Deep Single-View 3D Object Reconstruction with Visual Hull Embedding

Hanqing Wang,[*1,2] **Jiaolong Yang,**[2] **Wei Liang,**[1] **Xin Tong**[2]
[1]Beijing Institute of Technology, [2]Microsoft Research Asia
{hanqingwang,liangwei}@bit.edu.cn, {jiaoyan,xtong}@microsoft.com

Abstract

3D object reconstruction is a fundamental task of many robotics and AI problems. With the aid of deep convolutional neural networks (CNNs), 3D object reconstruction has witnessed a significant progress in recent years. However, possibly due to the prohibitively high dimension of the 3D object space, the results from deep CNNs are often prone to missing some shape details. In this paper, we present an approach which aims to preserve more shape details and improve the reconstruction quality. The key idea of our method is to leverage object mask and pose estimation from CNNs to assist the 3D shape learning by constructing a probabilistic single-view visual hull inside of the network. Our method works by first predicting a coarse shape as well as the object pose and silhouette using CNNs, followed by a novel 3D refinement CNN which refines the coarse shapes using the constructed probabilistic visual hulls. Experiment on both synthetic data and real images show that embedding a single-view visual hull for shape refinement can significantly improve the reconstruction quality by recovering more shapes details and improving shape consistency with the input image.

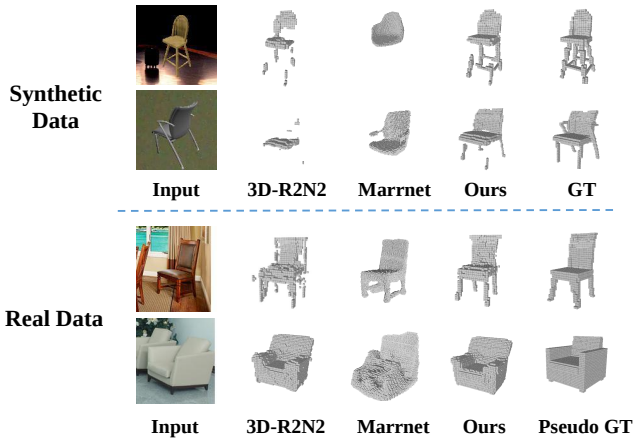

Figure 1: Some results reconstructed from synthetic data and real data by the baseline approach (Choy et al. 2016), MarrNet (Wu et al. 2017) and our approach on the chair category. Note the inconsistency with input images and missing parts in the results of the former two methods.

Introduction

Recovering the dense 3D shapes of objects from 2D imageries is a fundamental AI problem which has many applications such as robot-environment interaction, 3D-based object retrieval, recognition and functionality estimate (Wang, Liang, and Yu 2017; Zhu, Zhao, and Zhu 2015; Liang et al. 2016), *etc*. Given a single image of an object, a human can reason the 3D structure of the object reliably. However, single-view 3D object reconstruction is very challenging for computer algorithms.

Recently, a significant progress of single-view 3D reconstruction has been achieved by using deep convolutional neural networks (CNNs) (Choy et al. 2016; Girdhar et al. 2016; Wu et al. 2016; Yan et al. 2016; Fan, Su, and Guibas 2017; Tulsiani et al. 2017b; Zhu et al. 2017; Wu et al. 2017; Tulsiani, Efros, and Malik 2018). Most CNN-based methods reconstruct the object shapes using 2D and 3D convolutions in a 2D encoder-3D decoder structure with the volumetric 3D representation. The input to these CNNs are object images taken under unknown viewpoints, while the output shapes are often aligned with the canonical viewpoint in a single, pre-defined 3D coordinate system such that shape regression is more tractable.

Although promising results have been shown by these CNN-based methods, single-view 3D reconstruction is still a challenging problem and the results are far from being perfect. One of the main difficulties lies in the object shape variations which can be very large even in a same object category. The appearance variations in the input images caused by pose differences make this task even harder. Consequently, the results from CNN-based methods are prone to missing some shape details and sometimes generate plausible shapes which, however, are inconsistent with input images, as shown in Figure 1.

In this paper, we propose an approach to improve the fidelity of the reconstructed shapes by CNNs. Our method combined traditional wisdom into the network architecture. It is motivated by two observations: 1) while directly recovering all the shape details in 3D is difficult, extracting the projected shape silhouette on the 2D plane, *i.e*. segmenting

[*]Part of this work was done when HW was an intern at MSR.

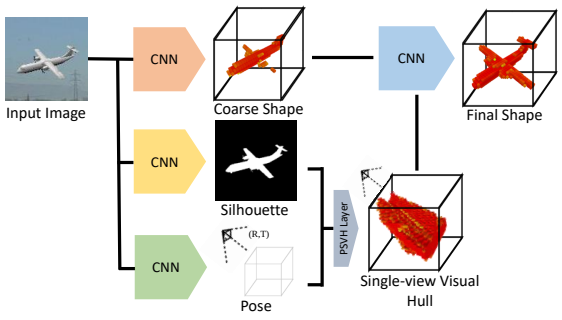

Figure 2: An overview of the proposed method. Given an input image, we first use CNNs to predict a coarse 3D volumetric shape, the silhouette and the object pose. The latter two are used to construct a single-view visual hull, which is used to refine the coarse shape using another CNN.

out the object from background in a relatively easy task using CNNs; 2) for some common objects such as chairs and cars whose 3D coordinate systems are well defined without ambiguity, the object pose (or equivalently, the viewpoint) can also be well estimated by a CNN (Su et al. 2015; Massa, Marlet, and Aubry 2016). As such, we propose to leverage the object silhouettes to assist the 3D learning by lifting them to 3D using pose estimates.

Figure 2 is a schematic description of our method, which is a pure GPU-friendly neural network solution. Specifically, we embed into the network a single-view visual hull using the estimated object silhouettes and poses. Embedding a visual hull can help recover more shape details by considering the projection relationship between the reconstructed 3D shape and the 2D silhouette. Since both the pose and segmentation are subject to estimation error, we opted for a "soft" visual-hull embedding strategy: we first predict a coarse 3D shape using a CNN, then employ another CNN to refine the coarse shape with the constructed visual hull. We propose a probabilistic single-view visual hull (PSVH) construction layer which is differentiable such that the whole network can be trained end-to-end.

In summary, we present a novel CNN-based approach which uses a single-view visual hull to improve the quality of shape predictions. Through our method, the perspective geometry is seamlessly embedded into a deep network. We evaluate our method on synthetic data and real images, and demonstrate that using a single-view visual hull can significantly improve the reconstruction quality by recovering more shape details and improving shape consistency with input images.

Related Work

Traditional methods. Reconstructing a dense 3D object shape from a single image is an ill-posed problem. Traditional methods resort to geometry priors for the otherwise prohibitively challenging task. For example, some methods leveraged pre-defined CAD models (Sun et al. 2013). Category-specific reconstruction methods (Vicente et al. 2014; Kar et al. 2015; Tulsiani et al. 2017a) reconstruct a 3D shape template from images of the objects in the same category as shape prior. Given an input image, these methods estimate silhouette and viewpoint from the input image and then reconstruct 3D object shape by fitting the shape template to the estimated visual hull. Our method integrates the single-view visual hull with deep neural network for reconstructing 3D shape from single image.

Deep learning for 3D reconstruction. Deep learning based methods directly learn the mapping from 2D image to a dense 3D shape from training data. For example, (Choy et al. 2016) directly trained a network with 3D shape loss. (Yan et al. 2016) trained a network by minimizing the difference between the silhouette of the predicted 3D shape and ground truth silhouette on multiple views. A ray consistency loss is proposed in (Tulsiani et al. 2017b) which uses other types of multi-view observations for training such as depth, color and semantics. (Wu et al. 2017) applied CNNs to first predict the 2.5D sketches including normal, depth and silhouette, then reconstruct the 3D shape. A reprojection consistency constraint between the 3D shape and 2.5D sketches is used to finetune the network on real images. (Zhu et al. 2017) jointly trained a pose regressor with a 3D reconstruction network so that the object images with annotated masks yet unknown poses can be used for training. Many existing methods have explored using pose and silhouette (or other 2D/2.5D observations) to supervise the 3D shape prediction (Yan et al. 2016; Tulsiani et al. 2017b; Gwak et al. 2017; Zhu et al. 2017; Wu et al. 2017; Tulsiani, Efros, and Malik 2018). However, our goal is to refine the 3D shape inside of the network using an estimated visual hull, and our visual hull construction is an inverse process of their shape-to-image projection scheme.

Generative models for 3D shape. Some efforts are devoted to modeling the 3D shape space using generative models such as GAN (Goodfellow et al. 2014) and VAE (Kingma and Welling 2013). In (Wu et al. 2016), a 3D-GAN method is proposed for learning the latent space of 3D shapes and a 3D-VAE-GAN is also presented for mapping image space to shape space. A fully convolutional 3D autoencoder for learning shape representation from noisy data is proposed in (Sharma, Grau, and Fritz 2016). A weakly-supervised GAN for 3D reconstruction with the weak supervision from silhouettes can be found in (Gwak et al. 2017).

3D shape representation. Most deep object reconstruction methods use the voxel grid representation (Choy et al. 2016; Girdhar et al. 2016; Yan et al. 2016; Tulsiani et al. 2017b; Zhu et al. 2017; Wu et al. 2017; 2016; Gwak et al. 2017), *i.e.*, the output is a voxelized occupancy map. Recently, memory-efficient representations such as point clouds (Qi et al. 2017), voxel octree (Häne, Tulsiani, and Malik 2017; Tatarchenko, Dosovitskiy, and Brox 2017) and shape primitives (Zou et al. 2017) are investigated. (Richter and Roth 2018) proposes a memory-efficient shape representation for high-resolution reconstruction. Specifically, their core idea is orthogonal to ours.

Visual hull for deep multi-view 3D reconstruction. Some recent works use visual hulls of color (Ji et al. 2017) or learned feature (Kar, Häne, and Malik 2017) for multi-view stereo with CNNs. Our method is different from theirs in

several ways. First, the motivations of using visual hulls differ: they use visual hulls as input to their multi-view stereo matching networks in order to reconstruct the object shape, whereas our goal is to leverage a visual hull to refine a coarse single-view shape prediction. Second, the object poses are given in their methods, while in ours the object pose is estimated by a CNN. Related to the above, our novel visual hull construction layer is made differentiable, and object segmentation, pose estimation and 3D reconstruction are jointly trained in one framework.

Our Approach

In this section, we detail our method which takes as input a single image of a common object such as car, chair and coach, and predicts its 3D shape. We assume the objects are roughly centered (*e.g*. those in bounding boxes given by an object detector).

Shape representation. We use voxel grid for shape representation similar to previous works (Wu et al. 2015; Yan et al. 2016; Wu et al. 2016; Zhu et al. 2017; Wu et al. 2017), *i.e*., the output of our network is a voxelized occupancy map in the 3D space. This representation is very suitable for visual hull construction and processing, and it is also possible to extend our method to use tree-structured voxel grids for more fine-grained details (Häne, Tulsiani, and Malik 2017; Tatarchenko, Dosovitskiy, and Brox 2017)

Camera model. We choose the perspective camera model for the 3D-2D projection geometry, and reconstruct the object in a unit-length cube located in front of the camera (*i.e*., with cube center near the positive Z-axis in the camera coordinate frame). Under a perspective camera model, the relationship between a 3D point (X, Y, Z) and its projected pixel location (u, v) on the image is

$$Z[u, v, 1]^T = \mathbf{K}\big(\mathbf{R}[X, Y, Z]^{\mathrm{T}} + \mathbf{t}\big) \tag{1}$$

where $\mathbf{K} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ is the camera intrinsic matrix with f being the focal length and (u_0, v_0) the principle point. We assume that the principal points coincide with image center (or otherwise given), and focal lengths are known. Note that when the exact focal length is not available, a rough estimate or an approximation may still suffice. When the object is reasonably distant from the camera, one can use a large focal length to strike a balance between perspective and weak-perspective models.

Pose parametrization. The object pose is characterized by a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ and a translation vector $\mathbf{t} = [t_X, t_Y, t_Z]^{\mathrm{T}} \in \mathbb{R}^3$ in Eq. 1. We parameterize rotation simply with Euler angles $\theta_i, i = 1, 2, 3$. For translation we estimate t_Z and a 2D vector $[t_u, t_v]$ which centralizes the object on image plane, and obtain $\mathbf{t}$ via $[\frac{t_u}{f} * t_Z, \frac{t_v}{f} * t_Z, t_Z]^{\mathrm{T}}$. In summary, we parameterize the pose as a 6-D vector $\mathbf{p} = [\theta_1, \theta_2, \theta_3, t_u, t_v, t_Z]^{\mathrm{T}}$.

Sub-nets for Pose, Silhouette and Coarse Shape

Given a single image as input, we first apply a CNN to directly regress a 3D volumetric reconstruction similar to previous works such as (Choy et al. 2016). We call this network the *V-Net*. Additionally, we apply another two CNNs for pose estimation and segmentation, referred to as *P-Net* and *S-Net* respectively. In the following we describe the main structure of these sub-networks.

V-Net: The V-Net for voxel occupancy prediction consists of a 2D encoder and a 3D decoder, as depicted in Fig. 3 (a). It is adapted from the network structure of (Choy et al. 2016), and the main difference is we replaced their LSTM layer designed for multi-view reconstruction with a simple fully connected (FC) layer. We denote the 3D voxel occupation probability map produced by the V-Net as $\mathcal{V}$.

P-Net: The P-Net for pose estimation is a simple regressor outputting 6-dimensional pose vector denoted as $\mathbf{p}$, as shown in Fig. 3 (b). We construct the P-Net structure simply by appending two FC layers to the encoder structure of V-Net, one with 512 neurons and the other with 6 neurons. Since the geometric variation among different object categories is huge and the viewpoint-appearance relationships significantly differ, we follow previous works (Su et al. 2015; Massa, Marlet, and Aubry 2016) to use category-specific final FC layers for multiple object categories.

S-Net: The S-Net for object segmentation has a 2D encoder-decoder structure, as shown in Fig. 3 (c). We use the same encoder structure of V-Net for S-Net encoder, and apply a mirrored decoder structure consisting of deconv and uppooling layers. The S-Net generates an object probability map of 2D pixels, which we denote as $\mathcal{S}$.

PSVH Layer for Probabilistic Visual Hull

Given the estimated pose $\mathbf{p}$ and the object probability map $\mathcal{S}$ on the image plane, we construct inside of our neural network a Probabilistic Single-view Visual Hull (PSVH) in the 3D voxel grid. To achieve this, we project each voxel location $\mathbf{X}$ onto the image plane by the perspective transformation in Eq. 1 to obtain its corresponding pixel $\mathbf{x}$. Then we assign $\mathcal{H}(\mathbf{X}) = \mathcal{S}(\mathbf{x})$, where $\mathcal{H}$ denotes the generated probabilistic visual hull. This process is illustrated in Fig. 3 (d).

The PSVH layer is differentiable, which means that the gradients backpropagated to $\mathcal{H}$ can be backpropagated to $\mathcal{S}$ and pose P, hence further to S-Net and P-Net. The gradient of $\mathcal{H}$ with respect to $\mathcal{S}$ is easy to discern: we have built correspondences from $\mathcal{H}$ to $\mathcal{S}$ and simply copied the values. Propagating gradients to $\mathbf{p}$ is somewhat tricky. According to the chain rule, we have $\frac{\partial l}{\partial \mathbf{p}} = \sum_{\mathbf{X}} \frac{\partial l}{\partial \mathcal{H}(\mathbf{X})} \frac{\partial \mathcal{H}(\mathbf{X})}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{p}}$ where l is the network loss. Obtaining $\frac{\partial l}{\partial \mathbf{p}}$ necessitates computing $\frac{\partial \mathcal{H}(\mathbf{X})}{\partial \mathbf{X}}$, *i.e*., the spatial gradients of $\mathcal{H}$, which can be numerically computed by three convolution operations with pre-defined kernels along X-, Y- and Z-axis respectively. $\frac{\partial \mathbf{X}}{\partial \mathbf{p}}$ can be derived analytically.

Refinement Network for Final Shape

With a coarse voxel occupancy probability $\mathcal{V}$ from V-Net and the visual hull $\mathcal{H}$ from the PSVH layer, we use a 3D CNN to refine $\mathcal{V}$ and obtain a final prediction, denoted by $\mathcal{V}^+$. We refer to this refinement CNN as *R-Net*. The basic structure of our R-Net is shown in Fig. 3 (e). It consists of five 3D conv layers in the encoder and 14 3D conv layers in the decoder.

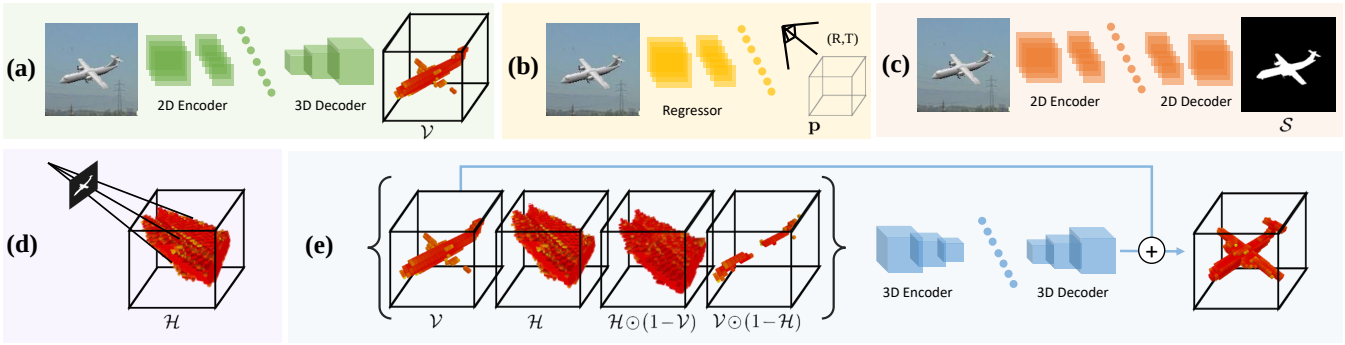

Figure 3: Network structure illustration. Our network consists of (a) a coarse shape estimation subnet V-Net, (b) an object pose estimation subnet P-Net , (c) an object segmentation subnet, (d) a probabilistic single-view visual hull (PSVH) layer, and finally (e) a shape refinement network R-Net.

A straightforward way for R-Net to process $\mathcal{V}$ and $\mathcal{H}$ is concatenating $\mathcal{V}$ and $\mathcal{H}$ to form a 2-channel 3D voxel grid as input then generating a new $\mathcal{V}$ as output. Nevertheless, we have some domain knowledge on this specific problem. For example, if a voxel predicted as occupied falls out of the visual hull, it's likely to be a false alarm; if the prediction does not have any occupied voxel in a viewing ray of the visual hull, some voxels may have been missed. This domain knowledge prompted us to design the R-Net in the following manners.

First, in addition to $\mathcal{V}$ and $\mathcal{H}$, we feed into R-Net two more occupancy probability maps: $\mathcal{V}\odot(1-\mathcal{H})$ and $\mathcal{H}\odot(1-\mathcal{V})$ where $\odot$ denotes element-wise product. These two probability maps characterize voxels in $\mathcal{V}$ but not in $\mathcal{H}$, and voxels in $\mathcal{H}$ but not in $\mathcal{V}$[1], respectively. Second, we add a residual connection between the input voxel prediction $\mathcal{V}$ and the output of the last layer. This way, we guide R-Net to generate an effective *shape deformation* to refine $\mathcal{V}$ rather than directly predicting a new $\mathcal{V}$, as the predicted $\mathcal{V}$ from V-Net is often mostly reasonable (as found in our experiments).

Network Training

We now present our training strategies, including the training pipeline for the sub-networks and their training losses.

Training pipeline. We employ a three-step network training algorithm to train the proposed network. Specifically, we first train V-Net, S-Net and R-Net separately, with input training images and their ground-truth shapes, silhouettes and poses. After V-Net converges, we train R-Net independently, with the predicted voxel occupancy probability $\mathcal{V}$ from V-Net and the ground-truth visual hull, which is constructed by ground-truth silhouettes and poses via the PSVH layer. The goal is to let R-Net learn how to refine coarse shape predictions with ideal, error-free visual hulls. In the last stage, we finetune the whole network, granting the subnets more opportunity to cooperate accordingly. Notably, the R-Net will adapt to input visual hulls that are subject to estimation error from S-Net and P-Net.

[1]A better alternative for $\mathcal{H}\odot(1-\mathcal{V})$ would be constructing another visual hull using $\mathcal{V}$ and $\mathbf{p}$ then compute its difference from $\mathcal{H}$. We choose $\mathcal{H}\odot(1-\mathcal{V})$ here for simplicity.

Training loss. We use the binary cross-entropy loss to train V-Net, S-Net and R-Net. Concretely, let p_n be the estimated probability at location n in either $\mathcal{V}$, $\mathcal{S}$ or $\mathcal{V}^+$, then the loss is defined as

$$l = -\frac{1}{N}\sum_n \left(p_n^* \log p_n + (1-p_n^*) log(1-p_n)\right) \qquad (2)$$

where p_n^* is the target probability (0 or 1). n traverses over the 3D voxels for V-Net and R-Net, and over 2D pixels for S-Net. The P-Net produces a 6-D pose estimate $\mathbf{p} = [\theta_1, \theta_2, \theta_3, t_u, t_v, t_Z]^{\mathrm{T}}$ as described before. We use the L_1 regression loss to train the network:

$$l = \sum_{i=1,2,3} \alpha|\theta_i - \theta_i^*| + \sum_{j=u,v} \beta|t_j - t_j^*| + \gamma|t_Z - t_Z^*|, \qquad (3)$$

where we set $\alpha = 1$, $\gamma = 1$ and $\beta = 0.01$, the Euler angles are normalized into $[0, 1]$. We found in our experiments the L_1 loss produces better results than an L_2 loss.

Experiments

Implementation details. Our network is implemented in TensorFlow. The input image size is 128×128 and the output voxel grid size is $32\times32\times32$. Batch size of 24 and the ADAM solver are used throughout the training. We use a learning rate of $1e{-}4$ for S-Net, V-Net and R-Net and divide it by 10 at the 20K-th and 60K-th iterations. The learning rate for P-Net is $1e-5$ and is dropped by 10 at the 60K-th iteration. When finetuning all the subnets together the learning rate is $1e-5$ and dropped by 10 at the 20K-th iteration.

Training and testing data. In this paper, we test our method on four common object categories: *car* and *airplane* as the representative vehicle objects, and *chair* and *couch* as furniture classes. Real images that come with precise 3D shapes are difficult to obtain, so we first resort to the CAD models from the ShapeNet repository (Chang et al. 2015). We use the ShapeNet object images rendered by (Choy et al. 2016) to train and test our method. We then use the PASCAL 3D+ dataset (Xiang, Mottaghi, and Savarese 2014) to evaluate our method on real images with pseudo ground truth shapes. This dataset only contains some pseudo ground truth shapes.

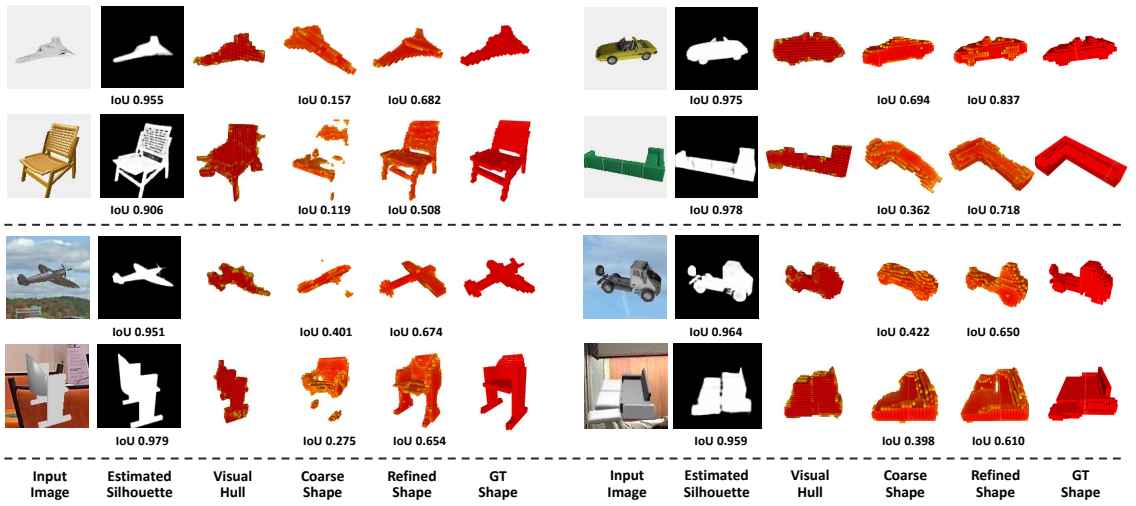

Figure 4: Qualitative results on the test set of the rendered ShapeNet objects. The top two rows and bottom two rows show some results on images with a clean background and real-image background, respectively. The color of the voxels indicates the predicted probability (red/yellow for high/low probability). While the coarse shapes may miss some shape details or be inconsistent with the image, our final shapes refined with single-view visual hulls are of high fidelity.

Table 1: The performance (shape IoU) of our method on the test set of the rendered ShapeNet objects. $\mathcal{H}$ denotes visual hull and GT indicates ground truth.

	car	airplane	chair	couch	Mean
Before Refine.	0.819	0.537	0.499	0.667	0.631
After Refine.	**0.839**	**0.631**	**0.552**	0.698	**0.680**
Refine. w/o $\mathcal{H}$	0.824	0.541	0.505	0.675	0.636
Refine. w. GT $\mathcal{H}$	0.869	0.701	0.592	0.741	0.726
Refine. w/o 2 prob.maps	0.840	0.610	0.549	0.701	0.675
Refine. w/o end-to-end	0.822	0.593	0.542	0.677	0.658

Results on Rendered ShapeNet Objects

The numbers of 3D models for the four categories are 7,496 for car, 4,045 for airplane, 6,778 for chair and 3,173 for table, respectively. In the rendering process of (Choy et al. 2016), the objects were normalized to fit in a radius-0.5 sphere, rotated with random azimuth and elevation angles, and placed in front of a 50-degree FOV camera. Each object has 24 images rendered with random poses and lighting.

Following (Choy et al. 2016), we use 80% of the 3D models for training and the rest 20% for testing. We train one network for all the four shape categories until the network converge. The rendered images are with clean background (uniform colors). During training, we blend half of the training images with random crops of natural images from the SUN database (Xiao et al. 2010). We binarize the output voxel probability with threshold 0.4 and report Intersection-over-Union (IoU).

Quantitative results. The performance of our method evaluated by IoU is shown in Table 1. It shows that the results after refinement (*i.e.*, our final results) are significantly better, especially for airplane and chair where the IoUs are improved by about 16% and 10%, respectively. Note that since our V-Net is adapted from (Choy et al. 2016) as mentioned previously, the results before refinement can be viewed as the 3D-R2N2 method of (Choy et al. 2016) trained by us.

To better understand the performance gain from our visual hull based refinement, we compute the IoU of the coarse and refined shapes for each object from the four categories. Figure 5 presents the comparisons, where the object IDs are

Table 2: The performance (shape IoU) of our method and PointOutNet (Fan, Su, and Guibas 2017).

	car	airplane	chair	couch	Mean
(Fan, Su, and Guibas 2017)	0.831	0.601	0.544	**0.708**	0.671
Ours	**0.839**	**0.631**	**0.552**	0.698	**0.680**

Table 3: The performance (shape IoU) of our method on the test set of the rendered ShapeNet objects with clean background and background from natural image crops.

Background	car	airplane	chair	couch	Mean
Clean	0.839	0.631	0.552	0.698	0.680
Image crop.	0.837	0.617	0.541	0.700	0.674

Table 4: The pose estimation and segmentation quality of our P-Net and S-Net on the rendered ShapeNet objects. Mean values are shown for each category.

	car	airplane	chair	couch	Mean
Rotation error	7.96°	4.72°	6.59°	10.41°	7.42°
Translation error	3.33%	2.60%	3.26%	3.41%	3.15%
Silhouette IoU	0.923	0.978	0.954	0.982	0.959

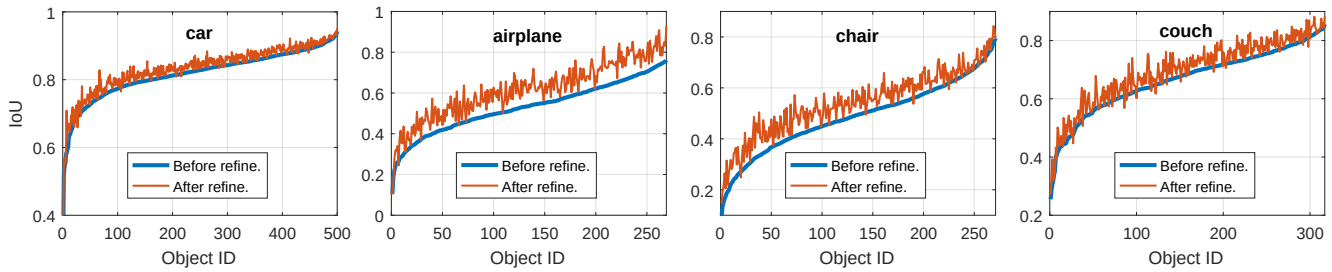

Figure 5: Comparison of the results before and after refinement on rendered ShapeNet objects.

uniformly sampled and sorted by the IoUs of coarse shapes. The efficacy of our refinement scheme can be clearly seen. It consistently benefits the shape reconstruction for most of the objects, despite none of them is seen before.

We further compare the numerical results with PointOutNet (Fan, Su, and Guibas 2017) which was also evaluated on this rendered dataset and used the same training/testing lists as ours. Table 1 shows that our method outperformed it on the three of the four categories (car, airplane and chair) and obtained a higher mean IoU over the four categories. Note that the results of (Fan, Su, and Guibas 2017) were obtained by first generating point clouds using their PointOutNet, then converting them to volumetric shapes and applying another 3D CNN to refine them.

Table 3 compares the results of our method on test images with clean background and those blended with random real images. It shows that with random real image as background the results are only slightly worse. Table 4 shows the quality of the pose and silhouette estimated by P-Net and S-Net.

Qualitative results. Figure 4 presents some visual results from our method. It can be observed that some object components especially thin structures (*e.g.* the chair legs in the second and fifth rows) are missed in the coarse shapes. Moreover, we find that although some coarse shapes appear quite realistic (*e.g.* the airplanes in the left column), they are clearly inconsistent with the input images. By leveraging the single-view visual hull for refinement, many shape details can be recovered in our final results, and they appear much more consistent with the input images.

We also compare our results qualitatively with MarrNet (Wu et al. 2017), another state-of-the-art single-view 3D object reconstruction method[2]. The authors released a MarrNet model trained solely on the chair category of the ShapeNet objects. Figure 6 presents the results on four chair images, where the first/last two are relatively good results from MarrNet/our method cherry-picked among 100 objects on our test set. It can be seen that in both cases, our method generated better results than MarrNet. Our predicted shapes are more complete and consistent with the input images.

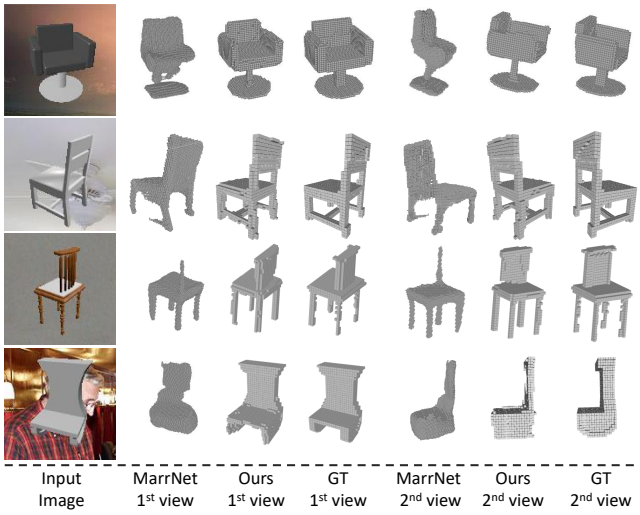

Figure 6: Result comparison with the MarrNet method on ShapeNet chairs (our testing split). Top two rows: cherry-picked results of MarrNet, compared against our results. Bottom two rows: cherry-picked results of our method, compared against MarrNet results.

Table 5: The performance (shape IoU) of our method on the test set of the PASCAL 3D+ dataset.

	car	airplane	chair	couch	Mean
Before Refine.	0.625	0.647	0.341	0.633	0.552
After Refine.	**0.653**	**0.690**	**0.341**	**0.664**	**0.587**

Table 6: The pose estimation and segmentation quality of P-Net and S-Net on PASCAL 3D+. Median values are reported. Note that the pseudo ground-truth poses and silhouettes used for evaluation are noisy.

	car	airplane	chair	couch
Rotation error	39.4°	25.5°	43.6°	34.0°
Translation error	8.6%	4.4%	12.7%	12.5%
Silhouette IoU	0.757	0.614	0.457	0.696

Results on the Pascal 3D+ Dataset

We now evaluate our method on real images from the PASCAL 3D+ dataset (Xiang, Mottaghi, and Savarese 2014).

[2]We were not able to compare the results quantitatively: Marr-

This dataset only have pseudo ground-truth shapes for real images, which makes it very challenging for our visual hull based refinement scheme. Moreover, the provided object poses are noisy due to the lack of accurate 3D shapes, making it difficult to train our pose network.

To test our method on this dataset, we finetune our network trained on ShapeNet objects on images in PASCAL 3D+. We simply set the focal length to be 2000 for all images since no focal length is provided. With this fixed focal length, we recomputed the object distances using the image keypoint annotations and the CAD models through reprojection error minimization.

Quantitative results. The quantitative results of our method are presented in Table 5 and Table 6. As can be seen in Table 6, the pose and silhouette estimation errors are much higher than the results on the ShapeNet objects. Nevertheless, Table 5 shows that our visual hull based refinement scheme still largely improved the coarse shape from V-Net for the car, airplane and couch categories. Note again that our V-Net is almost identical to the network in the 3D-R2N2 method (Choy et al. 2016). The refinement only yields marginal IoU increase for the chair category. We observed that the chair category on this dataset contains large intra-class shape variations (yet only 10 CAD shapes as pseudo ground truth) and many instances with occlusion; see the *suppl. material* for more details.

Qualitative results. Figure 7 shows some visual results of our method on the test data. It can be seen that the coarse shapes are noisy or contain erroneous components. For example, possibly due to the low input image quality, the coarse shape prediction of the car image in the second row of the left column has a mixed car and chair structure. Nevertheless, the final results after the refinement are much better.

Figure 1 shows some results from both our method and MarrNet (Wu et al. 2017). Visually inspected, our method produces better reconstruction results again.

More Ablation Study and Performance Analysis

Performance of refinement without visual hull. In this experiment, we remove the probabilistic visual hull and train R-Net to directly process the coarse shape. As shown in Table 1, the results are slightly better than the coarse shapes, but lag far behind the results refined with visual hull.

Performance of refinement with GT visual hull. We also trained R-Net with visual hulls constructed by ground-truth poses and silhouettes. Table 1 shows that the performance is dramatically increased: the shape IoU is increased by up to 30% from the coarse shape for the four object categories. The above two experiments indicate that our R-Net not only leveraged the visual hull to refine shape, but also can work remarkably well if given a quality visual hull.

Effect of two additional occupancy probability maps $\mathcal{V} \odot (1 - \mathcal{H})$ ***and*** $\mathcal{H} \odot (1 - \mathcal{V})$***.*** The results in Table 1 shows that, if these two additional maps are removed from the input of R-Net, the mean IoU drops slightly from 0.680 to 0.675, indicating our explicit knowledge embedding helps.

Effect of end-to-end training. Table 1 also presents the result without end-to-end training. The clear performance drop demonstrates the necessity of our end-to-end finetuning which grants the subnets the opportunity to better adapt to each other (notably, R-Net will adapt to input visual hulls that are subject to estimation error from S-Net and P-Net).

Net directly predicts the shapes in the current camera view which are not aligned with GT shapes; moreover, the training and testing splits for MarrNet are not disclosed in Wu et al. (2017)

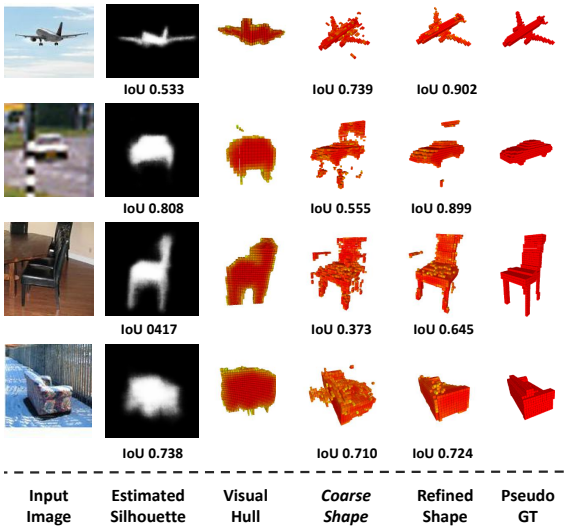

Figure 7: Qualitative results on the test set of PASCAL 3D+

Running Time

For a batch of 24 input images, the forward pass of our whole network takes 0.44 seconds on an NVIDIA Tesla M40 GPU, *i.e.*, our network processes one image with 18 milliseconds on average.

Conclusions

We have presented a novel framework for single-view 3D object reconstruction, where we embed the perspective geometry into a deep neural network to solve the challenging problem. Our key innovations include an in-network visual hull construction scheme which connects the 2D space and pose space to the 3D space, and a refinement 3D CNN which learns shape refinement with visual hulls. The experiments demonstrate that our method achieves very promising results on both synthetic data and real images.

Limitations and future work. Since our method involves pose estimation, objects with ambiguous pose (symmetric shapes) or even do not have a well-defined pose system (irregular shapes) will be challenging. For the former cases, using a classification loss to train the pose network would be a good remedy (Su et al. 2015), although this may render the gradient backpropagation problematic. For the latter, one possible solution is resorting to multi-view inputs and train the pose network to estimate *relative* poses.

Acknowledgement

This research is supported by a Natural Science Foundation of China(NSFC) grant No.61876020 and No.61472038.

References

Chang, A. X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; Xiao, J.; Yi, L.; and Yu, F. 2015. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*.

Choy, C. B.; Xu, D.; Gwak, J.; Chen, K.; and Savarese, S. 2016. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *ECCV*, 628–644.

Fan, H.; Su, H.; and Guibas, L. 2017. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 605–613.

Girdhar, R.; Fouhey, D. F.; Rodriguez, M.; and Gupta, A. 2016. Learning a predictable and generative vector representation for objects. In *ECCV*, 484–499.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2672–2680.

Gwak, J.; Choy, C. B.; Garg, A.; Chandraker, M.; and Savarese, S. 2017. Weakly supervised generative adversarial networks for 3D reconstruction. *arXiv:1705.10904*.

Häne, C.; Tulsiani, S.; and Malik, J. 2017. Hierarchical surface prediction for 3d object reconstruction. In *International Conference on 3D Vision (3DV)*.

Ji, M.; Gall, J.; Zheng, H.; Liu, Y.; and Fang, L. 2017. SurfaceNet: an end-to-end 3d neural network for multiview stereopsis. In *ICCV*, 2307–2315.

Kar, A.; Tulsiani, S.; Carreira, J.; and Malik, J. 2015. Category-specific object reconstruction from a single image. In *CVPR*, 1966–1974.

Kar, A.; Häne, C.; and Malik, J. 2017. Learning a multi-view stereo machine. In *NIPS*, 364–375.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Liang, W.; Zhao, Y.; Zhu, Y.; and Zhu, S.-C. 2016. What is where: Inferring containment relations from videos. In *IJCAI*.

Massa, F.; Marlet, R.; and Aubry, M. 2016. Crafting a multi-task cnn for viewpoint estimation. *The British Machine Vision Conference (BMVC)*.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 652–659.

Richter, S. R., and Roth, S. 2018. Matryoshka networks: Predicting 3d geometry via nested shape layers. *CoRR* abs/1804.10975.

Sharma, A.; Grau, O.; and Fritz, M. 2016. VConv-DAE: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision Workshop on Geometry Meets Deep Learning*, 236–250.

Su, H.; Qi, C. R.; Li, Y.; and Guibas, L. J. 2015. Render for CNN: Viewpoint estimation in images using cnns trained with rendered 3D model views. In *ICCV*, 2686–2694.

Sun, M.; Kumar, S. S.; Bradski, G.; and Savarese, S. 2013. Object detection, shape recovery, and 3d modelling by depth-encoded hough voting. *Computer Vision and Image Understanding (CVIU)* 117(9):1190–1202.

Tatarchenko, M.; Dosovitskiy, A.; and Brox, T. 2017. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2088–2096.

Tulsiani, S.; Kar, A.; Carreira, J.; and Malik, J. 2017a. Learning category-specific deformable 3d models for object reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39(4):719–731.

Tulsiani, S.; Zhou, T.; Efros, A. A.; and Malik, J. 2017b. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2626–2633.

Tulsiani, S.; Efros, A. A.; and Malik, J. 2018. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *CVPR*, 2897–2905.

Vicente, S.; Carreira, J.; Agapito, L.; and Batista, J. 2014. Reconstructing pascal voc. In *CVPR*, 41–48.

Wang, H.; Liang, W.; and Yu, L. F. 2017. Transferring objects: Joint inference of container and human pose. In *IEEE International Conference on Computer Vision*, 2952–2960.

Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; and Xiao, J. 2015. 3D ShapeNets: A deep representation for volumetric shapes. In *CVPR*, 1912–1920.

Wu, J.; Zhang, C.; Xue, T.; Freeman, B.; and Tenenbaum, J. 2016. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *NIPS*, 82–90.

Wu, J.; Wang, Y.; Xue, T.; Sun, X.; Freeman, W. T.; and Tenenbaum, J. B. 2017. MarrNet: 3d shape reconstruction via 2.5D sketches. In *NIPS*, 540–550.

Xiang, Y.; Mottaghi, R.; and Savarese, S. 2014. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 75–82.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492.

Yan, X.; Yang, J.; Yumer, E.; Guo, Y.; and Lee, H. 2016. Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *NIPS*, 1696–1704.

Zhu, R.; Galoogahi, H. K.; Wang, C.; and Lucey, S. 2017. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *ICCV*, 57–64.

Zhu, Y.; Zhao, Y.; and Zhu, S. C. 2015. Understanding tools: Task-oriented object modeling, learning and recognition. In *Computer Vision and Pattern Recognition*, 2855–2864.

Zou, C.; Yumer, E.; Yang, J.; Ceylan, D.; and Hoiem, D. 2017. 3D-PRNN: generating shape primitives with recurrent neural networks. In *ICCV*, 900–909.