Silhouette-Assisted 3D Object 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

The objective of our work is to reconstruct 3D object instances from a single RGB image of a cluttered scene. 3D object instance reconstruction is an ill-posed problem due to the presence of heavily occluded and truncated objects, and self-occlusions that lead to substantial regions of unseen areas. Previous works for 3D reconstruction take clues from object silhouettes to carve reconstructed outputs. In this paper, we explore two ways to include silhouette learnable in the network for 3D instance reconstruction from a single cluttered scene image. To this end, in the first approach, we automatically generate instance-specific silhouettes that are compactly encoded within our network design and used to improve the reconstructed 3D shapes; in the second approach, we find an efficient design to regularize object reconstruction explicitly. Experimental results on the SUNCG dataset show that our methods have better performance than the state-of-the-art.

1. Introduction

3D shape and pose estimation for 2D object instances is a fundamental tool for scene reasoning. Such a deep understanding about visual content can greatly help in applications such as robotics and augmented reality. Specifically, accurate 3D estimation is essential for problems like object grasping, navigation, and content augmentation. Despite recent attempts, this important problem is far from being solved and significant improvements in accuracy are required for 3D shape and pose estimation in cluttered scenes.

A recent research work named Factored3D [23] takes a single view of a cluttered scene and estimates the 3D shape and pose of instances without any supervision from the 2D scene structure. This approach factorizes room layout estimation and 3D instance reconstruction into two discrete steps to reduce the task complexity. However, it does not consider 2D structure constraints, such as an instance silhouette, that encompasses valuable shape information. Kundu *et al*. [9] uses a 2D silhouette constraint explicitly by a render-and-compare 3D-CAD model driven approach for

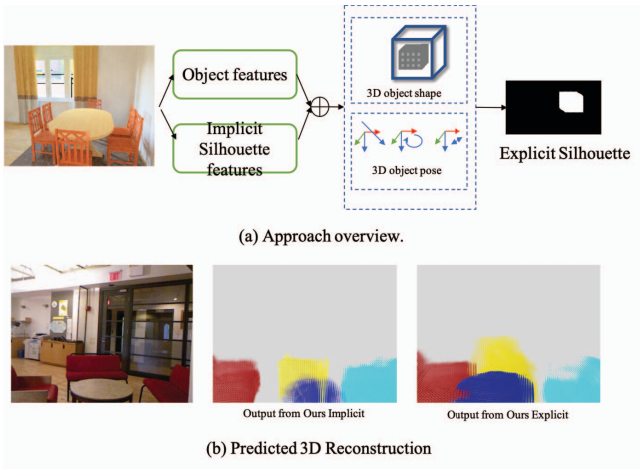


Figure 1: Overview of our work. Given a single cluttered RGB scene image, our goal is to estimate 3D object instance shape and pose. In this work, we emphasize the importance of silhouette for 3D instance reconstruction and have explored two ways, namely, implicit and explicit ones as briefly illustrated in (a). To visualize the estimation output, we represent 3D object shapes by a volume grid, the size of each voxel illustrates the probability of occupancy. Our approaches is able to reconstruct compact and accurate 3D representations of the 2D object instances even from real images, one example shown in (b). (see row 1, Fig. 6 for detailed comparison). *Best seen in color and high resolution.*

3D instance reconstruction. However, in an indoor scene scenario, extension of this method requires a much larger generalized CAD model space that is prohibitive to develop. In this work, we explore two ways to include silhouette as a 2D structure cue in our deep network. In the first approach, we present a method to efficiently generate an instance-level silhouette and **implicitly** include it in our network. Our approach also avoids the additional computation required for rendering the silhouette during inference time. While the use of an instance silhouette is intuitive, the acquisition of silhouette annotations at large-scale is non-trivial.

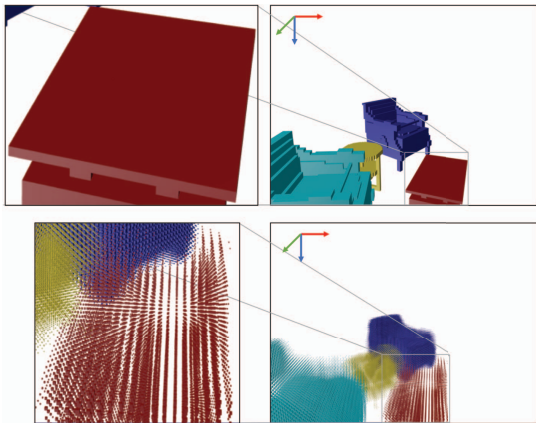
For scene instance segmentation annotation, the typical approach (e.g., [32, 16]) is to use human annotation acquired from crowd-sourcing platforms, e.g. Amazon Mechanical Turk service. However, this is an expensive process and may not be precise due to human perceptual bias. Here, we propose a way to generate instance-level amodal silhouettes automatically at a large-scale. It requires no human intervention and is precise. In the second approach, we propose an efficient method to **explicitly** regularize 2D structure of object instance through perspective silhouette projection without much memory and computation cost (in contrast to techniques like z-buffer). With our predefined single-object-only scene volume, we can generate the perspective silhouette projection efficiently and make it differentiable to regularize object shape and pose parameters.

Our work has four contributions: (1) We propose an approach to model the silhouette as a latent feature to ensure correct 2D projection of a 3D shape implicitly. The silhouette is only required at training time, not inference time and does not require manual labeling. (2) We also propose another approach to regularize 3D reconstruction explicitly through perspective silhouette projection and make it differentiable. (3) Experimental results show that both of our methods reduce 3D shape uncertainty as well as improving pose estimation accuracy.

2. Related Work

3D Indoor Scene Understanding. The goal of indoor scene understanding is to estimate the type of scene, category and location of objects, and the relationship between them [29, 14, 31, 7]. Previous works explore 3D indoor scene understanding by dealing with problems of depth or surface normal estimation [11, 3, 2, 13], 3D object detection [17, 5, 8, 18, 33, 12, 25], and 3D holistic scene understanding (joint object detection, room layout and camera pose estimation) [6]. In summary, the estimation of structure and arrangement of scene elements are critical problems for indoor scene understanding. After we obtain 3D instance shape and pose, the 3D object bounding box, depth and layout are easy to obtain. Further, since 3D object size and pose represent the location and geometry of a 3D object instance, support relations could also be inferred easily. Therefore, 3D object instance reconstruction is a key basis for 3D indoor scene understanding.

Single-Object Image Reconstruction. A single-object image means there exists only one object in the image. This scenario contains no object truncation or occlusion. One solution is to consider photo-consistency from different views, like traditional Space-carving method [10], or learning methods [15]. Smith *et al*. [15] propose to sculpt an object via the learning-based estimation of multi-view high resolution depth and silhouette. Other methods [24, 22, 26] use differentiable projection from new view prediction to learn 3D single-object shape. However, in a single-object instance scene, an instance is centered and well posed, therefore generating a multi-view silhouette is easy through enumerating azimuth, and elevation angles. In contrast, in a cluttered scene, the object instances has a large pose variations with each other, and visibility of object parts can not maintain through multiple view points due to the object occlusion and truncation. Therefore obtaining a ground-truth multi-view object silhouette or novel viewpoint prediction is not easy for one object in a cluttered scene. To address this problem, we propose a 2D silhouette in a constrained way of single view. Another category of learning based methods propose category-specific reconstruction by using a generative adversarial network [28], or an auto-encoder [20, 27, 21]. However, these methods learn separate models for different object categories.



(a) Zoomed in 3D object shape visualization.

(b) 3D objects visualization from a cluttered scene.

Figure 2: 3D object shape visualization. The first row is the ground-truth, second row is the prediction from [23]. The level of blur shows the uncertainty of a shape region.

3. Our Approach

To reconstruct a 3D object instance from a single cluttered scene image, we emphasize the importance of 2D silhouette structure. In this work, we propose two distinct approaches to incorporate silhouette information in the shape and pose estimation pipeline. The **first** approach is to append the silhouette representation as a latent feature to *implicitly* reduce uncertainty for 3D shape estimation and simultaneously improve pose estimation. The **second** approach is to obtain the silhouette from the estimated 3D object's shape and pose and *explicitly* enforce its conformity with the ground-truth silhouette projection in the scene and make it differentiable. Below, we sequentially explain both these approaches.

Notation. Our goal is to reconstruct 3D object instances

Figure 3: Cluttered scene image with color-coded amodal silhouette. Color is only for instance illustration.

from cluttered scene images. We represent 3D object shape as a volume V in a canonical coordinate system, and transformation pose for the scene as parameters of rotation (as a quaternion) q, translation t, and scale s, following the notation of [23].

3.1. Implicit Silhouette

This approach implicitly incorporates the instance-based silhouette information for 3D reconstruction. Firstly, we illustrate the process of amodal[1] silhouette generation in Sec. 3.1.1. Secondly, we explain the procedure to incorporate instance-centred silhouette in the network. Then, we illustrate the whole network architecture and our training strategy.

3.1.1 Instance-level Amodal Silhouette Generation

The common way to generate an exact instance-level silhouette from a multi-object scene is via manual annotations from humans. However, this approach is not scalable and highly expensive. Also, for a cluttered scene image, human annotation cannot deal with the problem of amodal silhouette generation. Human annotators cannot annotate silhouette accurately due to object occlusion. We propose a simple approach to automatically generate a large number of amodal silhouette images for a variety of indoor objects. Here, we take advantage of a large-scale synthetic dataset of indoor scenes, SUNCG, with abundant 3D annotations. For each cluttered scene image, this dataset has instance-level CAD models and relative poses, thanks to the Planner5D platform which is an online interior design interface [1]. Then, we obtain an amodal silhouette image for each object instance using the renderer `Blender`. Specifically, for each object instance in a training image, we have its corresponding CAD model and relative pose, that is used to obtain the amodal instance-specific silhouette image by rendering the instance separately. We use the same camera intrinsic and extrinsic parameters as [30]. Examples of rendered instance-level amodal silhouettes are shown with color-coded masks in Fig. 3.

[1] Amodal is a perception psychology term, defines the perception of the whole of a physical structure when only parts of it are visible.

3.1.2 Instance-centered Silhouette

A 3D object shape cannot be sculptured precisely based only on the high-level features of deep CNNs (e.g., Region-of-Interest (ROI) pooling features in [23]). We emphasize that such features encapsulate more global details about a shape and lack fine-grained details that specify the local structure of a 3D shape. This can be evidenced by visualizing the volume grid where each element shows the probability of occupancy, and the more blurry a surface area is, the more uncertain the shape. Our analysis of the output from [23] shows surface areas of predicted 3D object shape from ROI pooling features are blurry, as shown in Fig. 2.

Alternatively, we propose to use silhouette constraints to enhance the local shape details necessary for accurate instance reconstruction. Specifically, we propose a silhouette feature estimation network to calculate the 2D object silhouette and incorporate it in an end-to-end trainable network. As a rendered instance-level silhouette image from the cluttered scene is not instance-centered, we propose to add a specialized silhouette branch after the ROI pooling layer. Hence, the amodal silhouette is cropped by the instance bounding box to align it with an object instance appropriately.

3.1.3 Network Architecture

The complete network architecture is illustrated in Fig. 4. The whole model comprises of five main parts: (a) Global and local feature extraction, (b) Instance-centered feature extraction, (c) Implicit silhouette estimation and encoding, (d) Bounding box encoding, (e) Shape and pose prediction. Overall, our architecture concatenates features of global, ROI, silhouette, and bounding boxes to form a latent feature space. This latent space is then used to predict the 3D shape

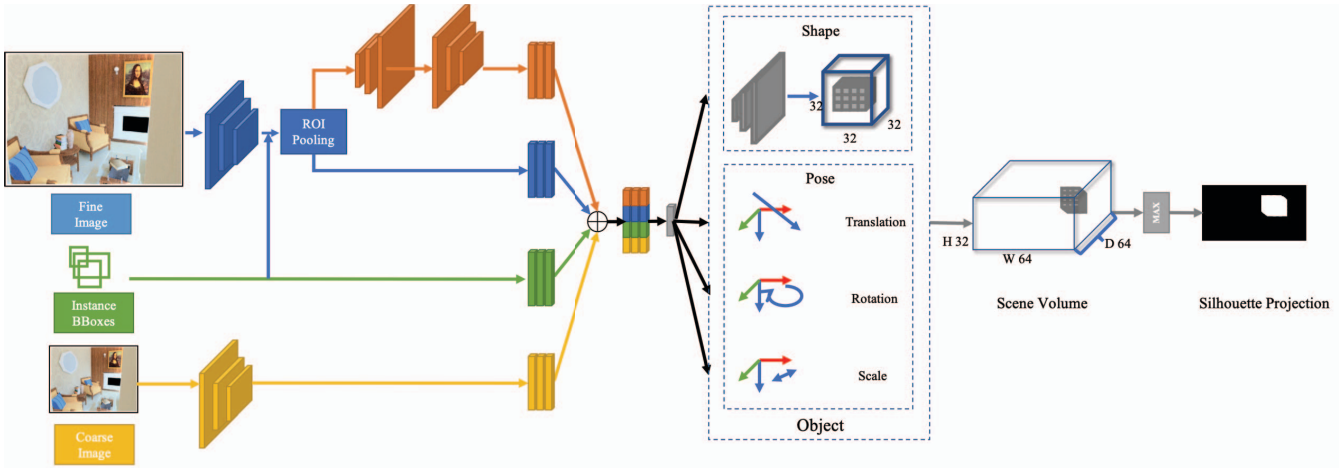


Figure 4: The whole network architecture of our work. Input is a single cluttered scene RGB image. Orange branch for instance-centered implicit silhouette estimation and encoding. Blue branch is for Instance-centered feature extraction. Green branch is for bounding boxes encoder. Yellow branch for coarse 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. Gray branch for explicit silhouette projection generation. *Best seen in color.*

and pose of each instance. Below, we outline the components of our proposed approach that are key to our contributions. Other components follow our baseline network [23] and please refer their paper for details.

Silhouette estimation and encoding. Based on the ROI pooled instance-centered features, we first estimate a 2D object structure using a silhouette estimator network and then compactly represent it using an encoder stage. The silhouette 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 an instance-centered silhouette sil. In the next stage, the silhouette encoder projects the estimated silhouette to a latent silhouette feature. Specifically, the encoder consists of two convolution and two 300-unit fully-connected layers.

3.1.4 Two Stage Training

We train our proposed network in two stages. The first stage is for training the silhouette estimator with instance-centered silhouette loss. The second is for training four feature branches, as shown in Fig. 4 with instance shape and pose losses, silhouette estimation and encoder trained together with the other parts. For 3D instance shape and pose output, we follow the object shape normalization and relative pose configuration in [23]. The objective functions used for the training are also elaborated below.

Silhouette Estimation Loss: The silhouette image is binary, so we compare the performance using binary cross entropy loss (BCE, Eq. 1) and mean square loss (MSE, Eq. 2) between the predicted instance-centered silhouette and the ground-truth. These two losses are given by:

$$L_{sil\_bce} = \frac{1}{N}\sum_i (s_i \log(\hat{s_i}) + (1 - s_i)\log(1 - \hat{s_i})) \quad (1)$$

$$L_{sil\_mse} = \frac{1}{N}\sum_i (s_i - \hat{s_i})^2 \quad (2)$$

where s is the ground-truth instance-centered silhouette, and $\hat{s}$ is the estimated silhouette, N is the number of pixels in s.

3D shape Loss: 3D shape is the voxel representation $V = \{v_i\}$, where $v_i \in \{0, 1\}$. $\hat{v_i}$ is the predicted voxel occupancy probability for the voxel at location. We use voxel-level Cross Entropy loss Eq. 3 to learn this representation.

$$L_V = \frac{1}{N}\sum_i (v_i \log \hat{v_i} + (1 - v_i)\log(1 - \hat{v_i})). \quad (3)$$

3D Pose Loss: The pose loss comprises of three terms that are described below:

- *Rotation.* Our objective function for rotation is the negative log-likelihood (Eq. 4) computed using the predicted probability of the ground-truth class $\hat{q}^g$. $\hat{q}$ is the predicted probability over all 24-bin classes,

$$L_q = -\log(\hat{q}^g). \quad (4)$$

- *Scale.* We use squared Euclidean distance, Eq. 5, between predicted scale values $\hat{s}$ and ground-truth s. This distance is calculated in logarithmic space to reduce the influence of magnitude.

$$L_s = \|\log(s) - \log(\hat{s})\|_2^2 \quad (5)$$

- *Translation.* The translation loss is represented in terms of Euclidean loss (Eq. 6) between prediction $\hat{t}$ and ground-truth t.

$$L_t = \left\| t - \hat{t} \right\|_2^2. \tag{6}$$

Two Stage Training: We train the silhouette branch with silhouette estimation loss first, and then train the silhouette estimator and encoder with all other modules using weighted 3D shape and pose losses together.

$$L = \sum_{b \in \mathcal{B}+} \Big(w_V L_V + w_q L_q + w_s L_s + w_t L_t - \ln(f) \Big) + \sum_{b \in \mathcal{B}-} \ln(1-f). \tag{7}$$

where w_V, w_q, w_s, w_t are the weights for corresponding loss functions, $\mathcal{B}+, \mathcal{B}-$ denote the set of positive and negative object bounding boxes respectively. Details are provided in experimental section.

3.2. Explicit Silhouette Projection

As an alternative to implicit silhouette estimation, we propose explicit silhouette projection to improve 3D object shape and pose estimation. The main idea is that the perspective projection of predicted 3D object should be as similar to the ground-truth projection as possible. With this insight, we propose an explicit perspective silhouette projection loss for our task. To this end, for each object instance, we obtain its perspective silhouette projection directly from it's 3D shape and pose parameters and make it differentiable. We use the loss given in Eq. 10 and 11 to minimize the reprojection error and regularize shape and pose parameter estimations. We explain this process below.

Instance-level Perspective Silhouette Projection Generation: To generate instance perspective silhouette projection p, we use the 3D object instance shape V and pose parameters q, s, t. Specifically, since the predicted 3D object pose denotes the transformation parameters for the object from canonical coordinates to camera coordinates, we can obtain the single-object-only 3D scene volume in camera coordinates through 3D transformation of object from canonical coordinate with the shape and pose parameters. Specifically, we use a predefined scene volume V^s with dimensions height, width and depth as $32 * 64 * 64$. p^s denotes 3D point coordinate for one occupied voxel V^s_{ijk} in V^s. We can obtain single object coordinates $\{p^s_i\}$ in the scene from 3D transformation of points $\{p^c_i\}$ in the canonical coordinate by Eq. 8.

$$p^s = \begin{bmatrix} R^q * diag(s) & t \\ \mathbf{0}^T & 1 \end{bmatrix} p^c, \tag{8}$$

where $\mathbf{0}$ is a three-zero-elements vector, R^q is the rotation matrix from quaternion q, and $diag(s)$ is the diagonal matrix with elements s.

Subsequently, we propose to take advantage of max pooling to calculate the perspective silhouette projection and make this process differentiable for shape and pose parameters regularization. This is through max pooling along the depth dimension of scene volume V^s in Eq. 9. This procedure is illustrated in the gray branch of Fig. 4.

$$s = \max_d V^s_{hwd}. \tag{9}$$

Projection Loss: Similar to the implicit silhouette section, here we use binary cross entropy loss Eq. 10 and mean square error loss Eq. 11,

$$L_{proj\_bce} = \frac{1}{N} \sum_i s_i \log \hat{s_i} + (1 - s_i) \log(1 - \hat{s_i}), \tag{10}$$

$$L_{proj\_mse} = \frac{1}{N} \sum_i (s_i - \hat{s_i})^2, \tag{11}$$

where $\hat{s}$ is the silhouette generated from the predicted shape and pose $\widehat{V^c}$, $\widehat{q}$, $\widehat{t}$, $\widehat{s}$.

Training: We train each projection loss Eq. 10 and 11 with shape loss Eq. 3 and pose losses Eq. 4, 5, and 6 in a single stage manner.

$$L = \sum_{b \in \mathcal{B}+} \Big(w_V L_V + w_q L_q + w_s L_s + w_t L_t + w_{proj} L_{proj} - \ln(f) \Big) + \sum_{b \in \mathcal{B}-} \ln(1-f). \tag{12}$$

Hyper-parameter details are provided in experimental section.

4. Experiments

Here we comprehensively evaluate our proposed implicit and explicit silhouette techniques and compare it with the state-of-the-art method Factored3d [23] on both synthetic and real data. Our datasets for 3D object reconstruction reflect real-world conditions such as cluttered scenes, occlusions and small objects. We evaluate our approach using some different metrics (following [23]) since 3D object reconstruction from a cluttered single scene image is a compounded problem that involves both 3D object shape and pose estimation. Our qualitative and quantitative results show that overall our proposed method outperforms the recent Factored3d [23]. We provide an in-depth ablation study in Sec. 4.3 to justify various design choices considered in our proposed approach.

4.1. Dataset

SUNCG [19]: For our experiments, we use SUNCG, a large-scale synthetic dataset. This dataset contains cluttered indoor scenes and large pose and shape variations that make 3D object shape and pose estimation difficult from

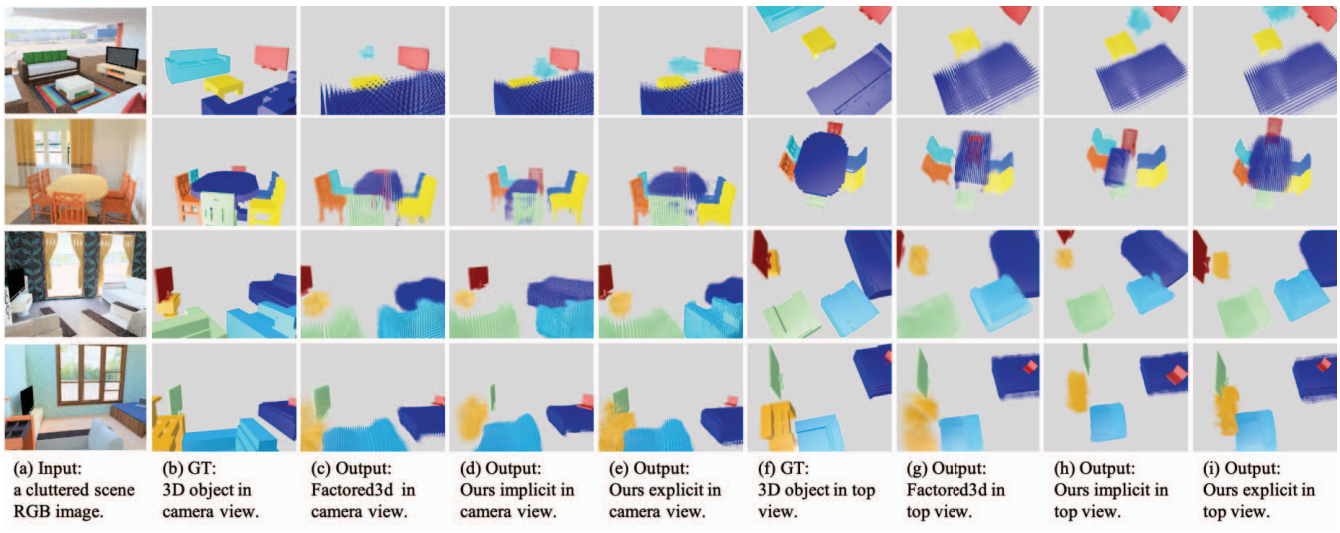


Figure 5: Visualization of 3D reconstruction with ground-truth bounding boxes on the SUNCG test dataset. Each row is one comparison between our work and [23]. Our predicted shape has lower uncertainty around object surface areas. Instance color is only to distinguish between object instances.

single RGB images. The RGB images belonging to cluttered scenes are rendered by a photo-realistic renderer proposed in [30]. For each object in a cluttered scene, we can obtain voxel-based 3D object shape in a canonical coordinate system and pose parameters relative to the camera coordinate system. The camera calibration parameters are fixed as [30] for simplicity. In the SUNCG dataset, there are 45,622 scenes with over 5M instances of 2,644 unique objects belonging to 84 object categories. Here, we select 6 categories of common 3D indoor scene objects (television, desk, sofa, table, bed, chair) similar to [23] to make a fair comparison. We use the split setting of [23], which randomly divides the dataset into training, validation and test sets with a ratio of 70%:10%:20%, respectively.

NYU depth v2 [14]: To clarify our works generality to real world scenarios, we analyze our network's inference ability on the NYU depth v2 dataset. This dataset contains 1449 real world indoor scene images. Thanks to Guo [4], we can use their 3D surface mesh annotation to obtain the ground truth of 3D object instances. Basically, we show qualitative and quantitative results on the NYU depth v2 dataset, based on the models trained only on the SUNCG training set.

Evaluation Metrics Tulsiani *et al*. [23] propose several quantitative evaluation metrics for the task of 3D object instance reconstruction from a single image. They aim to make a comprehensive study of 2D object detection, and 3D object shape and pose estimation. However, they use some loose thresholds $(\delta_V, \delta_q, \delta_t, \delta_s)$. (Please refer to [23] for details.) So we incorporate more strict thresholds alongside their original ones to show our generalizability.

Implementation Details Hyper-parameter For hyper-parameters of shape and pose loss weights w_V, w_q, w_s, w_t, we follow [23], namely (10, 1, 1, 1). For the perspective projection loss weight, we choose $w_p = 1$. **Training** Basically, we use ground trugh object bounding boxes and object proposals [34] to train the 2D object detection part as [23]. For the **implicit silhouette**, we train silhouette estimation based on ground truth bounding boxes with 1 epoch, then fine tune the model on object proposals with 1 epoch. Then we train the silhouette encoder together with object shape and pose with ground truth bounding boxes with 1 epoch and object proposals with 4 epochs. For the **explicit silhouette**, to make fair comparison with implicit silhouette encoding, we train the combined losses in the same manner, namely, 1 epoch with ground truth bounding boxes and 4 epochs with object proposals.

4.2. Comparisons with state-of-the-art

4.2.1 Reconstruction with Ground-truth 2D Box

To eliminate the effect of mislocalization of objects, we first feed objects cropped using the 2D ground-truth bounding boxes for 3D object reconstruction.

Qualitative results: We show the 3D object instance shape and pose estimation results in two views to give a better illustration of 3D information (Fig. 5). To illustrate the occupancy for each voxel, we visualize each voxel using a cubic mesh, whose size reflects the probability value. High sparsity of a volume grid means the shape estimation is more uncertain. We illustrate 3D object pose by showing the 2D projection of the 3D instance. The 2D position of the object can show the soundness of its 3D pose estima-

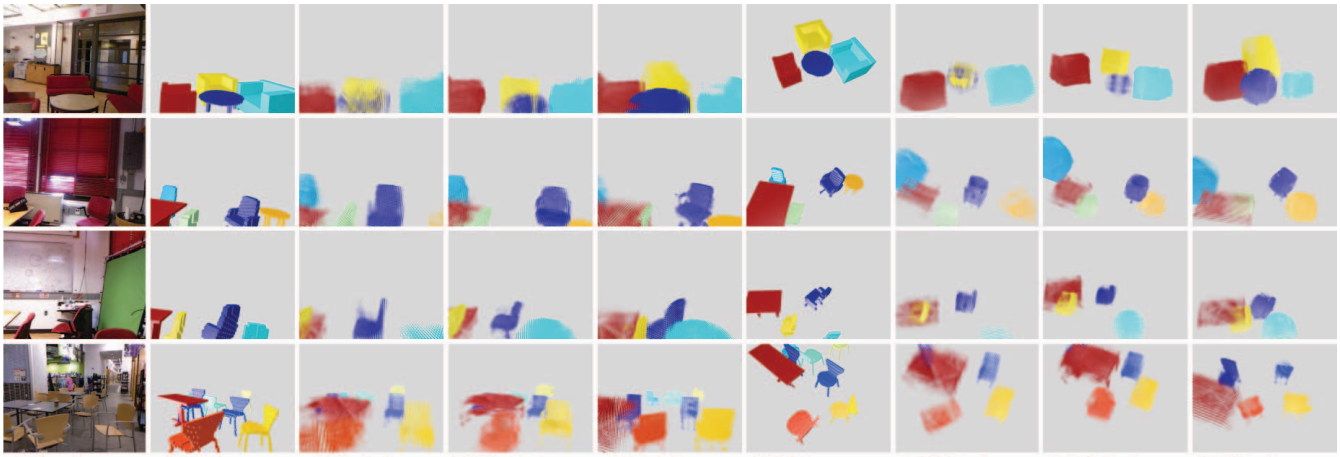


Figure 6: Visualization of 3D instance reconstruction from a real scene from NYU depth v2 validation dataset. Each row is a comparison between ours implicit and explicit methods with [23]. We can see that our results are more compact and accurate.

Dataset	Method	Shape			
		Median IoU ↑	Mean IoU ↑	$\%(\delta_V = 0.25) \uparrow$	$\%(\delta_V = 0.5) \uparrow$
SUNCG	Factored3d[23]	0.47	0.49	74.98	47.15
	Ours Implicit	0.55	0.55	78.54	55.21
	Ours Explicit	**0.60**	**0.58**	**80.52**	**58.97**
NYUv2	Factored3d[23]	0.10	0.14	15.53	2.40
	Ours Implicit	**0.11**	**0.16**	**20.18**	4.24
	Ours Explicit	0.09	0.15	18.17	**5.20**

Table 1: 3D shape estimation: median and mean IoU, precision % with thresholds $(\delta_V = 0.25, 0.5)$. Arrow direction means better performance.

tion. As shown in Fig. 5, our proposed methods obtain a more compact object shape than [23], this visually demonstrates our method decreases the uncertainty of object shape estimation. Besides, explicit method works better than implicit one. For 3D object instance pose, the relative pose difference in the 2D image shows one view of the 3D pose estimation difference. We can see from Fig. 5, our pose estimation of object instances is better, e.g., some chair instance estimations in row 3.

To compare our method with others in the real image scenario, we show some qualitative results based on the NYU depth v2 dataset [14] in Fig. 5 from our model and Factored3d [23], trained on the SUNCG synthetic dataset. There are more complicated lighting conditions and disordered object instances. Hence, reconstruction from a real scene image is harder than reconstruction from a synthetic scene image. We can this see in Fig. 6, in comparison with the qualitative results from Fig. 5, object shape estimation uncertainty is higher, and misdetection is higher too. However, we can still draw a similar conclusion as in Sec. 4.2.1: our proposed method has improved 3D instance reconstruction from a cluttered real single scene image.

Dataset	Method	Rotation				
		Median Err ↓	Mean Err ↓	$\%(\delta_q = 30) \uparrow$	$\%(\delta_q = 10) \uparrow$	$\%(\delta_q = 5) \uparrow$
SUNCG	Factored3d [23]	5.02	31.80	77.90	70.77	49.87
	Ours Implicit	5.05	31.66	78.22	71.12	49.65
	Ours Explicit	**4.79**	**28.89**	**80.22**	**73.31**	**51.70**
NYUv2	Factored3d[23]	15.61	47.34	62.37	36.03	16.41
	Ours Implicit	16.33	49.23	59.57	34.99	17.69
	Ours Explicit	**14.40**	**44.35**	**64.93**	**39.79**	**19.62**
Dataset	Method	Translation				
		Median Err ↓	Mean Err ↓	$\%(\delta_t = 1.) \uparrow$	$\%(\delta_t = 0.5) \uparrow$	$\%(\delta_t = 0.1) \uparrow$
SUNCG	Factored3d[23]	0.30	0.58	91.09	74.46	6.80
	Ours Implicit	**0.28**	**0.54**	**91.89**	**76.87**	**8.28**
	Ours Explicit	0.31	0.55	91.29	73.85	6.77
NYUv2	Factored3d [23]	0.73	1.03	65.81	31.55	0.64
	Ours Implicit	**0.72**	**0.99**	67.57	31.55	0.96
	Ours Explicit	0.73	1.18	**91.29**	**73.85**	**6.77**
Dataset	Method	Scale				
		Median Err ↓	Mean Err ↓	$\%(\delta_s = 0.5) \uparrow$	$\%(\delta_s = 0.3) \uparrow$	$\%(\delta_s = 0.2) \uparrow$
SUNCG	Factored3d [23]	0.12	0.23	87.67	75.79	64.43
	Ours Implicit	**0.11**	0.22	88.40	77.05	66.28
	Ours Explicit	0.12	**0.21**	**89.05**	**78.25**	**67.03**
NYUv2	Factored3d [23]	0.89	0.92	11.05	2.72	0.88
	Ours Implicit	**0.74**	**0.78**	**19.30**	**5.60**	**2.16**
	Ours Explicit	0.87	0.91	11.77	3.04	1.36

Table 2: 3D pose estimation: median and mean IoU, precision % with thresholds $(\delta_V = 0.25, 0.5)$. Arrow direction means better performance.

Quantitative Results: We evaluate 3D shape and pose estimation based on the evaluation metrics given in [23].

(a) Shape evaluation: We evaluate shape estimation on the metrics of the median and mean IoU, and IoU percentage precision % based on two thresholds $\%(\delta_V = 0.25, 0.5)$. While [23] only show results based on $\delta_v = 0.25$, we add a more strict IoU threshold $\delta_v = 0.5$. From Table 1, we can see that the joint modeling of object and silhouette has made a big improvement, especially for the more strict threshold setting. These results demonstrate that both 2D silhouette techniques helps reduce the 3D shape estimation uncertainty.

(b) Pose evaluation: We evaluate rotation, translation, and scale estimation and show results in Table 2. This table

Method	Factored3d [23]	Ours Implicit	Ours Explicit
$(\delta_V, \delta_q, \delta_t, \delta_s, \delta_d)$	(0.25, 30, 1, 0.5, 0.5)		
all	39.01	41.48	**43.00**
-shape	44.57	46.	**47.65**
-rot	45.74	49.75	**51.02**
-trans	40.42	42.80	**44.44**
-box2d	41.66	43.92	**45.90**
-scale	42.00	44.12	**45.45**
box2d	68.01	69.43	**69.78**
box2d+rot	52.51	53.33	**54.61**
box2d+trans	62.92	65.08	**65.12**
box2d+shape	52.64	56.06	**57.24**
box2d+scale	58.35	60.66	**61.60**
box2d+rot+shape	44.24	46.19	**47.63**

Table 3: mean Average Precision (mAP) for 2D detection and 3D reconstruction with three threshold settings on the SUNCG test dataset.

shows that our proposed method has similar performance to [23] for rotation estimation. We believe this is because the 2D silhouette does not provide as much information for rotation estimation. Intuitively, better results can be obtained by using 3D motion constraints that we will explore in future work. Our work outperforms [23] both in terms of error and precision measures for translation estimation. In the end, our method has better performance especially with more strict threshold δ_s setting for scale estimation.

4.2.2 Reconstruction with 2D Detection

Now we evaluate 3D reconstruction with 2D detection to show the results from this combination.

Quantitative Results: We follow the evaluation metrics proposed in Tulsiani *et al*. [23] for this setting. As shown in Table 3, our method outperforms [23] in every criterion. Our proposed method has improved 3D instance reconstruction from a cluttered single scene image for the combined task of 2D detection and 3D instance shape and pose estimation quantitatively.

4.3. Ablation study

In addition to the previous comparison with the baseline method, we performed an analysis of the impact of silhouette loss choices for both implicit and explicit settings. In Table 4, for implicit silhouette, binary cross entropy (BCE) loss outperforms mean square error (MSE) loss a little for almost all evaluation metrics. From this, we can draw a conclusion that it is better to treat implicit silhouette estimation as a binary regression problem instead of binary classification problem.

However in Table 5, for explicit silhouette, mean square error (MSE) loss obtains better or at least comparable performance compared to binary cross entropy (BCE). So unlike the above finding in implicit setting, we can draw the

Method	Shape			
	Median IoU ↑	Mean IoU ↑	$\%(\delta_V = 0.25)$ ↑	$\%(\delta_V = 0.5)$ ↑
MSE	0.54	0.54	78.36	54.04
BCE	**0.55**	**0.55**	**78.54**	**55.21**

Method	Rotation				
	Median Err ↓	Mean Err ↓	$\%(\delta_q = 30)$ ↑	$\%(\delta_q = 10)$ ↑	$\%(\delta_q = 5)$ ↑
MSE	5.11	32.11	77.78	70.46	49.22
BCE	**5.05**	**31.66**	**78.22**	**71.12**	**49.65**
Method	Translation				
	Median Err ↓	Mean Err ↓	$\%(\delta_t = 1.)$ ↑	$\%(\delta_t = 0.5)$ ↑	$\%(\delta_t = 0.1)$ ↑
MSE	0.31	0.56	91.72	74.45	6.63
BCE	**0.28**	**0.54**	**91.89**	**76.87**	**8.28**
Method	Scale				
	Median Err ↓	Mean Err ↓	$\%(\delta_s = 0.5)$ ↑	$\%(\delta_s = 0.3)$ ↑	$\%(\delta_s = 0.2)$ ↑
MSE	0.12	0.22	88.23	76.53	65.39
BCE	**0.11**	**0.22**	**88.40**	**77.05**	**66.28**

Table 4: Ablation study for implicit silhouette losses Eq. 1 and 2 on SUNCG test dataset.

Method	Shape			
	Median IoU ↑	Mean IoU ↑	$\%(\delta_V = 0.25)$ ↑	$\%(\delta_V = 0.5)$ ↑
MSE	**0.5986**	**0.5832**	**80.52**	**58.97**
BCE	0.5977	0.5765	80.02	58.70

Method	Rotation				
	Median Err ↓	Mean Err ↓	$\%(\delta_q = 30)$ ↑	$\%(\delta_q = 10)$ ↑	$\%(\delta_q = 5)$ ↑
MSE	4.79	**28.89**	**80.22**	**73.31**	51.70
BCE	**4.75**	29.34	79.98	73.08	**51.94**
Method	Translation				
	Median Err ↓	Mean Err ↓	$\%(\delta_t = 1.)$ ↑	$\%(\delta_t = 0.5)$ ↑	$\%(\delta_t = 0.1)$ ↑
MSE	0.3096	0.5547	91.29	**73.85**	**6.77**
BCE	**0.3095**	**0.5500**	**91.52**	73.67	6.57
Method	Scale				
	Median Err ↓	Mean Err ↓	$\%(\delta_s = 0.5)$ ↑	$\%(\delta_s = 0.3)$ ↑	$\%(\delta_s = 0.2)$ ↑
MSE	4.79	**28.89**	**80.22**	**73.31**	51.70
BCE	**4.75**	29.34	79.98	73.08	**51.94**

Table 5: Ablation study for explicit silhouette losses Eq. 10 and 11 on SUNCG test dataset.

conclusion that as a perspective projection silhouette assistant for 3D object instance reconstruction, a regression solution works better than a binary classification solution.

In summary, we choose the results from binary cross entropy loss for implicit silhouette, and mse loss for explicit silhouette and report qualitative and quantitative results in Table. 1, 2, 3 and Fig. 5, 6.

5. Conclusion

We present two methods to explore the importance of silhouettes for 3D instance reconstruction. In the first approach, we include a 2D implicit silhouette feature and combine it with other object features to make a compact 3D object reconstruction. In the second approach, we propose an efficient and differentiable way through explicit perspective silhouette projection to regularize object shape and pose. Qualitative and Quantitative results show that both our methods have improvement with a considerable margin and the explicit method works better. Automatic rendering and de-rendering in the network is an excellent direction to improve the performance further. Also, a further independent design to solve 3D reconstruction without 2D object detection is another promising direction to investigate.

References

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

[2] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[3] R. Garg, V. K. BG, G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *ECCV*, 2016.

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

[5] S. Gupta, P. Arbeláez, R. Girshick, and J. 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] S. Huang, S. Qi, Y. Xiao, Y. Zhu, Y. N. Wu, and S.-C. Zhu. Cooperative holistic scene understanding: Unifying 3d object, layout, and camera pose estimation. In *Advances in Neural Information Processing Systems*, 2018.

[7] S. H. Khan, M. Bennamoun, F. Sohel, and R. Togneri. Geometry driven semantic labeling of indoor scenes. In *ECCV*, 2014.

[8] S. H. Khan, X. He, M. Bennamoun, F. Sohel, and R. Togneri. Separating objects and clutter in indoor scenes. In *CVPR*, 2015.

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

[10] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International journal of computer vision*, 2000.

[11] M. Liu, M. Salzmann, and X. He. Discrete-continuous depth estimation from a single image. In *CVPR*, 2014.

[12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018.

[13] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *CVPR*, 2018.

[14] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[15] E. Smith, S. Fujimoto, and D. Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *NIPS*, 2018.

[16] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

[17] S. Song and J. Xiao. Sliding shapes for 3d object detection in depth images. In *ECCV*, 2014.

[18] S. Song and J. Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *CVPR*, 2016.

[19] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

[20] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *CVPR*, 2018.

[21] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017.

[22] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, 2018.

[23] S. Tulsiani, S. Gupta, D. Fouhey, A. A. Efros, and J. Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[24] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.

[25] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. 2019.

[26] O. Wiles and A. Zisserman. Learning to predict 3d surfaces of sculptures from single and multiple views. *International Journal of Computer Vision*, 2018.

[27] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *NIPS*, 2017.

[28] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in neural information processing systems*, 2016.

[29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[30] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *CVPR*, 2017.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.

[32] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017.

[33] W. Zhuo, M. Salzmann, X. He, M. Liu, et al. 3d box proposals from a single monocular image of an indoor scene. AAAI, 2018.

[34] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.