2021 IEEE Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-0477-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/WACV48630.2021.00222

Learning to Generate Dense Point Clouds with Textures on Multiple Categories

Tao Hu, Geng Lin, Zhizhong Han, Matthias Zwicker
University of Maryland, College Park
taohu@cs.umd.edu, geng@cs.umd.edu, h312h@umd.edu, zwicker@cs.umd.edu

Abstract

3D reconstruction from images is a core problem in computer vision. With recent advances in deep learning, it has become possible to recover plausible 3D shapes even from single RGB images. However, obtaining detailed geometry and texture for objects with arbitrary topology remains challenging. In this paper, we propose a novel approach for reconstructing point clouds from RGB images. Unlike other methods, we can recover dense point clouds with hundreds of thousands of points, and we also include RGB textures. In addition, we train our model on multiple categories, which leads to superior generalization to unseen categories compared to previous techniques. We achieve this using a two-stage approach, where we first infer an object coordinate map from the input RGB image, and then obtain the final point cloud using a reprojection and completion step. We show results on standard benchmarks that demonstrate the advantages of our technique.

1. Introduction

3D reconstruction from single RGB images has been a longstanding challenge in computer vision. While recent progress with deep learning-based techniques and large shape or image databases has been significant, the reconstruction of detailed geometry and texture for a large variety of object categories with arbitrary topology remains challenging. Point clouds have emerged as one of the most popular representations to tackle this challenge because of a number of distinct advantages: unlike meshes they can easily represent arbitrary topology, unlike 3D voxel grids they do not suffer from cubic complexity, and unlike implicit functions they can reconstruct shapes using a single evaluation of a neural network. In addition, it is straightforward to represent surface textures with point clouds by storing per-point RGB values.

In this paper, we present a novel method to reconstruct 3D point clouds from single RGB images, including the optional recovery of per-point RGB texture. In addition, our approach can be trained on multiple categories. The key idea of our method is to solve the problem using a two-stage approach, where both stages can be implemented using powerful 2D image-to-image translation networks: in the first stage, we recover an object coordinate map from the input RGB image. This is similar to a depth image, but it corresponds to a point cloud in object-centric coordinates that is independent of camera pose. In the second stage, we reproject the object space point cloud into depth images from eight fixed viewpoints in object space, and perform depth map completion. We can then trivially fuse all completed object space depth maps into a final 3D reconstruction, without requiring a separate alignment stage, for example using the iterative closest point algorithm (ICP) [3]. Since all networks are based on 2D convolutions, it is straightforward to achieve high resolution reconstructions with this approach. Texture reconstruction uses the same pipeline, but operating on RGB images instead of object space depth maps.

We train our approach on a multi-category dataset and show that our object-centric, two-stage approach leads to better generalization than competing techniques. In addition, recovering object space point clouds allows us to avoid a separate camera pose estimation step. In summary, our main contributions are as follows:

- A strategy to generate 3D shapes from single RGB images in a two-stage approach, by first recovering object coordinate images as an intermediate representation, and then performing reprojection, depth map completion, and a final trivial fusion step in object space.
- The first work to train a single network to reconstruct point clouds with RGB textures on multiple categories.
- More accurate reconstruction results than previous methods on both seen and unseen categories from ShapeNet [4] or Pix3D [33] datasets.

2. Related Work

Our method is mainly related to single image 3D reconstruction and shape completion. We briefly review previous works in these two aspects.

978-1-6654-0477-8/21/$31.00 ©2021 IEEE
DOI 10.1109/WACV48630.2021.00222

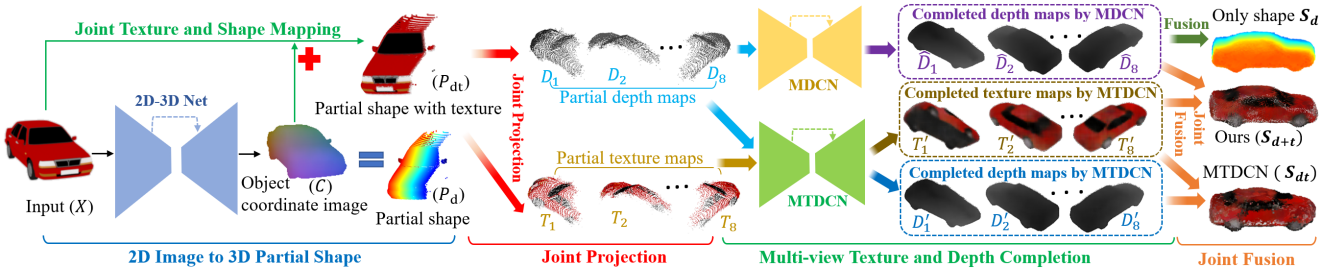


Figure 1: Approach overview. An image X is passed through a 2D-3D network to reconstruct the visible parts of the object, represented by an object coordinate image C. X and C represent the texture and 3D coordinates of a shape respectively, which yield a partial shape with texture P_{dt} when combined by a *Joint Texture and Shape Mapping* operator. Next, by *Joint Projection*, P_{dt} is jointly projected from 8 fixed viewpoints into 8 pairs of partial depth maps and textures maps, which are translated to completed maps by the Multi-view Texture-Depth Completion Net (MTDCN) that jointly completes texture and depth maps. Alternatively, Multi-view Depth Completion Net (MDCN) only completes the depth maps. Finally, the *Joint Fusion* operator fuses the completed multiple texture and depth maps into completed point clouds.

Single image 3D reconstruction. Along with the development of deep learning techniques, single image 3D reconstruction has made huge progress. Because of their regularity, early works mainly learned to reconstruct voxel grids from 3D supervision [7] or 2D supervision [34] using differentiable renderers [43, 37]. However, these methods can only reconstruct shapes at low resolution, such as 32^3 or 64^3, due to the cubic complexity of voxel grids. Although various strategies [13, 35] were proposed to increase the resolution, these methods were too complex to follow. Mesh based methods [39, 21] are also alternatives to increase the resolution. However meshes often have fixed topologies, which limits the space of representable 3D shapes. Point cloud based methods [10, 29, 46, 23] provide another direction for single image 3D reconstruction. However, some of these methods are also limited to low resolutions, which makes it hard to reconstruct small geometric details.

Besides low resolution, lack of texture is another issue that affects the realism of generated shapes. Current methods aim to map the texture from single images to the reconstructed shapes either represented by mesh templates [17] or point clouds using object coordinate maps [32]. [26] proposed a differentiable point feature rendering module named DIFFER to reconstruct 3D point clouds with colors from single images. The texture prediction pipeline of [17] samples pixels from input images directly and works on symmetric objects with a good viewpoint. Some other methods (e.g. [48, 34]) try to predict novel RGB views by view synthesis. Although these methods have shown promising results in some specific shape classes, their models were usually trained on one single category for category-specific reconstruction.

Recently implicit functions have also been used to represent shapes [27, 24, 25, 5, 11, 42, 6]. Fed with a latent code and a query point, a neural network is trained to predict the SDF value [24, 42] or the binary occupancy of the point [27, 25]. Though these methods can generate high resolution geometry by evaluating the learned implicit functions at query 3D points at arbitrary resolutions, they cannot reconstruct shapes and textures at the same time in their pipelines.

Different from all these methods, our method can jointly learn to reconstruct very dense point clouds with texture for multiple-category reconstruction by a two-stage reconstruction approach, leveraging object coordinate maps (also called NOCS maps [38, 32]) as intermediate representation. Different from previous methods [47, 46] that use depth maps as intermediate representation in a viewer-centered setting, our method works on object-centered coordinates. Besides the capability of predicting textures, compared with the implicit function-based methods, our approach generates 3D point clouds by multi-view back-projection, while implicit function-based methods usually need the marching cubes [22] algorithm as post-processing to extract surfaces.

Shape completion. Shape completion is to infer the whole 3D geometry from partial observations. Different methods use volumetric grids [8] or point clouds [45, 44, 1] as shape representation for this task. Point-based methods are often based on encoder-decoder structures that employ the PointNet architecture [28] as a backbone. Although these works have shown nice completed shapes, they are limited to low resolution. To resolve this issue, Hu et al. [14] introduced Render4Completion to cast the 3D shape completion problem into multiple 2D view completions, and they demonstrate promising potential on high resolution shape completion. Our method follows this direction, however, we perform image-based 3D reconstruction and not only learn geometry but also texture.

3. Approach

Most 3D point cloud reconstruction methods [23, 7, 9] solely focus on generating 3D shapes $\{P_i = [x_i, y_i, z_i]\}$

from input RGB images $X \in \mathbb{R}^{H\times W\times 3}$, where $H \times W$ is the image resolution and $[x_i, y_i, z_i]$ are 3D coordinates. Recovering the texture besides 3D coordinates is a more challenging task, which requires learning a mapping from $\mathbb{R}^{H\times W\times 3}$ to $\{P_i = [x_i, y_i, z_i, r_i, g_i, b_i]\}$, where $[r_i, g_i, b_i]$ are RGB values.

We propose a method to generate high resolution 3D predictions and recover textures from RGB images. At a high level, we decompose the reconstruction problem into two less challenging tasks: first, transforming 2D images to 3D partial shapes that correspond to the observed parts of the target object, and second, completing the unseen parts of the 3D object. We use object coordinate images to represent partial 3D shapes, and multiple depth and RGB views to represent completed 3D shapes.

As shown in Figure 1, our pipeline consists of four submodules: (1) 2D-3D Net, an image translation network which translates an RGB image X to a partial shape P_d (represented by object coordinate image C); (2) the Joint Projection module, which first jointly maps the partial shape P_d with texture X to generate P_{dt}, a partial shape mapped with texture, and then jointly projects P_{dt} into 8 pairs of partial depth $[D_1, \ldots, D_8]$ and texture views $[T_1, \ldots, T_8]$ from 8 fixed viewpoints (the 8 vertices of a cube); (3) the multi-view texture and depth completion module, which consists of two networks: Multi-view Texture-Depth Completion Net (MTDCN), which generates completed texture maps $[T'_1, \ldots, T'_8]$ and depth maps $[D'_1, \ldots, D'_8]$ by jointly completing partial texture and depth maps, and as an alternative, Multi-view Depth Completion Net (MDCN), which only completes depth maps and generates more accurate results $[\hat{D}_1, \ldots, \hat{D}_8]$; (4) the Joint Fusion module, which jointly fuses the completed depth and texture views into completed 3D shape with textures, like S_{d+t} and S_{dt}.

3.1. 2D RGB Image to Partial Shapes

We propose to use 3-channel object coordinate images to represent partial shapes. Each pixel on the object coordinate image represents a 3D point, where its (r, g, b) value corresponds to the point's location (x, y, z). An object coordinate image is aligned with the input image, as shown in Figure 1, and in our pipeline, it represents the visible parts of the target 3D object. With this image-based 3D representation, we formulate the 2D-to-3D transformation as an image-to-image translation problem, and propose a 2D-3D Net to perform the translation based on the U-Net [30] architecture as in [16].

Unlike the depth map representation used in [47, 46], which requires camera pose information for back-projection, the 3-channel object coordinate image can represent a 3D shape independently.

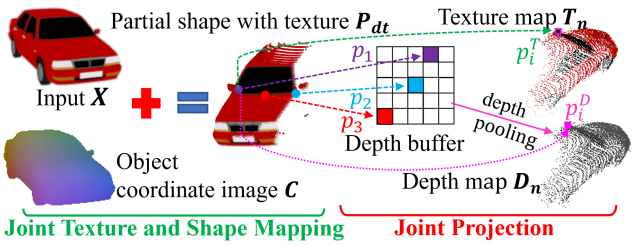


Figure 2: Partial shapes to multiple views.

3.2. Partial Shapes to Multiple Views

In this module, we transform the input RGB image X and the predicted object coordinate image C to a partial shape mapped with texture, P_{dt}, which is then rendered from 8 fixed viewpoints to generate depth maps and texture maps. The process is illustrated in Figure 2.

Joint Texture and Shape Mapping. The input RGB image X is aligned with the generated object coordinate image C. An equivalent partial point cloud P_{dt} can be obtained by taking 3D coordinates from C and texture from X.

We denote a pixel on X as $p_i^X = [u_i^X, v_i^X, r_i^X, g_i^X, b_i^X]$, where u_i^X and v_i^X are pixel coordinates, and similarly, a point on C as $p_i^C = [u_i^C, v_i^C, x_i^C, y_i^C, z_i^C]$. Given p_i^X and p_i^C appearing at the same location, which means $u_i^X = u_i^C$ and $v_i^X = v_i^C$, then p_i^X and p_i^C can be projected into 3D coordinates as $P_i = [x_i, y_i, z_i, r_i, g_i, b_i]$ on partial shape P_{dt}, where r_i, g_i, b_i are RGB channels and $x_i = x_i^C, y_i = y_i^C, z_i = z_i^C, r_i = r_i^X, g_i = g_i^X, b_i = b_i^X$.

Joint Projection. We render multiple depth maps $D = \{D_1, \ldots, D_8\}$ and texture maps $T = \{T_1, \ldots, T_8\}$ from 8 fixed viewpoints $V = \{V_1, \ldots, V_8\}$ of the partial shape P_{dt}, where $D_n \in \mathbb{R}^{H\times W}$, $T_n \in \mathbb{R}^{H\times W\times 3}$, $n \in [1, 8]$.

Given n, we denote a point on depth map D_n as $p_i^D = [u_i^D, v_i^D, d_i^D]$ where u_i^D and v_i^D are pixel coordinates and d_i^D is the depth value. Similarly, a point on T_n is $p_i^T = [u_i^T, v_i^T, r_i^T, g_i^T, b_i^T]$, where r_i^T, g_i^T, b_i^T are RGB values. Then, we transform each 3D point P_i on the partial shape P_{dt} into a pixel $p'_i = [u'_i, v'_i, d'_i]$ on depth map D_n by

$$p'_i = K(\Re_n P_i + \tau_n) \quad \forall i, \tag{1}$$

where K is the intrinsic camera matrix, $\Re_n$ and τ_n are the rotation matrix and translation vector of view V_n. Note that Eq. (1) only projects the 3D coordinates of P_i.

However, different points on P_{dt} may be projected to the same location $[u, v]$ on the depth map D_n. For example, in Figure 2, $p_1 = [u, v, d_1], p_2 = [u, v, d_2], p_3 = [u, v, d_3]$ are projected to the same pixel $p_i^D = [u_i^D, v_i^D, d_i^D]$ on D_n, where $u_i^D = u, v_i^D = v$. The corresponding point on the texture map T_n is $p_i^T = [u_i^T, v_i^T, r_i^T, g_i^T, b_i^T]$ where $u_i^T = u, v_i^T = v$.

To alleviate this collision effect, we implement a pseudo-rendering technique similar to [15, 20]. Specifically, for each point on P_{dt}, a depth buffer with a size of $U \times U$

is used to store multiple depth values corresponding to the same pixel. Then we implement a depth-pooling operator with stride $U \times U$ to select the minimum depth value. We set $U = 5$ in our experiments. In depth-pooling, we store the indices of pooling (j) and select the closest point from the view point V_n among $\{p_1, p_2, p_3\}$. For example, in Figure 2, pooling index $j = 1$, the selected point is p_1, and the corresponding point on P_{dt} is P_1. In this case, we copy the texture values from P_1 to p_i^T.

3.3. Multi-view Texture and Depth Completion

In our pipeline, a full shape is represented by depth images from multiple views, which are processed by CNNs to generate high resolution 3D shapes as mentioned in [20, 14].

Multi-view Texture-Depth Completion Net (MTDCN). We propose a Multi-view Texture-Depth Completion Net (MTDCN) to jointly complete texture and depth maps. MTDCN is based on a U-Net architecture. In our pipeline, we stack each pair of partial depth map D_n and texture map T_n into a 4-channel texture-depth map $Q_n = [T_n, D_n], Q_n \in \mathbb{R}^{H \times W \times 4}$, $n \in [1, 8]$. MTDCN takes Q_n as input, and generates completed 4-channel texture-depth maps $Q'_n = [T'_n, D'_n], Q'_n \in \mathbb{R}^{H \times W \times 4}$, where T'_n and D'_n are completed texture and depth map respectively. The completions of the car model are shown in Figure 3. After fusing these views, we get a completed shape with texture S_{dt} in Figure 1.

In contrast to the category-specific reconstruction in [17], which samples texture from input images, thus having its performance relying on the viewpoint of the input images and the symmetry of the target objects, MTDCN can be trained to infer textures on multiple categories and does not assume objects being symmetric.

Multi-view Depth Completion Net (MDCN). In our experiments, we found it very challenging to complete both depth and texture map at the same time. As an alternative we also train MDCN, which only completes partial depth maps $[D_1, \ldots, D_8]$ and can generate more accurate full depth maps $[\hat{D}_1, \ldots, \hat{D}_8]$. We then map the texture $[T'_1, \ldots, T'_8]$ generated by MTDCN to the MDCN-generated shape S_d to get a reconstructed shape with texture S_{d+t} as illustrated in Figure 1.

Different from the multi-view completion net in [14], which only completes 1-channel depth maps, MTDCN can jointly complete both texture and depth maps. In addition, there is no discriminator in MTDCN or MDCN, in contrast to [14].

3.4. Joint Fusion

With the completed texture maps $T' = [T'_1, \ldots, T'_8]$ and depth maps $D' = [D'_1, \ldots, D'_8]$ by MTDCN and more accurate completed depth maps $\hat{D} = [\hat{D}_1, \ldots, \hat{D}_8]$ by MDCN, we jointly fuse the depth and texture maps into a colored 3D point, as illustrated in Figure 1.

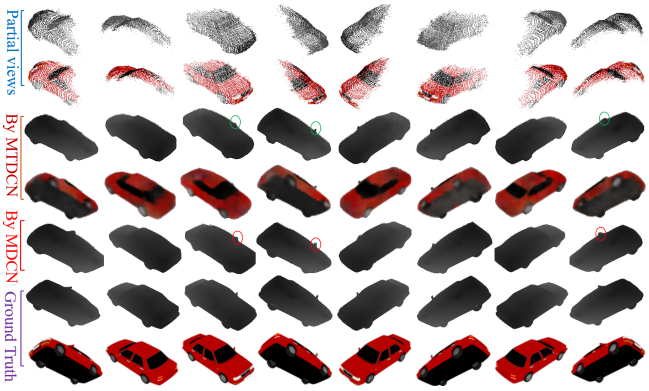


Figure 3: Completions of texture and depth maps.

Joint Fusion for MTDCN. Given one point $p_i^{D'} = [u_i^{D'}, v_i^{D'}, d_i^D]$ on D'_n, and the aligned point $p_i^{T'} = [u_i^{T'}, v_i^{T'}, r_i^{T'}, g_i^{T'}, b_i^{T'}]$ on the texture map T'_n, where $u_i^{D'} = u_i^{T'}$ and $v_i^{D'} = v_i^{T'}$, the back-projected point on S_{dt} is $P'_i = [x'_i, y'_i, z'_i, r'_i, g'_i, b'_i]$ by

$$P'_i = \Re_s^{-1}(K^{-1} p_i^{D'} - \tau_n) \quad \forall i. \tag{2}$$

Note that Eq. 2 only back-projects the depth map D'_n to the coordinates of P'_i, while the texture of P'_i is obtained from $p_i^{T'}$, where $r'_i = r_i^{T'}, g'_i = g_i^{T'}, b'_i = b_i^{T'}$. We also extract a completed shape S_d without texture.

Joint Fusion for MDCN. We map the texture $[T'_1, \ldots, T'_8]$ generated from MTDCN to the completed shape of MDCN S_{d+t}. The joint fusion process is similar. However, since texture and depth maps are generated separately, a valid point on a depth map may be aligned to an invalid point on the corresponding texture map, especially near edges. For such points, we take their nearest valid neighbor on the texture map. Since S_d is generated by direct fusion of depth maps $[\hat{D}_1, \ldots, \hat{D}_8]$, S_{d+t} has the same shape as S_d.

3.5. Loss Function and Optimization

Training Objective. We perform a two-stage training and train three networks: 2D-3D Net ($G1$), MTDCN (G_2), and MDCN (G_3). Given an input RGB image X, the generated object coordinate image is $C = G_1(X)$. The training objective of G_1 is

$$G_1{}^* = \arg\min_{G_1} ||G_1(X) - Y||_1, \tag{3}$$

where Y is the ground truth object coordinate image.

Given partial texture-depth images $Q_n = [T_n, D_n]$, $n \in [1, 8]$, the completed texture-depth images $Q'_n = G_2(Q_n)$, we get the optimal G_2 by

$$G_2{}^* = \arg\min_{G_2} ||G_2(Q_n) - Y'||_1, \tag{4}$$

where Y' is the ground truth texture-depth image.

MDCV only completes depth maps and takes 1-channel depth maps as input. Given a partial depth map D_n, the completed depth map $\hat{D}_n = G_3(D_n)$. G_3 is trained with

$$G_3{}^* = \arg\min_{G_3} ||G_3(D_n) - \hat{Y}||_1, \quad (5)$$

where $\hat{Y}$ is the ground truth depth image.

Optimization. We use Minibatch SGD and the Adam optimizer [18] to train all the networks. More details can be found in the supplementary material.

4. Experiments

We evaluate our methods (Ours-S_{d+t} generated by MDCN, and Ours-S_{dt} by MTDCN) on single-image 3D reconstruction and compare against state-of-the-art methods.

Dataset and Metrics. We train all our networks on synthetic models from ShapeNet [4], and evaluate them on both ShapeNet and Pix3D [33]. We render depth maps, texture maps and object coordinate images for each object. More details can be found in the supplementary material. The image resolution is 256×256. We sample 100K points from each mesh object as ground truth point clouds for evaluations on ShapeNet, as in [20]. For a fair comparison, we use Chamfer Distance (CD) [2] as the quantitative metric. Another popular option, Earth Mover's Distance (EMD) [10], requires that the generated point cloud has the same size as the ground truth, and its calculation is time-consuming. While EMD is often used as a metric for methods whose output is sparse and has fixed size, like 1024 or 2048 points in [9, 23], it is not suitable to evaluate our methods that generates very dense point clouds with varying numbers of points. We also report the performance in terms of F-score for overall evaluations.

4.1. Single Object Category

We first evaluate our method on a single object category. Following [43, 20], we use the chair category from ShapeNet with the same 80%-20% training/test split. We compare against two methods (Tatarchenko et al. [34] and Lin et al. [20]) that generate dense point clouds by view synthesis, as well as two voxels-based methods, Perspective Transformer Networks (PTN) [43] in two variants, and a baseline 3D-CNN provided in [43].

The quantitative results on the test dataset are reported in Table 1. Test results of other approaches are referenced from [20]. Our method (Ours-S_{d+t}) achieves the lowest CD in this single-category task. A visual comparison with Lin's method is shown in Figure 4, where our generated point clouds are denser and more accurate. In addition, we also infer the textures of the generated point clouds.

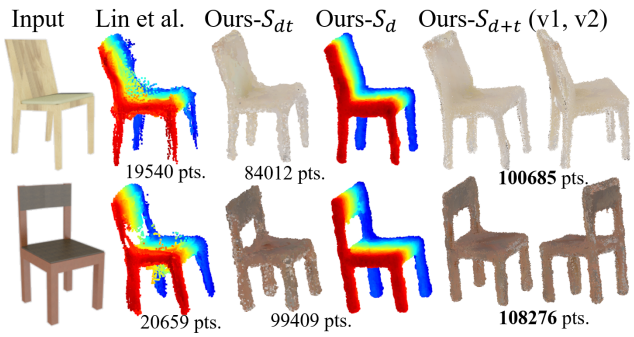


Figure 4: Reconstructions on single-category task.

4.2. General Object Categories from ShapeNet

We also simultaneously train our network on 13 categories (listed in Table 3) from ShapeNet and use the same 80%-20% training/test split as existing methods [7, 23].

Reconstructing novel objects from seen categories. We test our method on novel objects from the 13 seen categories and compare against (a) 3D-R2N2 [7], which predicts volumeric models with recurrent networks, and (b) PSGN [9], which predicts an unordered set of 1024 3D points by fully-connected layers and deconvolutional layers, and (3) 3D-LMNet which predicts point clouds by latent-embedding matching. We only compare methods that follow the same setting as 3D-R2N2, and do not include [20] which assumes fixed elevation or OptMVS [40]. We use the pretrained models readily provided by the authors, and the results of 3D-R2N2 and PSGN are referenced from [20]. We extract the surface voxels of 3D-R2N2 for evaluation.

Table 3 shows the quantitative results. Since most methods (e.g. [23, 9]) need ICP alignment as a post-processing step to achieve finer alignment with ground truth, we list the results without and with ICP. Note that PSGN predicts rotated point clouds, so we only list the results after ICP alignment. Ours-S_{d+t} outperforms the state-of-the-art methods on most categories. Specifically, we outperform 3D-LMNet on 12 categories out of 13 without ICP, and 7 with ICP. In addition, we achieve the lowest CD in average. Different from other methods, our methods do not rely too much on ICP, and more analysis can be found in Section 4.4.

We also visualize the predictions in Figure 6. It can be seen that our method predicts more accurate shapes with higher point density. Besides 3D coordinate predictions, our method also predicts textures. We demonstrate ours-S_{d+t} from two different views (v1) and (v2).

Reconstructing objects from unseen categories. We also evaluate how well our models generalizes to 6 unseen categories from ShapeNet: bed, bookshelf, guitar, laptop, motorcycle, and train. The quantitative comparisons with 3D-LMNet in Table 4 shows a better generalization of our method. We outperform 3D-LMNet on 4 categories out of

Method	CD
3D CNN (vol. loss only)	4.49
PTN (proj. loss only)	4.35
PTN (vol. & proj. loss)	4.43
Tatarchenko et al.	5.40
Lin et al.	3.53
Ours-S_{dt}	3.68
Ours-S_{d+t}	**3.04**

Table 1: CD on single-category task.

Category	P_d	Ours-S_{d+t}
airplane	10.53	4.19
bench	7.85	3.40
cabinet	19.07	4.88
car	11.14	2.90
chair	8.69	3.59
display	12.43	4.71
lamp	11.95	6.18
loudspeaker	20.26	6.39
rifle	9.47	5.44
sofa	10.86	4.07
table	8.83	3.27
telephone	9.83	3.16
vessel	9.08	3.79
mean	10.58	3.91
chair	9.04	3.04

Table 2: Mean CD of partial shape P_d and completed shape S_{d+t} to ground truth.

Category	3D-R2N2	PSGN	3D-LMNet	Ours-S_{dt}	Ours-S_{d+t}
airplane	(4.79)	(2.79)	6.16 (**2.26**)	**3.70** (3.37)	4.19 (3.66)
bench	(4.93)	(3.80)	5.79 (3.72)	4.27 (3.83)	**3.40** (**3.10**)
cabinet	(**4.04**)	(4.91)	6.98 (4.46)	6.77 (5.89)	**4.88** (4.50)
car	(4.81)	(3.85)	3.17 (2.91)	2.93 (2.95)	**2.90** (**2.90**)
chair	(4.93)	(4.24)	7.08 (3.74)	4.47 (4.12)	**3.59** (**3.22**)
display	(5.04)	(4.25)	7.89 (**3.72**)	5.55 (4.94)	**4.71** (3.85)
lamp	(13.03)	(**4.56**)	11.36 (4.57)	8.06 (7.13)	**6.18** (5.65)
loudspeaker	(6.69)	(6.00)	7.95 (**5.46**)	9.53 (8.28)	**6.39** (5.74)
rifle	(6.64)	(2.67)	**4.46** (**2.55**)	5.31 (4.28)	5.44 (4.30)
sofa	(5.50)	(5.38)	6.06 (4.44)	4.43 (3.93)	**4.07** (**3.57**)
table	(5.26)	(4.10)	6.65 (3.84)	4.59 (4.26)	**3.27** (**3.14**)
telephone	(4.61)	(3.50)	3.91 (3.10)	4.98 (4.72)	**3.16** (**2.90**)
vessel	(6.82)	(3.59)	6.30 (3.81)	4.13 (3.85)	**3.79** (**3.52**)
mean	(5.93)	(4.13)	6.14 (3.59)	4.68 (4.26)	**3.91** (**3.56**)

Table 3: Average CD of multiple-seen-category experiments on ShapeNet. Numbers beyond '()' are the CD before ICP, and in '()' are after ICP.

Category	3D-LMNet	Ours-S_{dt}	Ours-S_{d+t}
bed	13.56 (7.13)	12.82 (8.43)	**11.46** (**6.51**)
bookshelf	7.47 (**4.68**)	8.99 (7.96)	**5.63** (4.89)
guitar	8.19 (6.40)	7.07 (7.29)	**5.96** (**6.33**)
laptop	19.42 (**5.21**)	9.76 (7.58)	**7.08** (5.67)
motorcycle	**7.00** (5.91)	7.32 (6.75)	7.03 (**5.79**)
train	**6.59** (4.07)	9.16 (4.38)	9.54 (**3.93**)
mean	10.37 (5.57)	9.19 (7.06)	**7.79** (**5.52**)

Table 4: Average CD of multiple-unseen-category experiments on ShapeNet.

Category	PSGN	3D-LMNet	OptMVS	Ours-S_{dt}	Ours-S_{d+t}
chair	(8.98)	9.50 (**5.46**)	8.86 (7.23)	8.35 (7.40)	**7.28** (6.05)
sofa	(7.27)	**7.82** (**6.54**)	8.25 (8.00)	8.54 (7.18)	8.41 (6.83)
table	(8.84)	13.57 (**7.62**)	9.09 (8.88)	9.52 (9.06)	**8.53** (7.97)
mean-seen	(8.55)	9.73 (**6.04**)	8.75 (7.67)	8.54 (7.55)	**7.74** (6.53)
bed*	(9.23)	13.11 (9.02)	12.69 (9.01)	**10.91** (8.41)	11.04 (**8.19**)
bookcase*	(8.24)	8.32 (**6.64**)	**8.10** (8.35)	10.38 (9.72)	8.99 (8.44)
desk*	(8.40)	11.75 (7.72)	9.01 (8.50)	8.64 (8.16)	**7.64** (**7.18**)
misc*	(9.84)	13.45 (11.34)	13.82 (12.36)	12.58 (11.03)	**11.48** (**9.30**)
tool*	(11.20)	13.64 (9.09)	14.98 (11.27)	13.27 (11.70)	**12.18** (**9.02**)
wardrobe*	(7.84)	9.46 (**6.96**)	**6.96** (7.26)	9.15 (8.80)	8.33 (8.26)
mean-unseen	(8.81)	11.67 (8.22)	10.48 (8.83)	10.19 (8.86)	**9.57** (**8.07**)

Table 5: Average CD on both seen and unseen category on Pix3D dataset. All numbers are multiplied by 100. '*' indicates unseen category.

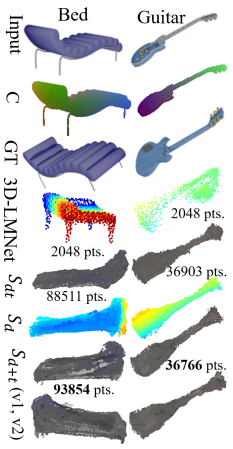


Figure 5: Results on unseen categories

6 before or after ICP. Qualitative completions are shown in Figure 5. Our methods perform reasonably well on the reconstruction of bed and guitar, while 3D-LMNet interprets the input as sofa or lamp.

4.3. Real-world Images from Pix3D

To test the generalization of our approach to real-world images, we evaluate our trained model on Pix3D [33]. We compare against the state-of-the-art methods, PSGN [9],

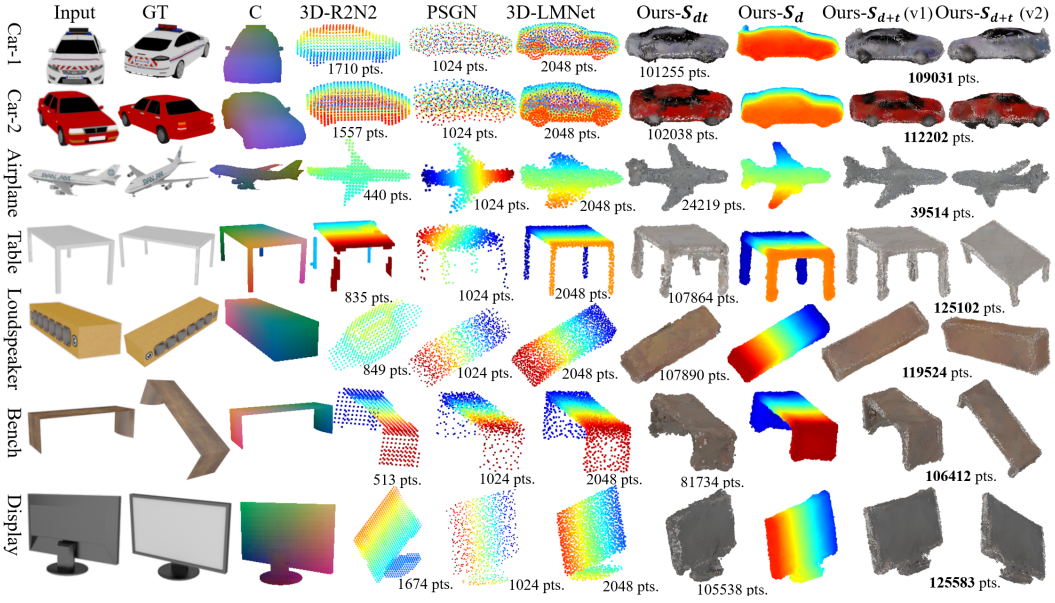


Figure 6: Reconstructions of the seen categories on ShapeNet dataset. 'C' is the generated object coordinate image, and 'GT' is another view of the target object. Ours-S_{dt} is generated by MTDCN, Ours-S_d and Ours-S_{d+t} are generated by MDCN.

Data	lmnet	ours
S-s	6.65	**4.63**
S-uns	10.15	**8.53**
P-s	8.85	**7.22**
P-uns	10.08	**9.59**

Table 6: 6D pose evaluations (ADD-S [41]). S: ShapeNet, P: Pix3D. '-s' and '-uns' are seen and unseen categories separately.

Data	lmnet	ours
S-s	0.42	**0.09**
S-uns	0.46	**0.29**
P-s	0.38	**0.16**
P-uns	0.30	**0.16**

Table 7: Relative CD improvements after ICP.

3D-LMNet [23] and OptMVS [40]. Following [23, 33], we uniformly sample 1024 points from meshes as ground truth point cloud to calculate CD, and remove images with occlusion and truncation. We have 4476 test images from seen categories, and 1048 from unseen categories.

Reconstructing novel objects from seen categories in Pix3D. We test the methods on 3 seen categories (chair, sofa, table) that co-occur in the 13 training sets of ShapeNet, and the results are shown in Table 5. Even on real-world data, our networks generate well aligned shapes, while other methods largely rely on ICP. Qualitative results are shown in Figure 7. Our method generates denser point clouds with reasonable texture. Besides more accurate shape alignment, our method also predicts better shapes, like the aspect ratio in the 'Table' example.

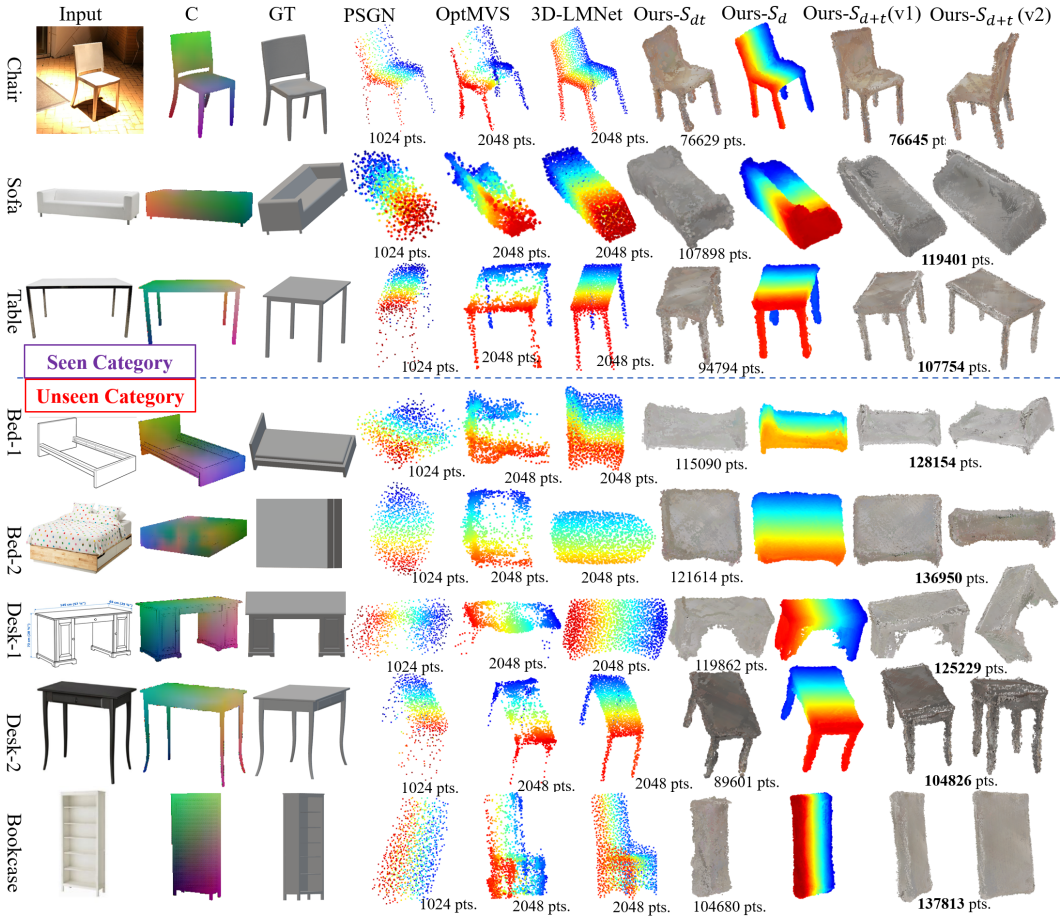
Reconstructing objects from unseen categories in Pix3D. We also test the pretrained models on 7 unseen categories (bed, bookcase, desk, misc, tool, wardrobe), and the results are shown in Table 5. Our methods outperform other approaches [9, 40, 23] in mean CD with or without ICP alignment. Figure 7 shows a qualitative comparison. For 'Bed-1' and 'Bed-2', our methods generate reasonable beds, while 3D-LMNet regards them as sofa or car-like objects. Similarly, we generate reasonable 'Desk-1' and recover the main structure of the input. For 'Desk-2', our method estimates the aspect ratio more accurately and recovers some details of the target object, like the curved legs. For 'Bookcase', ours generates a reasonable shape, while OptMVS or 3D-LMNet takes it as a chair.

4.4. More Experimental Results

Evaluations on F-score [19, 36] metric. Different from CD, F-score (the harmonic mean between precision and recall) is another evaluation metric for 3D reconstruction. In Table 8, we also provide the results on F-score calculated on 40K points. For F-score, the higher, the better. As suggested by [36], we take a distance threshold d of 0.5%, 0.8% and 1%. Our method outperforms 3D-LMNet [23] on both Pix3D and ShapeNet by a large margin.

Comparisons with GenRe [47] on generalization. GenRe has shown good generalization on reconstructing objects from unseen classes from single RGB images. We trained our model on the same dataset as [47], including ShapeNet cars, chairs, and airplanes, and tested on real images of beds, bookcases, desks, sofas, tables, and wardrobes from Pix3D. Results of AtlasNet [12], Shin et al. [31], and GenRe are from [47]. The CD reported is calculated on 1024 points. We outperform the three baseline methods: object-centered AtlasNet, and two viewer-centered methods [47, 31] on five out of seven classes.

6D Pose estimation. Besides CD, pose estimation should also be evaluated in the comparisons among different object-centered reconstruction methods. We compare 6D pose estimation accuracy by calculating the average closest point distance (ADD-S) [41]. The results are reported in Table 6 (S: ShapeNet, P: Pix3D), where our method outperforms 3D-LMNet on all test datasets. For ADD-S, the lower, the better. Note that since P and P' both are ground truth shapes under different poses, the ADD-S met-



d=0.5%	lmnet	ours
S-s	0.72	**2.55**
S-uns	0.37	**1.72**
P-s	0.41	**1.74**
P-uns	0.27	**1.39**
d=0.8%	lmnet	ours
S-s	1.55	**4.13**
S-uns	1.17	**2.95**
P-s	1.35	**2.92**
P-uns	0.91	**2.42**
d=1%	lmnet	ours
S-s	3.17	**4.86**
S-uns	1.89	**3.59**
P-s	2.17	**3.51**
P-uns	1.49	**2.92**

Table 8: F-score on ShapeNet and Pix3D. S: ShapeNet, P: Pix3D, s: seen, uns: unseen; lmnet: 3D-LMNet, ours: Ours-S_{d+t}. F-score is multiplied by 10.

Figure 7: Reconstructions on Pix3D. 'C' is object coordinate image, and 'GT' is ground truth.

	AtlasNet	Shin et al.	GenRe	Ours-S_{d+t}
chair	0.80	0.89	0.93	**0.57**
bed	1.14	1.06	1.13	**0.86**
bookc.	1.40	1.09	**1.01**	1.14
desk	1.26	1.21	**1.09**	1.18
sofa	0.95	0.88	0.83	**0.77**
table	1.34	1.24	1.16	**1.07**
wardr.	1.21	1.16	1.09	**1.06**

Table 9: Comparisons with GenRe on seen (chair) and unseen classes (the rest) from Pix3D. CD is multiplied by 10.

ric does not take the reconstruction completeness into account. More details about the experiment can be found in the supplementary material.

Ablation study 1: contributions of each reconstruction stage to the final shape. Considering both 2D-3D and view completion nets perform reconstruction, in Table 2, we compare the generated partial shape P_d with the completed shape Ours-S_{d+t} on the multiple-category and single-category tasks. For the former, the mean CD decreases from 10.58 to 3.91 after the second stage.

Ablation study 2: the impact of ICP alignment on the reconstruction results. Besides ADD-S, we also evaluate the pose estimations of 3D-LMNet and our methods by comparing the relative mean improvement of CD after ICP alignment in Table 7, which is calculated from Table 3, 4, 5. A bigger improvement means a worse alignment. Although the generated shapes of 3D-LMNet are assumed to be aligned with ground truth, its performance still relies heavily on ICP. But our methods rely less on ICP, which implies that our pose estimation is more accurate.

5. Conclusion

We propose a two-stage reconstruction method for 3D reconstruction from single RGB images by leveraging object coordinate images as intermediate representation. Our pipeline can generate denser point clouds than previous methods and also predict textures on multiple-category reconstruction tasks. Experiments show that our method outperforms the existing methods on both seen and unseen categories on synthetic or real-world datasets.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *ICML*, 2018. 2

[2] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI*, 1977. 5

[3] Paul Besl and H.D. McKay. A method for registration of 3-d shapes. ieee trans pattern anal mach intell. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14:239–256, 03 1992. 1

[4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 1, 5

[5] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5932–5941, 2019. 2

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. *ArXiv*, abs/2003.01456, 2020. 2

[7] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. *ArXiv*, abs/1604.00449, 2016. 2, 5

[8] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6545–6554, 2017. 2

[9] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2016. 2, 5, 6, 7

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017. 2, 5

[11] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T. Freeman, and Thomas A. Funkhouser. Learning shape templates with structured implicit functions. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7153–7163, 2019. 2

[12] Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. A papier-mache approach to learning 3d surface generation. In *CVPR*, pages 216–224, 06 2018. 7

[13] Christian Hane, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3D object reconstruction. In *International Conference on 3D Vision*, pages 412–420, 2017. 2

[14] Tao Hu, Zhizhong Han, Abhinav Shrivastava, and Matthias Zwicker. Render4completion: Synthesizing multi-view depth maps for 3d shape completion. *ArXiv*, abs/1904.08366, 2019. 2, 4

[15] Tao Hu, Zhizhong Han, and Matthias Zwicker. 3d shape completion with multi-view consistent inference. *ArXiv*, abs/1911.12465, 2019. 3

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 3

[17] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. *ArXiv*, abs/1803.07549, 2018. 2, 4

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014. 5

[19] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36:1–13, 07 2017. 7

[20] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3, 4, 5

[21] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. *The IEEE International Conference on Computer Vision*, 2019. 2

[22] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *SIGGRAPH '87*, 1987. 2

[23] Priyanka Mandikal, K L Navaneet, Mayank Agarwal, and R. Venkatesh Babu. 3d-lmnet: Latent embedding matching for accurate and diverse 3d point cloud reconstruction from a single image. *ArXiv*, abs/1807.07796, 2018. 2, 5, 7

[24] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019. 2

[25] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktash, and Anders P. Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *ArXiv*, abs/1901.06802, 2019. 2

[26] L. NavaneetK., Priyanka Mandikal, Varun Jampani, and R. Venkatesh Babu. Differ: Moving beyond 3d reconstruction with differentiable feature rendering. In *CVPR Workshops*, 2019. 2

[27] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 2

[28] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2

[29] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017. 2

[30] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]). 3

[31] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3061–3069, 2018. 7

[32] Srinath Sridhar, Davis Rempe, Julien Valentin, Sofien Bouaziz, and Leonidas J. Guibas. Multiview aggregation for learning category-specific shape reconstruction. In *Advances in Neural Information Processing Systems*, 2019. 2

[33] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B. Tenenbaum, and William T. Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 1, 5, 6, 7

[34] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Multi-view 3d models from single images with a convolutional network. In *ECCV*, 2015. 2, 5

[35] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3D outputs. In *IEEE International Conference on Computer Vision*, pages 2107–2115, 2017. 2

[36] Maxim Tatarchenko, Stephan Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, pages 3400–3409, 06 2019. 7

[37] Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. In *Computer Vision and Pattern Regognition*, 2018. 2

[38] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J. Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *CVPR*, 2019. 2

[39] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *European Conference on Computer Vision*, pages 55–71, 2018. 2

[40] Yi Wei, Shaohui Liu, Wang Zhao, Jiwen Lu, and Jie Zhou. Conditional single-view shape generation for multi-view stereo reconstruction. In *CVPR*, 2019. 5, 7

[41] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *ArXiv*, abs/1711.00199, 2017. 7

[42] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 2

[43] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *NIPS*, 2016. 2, 5

[44] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Interpretable unsupervised learning on 3d point clouds. *CoRR*, abs/1712.07262, 2017. 2

[45] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. *2018 International Conference on 3D Vision (3DV)*, pages 728–737, 2018. 2

[46] Wei Zeng, Sezer Karaoglu, and Theo Gevers. Inferring point clouds from single monocular images by depth intermediation. *ArXiv*, abs/1812.01402, 2018. 2, 3

[47] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2257–2268. Curran Associates, Inc., 2018. 2, 3, 7

[48] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and Bill Freeman. Visual object networks: Image generation with disentangled 3d representations. In *NeurIPS*, 2018. 2