NeRD++: Improved 3D-mirror symmetry learning from a single image

Yancong Lin
Silvia L. Pintea
Jan C. van Gemert

Computer Vision Lab,
Delft University of Technology,
Delft, Netherlands

Abstract

Many objects are naturally symmetric, and this symmetry can be exploited to infer unseen 3D properties from a single 2D image. Recently, NeRD \cite{erd} is proposed for accurate 3D mirror plane estimation from a single image. Despite the unprecedented accuracy, it relies on large annotated datasets for training and suffers from slow inference. Here we aim to improve its data and compute efficiency. We do away with the computationally expensive 4D feature volumes and instead explicitly compute the feature correlation of the pixel correspondences across depth, thus creating a compact 3D volume. We also design multi-stage spherical convolutions to identify the optimal mirror plane on the hemisphere, whose inductive bias offers gains in data-efficiency. Experiments on both synthetic and real-world datasets show the benefit of our proposed changes for improved data efficiency and inference speed.

1 Introduction

Symmetry exists in nature, in man-made environments, in science and arts. Mirror symmetry, also known as bilateral or reflection symmetry, is an intrinsic property of many objects, and it allows inferring the entire object from only partial view. This has been shown to be useful for shape completion \cite{f}, \cite{g} and single-view 3D reconstruction \cite{i}, \cite{j}, \cite{k}.

Deep learning approaches have achieved astonishing results on estimating mirror symmetries from single-view images, by learning dense features from convolutional networks instead of relying on local feature matching \cite{l}, \cite{m}. In addition to feature learning, deep networks can incorporate 3D mirror geometry as in NeRD \cite{erd}, the top-performing symmetry detection model. However, NeRD \cite{erd} builds a computationally 4D feature volumes during learning, resulting in high inference latency. Moreover, its performance also deteriorates substantially when limited training data is available.

In this paper, we make two improvements over NeRD \cite{erd} to increase its data and compute efficiency. Specifically, from learned semantic deep features we calculate a compact 3D correlation volume for each candidate plane. Correlations measure the similarity between the features and their mirrored versions, at every location in the featuremap. The optimal plane is characterized by the highest feature correlation. By adding explicit correlation computation into the model, we bypass the expensive 4D feature volumes in NeRD \cite{erd}, and thus substantially speed up the inference. Our second modification is on data-efficiency by...
Figure 1: **3D mirror symmetry detection.** We identify 3D mirror planes by measuring correlations between the input and its mirrors over depth. The mirrors are computed by 3D mirror geometry (a) as in NeRD [26], which localizes the reflections of a given pixel directly on the image plane (b). We measure the similarity between a pixel and its mirrors by explicitly calculating correlations which indicate to what extent a point resembles its correspondences. We also adopt multi-stage spherical convolutions to localize the optimal plane hierarchically on the hemisphere where all candidate planes are sampled (c). We improve both data and compute efficiency over NeRD [26] by introducing (b) and (c).

Our contributions are: (1) improving the data efficiency of NeRD, the top-performing method on 3D mirror symmetry detection from single images by introducing spherical convolutions; (2) reducing the inference latency significantly ($\times 20$), by calculating explicit correlations; (3) experimentally demonstrating the added-value of our improvements on two datasets: ShapeNet [2] and Pix3D [19].

2 Related work

**Planar symmetry detection.** A thorough overview on symmetry detection with focus on 2D symmetries, is given in [12]. Further work expands on this by including other types of symmetries such as medial-axis-like symmetries and by adding synthetic 3D data [4]. More recently, planar symmetry detection with deep networks achieves competitive results [3, 16]. However, for planar symmetry detection objects are typically front-facing, greatly simplifying the task. In addition, planar symmetry does not encode any 3D perspective information. Here we differ from these works, as we focus on 3D mirror symmetry from single-view images taken from any perspective.

**3D mirror symmetry detection.** 3D mirror symmetry is prevalent in both nature and the man-made environments. There has been excellent research on utilizing geometric transforms for detecting mirror symmetries from 3D inputs [1, 15, 17]. A 3D Hough transform proves effective at detecting mirror symmetry planes from point clouds, in [1]. Alternatively, planar reflective symmetry transform can find symmetry planes in 3D volumetric data [19].
3 Revisiting 3D mirror geometry and NeRD

3.1 3D mirror geometry

(i) 3D mirror planes. A plane $\psi$ is uniquely defined by its normal direction $n \in \mathbb{R}^3$ and offset $b \in \mathbb{R}$ as $n^T x + b = 0$, where $x \in \mathbb{R}^3$ denotes points on the plane. However, we are unable to determine $b$ from a single image due to scale ambiguity [14, 26]. This is because the scene can be moved arbitrarily along the normal direction $n$ and scaled accordingly, without affecting the image, as shown in Fig. 2(a). Therefore, $b$ is often set to 1 and the normal direction $n$ of the mirror plane is only the unknown to predict. Moreover, given that a normal direction $n$ is equivalent to a point on a unit hemisphere, we can further define a plane as a spherical point. Thus we can sample planes on a unit hemisphere.
**3D mirror transform.** Fig. 2(b) shows an illustration of 3D mirror geometry for a randomly sampled plane $\psi$ defined by $\mathbf{n}^\top \mathbf{x} + 1 = 0$. The corresponding 3D mirror transformation $\mathbf{M}_{\psi} \in \mathbb{R}^{4 \times 4}$ associated to plane $\psi : \mathbf{n}^\top \mathbf{x} + 1 = 0$ is uniquely defined by the normal direction of the plane $\mathbf{n}$ [4, 8]:

$$\mathbf{X}'' = \begin{pmatrix} 1 - 2\mathbf{n}^\top & -2\mathbf{n} \\ 0 & 1 \end{pmatrix} \mathbf{M}_{\psi} \mathbf{X}'$$

where $\mathbf{X}'' \in \mathbb{S}$ and $\mathbf{X}' \in \mathbb{S}$ are a pair of symmetric 3D points, and $\mathbb{S} \subset \mathbb{R}^4$ is the set of 3D points on the object surface in homogeneous coordinates.

Given the camera intrinsic matrix $\mathbf{K} \in \mathbb{R}^{4 \times 4}$, both $\mathbf{X}'$ and $\mathbf{X}''$ can be projected on the image plane by $\mathbf{x}' = \mathbf{KX}' / d'$ and $\mathbf{x}'' = \mathbf{KX}'' / d''$, where $d'$ and $d''$ are the corresponding depths in the camera space. Thus, the constraint between points $\mathbf{x}'$ and their projections $\mathbf{x}''$ can be derived as:

$$\mathbf{x}'' d'' = \mathbf{KM}_{\psi} \mathbf{K}^{-1} \mathbf{x}' d',$$

where $\mathbf{x}' = [x', y', 1, 1/d']$ and $\mathbf{x}'' = [x'', y'', 1, 1/d'']$ indicate the coordinates of the projected points in the pixel space. Eq. (2) enables us to find the symmetric correspondences of every pixel at various depths, given a sampled mirror plane, $\psi$.

### 3.2 NeRD overview

We briefly recap NeRD, the state-of-the-art model on symmetry detection, which incorporates 3D mirror geometry into learning. NeRD represents our starting point.

**(i) Feature learning.** NeRD first learns semantic features from RGBA images via a convolutional neural network, which results in a feature map $\mathcal{F}$ of size $[H \times W \times C]$, where $H$, $W$, $C$ indicate height, width, and number of channels, respectively.

**(ii) 3D mirroring.** Given a randomly sampled plane from the hemisphere, $\psi$, NeRD localizes the symmetric correspondences $(x', y')$ for each pixel $(x, y)$ at varying depth $d \in \mathcal{D}$ according to Eq. (2), where $\mathcal{D} = \{d_{min} + \frac{i}{D-1} (d_{max} - d_{min}) | i = \{0, 1, ..., D - 1\}\}$, $d_{min}$ and $d_{max}$ are the minimum and maximum depth values. Subsequently, NeRD concatenates the learned feature at each pixel $(x, y)$ and its correspondences $(x', y')$ across depth $d$, resulting in a 4D feature volume $\mathcal{V}$ of size $[32 \times D \times H \times W]$. $\mathcal{V}$ is further downsampled by 3D convolutions and then flattened into a 1D vector of size 16384 for further classification.

**(iii) Classification on the hemisphere.** To identify the optimal symmetry plane on the hemisphere, NeRD samples a number of candidate planes $\psi$ on the hemisphere hierarchically, and then employs huge fully connected layers ($\geq 8M$ parameters) to predict the likelihood of a candidate being the symmetry plane. The sampling is repeated 3 times in a coarse-to-fine manner, until the desired resolution is reached.

### 4 Data-efficient and fast 3D mirror symmetry detection

Built on top of NeRD [26], our model also takes as input an RGBA image and outputs sampled planes and their associated confidence for being a true mirror plane. We choose the plane with the highest confidence as our prediction. Although an object may admit
multiple symmetries, we only predict the principal mirror symmetry. We follow the design of NeRD \cite{26}, and decompose our model into three parts, as shown in Fig. 3: (i) feature extraction and correlation calculation, (ii) 3D mirror, (iii) plane identification by spherical convolutions. The 3D mirror follows the original design in \cite{26}, while the other two parts differ, as we make two essential modifications to improve both data and compute efficiency. We detail these changes below.

4.1 Compact correlation volume

Given the learned semantic features $\mathcal{F}$ of size $[H \times W \times C]$, where $H$, $W$, $C$ indicate height, width, and number of channels, we calculate intra-pixel correlations. We correlate all pairs of points in the $[H \times W]$ grid with each other by a dot product over the channel dimension. This produces a correlation tensor $\mathcal{C}$ of size $[H \times W \times H \times W]$. $\mathcal{C}$ encodes the extent to which a pixel resembles the others. Using Eq. (2), we index the correlation tensor $\mathcal{C}$ at $\mathcal{C}(x, y, x', y')$ via bi-linear interpolation, where $(x', y')$ are the correspondences of a point $(x, y)$ across a candidate symmetry plane $\psi$, at various depths $d$. This results in a compact 3D correlation volume $\mathcal{V}$ of size $[D \times H \times W]$, which substantially reduces the computations compared to the 4D feature volume in NeRD \cite{26}. We aggregate the information over the entire $\mathcal{V}$, and apply 3D convolutions to downscale $\mathcal{V}$, resulting in an output tensor of size $[H' \times W' \times D']$. In practice, we set $H$, $W$ and $C$ to 64.

The correlation volume $\mathcal{V}$ encodes the similarity between each input and its mirrors at all sampled depths. A higher similarity indicates that the given plane $\psi$ is more likely to be a mirror plane. We flatten the downscaled volume $\mathcal{V}$ to a 1D vector. Each sampled plane on the hemisphere $\psi$ is characterized by one such 1D vector.
4.2 Spherical convolutions for symmetry plane detection

Each candidate plane is equivalent to a sampled spherical point, and has an associated 1D feature descriptor obtained from the correlation volume $V$. Given that the planes lie on the hemisphere, we take advantage of spherical convolutions to learn the most probable candidate plane. We use EdgeConv [21] to extract features from the local neighborhood of a plane. Specifically, we treat each sampled spherical point as a node, and compute its top 16 nearest neighbors. We stack 3 layers of EdgeConv, followed by BatchNorm and LeakyReLU activations, to guarantee a sufficiently large receptive field.

Following [26], we also sample the planes in a multi-stage sequence by adopting the Fibonacci lattice [6]. In practice, we sample planes over 3 stages, at multiple scales, in a coarse-to-fine manner. $P_i = \{n^k_{i}\}_{k=1}^K \subset \mathbb{R}^3$ represents all sampled planes at the $i$th stage, where $n$ is the normal direction, and $K$ is the number of planes. The sampling at the $i$th stage satisfies $P_i = \{n^k : \arccos(|\langle n^k, \hat{n}_{i-1} \rangle|) \leq \delta_i\}_{k=1}^K$, where $\hat{n}_{i-1}$ is the optimal plane from previous stage and $\delta$ controls the sampling region.

During training, we minimize the binary cross-entropy loss at each stage and averaged over the positive and negative samples separately, due to the class imbalance. The positive samples are the spherical points that are closet to the ground truth at each stage, while the others are considered negative samples. At test time, we choose the plane with the highest estimated confidence at the final stage.

5 Experimental analysis

Datasets. We conduct experiments on synthetic ShapeNet [2] and real-world Pix3D [19] datasets. For both datasets the objects are aligned to the canonical space such that the Y-Z plane is the 3D mirror symmetry plane. On the synthetic ShapeNet dataset, we use the same subset as in [26] for fair comparison. Images are of size 256x256 px and split in 175,122/500/8,756 training/validation/test sets. For the real-world Pix3D dataset we preprocess the data as in [26]. We first crop the objects inside bounding boxes. And then, we rescale them to 256x256 px, and adjust the camera intrinsic matrix $K$ accordingly. This results in a dataset of 5,285 and 588 images for training and test respectively.

Evaluation. We follow [26] and evaluate all methods by measuring the angle difference of the plane normals between the ground-truth and predictions in the camera space. We calculate the percentage of the predictions that have a smaller angle difference than a given threshold and compute the area under the angle accuracy (AA) curves.

Implementation details. The $x = 0$ plane in the object space is the ground truth as it is explicitly aligned for each object [2]. We set $d_{\text{min}} = 0.64$, $d_{\text{max}} = 1.23$, and $D = 64$ for depth. We perform spherical convolutions at 3 scales and sample $\{128, 64, 64\}$ symmetry planes at each scale. We set the scale factor to be $\delta = \{90.0^\circ, 12.86^\circ, 3.28^\circ\}$. We train from scratch for each dataset on Nvidia RTX2080 GPUs with the Adam optimizer [8], for maximum 32 epochs. The learning rate and weight decay are set to be $3 \times 10^{-4}$ and $1 \times 10^{-7}$. We decay the learning rate by 10 after 24 epochs. To maximize the GPU usage, we set the batch size to 6.

Baselines. We primarily compare with the state-of-the-art: NeRD [26]. We also implement a standard baseline using direct regression to estimate the symmetry normal $n$. For this baseline we use a ResNet-50 [1] backbone with an $L_1$ loss. We additionally compare with Front2Back [24], which detects 3D mirror symmetry using a variant of the iterative closest
Figure 4: Exp 1: Data efficiency. (a) Quantitative comparison between our model and NeRD when training on various ShapeNet subsets. The difference of the two models is limited when the training data is ample (e.g., \(18000 \geq\)). However, our model outperforms NeRD when training on limited data (e.g. \(\leq 9000\) images). (b) Qualitative comparison on the 1% subset. We project detected symmetries on the image plane, where green is the true symmetry plane, and red indicates the prediction error. We use the ground truth scale for plotting, to resolve the scale ambiguity. Our model makes preciser predictions than NeRD when training on the 1% subset.

We evaluate the data efficiency of our model by reducing the number of training samples to \{50\%, 25\%, 10\%, 5\%, 2.5\%, 1\%\} on the ShapeNet dataset, which has approximately 200K training images in total. We train all models from scratch and compare the AA scores at 3° and 5° on the complete test set. We compare our model with NeRD which holds state-of-the-art result in Fig. 4(a), and display a few examples of detected symmetry planes in Fig. 4(b). The two models have a comparable amount of parameters, thus removing the impact of parameters. In general, our model shows superiority over NeRD with the decrease of training samples, and this advantage accentuates on subsets with fewer than 10K images. When training on 100\% – 10\% subsets we observe minimal differences, while on 10\% – 1\% subsets we observe a drastic difference (up to 10\% in AA). This indicates that our model is more effective at learning from limited data. Notably, our model can achieve similar results to NeRD with only half of the training data, as seen on the 2.5\% and 5\% subsets, thus demonstrating the data efficiency of our model.
Figure 5: Exp 2.1: Comparison on synthetic ShapeNet. Comparison with existing baselines ResNet [7], RotCon [25], DISN [23], NOCS [20], Front2Back [24] and the recent NeRD [26]. Our model shows competitive results with the top-performing model.

(a) Percentage of predictions per angular error. (b) Angle accuracy.

Figure 6: Exp 2.2: Comparison on real-world Pix3D dataset. Our model performs the best on this challenging real-world dataset, as the knowledge of 3D mirror no longer needs to be learned from massive data. Moreover our model is $\times 20$ faster than NeRD at inference time.

(a) Percentage of predictions per angular error. (b) Angle accuracy.

5.2 Exp 2: Comparison with state-of-the-art

Exp 2.1: Comparison on synthetic data. In Fig. 5 we compare with the state-of-the-art on ShapeNet. NeRD [22] has the best AA at 1°, when large training sets are available. Our model is competitive to NeRD. The prediction error of both is less than 1° in 80% of the test cases. The ResNet baseline using direct regression can only reach approximately 50% AA at 1°, indicating that naive convolutions lack the ability to exploit the mirror symmetry, even with ample training data. We also notice that end-to-end approaches outperform models relying on heavy post-processing, such as Front2Back [24].

Exp 2.2: Comparison on real-world data. To further validate the effectiveness of our model, we also test on the real-world Pix3D dataset [19], as shown in Fig. 6. It is worth noting that the prediction error on Pix3D is relatively larger than on ShapeNet. On one hand, there is limited training data: 5,000 images in total, which is significantly less than ShapeNet. On the other hand, the camera configuration differs from image to image, thus making it hard to make precise predictions. Our model outperforms all the other models consistently, thus demonstrates the superiority of our design. NeRD lags behind due to a high demand for training data. Moreover, our mode is $\times 20$ faster than NeRD at inference time, as showing in Fig. 6(b). Please see the supplementary material for qualitative results.
Table 1: **Exp 3: Ablation studies.** We quantitatively verify the added value of 3D mirror geometry, 3D correlation volumes, and spherical convolutions on the ShapeNet 1% subset. All these design choices are essential for the performance of our model.

<table>
<thead>
<tr>
<th>3D Mirror</th>
<th>Correlation volumes</th>
<th>Spherical convolutions</th>
<th>AA@1°</th>
<th>AA@5°</th>
</tr>
</thead>
<tbody>
<tr>
<td>a ✗</td>
<td>✗</td>
<td>✓</td>
<td>0.8</td>
<td>9.5</td>
</tr>
<tr>
<td>b ✓</td>
<td>✓</td>
<td>✗</td>
<td>15.8</td>
<td>44.4</td>
</tr>
<tr>
<td>c ✓</td>
<td>✗</td>
<td>✓</td>
<td>8.0</td>
<td>31.5</td>
</tr>
<tr>
<td>d ✓</td>
<td>✓</td>
<td>✓</td>
<td>22.1</td>
<td>53.5</td>
</tr>
</tbody>
</table>

### 5.3 Exp 3: Ablation studies

To verify the contribution of each component in our design, we conduct ablation studies, as shown in Tab. 1. All models are trained on the ShapeNet 1% subset. Model (a) is a simple baseline using direct regression and shows inferior results to the others in detecting 3D mirror geometry. We replace the spherical convolutions in our design (d) with $1 \times 1$ convolutions in model (b). Comparing (b) and (d), we find that spherical convolutions improve the results significantly. In model (c), we replace the convolutions over the 3D cost volumes $V$ by taking the max over the depth dimension. By doing so, we only obtain the correspondence with the highest correlation across different depths for each pixel, thus removing the 3D spatial information. However, model (c) substantially underperforms model (d), thus validating the necessity of 3D cost volumes. The ablation studies justify the added value of the 3D mirror, correlation volumes, and spherical convolutions.

### 6 Conclusions and drawbacks

This paper analyzes improvements for 3D mirror symmetry detection from single-view perspective images. We explicitly incorporate feature correlations and spherical convolutions into the state-of-the-art 3D mirror detection [26]. This provides the model with improved data efficiency and computation efficiency. Extensive experiments on both synthetic and real-world datasets demonstrate the benefits of our proposed changes when compared to state-of-the-art.

One of the drawbacks of this work is that if the found point correspondences are not sufficiently similar in appearance, the mirror plane detection will be erroneous. A second drawback is the incapability in detecting multiple symmetry planes, and this is restrictive as certain objects may display multiple symmetries, such as local symmetries, translational and rotational symmetries. In addition, mirror symmetries sometimes cannot be characterized by a single plane, such as intrinsic symmetries commonly seen in non-rigid deformable objects: e.g. human bodies. Extending the current work for detecting multiple types of symmetries, such as rotation symmetry and translation symmetry is a viable future direction. Another promising future research direction is exploring the usage of 3D mirror in single-view 3D reconstruction, such as weakly-supervised depth estimation and shape completion.
References


