Geometry to the Rescue: 3D Instance Reconstruction from a Cluttered Scene

Lin Li
Data61-CSIRO, ANU
lin.li@anu.edu.au

Salman Khan
IIAI, ANU
salman.khan@anu.edu.au

Nick Barnes
ANU
nick.barnes@anu.edu.au

Abstract

3D object instance reconstruction from a cluttered 2D scene image is an ill-posed problem. The main challenge is posed by the lack of geometric information in color images and heavy occlusions that lead to incomplete shape details. To deal with this problem, existing works on 3D instance reconstruction directly learn the mapping between the intensity image and the corresponding 3D volume model. Different from these works, we propose to explicitly incorporate 2.5D geometric cues, such as the surface normal, relative depth, and height, while generating full 3D shapes from 2D images. With an intermediate step focused on estimating these 2.5D geometric features, we propose a novel convolutional neural network design that progressively moves from 2D to full 3D estimation. Our model automatically generates instance-specific surface normal maps, relative depth, and height that are compactly encoded within our network design and consequently used to improve the 3D instance reconstruction. Our experimental results on the large-scale synthetic SUNCG dataset and the real-world NYU depth v2 dataset demonstrate the effectiveness of the proposed approach where it beats the state-of-the-art Factored3D network [15].

1. Introduction

3D instance reconstruction provides valuable information about an object's shape and pose in the real-world. Such details are fundamental to scene understanding that, in turn, helps a diverse set of important applications such as robotic navigation, object grasping, context-aware digital assistants, and augmented reality. 3D reconstruction from a single monocular image is, however, an ill-posed problem that is further complicated by the heavy occlusions, cluttered regions, illumination variation, and the diverse range of object types commonly present in indoor scenes.

Recent research proposes to solve this problem by first performing 2D object detection followed by 3D single instance reconstruction [15]. They leverage the large-scale synthetic dataset SUNCG [14] with its abundant 3D annotations to learn their predefined 3D object pose and shape parameters. However, they do not 'explicitly' consider any geometric cues, which are essential for 3D reconstruction. In the literature, 2.5D geometric representations have been identified as informative cues to better constrain the reconstruction task [10]. In this work, we propose an efficient way to use geometry-driven representations as complimentary features for 3D object instance reconstruction. First, since the surface shape is a view-invariant geometric representation for 3D objects, here we select surface normals to distinctly characterize the 3D surfaces. In contrast to the traditional generation of surface normal approximation from the depth image, we utilize 3D mesh models to generate an accurate surface normal. The features based on surface normals are efficiently computed from the input monocular images and subsequently used for improved 3D reconstruction. Second, the importance of object height and depth for object reconstruction in human vision has been advocated by [10, 17]. Motivated by Marr's work [10], we compute relative height and depth as additional geometric cues in-

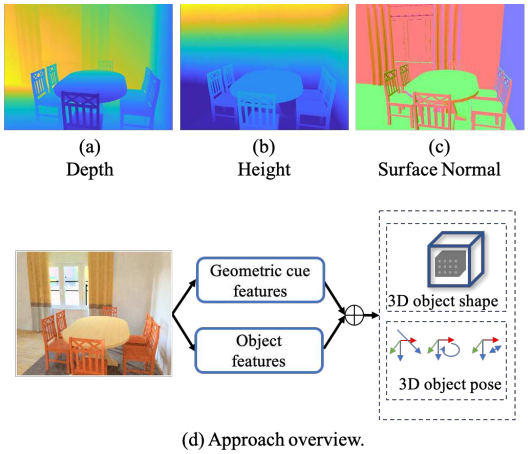


Figure 1. Illustration of the geometric cues (*top*) and an overview of our approach (*bottom*). The geometric cues we propose to use are (a) depth map, (b) height map, and (c) surface normal map. Our network architecture shown in (d) compactly encodes geometric and object features to estimate 3D instance shape and pose.

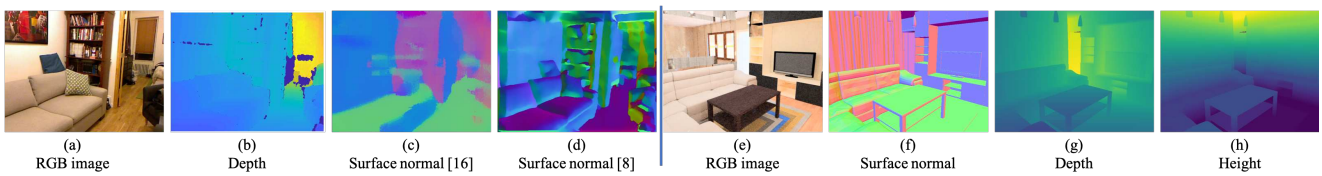


Figure 2. Comparison between geometric images obtained from **real** (a-d) and **synthetic** (e-h) data. For real data, NYU depth v2 [13] is used, (b) is the raw depth image, (c) is generated from depth image through a least-square solution [16], (d) is a more smoothed version using depth [8]. For synthetic data SUNCG [14] is used, (e) is the RGB image, (f) surface normal, (g) depth and (h) height map.

stead of absolute height and depth for 3D object instance reconstruction from a cluttered scene.

The key features of our work are: (1) We propose a way to generate and model surface normals, relative height, and depth as latent features to ensure that the reconstructed 3D shape implicitly conforms with the local geometry. (2) The surface normals, relative height, and depth are only required at training time, not inference time, and do not require manual labeling. (3) Experimental results show that our method reduces 3D shape uncertainty as well as improves pose estimation accuracy over the state-of-the-art methods.

2. Related Work

Several works have proposed to use geometry information for vision problems. [3, 18] recommended using stereo left-right geometry consistency constrains for single view depth estimation. However, the stereo setting is quite sensitive to texture and lighting. To reconstruct 3D vertices of a non-rigid surface from a single view, [11] proposed to combine the estimation of 2D depth and 3D vertices through the Procrustes transformation. However, their methods only deal with the non-rigid surfaces, like paper sheets, while indoor object instances are usually with planar surfaces [9, 4] and have a compact form in the 3D space. [12] tries to learn depth and surface normal estimation in a cyclic style to regularize the rough estimates with the local tangent surface consistency. However, this method requires a reasonable initial depth and surface normal estimation first.

3. Our Approach

In this work, we demonstrate the importance of geometric information, encoded in the form of explicit feature maps, as shown as (a-c) in Fig. 1, for 3D object instance reconstruction from a single color image of a cluttered scene. Specifically, we investigate the use of a surface normal map, relative height, and depth as the geometric primitives for the representation of a 3D surface. The main challenge is how to effectively use surface normals, relative height, and depth information for the generic object instance reconstruction task. To this end, we propose an efficient architecture that explicitly learns a 2.5D mapping that is subsequently used for an improved 3D reconstruction. Our network design requires no extra annotation and needs only a single intensity image to infer a complete 3D shape.

Below, we first elaborate on our geometric features, followed by the network architecture and the training strategy.

3.1. Geometric Features

Previous works [6, 7, 16, 5, 12] deal with surface normal estimation based on real datasets. They capture real depth images by laser scanners, commodity depth sensors, or stereo cameras. However, all these captured depth contain noise or scarcity, and an error will be accumulated after local surface normal approximation through a least-square calculation. Even though [8] proposed to denoise local surface normal estimation through optimization to keep surface edges, the method still has high computation cost and problems with reflective surfaces and insufficient local neighbor depth measurement. In contrast, we utilize synthetic data to efficiently generate more accurate and smooth surface normal, depth, and height images. To illustrate this discrepancy, we show some geometric image examples in Fig. 2.

3.1.1 Instance-centered Surface Normal

The surface shape is a view-invariant geometric primitive for a 3D object. The surface normal per 2D pixel is an efficient denotation of the surface compared to complex surface representations such as a 3D mesh representation (3D vertices, edges, and faces). We propose to harness readily available synthetic data to obtain the exact surface normal maps using a surface mesh technique. Hence the surface normal maps we use are more accurate than the previously used least-square solution. Furthermore, to exclude influences from each other, we obtain the normalized instance-centric surface normal vector $\vec{n} \in \mathbb{R}^3$ for each instance.

3.1.2 Instance-centered Relative Height

As described by the Manhattan world assumption [2], human-made environments exhibit a regular structure, e.g., most objects lie in parallel to the ground surface. Here, we design a height above ground feature $h \in \mathbb{R}$ to emphasize on the 3D space organization from the top view for each visible pixel. However, direct use of an absolute height

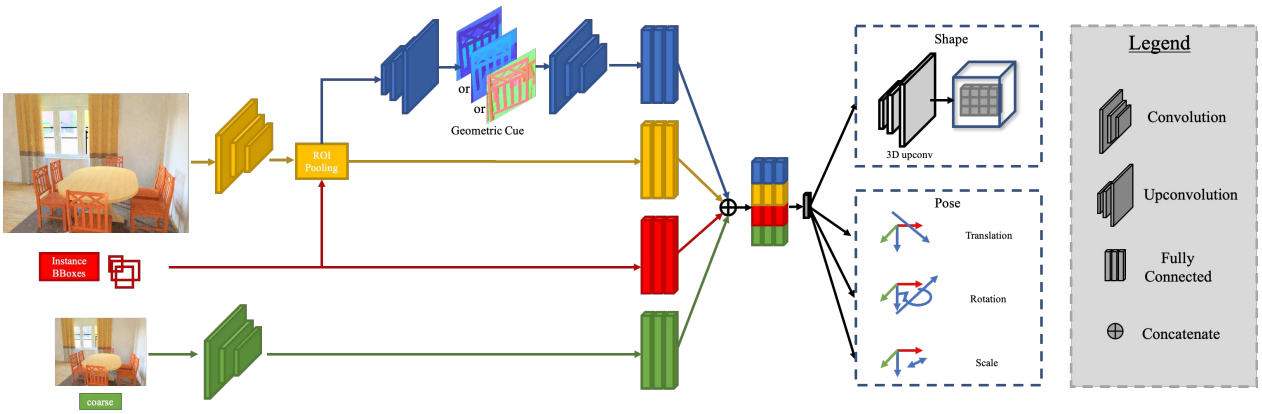


Figure 3. The complete network architecture of our work. Input is a single cluttered scene RGB image. Blue branch is for Instance-centered geometric feature estimation and encoding, namely, surface normal, relative depth or height. Yellow branch is for ROI pooled features. Red branch is for bounding box features. Green branch for coarse feature extraction. All the above features are concatenated to a latent feature space. Then shape and pose predictors estimate 3D instance shape and pose separately. *Better seen in color.*

measure does not provide a sound geometric feature due to the changes in viewpoint across different scenes. For single object reconstruction, we argue that relative height per object is a more proper geometric cue instead of the absolute height. Specifically, each object's height is normalized by the total height variation of itself in a scene. We compute the relative height $h = (h - \min(h))/(\max(h) - \min(h))$ from mesh model of synthetic data.

3.1.3 Instance-centered Relative Depth

As a complementary feature to h, depth $d \in \mathbb{R}$ captures the 3D organization from the camera view. For objects in a cluttered scene, the original absolute depth is not that important for 3D reconstruction, while relative depth d of object instance is more helpful, here we propose to encode depth to relative depth $d = (d - \min(d))/(\max(d) - \min(d))$ as the depth offset normalized by object size.

3.2. Network Architecture

We treat instance object reconstruction from a single image of a cluttered scene as a combination of 2D object detection and 3D reconstruction tasks, following the initial network design of Factored3D [15]. To add geometric cues into this architecture, we propose three branches, respectively, for surface normal, height, and depth estimation and encoding. The complete network architecture is shown in Fig. 3. Our model consists of the following parts: (1) Global and local feature extraction, (2) Instance-centered feature extraction, (3) Geometric cues estimation and encoding, (4) Bounding box encoding, and (5) 3D object shape and pose prediction. Here, we only focus on the surface normal, relative height and depth estimation and encoding part, please refer [15] for details on other parts.

Based on the ROI pooled instance-centered features, we first estimate 2D object structure using a surface normal estimator network and then compactly represent it through an encoder stage. The surface normal estimator can be understood as a decoder (or a generator) that contains one layer of 2D up-convolution, one layer of 2D convolution, and one sigmoid layer towards the end. It generates a three-channel instance-centered surface normal map $\vec{n}$. In the next stage, the surface normal encoder projects the estimated normal to a latent surface normal feature. Precisely, the encoder consists of two convolution and two 300-unit fully-connected layers. The network architecture of height and depth estimator is similar to the surface normal network, except the final map predicted by the decoder is one channel.

For object-centric features, bounding box features, and coarse object features, we follow the architectures proposed in [15]. Afterwards, all the features are concatenated.

3.3. Network Training

We train our proposed network design in two stages. The first stage is for training the geometric cue estimators separately with the corresponding losses. The second is for training all feature branches, as in Fig. 3 with instance, shape and pose losses, training together with the other parts. For 3D instance, shape, and pose output, we follow the object shape normalization and relative pose configuration in [15]. We explore each surface normal, depth, and height features separately. The objective functions used for the training are explained below.

Surface Normal Estimation Loss: For surface normal learning, we use a cosine angle loss Eq. 1 between the predicted instance-centered surface normal and the ground-truth. The loss is given by:

$$L_n = \langle \vec{n}, \hat{\vec{n}} \rangle, \tag{1}$$

where, $\vec{n}$ is ground-truth instance-centered surface normal,

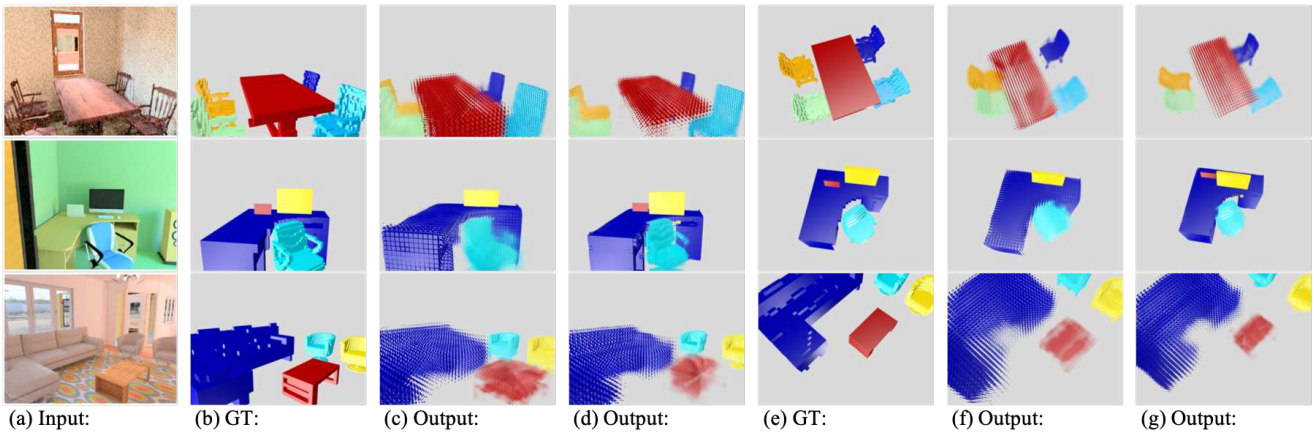


Figure 4. Visualization of 3D reconstruction with ground-truth bounding boxes on the SUNCG test dataset. Each row is one comparison between our surface normal branch and [15]. Our predicted shape has better object shape, pose estimations, more details in Section 4.4 qualitative part. Instance color is only to distinguish between object instances.

$\hat{\hat{n}}$ is the estimated surface normal.

Height Estimation Loss: For relative height loss, we utilize a mean square error loss Eq. 2 between the predicted instance-centered relative height and the ground-truth,

$$L_h = \frac{1}{N}\sum_i (h_i - \hat{h}_i)^2. \tag{2}$$

Depth Estimation Loss: For relative depth loss, we utilize a mean square error loss Eq. 3 between the predicted instance-centered relative depth and the ground-truth,

$$L_d = \frac{1}{N}\sum_i (d_i - \hat{d}_i)^2. \tag{3}$$

3D Shape Loss: We use a voxel representation $V = \{v_i\}$ for 3D shape, where $v_i \in \{0, 1\}$. $\hat{v_i}$ is the predicted voxel occupancy probability for voxel at location i. We treat shape estimation as a voxel-level binary classification problem, so we apply a voxel-level cross entropy loss Eq. 4 to learn this representation,

$$L_V = \frac{1}{N}\sum_i (v_i \log \hat{v_i} + (1 - v_i)\log(1 - \hat{v_i})). \tag{4}$$

3D Pose Loss

- *Rotation.* Our objective loss for rotation is the negative logarithm likelihood loss Eq. 5 for the predicted probability of the ground-truth class $\hat{q}^g$. $\hat{q}$ is the predicted probability over all 24-bin classes,

$$L_q = -\log(\hat{q}^g). \tag{5}$$

- *Scale.* We use squared Euclidean distance Eq. 6 between predicted scale values $\hat{s}$ and ground-truth s. This distance is calculated in the logarithm space to reduce the influence of magnitude,

$$L_s = \|\log(s) - \log(\hat{s})\|_2^2. \tag{6}$$

- *Translation.* Translation loss is depicted as Euclidean loss Eq. 7 between predicted $\hat{t}$ and ground-truth t.

$$L_t = \left\|t - \hat{t}\right\|_2^2. \tag{7}$$

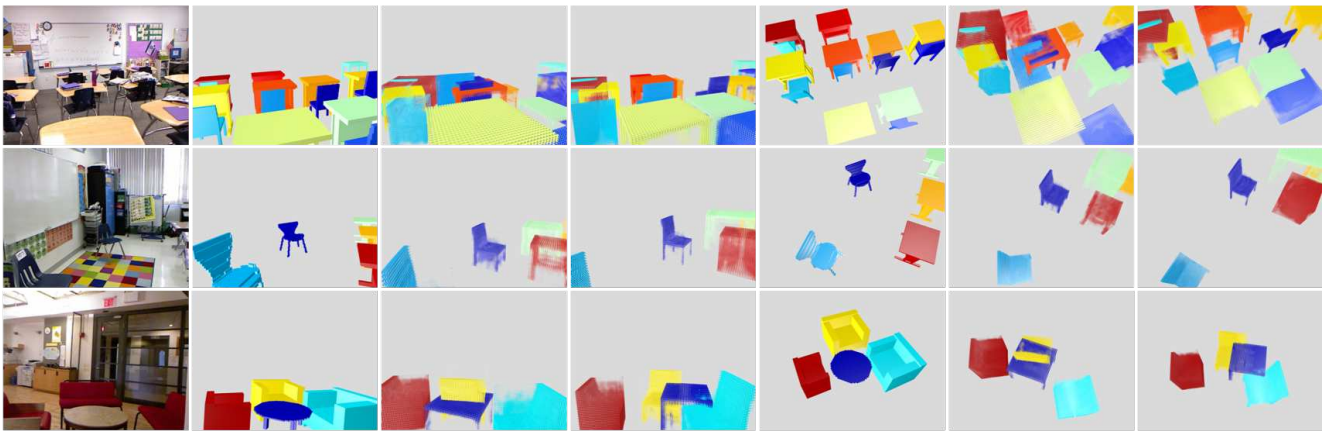
Training: We train geometric cue estimation with its correspondent estimation loss first, and then train the geometric cue estimator and encoder with all other modules together using weighted 3D shape and pose loss,

$$L = \sum_{b\in\mathcal{B}+} (w_V L_V + w_q L_q + w_s L_s + w_t L_t - \ln(f)) + \sum_{b\in\mathcal{B}-} \ln(1 - f). \tag{8}$$

Here w_V, w_q, w_s, w_t are the loss weights and $\mathcal{B}+, \mathcal{B}-$ means positive and negative bounding boxes, respectively.

4. Experiments

Here, we show our experimental results for 3D object instance reconstruction from a single cluttered image. We use the SUNCG [14] synthetic indoor scene dataset for network training as there are complete 2D and 3D annotation of objects. To explore our method's generalization ability to a real scenario, we show the evaluation results on the NYU depth v2 dataset [13]. As we show in Fig. 2, there exist a lot of noise in the surface normal and depth images from the real dataset. Hence, for NYU depth v2 dataset, we fine-tune the whole network trained on SUNCG dataset. We compare our method with others both qualitatively and quantitatively on the SUNCG dataset and NYU depth v2 datasets.



(a) Input: a cluttered scene RGB image. (b) GT: 3D object instance in camera view. (c) Output: Factored3d in camera view. (d) Output: ours in camera view. (e) GT: 3D object instance in top view. (f) Output: Factored3d in top view. (g) Output: ours in top view.

Figure 5. Visualization of 3D reconstruction with ground-truth bounding boxes on NYU depth v2 test set. Each row is one comparison between our surface normal branch and [15]. Our predicted output has better object shape, pose estimations, more details in section qualitative text part of Sec. 4.4. Instance color is only to distinguish between object instances.

We investigate the effectiveness of each of the three geometric cues. We follow the evaluation criteria from [15] for a fair comparison. The qualitative and quantitative results under these criteria show that our method performs better than state-of-the-art methods.

4.1. Dataset

SUNCG Dataset: This is a large-scale synthetic indoor scene dataset [14], containing 45,622 3D houses, 2644 object instance CAD models over 84 object categories. It is composed of simulated rooms and furniture from [1]. This furniture is well selected and arranged to simulate real scenes so that 3D object shape and pose are credible. For a fair comparison, we select the same six object categories - bed, chair, desk, sofa, table, television as [15]. We use the same split 70/10/20 train/test/val as [15]. **Surface normal generation:** Thanks to synthetic data, we calculate surface normal directly from the object CAD models. Specifically, for each surface pixel, we have a corresponding 3D triangle polygon and its normal. We make sure each normal orientation is consistent with view point angle, otherwise we flip it to ensure consistency. **Height generation:** For relative height generation, in the synthetic dataset like SUNCG, we can obtain the ground height, as the dataset is designed to meet the Manhanttan assumption, therefore the y axis of real world coordinate (X, Y, Z) is along the gravity direction. So absolute height for each 3D point (x, y, z) could be calculated by the subtraction $h_{ab} = y - y_g$ between y and ground floor height y_g. Then the relative height per each object h is calculated by $h = (h_{ab} - \min(h_{ab}))/(\max(h_{ab}) - \min(h_{ab}))$. **Depth generation:** To generate relative depth d, we calculate $d = (d - \min(d))/(\max(d) - \min(d))$. This depth d can be directly calculated by the distance between the 3D point of camera and each point from synthetic dataset.

Dataset	Method	Shape(IoU)			
		Median IoU ↑	Mean IoU ↑	%(IoU > 0.25) ↑	%(IoU > 0.5) ↑
SUNCG	Factored3D	0.48	0.49	75.06	47.49
	surface normal	0.61	**0.594**	**81.29**	59.64
	height	**0.61**	0.588	80.66	**60.19**
	depth	0.61	0.587	81.04	59.74
NYUv2	Factored3D	0.48	**0.53**	77.37	49.17
	surface normal	0.51	0.52	**78.56**	50.96
	height	**0.53**	0.52	77.37	**51.75**
	depth	0.51	0.51	78.29	50.69

Table 1. 3D shape estimation results with ground-truth bounding boxes on the test set of SUNCG dataset and NYU depth v2 dataset. Factored3D is the work from [15].

NYU depth v2 Dataset: [13] uses a Kinect sensor to capture a variety of indoor scenes. There are 1449 images with camera information, with 795 images for training and 654 images for testing. Guo *et al*. [4] offer 3D surface mesh annotation for object instances. We select the same object categories as we do for the SUNCG dataset. Since then, we have the 3D annotations to make a quantitative and qualitative comparison on this dataset. We fine-tune the network trained on SUNCG and show our approach's generality.

4.2. Evaluation Criteria

As object instance reconstruction from cluttered scenes is a composition of 2D object detection and 3D object shape and pose estimation, we need to evaluate object detection, single object shape, or 6D pose estimation separately for each result along with ground-truth bounding boxes. So for each result with shape (V) and pose (quaternion q, scale s, and translation t) estimation, we make a separate evaluation over them through different thresholds δ. We follow the basic settings of [15]. Besides, we found these thresholds

Dataset	Method	Translation (meters)				
		MedErr ↓	MeanErr ↓	%(Err < 1.) ↑	%(Err < 0.5) ↑	%(Err < 0.1) ↑
SUNCG	Factored3D	0.303	0.578	91.09	74.46	6.80
	normal	0.267	0.463	**93.22**	78.81	9.66
	height	**0.261**	**0.461**	92.99	78.76	**9.97**
	depth	0.265	0.466	93.03	**78.85**	9.46
NYUv2	Factored3D	**0.55**	**0.71**	**79.42**	**44.14**	1.46
	normal	0.56	0.75	76.97	44.01	1.59
	height	0.56	0.75	77.56	42.95	1.46
	depth	0.56	0.75	77.23	43.35	**1.85**
Dataset	Method	Rotation (degrees)				
		MedErr ↓	Mean Err ↓	%(Err<30) ↑	%(Err < 10) ↑	%(Err< 5) ↑
SUNCG	Factored3D	5.02	31.80	77.90	70.77	49.87
	normal	4.41	24.61	83.83	77.14	55.12
	height	**4.382**	**24.29**	**84.07**	**77.44**	**55.51**
	depth	4.385	24.45	83.92	77.35	55.31
NYUv2	Factored3D	15.92	43.43	62.87	32.23	9.07
	normal	**14.82**	41.40	**66.51**	**34.35**	**9.53**
	height	15.19	**41.00**	65.32	33.69	8.54
	depth	14.98	42.26	65.25	32.69	8.54
Dataset	Method	Scale				
		MedErr ↓	Mean Err ↓	%(Err<0.5) ↑	%(Err<0.3) ↑	%(Err<0.2) ↑
SUNCG	Factored3D	0.1208	0.2263	87.67	75.79	64.43
	normal	0.1051	0.1978	90.40	79.60	68.38
	height	0.1058	0.1995	90.28	79.48	**68.69**
	depth	**0.1041**	**0.1977**	**90.44**	**79.63**	68.67
NYUv2	Factored3D	0.39	0.422	68.96	33.55	14.76
	normal	0.39	0.43	67.17	33.16	**16.15**
	height	0.39	0.43	67.37	32.69	14.16
	depth	**0.38**	**0.421**	**70.68**	**35.14**	16.02

Table 2. 3D pose estimation results with ground-truth bounding boxes on the test set of SUNCG dataset and NYU depth v2 dataset. Factored3D is the work from [15].

are a little loose and added more strict settings for them. We will denote all the details in the quantitative part.

4.3. Implementation Details

For the model trained on the SUNCG dataset, we train with ground-truth bounding boxes for four epochs and proposals [19] with one epoch for each geometric cue estimator. Then, we train each estimator separately with an encoder and other network parts together with ground-truth bounding boxes for four epochs and object proposals for five epochs. We fine-tune the network parts from [15] and fine-tune the shape decoders during the second training stage. We keep the hyper-parameter settings as [15]. Then for the NYU depth v2 dataset, since in real scenario, objects are much more cluttered than synthetic datasets, we only fine-tune the models trained on SUNCG with ground-truth boxes and their corresponding 3D annotation. For the SUNCG trained models, we use batch size 8, while for the model fine-tuned on the NYU depth v2, we use batch size 16. For all other parameters we follow [15].

4.4. Compare with State-of-the-art

To analyze 3D instance reconstruction and 2D object detection from a cluttered scene image separately, we directly use the ground-truth bounding boxes to evaluate reconstruction performance.

Qualitative results: For qualitative results, we show some examples from surface normal cue as ours indicator in Fig. 4 on SUNCG dataset and Fig. 5 on NYU depth v2 dataset. Object instance reconstruction results are represented by volume, and a voxel size of each volume denotes objectiveness occupancy as [15]. Each row is a result representation for one single cluttered scene image. To have a comprehensive comparison, we show 2D rendered images of 3D object instances in two views, namely, camera view column (b-d) and top view column (e-g). First, our methods have better object shape detail estimation. For example, in row 2, the reconstructed table leg from ours is more similar to the ground-truth than the ones from [15]. Also, our method has better small object estimation quality, as shown in rows 3; better pose estimation quality as shown in row 1; better estimation quality from bad lighting image as shown in row 1. We will show quantitative results in the following quantitative analysis section to clarify our advantages of all geometric cues.

Quantitative Results: We evaluate the 3D shape and pose estimation based on the evaluation criteria given in Sec. 4.2.

(a) Shape evaluation: We evaluate shape estimation on criteria of the median and mean IoU, and precision $\%$ based on two thresholds $\%(\delta_V = 0.25, 0.5)$. From Table 1, we can see that the joint modeling of object and surface normal has led to a substantial improvement, especially under the more strict threshold. These results demonstrate that the geometric cues help reduce the 3D shape estimation uncertainty. Besides, we found all these geometric cues do improve the baseline method, while the surface normal is slightly better or at least comparable to others.

(b) Pose evaluation: We evaluate rotation, translation, and scale estimation and show results in Table 2. Our work outperforms [15] both in terms of error and precision measures for rotation, translation, and scale estimation. Specifically, for rotation, the fine-tuned model on NYU depth v2 shows surface normal is better than other geometric cues. For scale, interestingly, we found the depth cue works better than other geometric cues, we assume this is due to the its relevance to object size. For translation, we found that the existing Factored3D model works pretty well.

5. Conclusion

We present an effective way to generate accurate geometric cues, specifically, surface normals, height, and depth maps. To include these geometric cues in 3D instance reconstruction from a single image, we propose an efficient way to encode latent features to concatenate with other features for 3D instance reconstruction. We find this strategy improves the baseline method. For shape, rotation, and scale, there is a clear improvement. In summary, for 3D reconstruction, we find that training using geometric cues from synthetic data improves results for learning-based methods. For future work, we will explore the use of geometric cues in the 3D output space directly.

References

[1] planner5d. https://planner5d.com/.

[2] James M Coughlan and Alan L Yuille. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In *NIPS*, 2001.

[3] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

[4] Ruiqi Guo and Derek Hoiem. Support surface prediction in indoor scenes. In *ICCV*, 2013.

[5] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[6] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013.

[7] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014.

[8] L Ladickỳ, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, 2014.

[9] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *CVPR*, 2018.

[10] David Marr. A computational investigation into the human representation and processing of visual information, 1982.

[11] Albert Pumarola, Antonio Agudo, Lorenzo Porzi, Alberto Sanfeliu, Vincent Lepetit, and Francesc Moreno-Noguer. Geometry-aware network for non-rigid shape prediction from a single view. In *CVPR*, 2018.

[12] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[13] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[14] 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.

[15] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[16] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015.

[17] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *NIPS*, 2017.

[18] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[19] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.