NeRD: Neural 3D Reflection Symmetry Detector

Yichao Zhou
Univ. of California, Berkeley
zyc@berkeley.edu

Shichen Liu
Univ. of Southern California
liushich@usc.edu

Yi Ma
Univ. of California, Berkeley
yima@eecs.berkeley.edu

Abstract

*Recent advances have shown that symmetry, a structural prior that most objects exhibit, can support a variety of single-view 3D understanding tasks. However, detecting 3D symmetry from an image remains a challenging task. Previous works either assume the symmetry is given or detect the symmetry with a heuristic-based method. In this paper, we present NeRD, a **Ne**ural 3D **R**eflection Symmetry **D**etector, which combines the strength of learning-based recognition and geometry-based reconstruction to accurately recover the normal direction of objects' mirror planes. Specifically, we enumerate the symmetry planes with a coarse-to-fine strategy and find the best ones by building 3D cost volumes to examine the intra-image pixel correspondence from the symmetry. Our experiments show that the symmetry planes detected with our method are significantly more accurate than the planes from direct CNN regression on both synthetic and real datasets. More importantly, we also demonstrate that the detected symmetry can be used to improve the performance of downstream tasks such as pose estimation and depth map regression by a wide margin over existing methods. The code of this paper has been made public at* https://github.com/zhou13/nerd.

1. Introduction

Recovering the 3D orientation of objects in an image is a fundamental problem in 3D vision, which plays important roles in tasks such as robotics, autonomous driving, virtual reality (VR), augmented reality (AR), and 3D scene understanding. Traditionally, such a problem is hard to solve. Researchers can to RGB-D input captured with time-of-flight cameras or structured light [5, 27, 29]. Unfortunately, depth cameras often have limited range and can be interfered with by other light sources, and the requirement of owning a depth camera is inconvenient for average users, which severely restricts its applications.

Recent advances in convolutional neural networks in object detection and instance segmentation have shown good potential in inferring object-level information from RGB images by leveraging supervised learning. Nowadays, single-view neural network-based methods are able to predict the object pose under different settings. Some work explores the *instance-level 3D pose estimation* problem [21, 28, 32] in which the CAD models of the objects are known beforehand. However, these settings are rather limited because in practice we do not have CAD models for many objects. Therefore, other work tries to tackle the *category-level 3D pose estimation* problem [4, 25, 39] without relying on the exact CAD models of objects. Unlike the cases where either depth information or CAD models are available, previous single-view category-level 3D pose estimation methods can hardly exploit the geometric constraints between the input RGB image and the 3D shape and predict the pose solely by interpolating the training data. Hence, such formulation is ill-posed, which leads to inaccurate pose recovery [31].

To address this difficulty, we identify a structure that commonly exists in man-made objects, the *reflection symmetry*, as a geometric connection between the object poses and the

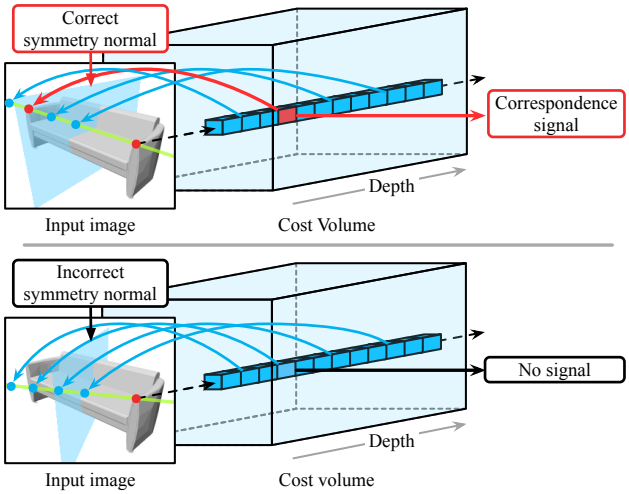


Figure 1: Illustration of the symmetry detection process in NeRD. For each pixel, we enumerate its depth and warp features along the line according to the symmetry plane hypothesis. If the hypothesis is correct, there should be matched features for most of the pixels.

images. We observe that the canonical space of objects often is determined by aligning the Y-Z plane to the symmetry planes of objects [2, 30], so the normal direction of the symmetry plane encodes most of the geometric information regarding the pose of the object. To this end, we propose the NeRD network to detect the reflection symmetry from RGB images. *NeRD* combines the strength of learning-based recognition and geometry-based reconstruction methods. It first enumerates the normal direction of the mirror plane from the image with a coarse-to-fine strategy and then verifies their correctness with a geometric-based neural network. More specifically, we incorporate the concept of reflection symmetry into deep networks through plane-sweep cost volumes built from features of corresponding pixels, as shown in Figure 1. This allows us to accurately recover the normal direction of the mirror plane under the principle of shape-from-symmetry [13].

The network (see Figure 3) consists of a backbone feature extractor, a differentiable warping module for building the 3D cost volumes, and a cost volume network. This framework naturally enables neural networks to utilize the information from corresponding pixels of reflection symmetry inside a single image. We evaluate our method on the ShapeNet dataset [2] and Pix3D dataset [30]. Extensive comparisons and analysis show that by detecting and utilizing intra-image pixel correspondence from reflection symmetry, our method has better accuracy for recovering the normal direction of the symmetry plane and hence the object pose, even when the object is not perfectly symmetric.

Our main contributions are summarized as below:

- we identify the problem of learning neural 3D reflection symmetry detector, in which the intra-image pixel correspondence of symmetry can be utilized for accurate plane normal estimation;
- we propose a novel framework that leverages single-view dense feature matching to estimate symmetry planes, significantly outperforming previous methods;
- we show that the learned symmetry planes benefit tremendously a variety of downstream tasks, including single-view pose recovery and depth estimation.

2. Related Work

3D Reflection Symmetry. For many years, scientists from vision science and psychology have found that symmetry plays an important role in the human vision system [33, 35]. People have exploited different kinds of symmetry for tasks such as texture impainting [20], unsupervised shape recovering [38], and image manipulation [45]. Researchers have utilized the correspondences of symmetry to reconstruct shapes in different representations, such as points [13], curves [14], and recent deep implicit fields [41]. However, these methods either assume that the input camera pose or the symmetry plane is given or require its correspondence points. This is because detecting 3D symmetry from a single view is challenging.

(a) 2D reflection symmetry (b) 3D reflection symmetry

Figure 2: Examples of 2D and 3D reflection symmetry reconstruction. 2D symmetries are not helpful for 3D understading due to lack of perspective distortion.

Symmetry Detection. [8] is a recent survey of existing 2D/3D symmetry detection methods. On one hand, most of the geometry-based symmetry detection methods use hand-crafted features and only work for 2D planar and front-facing objects [19, 22, 44] as shown in Figure 2a. The extracted 2D symmetry axes and correspondences cannot provide enough geometric cues for depth reconstruction. In order to make reflection symmetry useful for depth reconstruction, it is necessary to detect the 3D mirror plane and corresponding points of symmetric objects (Figure 2b) from perspective images. On the other hand, recent single-image processing neural networks [2, 15, 36, 42, 46] can approximately recover the camera orientation with respect to the canonical pose, which gives a mirror plane of symmetry. However, the camera poses from those data-driven networks are not accurate enough [9], because they cannot exploit the geometric constraints of symmetry. To remedy the above issues, our NeRD tries to take the best of both worlds. The proposed method first detects the 3D mirror plane of a symmetric object from an image and then recovers the depth map by finding the pixel-wise correspondence with respect to the symmetry plane, all of which are supported with geometric principles. Our experiment (Section 4) shows that NeRD is indeed much more accurate for 3D symmetry plane detection, compared to previous learning-based methods [40, 46].

Learning-Based Single-Image 3D Understanding. Inspired by the success of CNNs in image classification and object detection, multiple single-view learning-based 3D understanding tasks have been explored, including depth estimation [3, 7], camera pose recovery, etc. Although these methods demonstrate promising results on benchmark datasets, the inferred results are not accurate enough for most subsequent 3D reconstruction purposes. To alleviate this issue, our method leverages the symmetry prior by matching pixel-level features for accurate single-view 3D understanding.

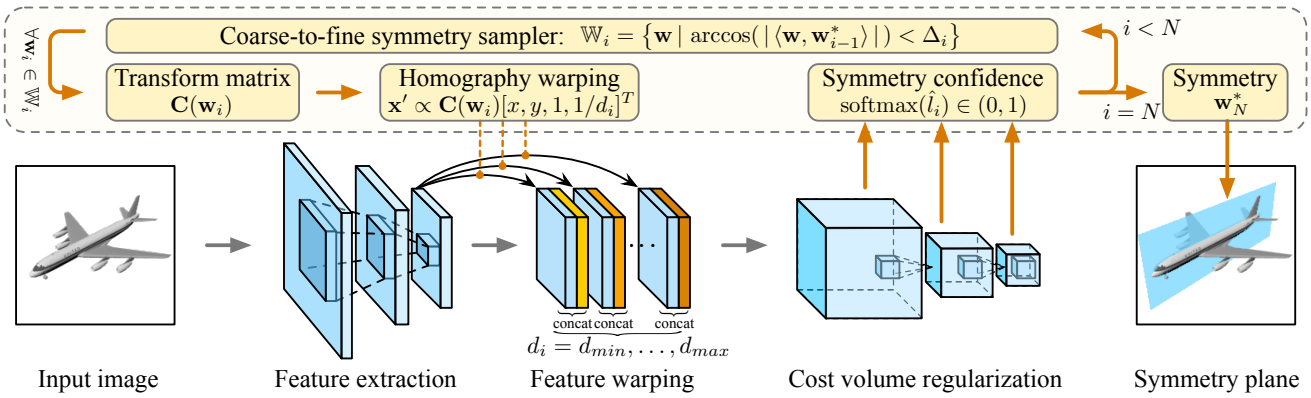


Figure 3: Overview of the NeRD. During inference, the coarse-to-fine symmetry sampler gives a list of candidate normal directions of the symmetry plane. For each candidate symmetry plane, a warping transformation matrix $\mathbf{C}$ is computed according to Equation (5). Input images first go through the feature extraction (backbone) network. Features are then warped by a warping module based on the symmetry transformation $\mathbf{C}$ and depth d_i. A cost volume is constructed by fusing the warped features and feeding into a 3D convolutional neural network for refinement. The final confidence of each symmetry plane is predicted by aggregating the resulting depth probability tensor.

3. Methods

3.1. Camera Model and 3D Symmetry

Let $\mathbb{O} \subset \mathbb{R}^4$ be the set of points in the homogeneous coordinate that are on the surface of an object. If we say $\mathbb{O}$ admits the *symmetry*[1] with respect to a rigid transformation $\mathbf{M} \in \mathbb{R}^{4\times4}$, it means that

$$\forall \mathbf{X} \in \mathbb{O} : \mathbf{MX} \in \mathbb{O}, \quad \text{and} \quad \mathcal{F}(\mathbf{X}) = \mathcal{F}(\mathbf{MX}), \tag{1}$$

where $\mathbf{X}$ is homogeneous coordinates of a point on the surface of the object, $\mathbf{MX}$ is the corresponding point of $\mathbf{X}$ with respect to the symmetry, and $\mathcal{F}(\cdot)$ represents the surface properties at a given point, such as the surface material and texture. For example, if an object has reflection symmetry with respect to the Y-Z plane in the world coordinate, then we have its transformation $\mathbf{M}_x = \text{diag}(-1, 1, 1, 1)$. Figure 2 shows an example of 3D reflection symmetry.

Given two 3D points $\mathbf{X}, \mathbf{X}' \in \mathbb{O}$ in the homogeneous coordinate that are associated by the symmetry transform $\mathbf{X}' = \mathbf{MX}$, their 2D projections $\mathbf{x}$ and $\mathbf{x}'$ must satisfy the following conditions:

$$\mathbf{x} = \mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{X}/d, \quad \text{and} \quad \mathbf{x}' = \mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{X}'/d'. \tag{2}$$

Here, we keep all vectors in $\mathbb{R}^4$. $\mathbf{x} = [x, y, 1, 1/d]^T$ and $\mathbf{x}' = [x', y', 1, 1/d']^T$ represent the 2D coordinates of the points in the pixel space, d and d' are the depth in the camera space, $\mathbf{K} \in \mathbb{R}^{4\times4}$ is the camera intrinsic matrix, and $\mathbf{R}_\mathbf{t} = \left[\begin{smallmatrix}\mathbf{R} & \mathbf{t}\\ 0 & 1\end{smallmatrix}\right]$ is the camera extrinsic matrix that rotates and translates the coordinate from the object space to the camera space.

[1]An object might admit multiple symmetries. For example, a rectangle has two reflective symmetries and one rotational symmetry. We here only consider the principle symmetry.

From Equation (2), we can derive the following constraint for their 2D projections $\mathbf{x}$ and $\mathbf{x}'$:

$$\mathbf{x}' \propto \underbrace{\mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{M}\mathbf{R}_\mathbf{t}{}^{-1}\mathbf{K}^{-1}}_{\mathbf{C}}\mathbf{x} \doteq \mathbf{C}\mathbf{x}. \tag{3}$$

We use the proportional symbol here as the 3rd dimension of $\mathbf{x}'$ can always be renormalized to one so the scale factor does not matter. The constraint in Equation (3) is valuable to us because the neural network now has a geometrically meaningful way to check whether the estimated depth d is reasonable at (x, y) by comparing the image appearance at (x, y) and (x', y'), where (x', y') is computed from Equation (3) given x, y, and d. If d is a good estimation, the two corresponding image patches should be similar due to $\mathcal{F}(\mathbf{X}) = \mathcal{F}(\mathbf{X}')$ from the symmetry constraint in Equation (1). This is often called *photo-consistency* in the literature of multi-view steropsis [10].

An alternative way to understand Equation (3) is to substitute $\mathbf{X}' = \mathbf{MX}$ into Equation (2) and treat the later equation as the projection from another view. By doing that, we reduce the problem of shape-from-symmetry to two-view stereopsis, only that the stereo pair is in special positions.

Reflection Symmetry in 3D. Equation (3) gives us a generalized way to represent any types of symmetry with matrix $\mathbf{C} = \mathbf{K}\mathbf{R}_\mathbf{t}\mathbf{M}\mathbf{R}_\mathbf{t}{}^{-1}\mathbf{K}^{-1}$. For reflection symmetry, a more intuitive parametrization is to use the equation of the symmetry plane in the camera space. Let $\tilde{\mathbf{x}} \in \mathbb{R}^3$ be the coordinate of a point on the symmetry plane in the camera space. The equation of the symmetry plane can be written as

$$\mathbf{w}^T\tilde{\mathbf{x}} + 1 = 0, \tag{4}$$

where we use $\mathbf{w} \in \mathbb{R}^3$ as the parameterization of symmetry.

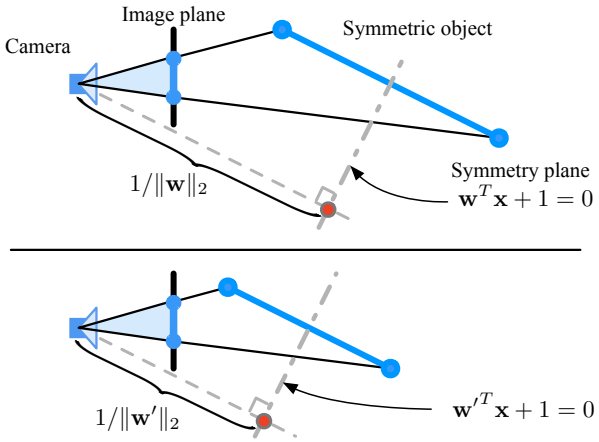


Figure 4: Illustration of scale ambiguity. We display two scenes that only differ by a scale c. The images of the two scenes are exactly the same, but the distances between the origin and two symmetry planes are different, i.e., $\|\mathbf{w}\|_2 = c\|\mathbf{w}'\|_2$.

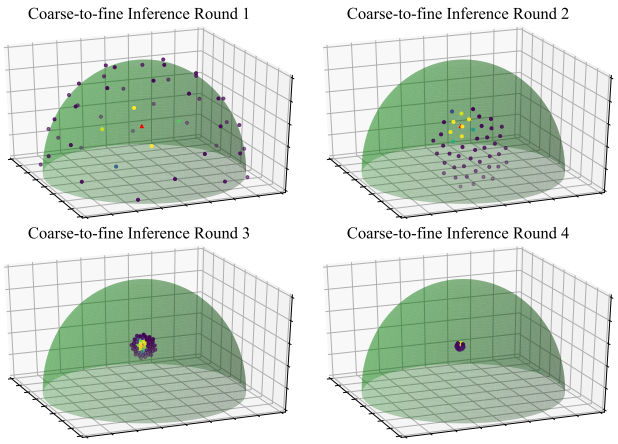


Figure 5: Illustration of the process of coarse-to-fine inference. We show the sampled normal direction in a 4-round coarse-to-fine inference. The color of points represents the scores from symmetry confidence network.

The relationship between $\mathbf{C}$ and $\mathbf{w}$ is

$$\mathbf{C}(\mathbf{w}) = \mathbf{K}\left(\mathbf{I} - \frac{2}{\|\mathbf{w}\|_2^2}\begin{bmatrix}\mathbf{w}\\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{w}^T & \mathbf{1}\end{bmatrix}\right)\mathbf{K}^{-1}. \quad (5)$$

We derive Equation (5) in the supplementary material. The goal of reflection symmetry detection is to recover $\mathbf{w}$ from images.

On the first impression, one may wonder why $\mathbf{R_t}$ (i.e., camera poses) in Equation (3) has 6 degrees of freedoms (DoFs) while $\mathbf{w}$ only has 3. This is due to the specialty of reflection symmetry. Rotating the camera with respect to the normal of the symmetry plane (1 DoF) and translating the camera along the symmetry plane (2 DoFs) cannot change the relative pose of the camera with respect to the symmetry plane. Therefore the number of DoFs in reflection symmetry is indeed $6-1-2=3$.

Scale Ambiguity. Similar to structure-from-motion in which it is impossible to determine the absolute size of scenes [23], shape-from-symmetry also has a scale ambiguity. This is demonstrated in Figure 4. In the case of reflection symmetry, we cannot determine the value of $\|\mathbf{w}\|_2$, i.e., the symmetry plane's distance from the origin, from a single image without relying on size priors, as it is always possible to scale the scene by a constant (and thus scale $\|\mathbf{w}\|_2$) without affecting images. Therefore, we fix $\|\mathbf{w}\|_2$ to be a constant and leave the ambiguity as it is. In other words, NeRD is designed only to recover the normal direction of the symmetry plane. For real-world applications, this scale ambiguity can be resolved when the object size or the distance between the object and the camera is known.

3.2. Overall Pipeline of NeRD

Motivation. Section 3.1 provides us a geometric way to verify whether a given $\mathbf{w}$ is valid: For each pixel (x, y), we check if there exists a d so that the image feature at (x, y) and its mirror point (x', y') are similar, where (x', y') are computed with Equation (3). If $\mathbf{w}$ is correct, then for pixels whose mirror parts are not occluded, we should be able to find their corresponding pixels that are similar to themselves. To utilize such an idea, we turn the problem of regressing $\mathbf{w}$ into a classification problem: We first enumerate possible plane normal directions and use a neural network to verify whether these directions are closed to the real symmetry planes or not.

Methods. Figure 3 illustrates the overall pipeline of NeRD during inference. For each input image, we compute its 2D feature map (Section 3.3) and generate a list of candidate normal directions of its symmetry plane. For each candidate normal $\mathbf{w}$, we use it to warp the 2D feature map and construct an initial 3D cost volume (Section 3.4) for photo-consistency matching. After that, the cost volume network (Section 3.5) converts the cost volume tensor into a confidence value. We pick $\mathbf{w}$ with the highest confidence as the resulting normal direction of the symmetry plane.

A brute-force enumeration of $\mathbf{w}$ is slow, especially when high precision is needed. To accelerate it, NeRD uses a coarse-to-fine strategy, which we will describe in detail in Section 3.6. Figure 5 illustrates the process of coarse-to-fine inference. In ith round of inference, the coarse-to-fine sampler samples N candidates symmetry plane $\{\mathbf{w}_i^k\}_{k=1}^K$ uniformly and evaluate their confidence with our neural network. Then, we find the pose $\mathbf{w}_i^*$ with the highest confidence score and limit the symmetry sampler to the nearby region around it. This process is repeated until we achieve the desired accuracy.

3.3. Backbone Network

The goal of the *backbone network* is to extract 2D features from images. We use a modified ResNet-like network as our backbone. To reduce the memory footprint, we first down-sample the image with a stride-2 5×5 convolution. After that, the network has 8 *basic blocks* [12] with ReLU activation. The 5th basic block uses stride-2 convolution to further downsample the feature maps. The number of channels is 64. The output feature map $\mathbf{F}$ has dimension $\lfloor \frac{H}{4} \rfloor \times \lfloor \frac{W}{4} \rfloor \times 64$. The network structure diagram is shown in the supplementary materials.

3.4. Feature Warping Module

The function of the *feature warping module* is to construct the initial 3D cost volume tensor $\mathbf{V}(x, y, d)$ for photo-consistency matching. We discretize d so that $d \in \mathcal{D} = \{d_{\min} + \frac{i}{D-1}(d_{\max} - d_{\min}) \mid i = 0, 1, \ldots, D-1\}$ to make the 3D cost volume homogeneous to 3D convolution, in which $d_{\min}$ and $d_{\max}$ is the minimal and maximal depth we want to predict and D is the number of sampling points for depth. As mentioned in Section 3.1, the correctness of d at (x, y) correlates with the appearance similarity of the image patch at pixels represented by $\mathbf{x}$ and $\mathbf{Cx}$. Therefore, we set $\mathbf{V}$ by concatenating the backbone features at these two locations, i.e.,

$$\mathbf{V}(x, y, d) = \Big[\mathbf{F}(x, y),\ \mathbf{F}(x', y')\Big], \tag{6}$$

where $[\,x', y', 1, 1/d'\,]^T \propto \mathbf{C}[x, y, 1, 1/d]^T$, i.e., (x', y') being the projection of the mirror point of the pixel (x, y) assuming its depth is d. Here $\mathbf{F}$ is the backbone feature, and $\mathbf{C}$ is computed from the sampled symmetry plane $\hat{\mathbf{w}}$. We apply bilinear interpolation to access the features at non-integer coordinates. The dimension of the cost volume tensor is $\lfloor \frac{H}{4} \rfloor \times \lfloor \frac{W}{4} \rfloor \times D \times 32$.

3.5. Cost Volume Network

The goal of the cost volume network is to turn the initial 3D cost volume tensor $\mathbf{V}$ from the feature warping module into a confidence value representing whether the current pose $\mathbf{w}$ is close to the ground truth. It may also predict a depth probability tensor $\mathbf{P}(x, y, d) := \Pr[\mathbf{D}(x, y) = d]$ for downstream tasks (Section 3.7). The cost volume network uses matrix multiplication on the channel dimension to check for the photo-consistency on $\mathbf{V}$. However, the initial cost volume aggregated from image features can be noisy. Thus, we use a network consists of multiple 3D convolution layers that are capable of regularizing the cost volume information. We aggregate the multi-resolution encoder features with max-pool operators and then apply the sigmoid function to normalize the confidence values into $[0, 1]$.

3.6. Symmetry Sampler

Inference. As shown in Figure 5, the symmetry sampler uniformly samples $\{\mathbf{w}_i^k\}_{k=1}^K$ from $\mathbb{W}_i \subset \mathbb{R}^3$ using the Fibonacci lattice [11, 47], where $\mathbb{W}_i$ is the sampling space of the ith round of inference. In the first round, candidates are sampled from the surface of a unit hemisphere. For the following rounds, we set $\mathbb{W}_i = \{\mathbf{w} \in \mathbb{S}^2 \mid \arccos(|\langle \mathbf{w}, \mathbf{w}_{i-1}^* \rangle|) < \Delta_i\}$ to be a spherical cap, where $\mathbf{w}_{i-1}^*$ is the optimal $\mathbf{w}$ from the previous round and Δ_i is a hyper-parameter.

Training. During training, we sample symmetry planes for each image according to the hyper-parameter Δ_i. For the ith level, symmetry candidates are sampled from $\{\hat{\mathbf{w}} \in \mathbb{S}^2 \mid \arccos(|\langle \mathbf{w}, \hat{\mathbf{w}} \rangle|) \le \Delta_i\}$, where $\mathbf{w}$ is the ground truth symmetry pose. We also add a random sample $\hat{\mathbf{w}} \in \mathbb{S}^2$ to reduce the sampling bias. For each sampled $\hat{\mathbf{w}}$, its confidence labels is $l_i = \mathbf{1}[\arccos(|\langle \mathbf{w}, \hat{\mathbf{w}} \rangle|) < \Delta_i]$ for the ith level. The training error could be written as

$$L_{\text{cls}} = \sum_i \text{BCE}(\hat{l}_i, l_i),$$

where BCE represents the binary cross entropy error, and $\hat{l}_i$ is predicted confidence of $\hat{w}$ for the ith level in the coarse-to-fine inference.

3.7. Applications

In this section, we introduce some potential applications of reflection symmetry detection that benefit from the accurate normal direction of the reflection symmetry plane.

Pose Recovery. In the problem of pose recovery, the goal is to find the pose of an object from an RGB image, in which people normally set up the canonical space of objects so that objects are symmetric with respect to the X-Z plane or the Y-Z plane [2]. Because NeRD is able to pinpoint the normal direction of the symmetry plane, we can accurately determine 2 DoFs of the 6 DoFs pose with our geometry-based method. For the rest 4 DoFs, we can still resort to data-driven approaches (e.g., direct regression) with neural networks.

Depth Estimation. As we construct cost volumes (i.e., depth probability tensors) in the symmetry detection pipeline (Section 3.5), it is straightforward to use it for a geometry-based depth estimation. With the estimated $\mathbf{w}^*$, we compute the expectation of depth from the probability tensor $\mathbf{P}$ as the depth map prediction $\hat{\mathbf{D}}$. This is sometimes referred as *soft argmin* [17]. Mathematically, we have

$$\hat{\mathbf{D}}(x, y) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} d\mathbf{P}(x, y, d). \tag{7}$$

We rescale the ground truth depth according to $\|\hat{\mathbf{w}}\|_2$ and add an additional ℓ_1 term to the training loss as the supervision

of depth:

$$L_{\text{dpt}} = \frac{1}{n} \sum_{x,y} \left| \hat{\mathbf{D}}(x,y) - \mathbf{D}(x,y) \right|, \tag{8}$$

where n is the number of pixels.

4. Experiments

4.1. Datasets

We conduct experiments on the synthetic ShapeNet dataset [2] and real-world Pix3D dataset [30], in which models have already been processed so that in their canonical poses the Y-Z plane is the plane of the reflection symmetry.

ShapeNet. We use the same camera pose, intrinsic, and train/validation/test split from a 13-category subset of the dataset as in R2N2 and others [6, 16, 37] to make the comparison easy and fair. We exclude the lamp category as it contains many asymmetric objects. We use Blender to render the images with resolution 256×256.

Pix3D. Pix3D [30] is a real-world dataset containing image-shape pairs with 2D-3D registrations. To demonstrate the versatility of NeRD, we test NeRD on the Pix3D dataset. We assume that the bounding boxes of objects have been detected, and we use them to crop the images for removing the background while maintaining the aspect ratio. We rescale the resulting images to 256×256 and adjust the camera intrinsic matrix $\mathbf{K}$ accordingly and reject images extraordinary with focal lengths and depth values. We randomly split the remaining data into train and test sets, which contain 5285 and 588 images, respectively.

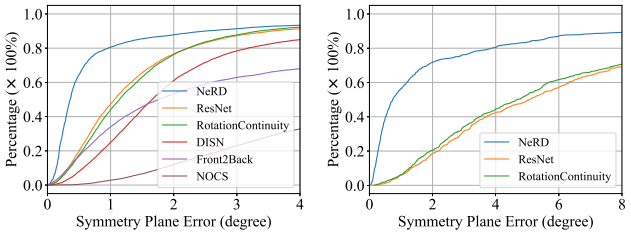
4.2. Implementation Details

We implement NeRD in PyTorch. We use the plane $x = 0$ in the object space as the ground truth symmetry plane because it is explicitly aligned for each model by authors of ShapeNet. We set $d_{\min}$ and $d_{\max}$ according to the depth distribution of the dataset, and use $D = 64$ for the depth of the cost volume. We use $N = 4$ rounds in the coarse-to-fine inference, in each of which $K = 32$ normal directions are sampled. We choose $\Delta = [20.7°, 6.44°, 1.99°, 0.61°]$ according to the gap between near directions on the Fibonacci lattice. Our experiments are conducted on two NVIDIA RTX 2080Ti GPUs. We use Adam [18] for training. The learning rate is set to 3×10^{-4} and batch size is set to 16 per GPU. We train the NeRD for 40 epochs and decay the learning rate by a factor of 10 at the 30th epoch. The overall inference speed is about 1 image per second per GPU.

Metrics. To better understand the performance of symmetry detection, we show two forms of metrics. We plot a performance curve for each detector-dataset pair, in which the x-axis represents the angle accuracy and the y-axis represents the proportion of the data whose error is less than that. We also report quantitative metrics, including the median and mean of the angle difference, and the percentages of testing images whose error is smaller than $0.5°$, $1.0°$, $2.0°$, and $4.0°$, for the ease of comparison.

	backbone (sec 3.3)	cost volume (sec 3.5)	feature warping var	avg	cat	error metrics avg	med	$<1°$	$<2°$
a		✓			✓	7.12°	**0.54°**	66.8%	77.2%
b	✓				✓	6.82°	0.99°	50.1%	70.1%
c	✓	✓	✓			6.33°	0.57°	68.1%	81.5%
d	✓	✓		✓		6.41°	0.66°	63.7%	77.7%
e	✓	✓			✓	**5.41°**	0.56°	**68.2%**	**81.5%**

Table 1: Ablation study of 3D reflection symmetry detection on ShapeNet.



(a) Curves on ShapeNet (synthetic) (b) Curves on Pix3D (real-world)

Figure 6: Performance curves of symmetry detection and camera pose recovery networks. *Higher is better.*

4.3. Ablation Studies

We conduct ablation studies to justify each component in NeRD. In Table 1, we analyze the function of three main components of NeRD: the 2D backbone network (Section 3.3), feature warping module (Section 3.4), and the cost volume network (Section 3.5). The second column of Table 1 represents whether we use the feature from the 2D backbone or just RGB values with a single 1×1 convolution to construct the cost volume. Comparing (a) and (e), we find that removing the 2D backbone degrades the performance, especially at the region $> 2°$. We think this is because the 2D backbone network increases the receptive field, which makes our method more robust. The third column represents whether we want to replace the cost volume network with a simple max-pool layer. Results in (b) and (e) show that the cost volume network is the key component for an accurate symmetry detector. Finally, we study the different pooling schemes in the feature warping module. From (c), (d), and (e), we find that the feature concatenation and variance pooling gives the best results, while the average pooling performs poorly in the high-precision region ($< 1°$). This matches our intuition in Section 3.1 that NeRD compares the feature to check photo-consistency.

4.4. Symmetry Detection on Synthetic Datasets

Baselines. We briefly introduce some state-of-the-art single-view symmetry detection and pose estimation baselines. Probably the plainest way to estimate the 3D symmetry plane $\mathbf{w}$ is direct regression [9]. We implement it

	avg	med	$<0.5°$	$<1.0°$	$<2.0°$
DISN [40]	2.80°	1.65°	7.96%	24.9%	61.0%
ResNet [12]	2.08°	1.06°	19.7%	47.3%	76.6%
RotationContinuity [46]	1.94°	1.14°	17.6%	43.9%	76.2%
Front2Back [42]	9.41°	1.76°	16.8%	34.0%	53.2%
NOCS [36]	9.95°	6.18°	0.39%	2.83%	11.9%
NeRD	**1.58°**	**0.36°**	**64.5%**	**80.6%**	**87.8%**

Table 2: Performance of symmetry detection and object pose recovery algorithms on ShapeNet. We report the normal direction error of the predicted symmetry planes. We note that NOCS [36] requires ground truth object shapes as input.

	avg	med	$<1.0°$	$<2.0°$	$<4.0°$
ResNet [12]	8.01°	5.06°	5.78%	18.5%	42.3%
RotationContinuity [46]	7.91°	4.67°	6.12%	20.4%	44.3%
NeRD	**3.37°**	**0.73°**	**56.3%**	**71.9%**	**80.4%**

Table 3: Performance of symmetry detection and object pose recovery algorithms on real-world dataset Pix3D [30]. We report the normal direction error of the predicted symmetry planes.

with ResNet-50 [12] and train it with L1 loss. RotationContinuity [46] identifies a 6D representation of rotation which they claim is more suitable for learning. We also implement it and train with L1 loss. DISN [40] also implements its 6D representation for ShapeNet but is trained with L2 loss. We report the performance of their pre-trained model. Front2Back [42] is a recent work that detects the 3D symmetry plane, which first predicts a depth map and then fits the symmetry plane with a traditional method [24]. We report the performance of their results provided by the authors. NOCS [36] predicts a coordinate of normalized object coordinate space for each pixel and recovers the pose with Umeyama algorithm [34]. Following their paper, we train the NOCS estimator on ShapeNet and use their code to recover the orientation of objects from prediction.

Results. Table 2 and Figure 6a show the comparison on ShapeNet. By utilizing geometric cues from symmetry, our approach significantly outperforms previous state-of-the-art methods. The performance gap is larger in the region of higher precision ($< 1°$). For example, NeRD can achieve an accuracy of $0.5°$ on about 70% of testing cases, while direct regression with ResNet and other baselines can only reach that on less than 20% of data. Such phenomena indicate that the intra-image correspondence does help algorithms recover symmetry planes more accurately, while naive CNNs can only roughly predict the plane normal by interpolating from training data. We also find that end-to-end approaches that directly predict the symmetry plane (ResNet, DISN, NeRD, etc) performs better than the methods which require heavier post-processing (NOCS and Front2Back). This hints us that using a loss function that is more directly related to the goal has an advantage.

	absRel	sqRel	rmse	mae	$<\delta^1$	$<\delta^2$	$<\delta^3$
DORN [7]	0.028	0.0014	0.026	0.020	30.8%	54.1%	69.0%
GeoNet [43]	0.028	0.0013	0.025	0.019	29.7%	53.4%	69.2%
Hourglass [26]	0.026	0.0012	0.024	0.018	33.0%	56.9%	71.5%
DenseDepth [1]	0.024	0.0011	0.022	0.017	36.3%	60.5%	74.6%
Pixel2Mesh [37]	0.102	0.0546	0.032	0.073	28.6%	49.2%	62.3%
DISN [40]	0.040	0.0030	0.038	0.028	24.0%	43.4%	57.8%
NeRD	**0.019**	**0.0009**	**0.021**	**0.011**	**49.5%**	**71.9%**	**82.3%**
NeRD*	0.015	0.0006	0.018	0.011	60.2%	78.7%	86.5%

Table 4: Quantitative comparison of NeRD and other baseline methods on ShapeNet. We set $\delta = 1.01$. NeRD* uses the ground truth symmetry plane as input.

(a) ResNet (b) NeRD (a) ResNet (b) NeRD

Figure 7: Qualitative results on the task of symmetry detection on Pix3D. We show the detected symmetry planes from ResNet and our NeRD. Errors of symmetry planes (pixels between the predicted and ground truth planes) are highlighted.

4.5. Symmetry Detection on Real-World Datasets

Table 3 and Figure 6b show the comparison on the real-world Pix3D dataset. NeRD outperforms the naive CNN regression, and the margin is even bigger compared to the results on ShapeNet. We hypothesize that this is because images in Pix3D use a larger number of camera configurations, including different focal lengths and object positions with respect to the focal center, while the dataset has fewer images. This requires more generalizability from the algorithms. Our geometry-based approach shines here because it can rely on the cues from correspondence to find the symmetry planes. Also, it is hard for naive convolutional neural networks to make use of the camera intrinsics, which varies from images to images, unlike ShapeNet. In contrast, NeRD uses camera intrinsic matrices in the feature warping module (Section 3.4) and thus generalizes better when dealing with different camera configurations.

4.6. Depth Estimation as an Application

As mentioned in Section 3.7, NeRD can be modified as a symmetry-guided depth estimator. We compare it with popular monocular depth estimation networks [1, 7, 26, 43]

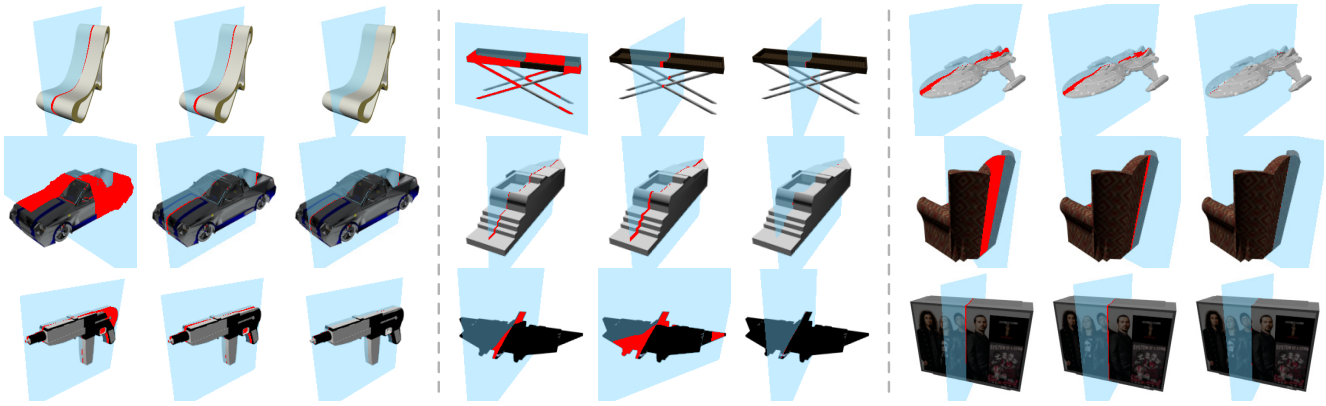


Figure 8: Qualitative results on ShapeNet. Errors of symmetry planes (pixels between the predicted and ground truth planes) are highlighted.

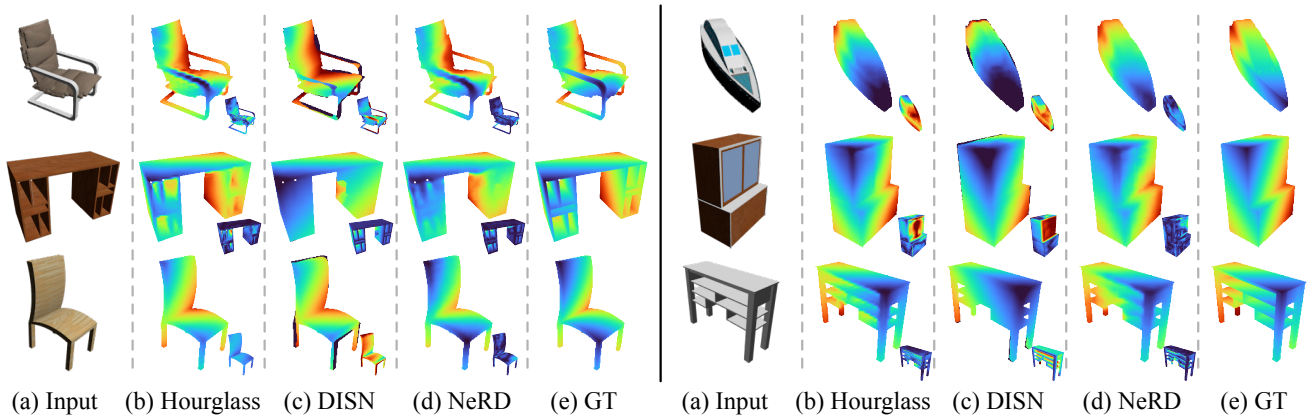


Figure 9: Qualitative results on the task of of depth estimation. We visualize the depth maps from Pixel2Mesh [37], DISN [40], and our NeRD on ShapeNet. The per-pixel errors are plotted at the lower right corner. Bluish color represents smaller values for error and depth.

and shape reconstruction networks [37, 40]. The results on the task of *depth estimation* are shown in Table 4. NeRD outperforms both monocular depth estimation networks and shape reconstruction networks. Besides, NeRD*, the variant of NeRD that uses the ground truth symmetry plane instead of the one predicted in coarse-to-fine inference, only slightly outperforms the standard NeRD. These behaviors indicate that detecting symmetry planes and incorporating photo-consistency priors of reflection symmetry into the neural network makes the task of single-view reconstruction less ill-posed and thus can improve the performance.

4.7. Visualization

We visualize our results in Figure 7 and Figure 8. We have the following observations: 1) our method outperforms previous methods on unusual objects, e.g. chairs in atypical shapes. This indicates that previous learning-based methods need to extrapolate from seen patterns and cannot generalize to unusual images well, while our method relies more on geometry cues from symmetry, a more reliable source of information for 3D understanding. 2) NeRD gives accurate symmetry planes even on challenging camera poses such as the orientation from the back of chairs. We believe that this is because geometric information from correspondence helps to pinpoint the normal of symmetry planes.

In Figure 9, we show sampled results of depth maps. Visually, NeRD gives the most accurate results among all the tested methods. For example, it can capture the details of desk frames and the shapes of ship cabins. Results from the hourglass network are also sharp but are less accurate, which may be the sign of overfitting. In the region such as the chair armrests and table legs, NeRD can recover the depth more accurate compared to the baseline methods. This is because, for NeRD, pixel-matching based on photo-consistency in those areas is easy and can provide a strong signal.

Acknowledgement

This work is supported by the research grant from Sony, the ONR grant N00014-20-1-2002, and the joint Simons Foundation-NSF DMS grant 2031899. We also thank Li Yi from Google Research for his comments.

References

[1] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 7

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. ShapeNet: An information-rich 3D model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2, 5, 6

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, 2018. 2

[4] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *CVPR*, 2016. 1

[5] Changhyun Choi and Henrik I Christensen. 3D pose estimation of daily objects using an RGB-D camera. In *IROS*, 2012. 1

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 6

[7] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *CVPR*, 2018. 2, 7

[8] Christopher Funk, Seungkyu Lee, Martin R Oswald, Stavros Tsogkas, Wei Shen, Andrea Cohen, Sven Dickinson, and Yanxi Liu. 2017 ICCV challenge: Detecting symmetry in the wild. In *ICCV Workshops*, 2017. 2

[9] Christopher Funk and Yanxi Liu. Beyond planar symmetry: Modeling human perception of reflection and rotation symmetries in the wild. In *ICCV*, 2017. 2, 6

[10] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 2009. 3

[11] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 2010. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 7

[13] W. Hong, A. Y. Yang, and Y. Ma. On group symmetry in multiple view geometry: Structure, pose and calibration from single images. *IJCV*, 2004. 2

[14] W. Hong, Y. Yu, and Y. Ma. Reconstruction of 3D symmetric curves from perspective images without discrete features. In *ECCV*, 2004. 2

[15] Eldar Insafutdinov and Alexey Dosovitskiy. Unsupervised learning of shape and pose with differentiable point clouds. In *NIPS*, 2018. 2

[16] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *NIPS*, 2017. 6

[17] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *ICCV*, 2017. 5

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] N. Kiryati and Y. Gofman. Detecting symmetry in grey level images: The global optimization approach. *IJCV*, 1998. 2

[20] Thommen Korah and Christopher Rasmussen. Analysis of building textures for reconstructing partially occluded facades. In *ECCV*, 2008. 2

[21] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *ICCV*, 2019. 1

[22] Gareth Loy and Jan-Olof Eklundh. Detecting symmetry and symmetric constellations of features. In *ECCV*, 2006. 2

[23] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An Invitation to 3D Vision: From Images to Geometric Models*. Springer Science & Business Media, 2012. 4

[24] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. *TOG*, 2006. 7

[25] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3D bounding box estimation using deep learning and geometry. In *CVPR*, 2017. 1

[26] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 7

[27] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *CVPR*, 2018. 1

[28] Mahdi Rad and Vincent Lepetit. BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In *ICCV*, 2017. 1

[29] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3D object detection in RGB-D images. In *CVPR*, 2016. 1

[30] 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. In *CVPR*, 2018. 2, 6, 7

[31] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3D reconstruction networks learn? In *CVPR*, 2019. 1

[32] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6D object pose prediction. In *CVPR*, 2018. 1

[33] N.F. Troje and H.H. Bulthoff. How is bilateral symmetry of human faces used for recognition of novel views. *Vision Research*, 1998. 2

[34] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *PAMI*, 1991. 7

[35] T. Vetter, T. Poggio, and H. H. Bulthoff. The importance of symmetry and virtual views in three-dimensional object recognition. *Current Biology*, 1994. 2

[36] 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, 7

[37] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3D mesh models from single RGB images. In *ECCV*, 2018. 6, 7, 8

[38] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *CVPR*, 2020. 2

[39] Yu Xiang, Wongun Choi, Yuanqing Lin, and Silvio Savarese. Data-driven 3D voxel patterns for object category recognition. In *CVPR*, 2015. 1

[40] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. DISN: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NIPS*, 2019. 2, 7, 8

[41] Yifan Xu, Tianqi Fan, Yi Yuan, and Gurprit Singh. Ladybird: Quasi-monte carlo sampling for deep implicit field based 3d reconstruction with symmetry. In *ECCV*, 2020. 2

[42] Yuan Yao, Nico Schertler, Enrique Rosales, Helge Rhodin, Leonid Sigal, and Alla Sheffer. Front2Back: Single view 3d shape reconstruction via front to back prediction. In *CVPR*, 2020. 2, 7

[43] Zhichao Yin and Jianping Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018. 7

[44] H. Zabrodsky, S. Peleg, and D. Avnir. Symmetry as a continuous feature. *PAMI*, 1995. 2

[45] Xuaner Cecilia Zhang, Yun-Ta Tsai, Rohit Pandey, Xiuming Zhang, Ren Ng, David E Jacobs, et al. Portrait shadow manipulation. *TOG*, 2020. 2

[46] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 2, 7

[47] Yichao Zhou, Haozhi Qi, Jingwei Huang, and Yi Ma. Neurvps: Neural vanishing point scanning via conic convolution. In *NIPS*, 2019. 5