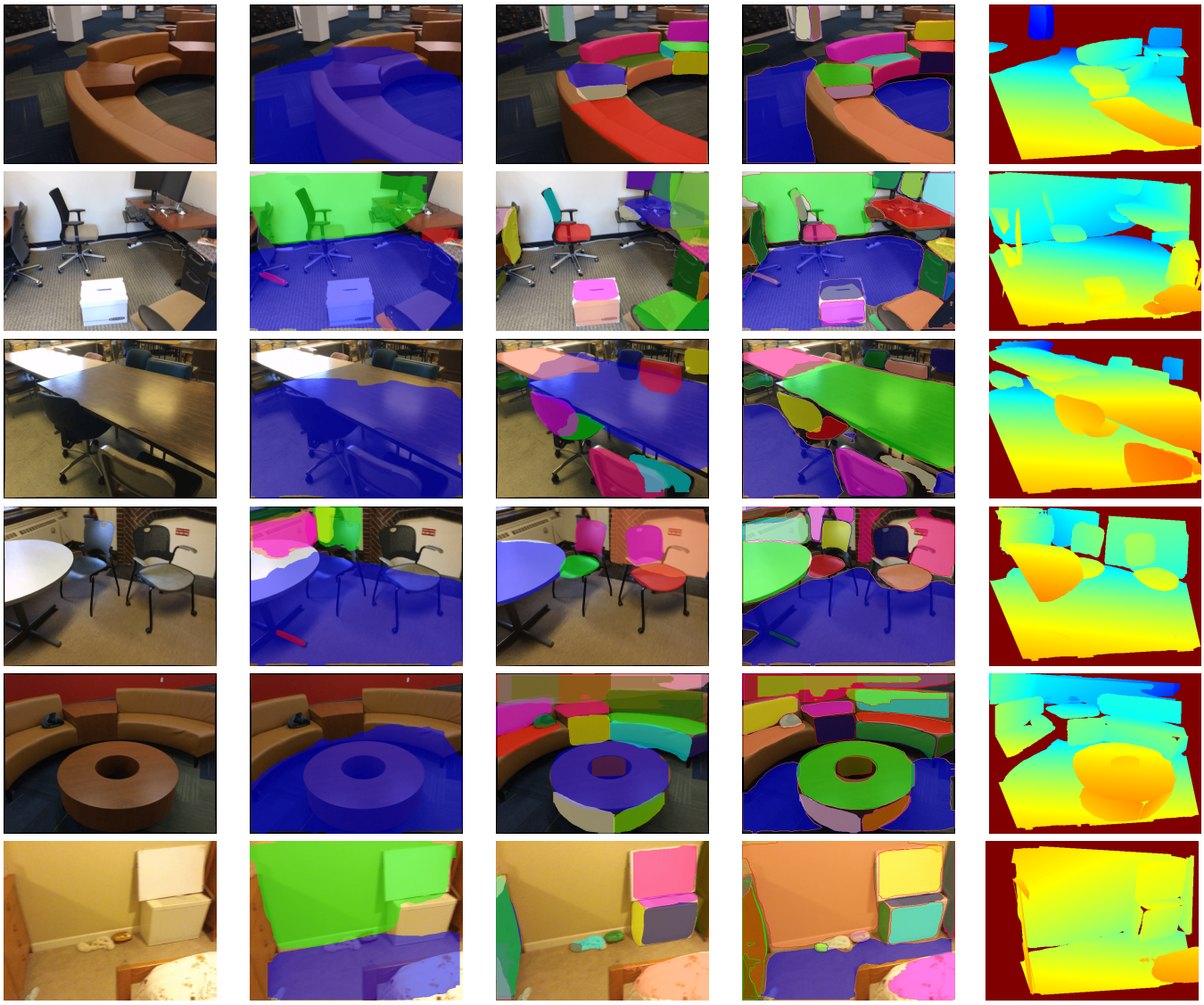


Figure 7: Visualization of proposed method, The first column is the input image. The second column and third column show the complete plane prediction from layout branch and that from object branch respectively. The fourth column demonstrates visible planes and the last column is the novel view synthesis. As can be viewed in our examples, the proposed method is able to predict both visible and complete area from single image, which provides a better representation for indoor scenes.

problem (see examples in Fig. 9d). We also report quantitative results in Tab. 1. Specifically, Semantic Merging improves AP values by [1.6%, 1.8%, 1.9%] compared to NMS. Combining DualRPN with the Semantic Merging, we further pushes APH by [2.9%, 2.9%, 2.6%] and AP by [1.2%, 1.1%, 0.9%].

DualRPN with semantic merging helps depth prediction We further demonstrate that DualRPN benefits the depth prediction. Specifically, we convert the output plane representation to a depth map and report the depth prediction performance on visible areas.As shown in Tab. 3, PlaneRCNN-OR improves Acc1 by 1% and our proposed method further boosts both RMSE and Acc1 by 2%. Our results prove that occlusion reasoning and 3D understanding are mutually beneficial.

4.4. Plane warping module

By enforcing the consistency with neighboring views, our proposed plane warping module helps to improve precision on predicting the complete mask on both visible and hidden regions. As shown in Tab. 1, our proposed method improves AP 0.9 and APH 0.9 by 1% and 0.4%, respectively.

We qualitatively compare our model with PlaneRCNN-OR [10] in Fig. 8, where our model almost always performs

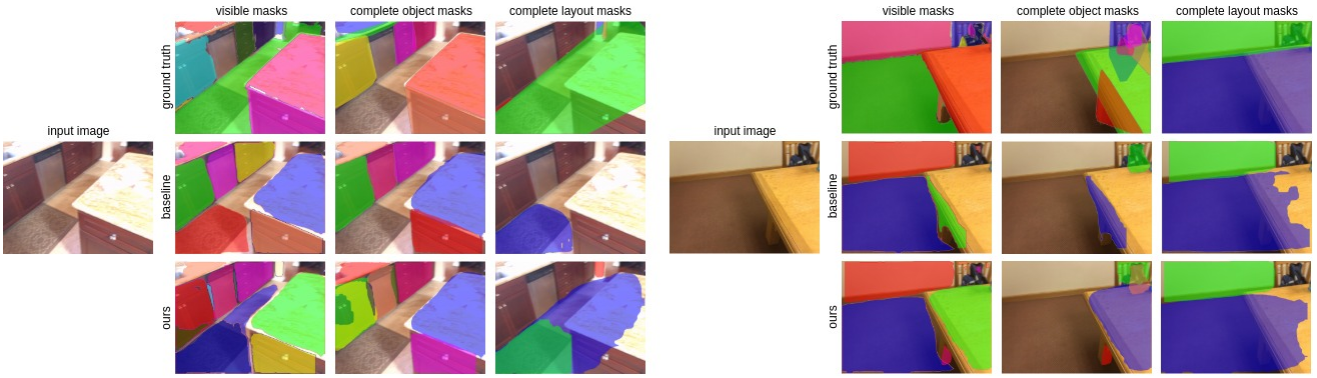


Figure 8: Qualitatively comparison of our method with PlaneRCNN-OR. For each sample, *Left*: The input image. *Right*: From top to bottom - ground truth, baseline prediction, proposed method prediction for visible and complete masks.

Table 1: Ablation study of proposed methods: Dual RPN, Semantic Merging, Channel-wise Attention Module and Occlusion-aware Warping Module. The first row without any proposed modules denotes the PlaneRCNN-OR [10]. The adopted metrics are AP and APH under κ_{iou} set as 0.5 and κ_{depth} set as [0.4m, 0.6m, 0.9m]. A relative improment of 42.65% on APH0.4 is achieved by comparing 0.068 (baseline) with 0.097 (our result).

Dual RPN	Semantic Merging	Plane Warping	AP 0.4	AP 0.6	AP 0.9	APH 0.4	APH 0.6	APH 0.9
			0.319	0.364	0.386	0.068	0.080	0.088
✓			0.315	0.357	0.376	0.092	0.104	0.109
✓	✓		0.331	0.375	0.395	**0.097**	0.109	0.114
✓	✓	✓	**0.334**	**0.382**	**0.405**	**0.097**	**0.111**	**0.118**

Table 2: Evaluation of PlaneRCNN-OR [10] when training and testing on different datasets with [AP0.4, AP0.6, AP0.9] and [APH0.4, APH0.6, APH0.9]

Testing set	Origin		Refine	
Training set	Origin	Refine	Origin	Refine
AP 0.4	0.284	0.284	0.314	**0.327**
AP 0.6	0.316	**0.319**	0.352	**0.371**
AP 0.9	0.334	**0.338**	0.374	**0.395**
APH 0.4	0.007	**0.019**	0.030	**0.068**
APH 0.6	0.009	**0.021**	0.035	**0.080**
APH 0.9	0.012	**0.023**	0.040	**0.088**

(a) (b) (c) (d)

Figure 9: Qualitative results for PlaneRCNN-OR [10] w/ or w/o DualRPN: (a) input image (b) w/o DualRPN (c) w/ DualRPN (d) w/ DualRPN and Semantic Merging.

Table 3: Results for depth prediction.

Method	RMSE↓	Acc1↑	Acc2↑	Acc3↑
PlaneRCNN[10]	0.34	0.78	0.95	0.99
PlaneRCNN-OR[10]	0.34	0.79	0.95	0.99
DualRPN	**0.32**	**0.81**	**0.95**	**0.99**

better in terms of complete mask prediction.

5. Conclusion

This paper proposes to address the occlusion reasoning problem in indoor scenes with efficient plane representations. We firstly generate a dataset where the ground-truth of our occlusion-aware representations are available. Our proposed model separates the prediction for foreground and layout planes for a more effective mask prediction in hidden regions. When multiple views are available at train time, a novel plane warping loss is also introduced to handle occlusion scenarios. Finally, we propose a novel evaluation metric for measuring the performance specifically on hidden regions. Compared to existing methods, our proposed method achieves large relative improvements in hidden regions.

References

[1] 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. 1, 2, 3, 6

[2] Helisa Dhamo, Nassir Navab, and Federico Tombari. Object-driven multi-layer scene decomposition from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5369–5378, 2019. 1, 2, 3

[3] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 125:333–340, 2019. 1, 2

[4] Vikas Dhiman, Quoc-Huy Tran, Jason J. Corso, and Manmohan Chandraker. A continuous occlusion model for road scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1

[5] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3D Traffic Scene Understanding from Movable Platforms. *PAMI*, 2014. 1, 2

[6] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3, 5

[7] Varsha Hedau, Derek Hoiem, and David Forsyth. Recovering the spatial layout of cluttered rooms. In *2009 IEEE 12th international conference on computer vision*, pages 1849–1856. IEEE. 2

[8] Chen-Yu Lee, Vijay Badrinarayanan, Tomasz Malisiewicz, and Andrew Rabinovich. Roomnet: End-to-end room layout estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4865–4874, 2017. 2

[9] Buyu Liu, Xuming He, and Stephen Gould. Joint semantic and geometric segmentation of videos with a stage model. In *IEEE Winter Conference on Applications of Computer Vision*, pages 737–744. IEEE, 2014. 2

[10] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 3, 4, 5, 6, 7, 8

[11] 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. 2, 3, 6

[12] Miaomiao Liu, Xuming He, and Mathieu Salzmann. Building scene models by completing and hallucinating depth and semantics. In *European Conference on Computer Vision*, pages 258–274. Springer, 2016. 2

[13] Gellért Máttyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. HD Maps: Fine-grained Road Segmentation by Parsing Ground and Aerial Images. In *CVPR*, 2016. 1

[14] Suryanarayana M Muddala, Märten Sjöström, and Roger Olsson. Depth-based inpainting for disocclusion filling. In *2014 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2014. 2

[15] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 5

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 2015. 2, 3

[17] Samuel Schulter, Menghua Zhai, Nathan Jacobs, and Manmohan Chandraker. Learning to look around objects for top-view representations of outdoor scenes. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[18] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. 1998. 1, 2

[19] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas Funkhouser. Semantic Scene Completion from a Single Depth Image. In *CVPR*, 2017. 1

[20] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[21] Joseph Tighe, Marc Niethammer, and Svetlana 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

[22] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 1, 2

[23] Huayan Wang, Stephen Gould, and Daphne Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 2

[24] Ziyan Wang, Buyu Liu, Samuel Schulter, and Manmohan Chandraker. A parametric top-view representation of complex road scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[25] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[26] Yi Yang, Sam Hallman, Deva Ramanan, and Charless Fowlkes. Layered object detection for multi-class segmentation. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3113–3120. IEEE, 2010. 2

[27] Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. Structured3d: A large photo-realistic dataset for structured 3d modeling. *arXiv preprint arXiv:1908.00222*, 2019. 2

[28] M. Zeeshan Zia, Michael Stark, and Konrad Schindler. Explicit Occlusion Modeling for 3D Object Class Representations. In *CVPR*, 2013. 1