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

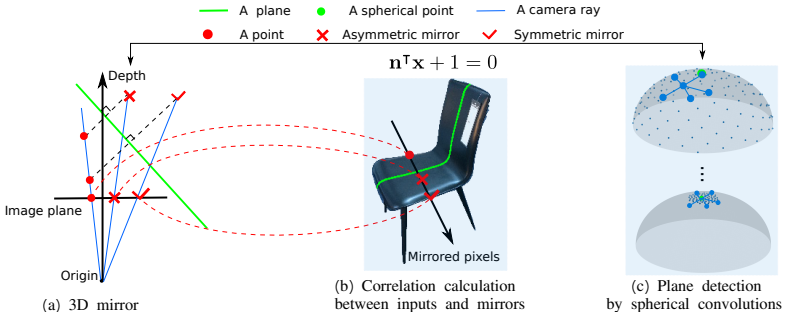
Many objects are naturally symmetric, and this symmetry can be exploited to infer unseen 3D properties from a single 2D image. Recently, NeRD [26] is proposed for accurate 3D mirror plane estimation from a single image. Despite the unprecedented accuracy, it relies on large annotated datasets for training 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 infering the entire object from only partial view. This has been shown to be useful for shape completion [5, 13] and single-view 3D reconstruction [22, 23, 24].

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 [5, 18, 26]. In addition to feature learning, deep networks can incorporate 3D mirror geometry as in NeRD [26], the top-performing symmetry detection model. However, NeRD [26] 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 [26] 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 [26], and thus substantially speed up the inference. Our second modification is on data-efficiency by



(a) 3D mirror (b) Correlation calculation between inputs and mirrors (c) Plane detection by spherical convolutions

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).

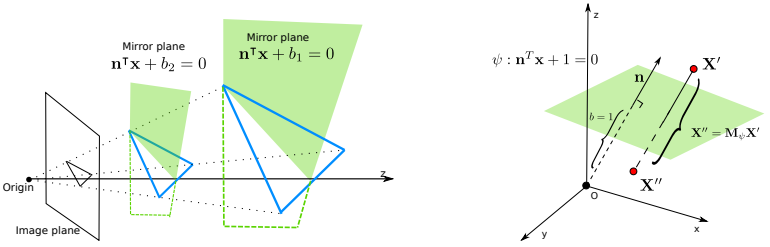
exploiting geometric priors [10, 11] where we use spherical convolutions on the hemisphere. The hemisphere is the space containing all the candidate symmetry planes. Rather than using a huge fully connected layer as in NeRD to locate the optimal symmetry plane, we make use of the geometric prior that the shape of the search-space is a hemisphere: We use spherical convolutions as a prior [11]. These two well-chosen inductive biases, are what contributes to both the data-efficiency and computational efficiency of our model. Fig. 1 summarizes our approach.

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 [15].



(a) Scale ambiguity (b) 3D mirror geometry

Figure 2: **(a) Scale ambiguity.** The two objects (in blue) only differ in scale, but their projections on the image plane are the same. Therefore, we are unable to determine the scale of the object, or the value of the plane offset. In practice, we set $b = 1$. **(b) 3D mirror geometry.** 3D points $\mathbf{X}'$ and $\mathbf{X}''$ are symmetric with respect to the given plane ψ defined by $\mathbf{n}^\intercal\mathbf{x}+1=0$. We represent $\mathbf{X}' \in \mathbb{R}^4$ and $\mathbf{X}'' \in \mathbb{R}^4$ in the homogeneous coordinate, where $\mathbf{X}'' = \mathbf{M}_\psi\mathbf{X}'$ as defined in Eq. (1). $\mathbf{M}_\psi \in \mathbb{R}^{4\times4}$ is the 3D mirror transformation uniquely determined by the normal direction $\mathbf{n}$ [11, 9].

Similarly, we also focus on 3D geometric priors, and specifically on how to improve existing priors, but instead of relying on 3D data we start from single-view images.

Recently, deep networks have been used for leveraging large datasets for learning 3D symmetries [6, 18]. Despite being able to detect multiple symmetries, they rely on heavy post-processing procedures to find the optimal symmetry plane. Moreover, these models have only been tested on synthetic 3D datasets with voxelized volumes or RGB-D data (i.e. ShapeNet [2]). In contrast, we propose improvements for end-to-end 3D mirror symmetry detection, and test on both synthetic and real-world 2D images.

3D mirror symmetry from single-view images. A 2-stage approach can be effective for 3D mirror symmetry detection from 2D images, by first matching image correspondences and then applying RANSAC to identify the best symmetry plane [9]. However, this strategy is no longer applicable in the absence of texture, or on smooth surfaces, or repetitive patterns, because of incorrect correspondences. Rather than relying on local feature matching, NeRD [26] makes use of neural networks to learn dense features, making it the top-performing model. Here, we use NeRD [26] as our starting point and make two essential changes to the model: we explicitly compute correlations between correspondences and we use spheric convolutions to correctly localize the optimal symmetry plane on the hemisphere. These two extensions greatly reduce computations and make our model data-efficient.

3 Revisiting 3D mirror geometry and NeRD

3.1 3D mirror geometry

(i) 3D mirror planes. A plane ψ is uniquely defined by its normal direction $\mathbf{n} \in \mathbb{R}^3$ and offset $b \in \mathbb{R}$ as $\mathbf{n}^\intercal\mathbf{x}+b=0$, where $\mathbf{x} \in \mathbb{R}^3$ 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 $\mathbf{n}$ and scaled accordingly, without affecting the image, as shown in Fig. 2(a). Therefore, b is often set to 1 and the normal direction $\mathbf{n}$ of the mirror plane is only the unknown to predict. Moreover, given that a normal direction $\mathbf{n}$ 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.

(ii) 3D mirror transform. Fig. 2(b) shows an illustration of 3D mirror geometry for a randomly sampled plane ψ defined by $\mathbf{n}^\intercal \mathbf{x} + 1 = 0$. The corresponding 3D mirror transformation $\mathbf{M}_\psi \in \mathbb{R}^{4\times4}$ associated to plane $\psi : \mathbf{n}^\intercal \mathbf{x} + 1 = 0$ is uniquely defined by the normal direction of the plane $\mathbf{n}$ [1, 9]:

$$\mathbf{X}'' = \underbrace{\begin{pmatrix} \mathbf{I} - 2\mathbf{n}\mathbf{n}^\intercal & -2\mathbf{n} \\ \mathbf{0} & 1 \end{pmatrix}}_{\mathbf{M}_\psi} \mathbf{X}', \tag{1}$$

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 $\mathbf{K} \in \mathbb{R}^{4\times4}$, both $\mathbf{X}'$ and $\mathbf{X}''$ can be projected on the image plane by $\mathbf{x}' = \mathbf{K}\mathbf{X}'/d'$ and $\mathbf{x}'' = \mathbf{K}\mathbf{X}''/d''$, 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:

$$\mathbf{x}''d'' = \mathbf{K}\mathbf{M}_\mathbf{P}\mathbf{K}^{-1}\mathbf{x}'d', \tag{2}$$

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, ψ.

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, ψ, 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 ψ 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

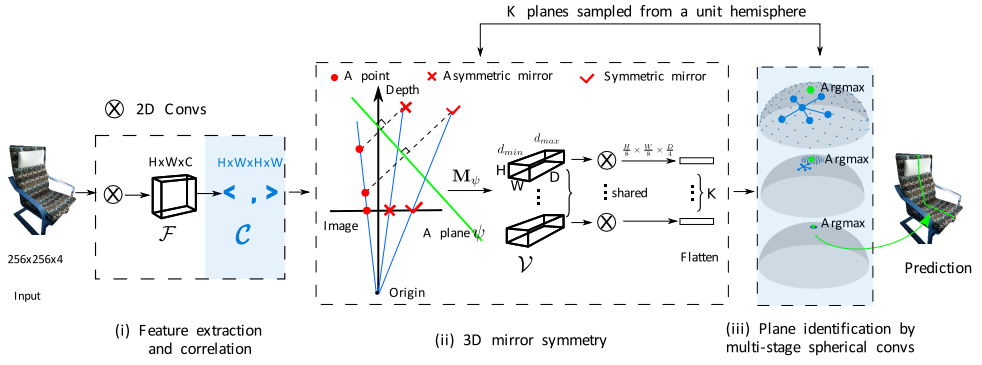


Figure 3: **Overview.** Our model follows [26] and includes three components: (i) feature extraction and correlation calculation, (ii) 3D mirror, and (iii) plane identification by spherical convolutions. The model first calculates intra-pixel correlations $\mathcal{C}$ using learned features $\mathcal{F}$. Then builds a 3D correlation volume $\mathcal{V}$ for each sampled plane ψ, which is then flattened as a feature descriptor. Here we incorporate explicit correlations to reduce computations compared to [26]. We additionally, adopt spherical convolutions on uniformly sampled planes on the hemisphere to locate the optimal plane (depicted in green). We highlight in blue our changes with respect to [26].

multiple symmetries, we only predict the principal mirror symmetry. We follow the design of NeRD [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 [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 ψ, 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 [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 $\left[\frac{H}{8} \times \frac{W}{8} \times \frac{D}{4}\right]$. 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 ψ 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 ψ 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 $\mathcal{V}$. 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 filed.

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. $\mathbb{P}_i = \{\mathbf{n}_i^k\}_{k=1}^K \subset \mathbb{R}^3$ represents all sampled planes at i^{th} stage, where $\mathbf{n}$ is the normal direction, and K is the number of planes. The sampling at i^{th} stage satisfies $\mathbb{P}_i = \{\mathbf{n}^k : \arccos(|\langle \mathbf{n}^k, \hat{\mathbf{n}}_{i-1} \rangle|) \leq \delta_i\}_{k=1}^K$, where $\hat{\mathbf{n}}_{i-1}$ is the optimal plane from previous stage and δ 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.

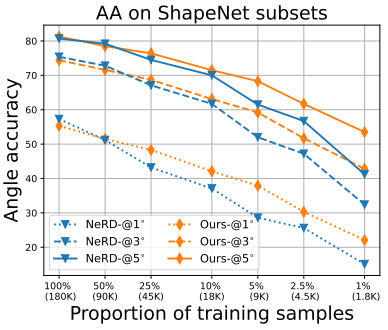
5 Experimental analysis

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 pre-process 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 $\mathbf{K}$ accordingly. This results in a dataset of 5,285 and 588 images for training and test respectively.

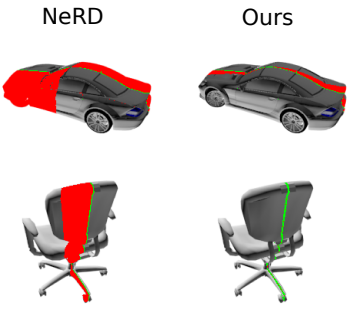
Evaluation. We follow [26] and evaluate all methods by measuring the angle difference of the plane normals between the ground-truth and predictions in the camera space. We calculate the percentage of the predictions that have a smaller angle difference than a given threshold and compute the area under 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 $d_{min} = 0.64$, $d_{max} = 1.23$, 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 $\delta = \{90.0°, 12.86°, 3.28°\}$. 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 3×10^{-4} and 1×10^{-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 $\mathbf{n}$. For this baseline we use a ResNet-50 [7] backbone with an L_1 loss. We additionally compare with Front2Back [24], which detects 3D mirror symmetry using a variant of the iterative closest



(a) Angle accuracy on ShapeNet subsets.



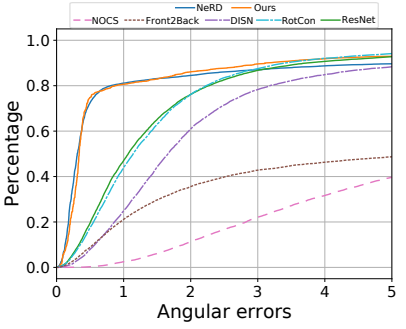
(b) Examples of prediction errors.

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., $18K \geq$). However, our model outperforms NeRD when training on limited data (e.g. $\leq 9K$ 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.

point approach. However, Front2Back requires prior depth maps, and has only been tested on ShapeNet. We also compare with RotCon [25] which proposes a continuous representation for estimating 3D rotation. We use the L_1 loss for training and report its performance on both datasets. We further consider DISN [23] and NCOS [20]. DISN learns 6D rotation representation for estimating camera poses on ShapeNet. We recover the normal of the mirror plane from camera poses and report the performance of their pre-trained models on ShapeNet. NCOS defines a *normalized object coordinate space (NOCS)* and identifies 6D representations of camera poses. We use NOCS to estimate object orientation in ShapeNet.

5.1 Exp 1: Data efficiency

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.

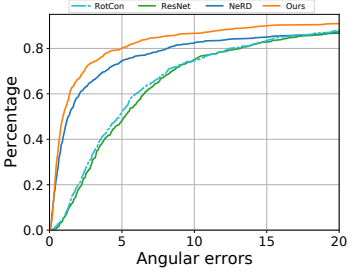


(a) Percentage of predictions per angular error.

Datasets	ShapeNet[0]		
Metrics	AA@1°	AA@3°	AA@5°
ResNet [0]	20.8	55.7	69.5
RotCon [25]	18.7	54.6	69.4
DISN [23]	9.3	26.1	34.1
NOCS [20]	0.6	7.9	17.3
Front2Back [24]	9.3	26.1	34.1
NeRD [26]	**57.3**	75.4	80.7
Ours	55.7	**75.5**	**82.0**

(b) Angle accuracy.

Figure 5: **Exp 2.1: Comparison on synthetic ShapeNet.** Comparison with existing baselines ResNet [0], 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.

Datasets	Speed	Pix3D[19]		
Metrics	(FPS)	AA@1°	AA@5°	AA@10°
ResNet [0]	66	1.5	12.3	23.6
Rotcon [25]	-	2.2	14.1	25.8
NeRD [26]	1.4	22.7	46.0	55.8
Ours	25	**27.5**	**53.0**	**62.8**

(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.

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 [26] 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.

	3D Mirror	Correlation volumes	Spherical convolutions	AA@1°	AA@5°
a	✗	✗	✗	0.8	9.5
b	✓	✓	✗	15.8	44.4
c	✓	✗	✓	8.0	31.5
d	✓	✓	✓	**22.1**	**53.5**

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.

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 1×1 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 $\mathcal{V}$ 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

[1] David Cailliere, Florence Denis, Danielle Pele, and Atilla Baskurt. 3d mirror symmetry detection using hough transform. In *2008 15th IEEE International Conference on Image Processing*, pages 1772–1775. IEEE, 2008.

[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.

[3] Christopher Funk and Yanxi Liu. Beyond planar symmetry: Modeling human perception of reflection and rotation symmetries in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 793–803, 2017.

[4] 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.

[5] L. Gao, L. X. Zhang, H. Y. Meng, Y. H. Ren, Y. K. Lai, and L. Kobbelt. Prs-net: Planar reflective symmetry detection net for 3d models. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020. doi: 10.1109/TVCG.2020.3003823.

[6] Álvaro González. Measurement of areas on a sphere using fibonacci and latitude–longitude lattices. *Mathematical Geosciences*, 42(1):49–64, 2010.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[9] Kevin Köser, Christopher Zach, and Marc Pollefeys. Dense 3d reconstruction of symmetric scenes from a single image. In *Joint Pattern Recognition Symposium*, pages 266–275. Springer, 2011.

[10] Yancong Lin, Silvia L Pintea, and Jan C van Gemert. Deep hough-transform line priors. In *ECCV*. Springer, 2020.

[11] Yancong Lin, Ruben Wiersma, Silvia L Pintea, Klaus Hildebrandt, Elmar Eisemann, and Jan C van Gemert. Deep vanishing point detection: Geometric priors make dataset variations vanish. In *CVPR*, 2022.

[12] Jingchen Liu, George Slota, Gang Zheng, Zhaohui Wu, Minwoo Park, Seungkyu Lee, Ingmar Rauschert, and Yanxi Liu. Symmetry detection from realworld images competition 2013: Summary and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 200–205, 2013.

[13] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020.

[14] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012.

[15] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. In *ACM SIGGRAPH 2006 Papers*, pages 549–559. 2006.

[16] Ahyun Seo, Woohyeon Shim, and Minsu Cho. Learning to discover reflection symmetry via polar matching convolution. In *ICCV*, 2021.

[17] Konstantinos Sfikas, Theoharis Theoharis, and Ioannis Pratikakis. Rosy+: 3d object pose normalization based on pca and reflective object symmetry with application in 3d object retrieval. *International Journal of Computer Vision*, 91(3):262–279, 2011.

[18] Yifei Shi, Junwen Huang, Hongjia Zhang, Xin Xu, Szymon Rusinkiewicz, and Kai Xu. Symmetrynet: Learning to predict reflectional and rotational symmetries of 3d shapes from single-view rgb-d images. *ACM Transactions on Graphics (SIGGRAPH Asia 2020)*, 39(6), 2020.

[19] 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.

[20] 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.

[21] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

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

[23] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019.

[24] 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.

[25] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[26] Yichao Zhou, Shichen Liu, and Yi Ma. NeRD: Neural 3d reflection symmetry detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15940–15949, 2021.