E3Sym: Leveraging E(3) Invariance for Unsupervised 3D Planar Reflective Symmetry Detection

Ren-Wu Li[1,2] Ling-Xiao Zhang[1] Chunpeng Li[1*] Yu-Kun Lai[3] Lin Gao[1,2]

[1]Beijing Key Laboratory of Mobile Computing and Pervasive Device,
Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3] School of Computer Science and Informatics, Cardiff University

renwuli1024@gmail.com, {zhanglingxiao, cpli, gaolin}@ict.ac.cn, LaiY4@cardiff.ac.uk

Abstract

Detecting symmetrical properties is a fundamental task in 3D shape analysis. In the case of a 3D model with planar symmetries, each point has a corresponding mirror point w.r.t. a symmetry plane, and the correspondences remain invariant under any arbitrary Euclidean transformation. Our proposed method, E3Sym, aims to detect planar reflective symmetry in an unsupervised and end-to-end manner by leveraging E(3) invariance. E3Sym establishes robust point correspondences through the use of E(3) invariant features extracted from a lightweight neural network, from which the dense symmetry prediction is produced. We also introduce a novel and efficient clustering algorithm to aggregate the dense prediction and produce a detected symmetry set, allowing for the detection of an arbitrary number of planar symmetries while ensuring the method remains differentiable for end-to-end training. Our method also possesses the ability to infer reasonable planar symmetries from incomplete shapes, which remains challenging for existing methods. Extensive experiments demonstrate that E3Sym is both effective and robust, outperforming state-of-the-art methods.

1. Introduction

Symmetry is a ubiquitous phenomenon in the real world, from microscopic to macroscopic, from virtual to the natural world, from architecture to art. Over the last few decades, considerable attention has been paid to symmetry detection from 2D to 3D. As a valuable property in 3D shape analysis, symmetry is employed in a wide range of tasks such as 3D reconstruction [26, 45, 47, 10], pose estimation [1], shape matching [17], etc. As a result, robust and accurate symmetry detection is required for various applications. In this paper, we focus on detecting global planar symmetry of 3D shapes.

Most of the early advances fall into one of the three categories: matching-based [27, 49, 46], sampling-based [43, 7, 31, 16, 20] and regression-based [15, 11, 42] methods. Matching-based methods detect symmetries by matching local shape signatures in pairs, generating potential symmetry planes, and optimizing them to obtain reasonable solutions. Therefore, the key of matching-based methods is to make the local shape signatures robust or invariant w.r.t. Euclidean transformations. By sampling numerous candidates in the transformation space and further verification, the main issue of sampling-based methods is their high computational complexity if exhaustive sampling is used, or generating inaccurate candidates when heuristic sampling is used, necessitating complicated optimization. With the help of expressive power of deep neural networks and large amounts of training data, regression-based methods detect symmetries by regressing symmetry parameters or symmetry point coordinates in a supervised [15, 42] or unsupervised [11] fashion.

It is evident that if we can establish accurate symmetric correspondences from the shape, it is straightforward to find the planar symmetries. Generally, the symmetric correspondences with similar appearances or geometric structures are in a way that is position and orientation invariant. The correspondence matching is based on the fact that local shape patterns of symmetric counterparts are invariant and should not be sensitive to symmetry transformations. Although this seems intuitive, how to establish the correspondences accurately and robustly is still challenging, which

*Corresponding author

is also the crucial requirement of matching-based methods. It can be even more challenging for symmetry detection of partial shapes where part of the otherwise globally symmetric shape is missing.

In this paper, we introduce a novel approach to estimating 3D global planar symmetry, called E3Sym, in a fully *unsupervised* fashion. Formally, for a given shape, we use a point cloud as its representation for input. Specifically, for each point in the point cloud, we retrieve its neighborhood to discriminatively describe its local pattern. By extracting E(3) invariant features (which are invariant under rotation, reflection and translation transformations in the 3D Euclidean space) of each local patch with an encoder network, the local geometric patterns are invariant to Euclidean transformations, from which we can easily and effectively establish the dense symmetric point correspondences. Once the dense symmetric correspondences are established, the dense potential symmetry planes are deduced. To make our method fully differentiable and trainable in an end-to-end manner, we propose a novel and efficient clustering algorithm to aggregate the dense prediction and produce the detected symmetry set.

To summarize, the main contributions of our work can be summarized as follows:

- We propose leveraging E(3) invariance to detect 3D planar symmetries in an end-to-end and unsupervised manner.
- Our approach is capable of handling shapes with an arbitrary number of planar symmetries and achieves state-of-the-art performance in symmetry detection.
- Our proposed method exhibits the ability to identify reasonable planar symmetries from incomplete shapes, which has been a persistently challenging task for existing methods.

2. Related Work

Symmetries can be classified as global [16, 20, 24, 30, 56, 12] and partial [27, 35, 46, 54] symmetry, as well as intrinsic [32, 54, 18, 23, 39, 53, 29] and extrinsic [56, 16, 35, 27, 24, 46, 20, 30, 12] symmetry. In this paper, we focus on global extrinsic planar reflective symmetries.

Non-learning-based 3D Symmetry Detection. Some methods deal with exact symmetries that map the shape exactly to itself. Martinet et al. [24] examine the extrema and spherical harmonic coefficients of the generalized moments to detect exact symmetries, but real-world objects may only be partially or approximately symmetric. Rajendra et al. [30] propose a manifold optimization method to detect mirror symmetry of point clouds, but cannot handle incomplete point clouds. The above methods can be impractical due to their sensitivity to noise because they only cope with exact symmetries. On the contrary, other methods deal with certain level of approximate symmetries. Zabrodsky et al. [56] define the symmetry distance of a shape to measure the approximate symmetries, which has been adopted and extended by follow-up works. It detects symmetries in $\mathcal{O}(n^6)$ for a shape discretized by an $n \times n \times n$ grid, and takes extremely high computation times with bigger n. Kazhadan et al. [16] introduce a more efficient reflective symmetry descriptor to detect symmetries using a fast Fourier transform-like approach and performs symmetry detection in $\mathcal{O}(n^4)$. Podolak et al. [35] further propose a planar reflective symmetry transform (PRST) to capture the symmetries using a Monte Carlo sampling algorithm so that partial symmetries can be captured. Mitra et al. [27] introduce a algorithm by matching each point through handcrafted local shape signatures, finding the corresponding symmetry points, and voting in the transformation space. Ivan et al. [46] detect symmetry of incomplete 3D mesh models in a similar way. Lipman et al. [22] introduce new tools to analyze and represent symmetries in a point set. Korman et al. [20] detect approximate symmetry using a designed bounded sampling algorithm performed in the transform space and achieve fast detection in $\mathcal{O}(kn^3)$ compared to [16], where $k \ll n$. Hruda et al. [12] define a metric to measure the quality of the candidate symmetry planes, which are produced by brute force matching, making their approach suffer from high computational complexity. These traditional matching-based and sample-based methods are usually time-consuming due to large samples and complex feature computation, and some methods with several hyperparameters are sensitive for various inputs. Our method simply uses a Multi-Layer Perceptron (MLP) to extract features with fewer samples, so it is more efficient than these methods.

Learning-based 3D Symmetry Detection. With the vigorous development of neural networks and deep learning in recent years, some data-driven symmetry detection methods using deep learning have also been proposed. SymNet [15] uses PointNet [37] to extract the features of the point cloud, and predicts the points lying on the symmetry plane, from which the reflective symmetry plane is reasoned afterwards. SymmetryNet [42] predicts point-wise symmetrical positions and the foot points on the symmetry plane (or symmetry axis), and performs clustering and filtering on predicted symmetries during inference. However, the above two methods require a large amount of annotated data for supervised learning, which limits the development and applications of such methods. PRS-Net [11] is the first unsupervised symmetry detection approach, and can detect reflective and rotational symmetry at the same time; however, the numbers of reflective planes and rotational axes are

restricted, and the network parameters increases approximately linearly as the numbers of predicted reflective planes and rotational axes grow.

Equivalent and Invariant Point Cloud Analysis. Learning based methods have been extensively studied for point cloud analysis [37, 38, 48], which achieves amazing performance in many tasks, especially in classification [51], segmentation [25] and registration [50]. However, these methods are built on the assumption that the point clouds are pre-aligned, and their performance may drop when the point clouds are with arbitrary poses. Recently, equivalence and invariance are introduced in point cloud analysis [14, 44, 36, 21, 6, 52, 40, 9, 4, 41] as an inductive bias, aiming to make networks robust to rotation and/or translation.

3. Method

3.1. Background

The E(3) group consists of all rotations, reflections and translations in three-dimensions. E(3) can be defined as

$$\mathrm{E}(3) = \mathrm{O}(3) \ltimes \mathbb{R}^3, \tag{1}$$

where O(3) is the orthogonal matrix group for three-dimensions whose determinant is 1 or -1, which can be viewed as rotation or reflection respectively. Formally, an element g in E(3) can be written by a pair $g = (\mathbf{R}, \mathbf{t})$, where $\mathbf{R}$ is an orthogonal matrix in $\mathbb{R}^{3\times3}$ for rotation or reflection, and $\mathbf{t}$ is a vector in $\mathbb{R}^3$ for translation.

We call a function ϕ, e.g. implemented as a neural network, E(3)-invariant if the extracted feature is invariant no matter which 3D Euclidean transformation $g \in \mathrm{E}(3)$ is applied to the input x.

$$\phi(x) = \phi(gx), \ \forall g \in \mathrm{E}(3). \tag{2}$$

Group invariance can be made simply by averaging over the group [55, 28] G with respect to a general function φ:

$$\phi(x) = \frac{1}{|G|} \sum_{g\in G} \varphi(g^{-1}x), \tag{3}$$

and the resulting $\phi(x)$ is G-invariant. But unfortunately, for the continuous 3D Euclidean group $G = \mathrm{E}(3)$, whose *cardinality* $|G|$ is infinite, and it is impossible to enumerate over the group. Instead of discretizing the continuous group or sampling over the group densely, Frame Averaging (FA) [36] proposed to construct a critical subset of the group, which only depends on the input itself, to replace the whole group G, making the group averaging efficient and expressive.

For a given vector space X and a group G, a *frame* is defined as a mapping $\mathcal{F} : X \to 2^G \setminus \emptyset$, which is only correlated with the input $x \in X$. Subsequently, the group averaging is replaced by frame averaging:

$$\phi(x) = \frac{1}{|\mathcal{F}(x)|} \sum_{g\in\mathcal{F}(x)} \varphi(g^{-1}x). \tag{4}$$

Frame can be defined in different fashions. FA defines the *frame* of a point cloud by leveraging Principal Component Analysis (PCA). Formally, given a point cloud $\mathcal{P} \in \mathbb{R}^{N\times3}$ that consists of N points, the covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ is computed, from which the eigenvalues (i.e., $\lambda_1, \lambda_2, \lambda_3$, $\lambda_1 < \lambda_2 < \lambda_3$, assuming eigenvalues are unique; see more discussions later) with their corresponding eigenvectors (i.e., e_1, e_2, e_3) are decomposed:

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (p_i - \overline{\mathcal{P}})^T (p_i - \overline{\mathcal{P}}), \tag{5}$$

where $\overline{\mathcal{P}}$ is the barycenter of the point cloud and p_i is the i-th point. Then the *frame* is defined as :

$$\mathcal{F}(\mathcal{P}) = \{([\pm e_1, \pm e_2, \pm e_3], \overline{\mathcal{P}})\}. \tag{6}$$

3.2. Unsupervised 3D Planar Symmetry Detection

In this section, we present the pipeline of our method, depicted in Figure 1. The pipeline begins with the extraction of E(3) invariant features $\mathcal{H}$ from the input point cloud. To establish correspondences and determine symmetry planes $\mathcal{S}$, we leverage the Differentiable Top-K method [5]. Then we propose a novel differentiable plane grouping method, named SymGroup. This method produces M clustered planes $\mathcal{T}$, where the number of clusters M is automatically determined during the grouping process.

E(3) Invariant Features Extraction. For the input point cloud $\mathcal{P} \in \mathbb{R}^{N\times3}$ with N points, we model the local patch around the i-th point $p_i \in \mathbb{R}^3$ as a spherical neighborhood by ball query [38]. To make each patch have the same number of points for network forwarding, we apply random dropping or duplication. The neighborhood is denoted as $\mathcal{B}_i = \{p_j \in \mathcal{P} \mid \|p_i - p_j\| \le r\}$, where r is the radius of the ball, and $|\mathcal{B}_i| = K$ indicates the cardinality of the local point cloud patch. We would like to derive an inductive representation from the local point cloud patch, aiming to discriminatively encode the underlying geometry information of point p_i. Without loss of generality, we denote the feature extraction network as φ. The E(3) invariant feature $h_i \in \mathcal{H}$ of p_i is computed as:

$$h_i = \frac{1}{|\mathcal{F}_i|} \sum_{k=1}^{|\mathcal{F}_i|} \mathcal{A}\left(\varphi(\mathcal{F}_i^k \mathcal{B}_i)\right), \tag{7}$$

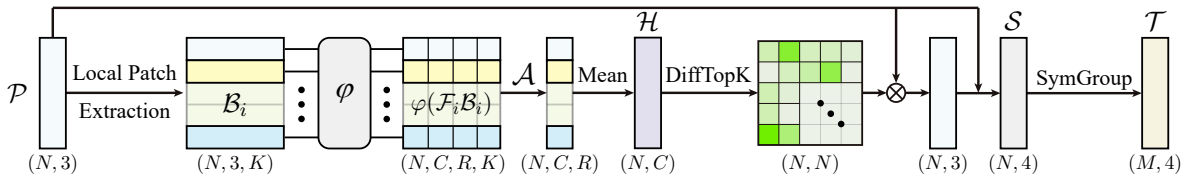


Figure 1: Overview of our method. Local patches $\mathcal{B}_i$ each with K points are extracted by ball query (with random dropping or duplication) from the input point cloud $\mathcal{P}$, and then E(3) invariant feature $\mathcal{H}$ is extracted by Frame Averaging [36] via the encoder φ with channel size C. R is the number of *frames*. Then we perform 1-NN search within the feature space using Differentiable Top-K method [5] to find the nearest neighbor, yielding correspondences to determine planes $\mathcal{S}$, where each plane s_i is parameterized in implicit form $a_ix+b_iy+c_iz+d_i=0$. Then we propose a differentiable plane grouping method, named SymGroup, to output the M clustered planes $\mathcal{T}$, where M is automatically determined during grouping. We train the whole network using symmetry distance loss L in an unsupervised manner.

where $\mathcal{F}_i^k$ is the k-th element of the *frame*s defined by Eq. (6) taking as input $\mathcal{B}_i$. $\mathcal{A}$ is a permutation invariant aggregation function implemented as max-pooling on neighbors. For the general encoding function φ, we choose to use a shared MLP with final channel size $C=64$, whose architecture is similar to the representative PointNet [37] in point cloud analysis.

Correspondence Establishment. When the E(3) invariant features are extracted, we find the symmetrical point for each point $p_i \in \mathcal{P}$ in the feature space. As a result, for p_i, we perform 1-NN (nearest neighbor) search within the feature space to find its nearest neighbor p_j, yielding a point pair (p_i, p_j), i.e, a correspondence. Since the NN search is not differentiable, we utilize the Differentiable Top-K method [5] instead. A plane can be uniquely determined and parameterized by the point pair in the implicit form: $a_{ij}x+b_{ij}y+c_{ij}z+d_{ij}=0$. The parameters $(a_{ij}, b_{ij}, c_{ij}, d_{ij})$ can be determined by:

$$\mathbf{n}_{ij}=\frac{p_i-p_j}{\|p_i-p_j\|}, d_{ij}=-\frac{\mathbf{n}_{ij}\cdot(p_i+p_j)}{2}, \tag{8}$$

where $\mathbf{n}_{ij}=(a_{ij}, b_{ij}, c_{ij})$, which is of unit length. When the correspondences in a point cloud $\mathcal{P}\in\mathbb{R}^{N\times 3}$ are established, we would get N point pairs, that is, N planes. Note that if there are identical eigenvalues of covariance matrix of $\mathcal{B}_i$, the corresponding eigenvectors are not uniquely determined, so we discard h_i in correspondence establishment. As a planar symmetry is likely to be related to many corresponding pairs, dropping pairs in such rare cases does not affect the performance of our method.

Symmetry Grouping. Inevitably, repeating or approximate symmetry planes can be identified in N planes of the point cloud $\mathcal{P}$. Therefore, we aggregate the planes to get a collection of candidate symmetry planes. We propose a novel plane grouping method named SymGroup that employs simple and effective graph clustering. The core idea is depicted in Figure 2 and Algorithm 1.

Algorithm 1: Symmetry plane clustering.

Input:
planes $\mathcal{S}=\{s_1, s_2, \ldots, s_N\}\in\mathbb{R}^{N\times 4}$;
distance threshold τ and cluster size threshold ξ.
Output:
clustered planes: $\mathcal{T}=\{t_1, t_2, \ldots, t_M\}\in\mathbb{R}^{M\times 4}$.

initialize graph $\mathcal{G}=(\mathbf{V}\leftarrow\mathcal{S}, \mathbf{E}\leftarrow\emptyset)$
for $i\leftarrow 1$ **to** N **do**
  initialize node weight $\mathbf{w}_i=0$
  for $j\leftarrow 1$ **to** N, $i\neq j$ **do**
    if $\delta_{ij}\leq\tau$ **then**
      assign an edge $e_{ij}=(s_i, s_j)$ with weight $w_{ij}=1-\delta_{ij}$
      $\mathbf{w}_i\leftarrow\mathbf{w}_i+w_{ij}$, $\mathbf{E}\leftarrow\mathbf{E}\cup\{e_{ij}\}$
    end if
  end for
end for
get connected components $\mathbf{C}$ by BFS algorithm
initialize clustered planes: $\mathcal{T}=\emptyset$
for $c\in\mathbf{C}$ **do**
  if $|c|\geq\xi$ **then**
    get the node t with maximum weight
    $\mathcal{T}\leftarrow\mathcal{T}\cup\{t\}$
  end if
end for
return $\mathcal{T}$

Since the parameters of a plane are of ambiguity, that is, (a, b, c, d) and $(-a, -b, -c, -d)$ actually define the same plane, given two planes $\mathbf{u}$ and $\mathbf{v}$, parameterized by $\mathbf{u}=(a_u, b_u, c_u, d_u)$ and $\mathbf{v}=(a_v, b_v, c_v, d_v)$, we define the distance function between them as

$$\delta_{\mathbf{uv}}=\min(\|\mathbf{u}-\mathbf{v}\|, \|\mathbf{u}+\mathbf{v}\|), \tag{9}$$

where $\|\cdot\|$ is the L_2 norm. Note that $\delta_{\mathbf{uv}}=\delta_{\mathbf{vu}}$, the

distance function taking as input two planes is a symmetric function. In particular, for each plane $s_i \in \mathcal{S}$ where $\mathcal{S}$ is the set of candidate planes, we calculate the distance δ_{ij} between s_i and $s_j, \forall j \neq i$, and we assign an edge $e_{ij} = (s_i, s_j)$ with weight $w_{ij} = 1 - \delta_{ij}$ if $\delta_{ij} < \tau$. And $\mathbf{w}_i = \sum_{\forall j \neq i} w_{ij}$ is the weight of the plane s_i. After iterating over all planes in $\mathcal{S}$, we build a weighted undirected graph, and the connected components $\mathbf{C}$ are found by BFS (Breadth First Search) algorithm. Then the plane with the maximum weight $\max_{i \in c_j} \mathbf{w}_i$, is the cluster center of each component $c_j \in \mathbf{C}$ if $|c_j| \geq \xi$, which constitutes the detected reflective symmetry planes $\mathcal{T}$. Formally, $\mathcal{T} = \text{SymGroup}(\mathcal{S})$. To make the SymGroup differential, we define the partial derivative of the output $t_j \in \mathcal{T}$ with respect to the input $s_i \in \mathcal{S}$ as:

$$\frac{\partial t_j}{\partial s_i} = \begin{cases} 1 & \text{if } s_i = t_j \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Network Training. To train the network unsupervisedly, we use the loss $L = \sum_{k=1}^{M} L_k$ similar as symmetry distance loss in PRS-Net [11] to train the network. In particular, L_k calculates the Chamfer Distance between input point cloud $\mathcal{P}$ and the mirror points $\mathcal{Q}_k$ w.r.t. the detected reflection plane $t_k \in \mathcal{T}$. It can be expressed by the following:

$$L_k = \frac{1}{|\mathcal{Q}_k|} \sum_{q_j \in \mathcal{Q}_k} \min_{p_i \in \mathcal{P}} \|q_j - p_i\| + \frac{1}{|\mathcal{P}|} \sum_{p_i \in \mathcal{P}} \min_{q_j \in \mathcal{Q}_k} \|p_i - q_j\|, \tag{11}$$

$$\mathcal{Q}_k = \{p_i - 2\frac{p_i \cdot \mathbf{n}_k + d_k}{\|\mathbf{n}_k\|^2}\mathbf{n}_k, p_i \in \mathcal{P}\}, \tag{12}$$

where t_k: $a_k x + b_k y + c_k z + d_k = 0$, and $\mathbf{n}_k = (a_k, b_k, c_k)$. Eq. 12 means applying the transformation to each point of $\mathcal{P}$ w.r.t. t_k to obtain the transformed point set $\mathcal{Q}_k$. During training, we set the batch size to be 32, learning rate $l_r = 0.001$, total epochs as 100, and use the ADAM [19] optimizer to train our network. E3sym is implemented in PyTorch [1] [34] and Jittor [2] [13].

3.3. Planar Symmetry Inference

Similar to PRS-Net [11], for two detected symmetry planes, if the dihedral angle is less than $\pi/6$, we remove the one with larger symmetry distance error. Also, we introduce a simple validation method in the inference stage to handle the issue that some approximate symmetries that are not good enough may be detected. Different from PRS-Net [11] that uses 0.0004 of symmetry distance error as the threshold to drop imperfect symmetries, our validation strategy depends on whether the input shape is expected to be complete or incomplete. If the input shape is complete, we use the same symmetry distance error threshold to filter out imperfect ones. When the input shape is incomplete with large region missing, the symmetry distance error may be large even when the detected symmetry plane is accurate enough. So for each detected symmetry plane t_k and for each point $p_i \in \mathcal{P}$, we find the nearest neighbor $p_i^{'}$ of its symmetrical point $q_i \in \mathcal{Q}_k$, if $p_i^{'}$ and q_i are close enough, we consider p_i to be an *inlier*. If the *inlier ratio* exceeds a given threshold, we regard t_k to be a valid symmetry.

[1] https://github.com/renwuli/e3sym
[2] https://github.com/renwuli/je3sym

4. Experiments

4.1. Datasets

ShapeNet. ShapeNet [2] offers a huge dataset containing 55 categories of man-made objects, 51300 objects in total. For training-test split, we follow the settings in PRS-Net [11], i.e., 80% models as the training set and the rest 20% as testing set, and the ground truth of symmetries are from PRS-Net [11] as well. For each shape in ShapeNet, we sample 512 points on the surface as the input of our method, which is randomly rotated during both training and testing.

MVP. Considering many scanned models are not perfectly complete in geometry, an experiment is performed to evaluate our method handling incomplete shapes. The incomplete point clouds are retrieved from MVP [33], a large dataset focusing on tasks of point cloud completion and partial-to-partial registration. MVP consists of 16 categories, and for each category, we randomly sample 100 point clouds for evaluation, 1600 point clouds in total. For each point cloud in MVP, we sample 512 points as the input of our method.

4.2. Metrics

Symmetry Distance Error. Following PRS-Net [11], we use Symmetry Distance Error (SDE) to measure how close the 3D shape is to its symmetrical copy with respect to a certain planar symmetry plane. And the SDE is defined as:

$$\text{SDE}(O, t_i) = \frac{1}{|O|} \sum_{o \in O} \min_{\hat{o} \in t_i(O)} \|o - \hat{o}\|_2^2. \tag{13}$$

O is the input 3D shape and t_i is a detected symmetry plane, $t_i(O)$ is the symmetrical copy of O. Eq. 13 is equivalent to the original SDE in PRS-Net [11] for meshes in ShapeNet, and equivalent to the Chamfer Distance [8] for point clouds in MVP.

F-score. To measure the effectiveness of correctly detecting symmetry planes, F-score is used as a metric. Given a detected symmetry plane t, if we can find a symmetry plane

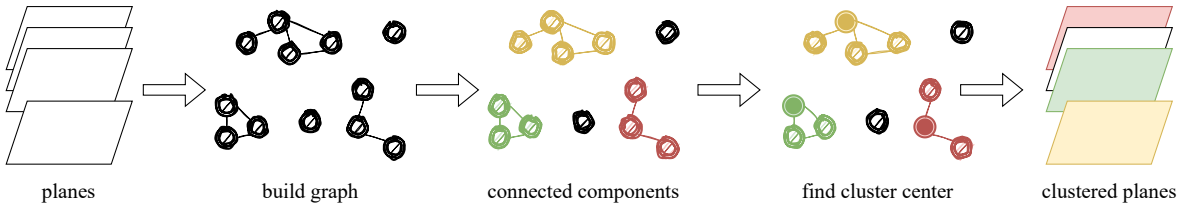


Figure 2: We presents a novel and effective planar symmetry grouping algorithm. We consider each plane established from point correspondences as a node and construct a weighted undirected graph by assigning edges between planes that are close to each other, with weights decreasing as the distances between planes increase. The weight of a node is defined as the sum of its adjacency edges. Using this graph representation, we identify connected components and treat each as a cluster. We extract the cluster centers by selecting nodes with the maximum weight and derive the corresponding clustered planes as detected symmetry planes.

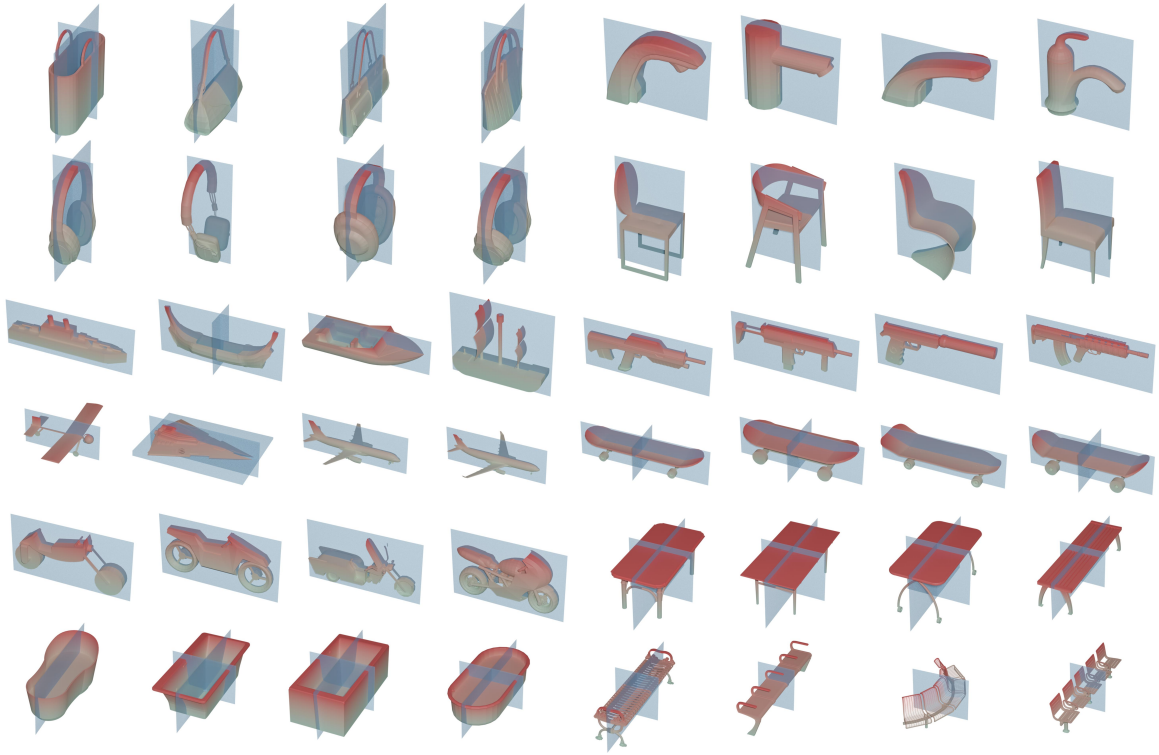


Figure 3: Qualitative results for reflective symmetry detection on the test set of ShapeNet [2]. From left to right, top to down are of the categories: bag, water tap, earphone, chair, boat, rifle, airplane, skateboard, motorcycle, table, bathtub and bench. 4 instances are displayed for each category. Our method is able to detect multiple symmetry planes accurately. For better visualization, we paint the shapes with gradient color.

Method	PCA	OBB [3]	RSD [16]	ADS [24]	PASD [27]	PRST [35]	PRST with GEDT [35]	PAS [20]	PRS-Net [11]	Ours
SDE ($\times 10^{-4}$)	3.32	1.25	0.90	3.95	14.2	1.78	1.60	1.75	0.86	**0.46**
F-score	0.692	0.740	0.684	0.694	0.322	0.619	0.646	0.678	0.712	**0.753**

Table 1: The SDE ($\times 10^{-4}$) and Recall(%) measured with different methods on ShapeNet [2].

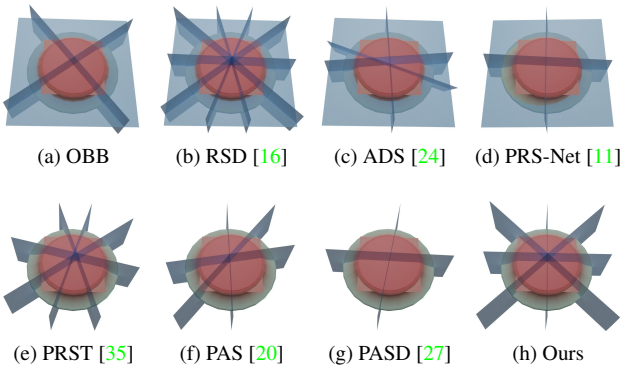
$\hat{t}$ among ground-truth symmetry plane set, such that the distance between them is small enough, we treat t as a true positive (TP), otherwise false positive (FP), if no detected symmetry plane is close to a ground-truth symmetry $\hat{t}$, we treat $\hat{t}$ as a false negative (FN). The distance function between two distinct planes is defined the same as Eq. 9. For a given plane distance threshold, the precision and recall are defined as $PR = \frac{TP}{TP+FP}$ and $RE = \frac{TP}{TP+FN}$ respectively, and F-score is computed by $\frac{2PR \times RE}{PR+RE}$. After iterating over all thresholds in the range of $[0, 0.2]$, we use the average F-score as the measurement.

4.3. Evaluation on Complete Shapes

Symmetry detection on ShapeNet. We take PCA, OBB [3], PSD. [16], ADS [24], PASD [27], PRST [35], PRST with GEDT (Gaussian Euclidean Distance Transform), PAS [20] and PRS-Net [11] as baselines. Our method is qualitatively evaluated in ShapeNet [2] with SDE against these baselines. Since models in ShapeNet are prealigned and share the common symmetry planes as a prior, we apply a random rotation to each model to demonstrate the effectiveness of our method more convincingly. Comparison results are shown in Table 1, from which we can observe that our method outperforms other methods. Qualitative results conducted on ShapeNet are also provided in Figure 3 for better comprehension, which displays 4 shapes in each category and 12 different categories in total. Our method is able to detect reasonable and accurate reflective symmetry planes. Although it is challenging to detect accurate symmetries for the rifles and motorcycles with thin structures, our method produces decent results as well. Figure 4 shows the visualization results compared with baselines, obviously, 4 distinct reflective symmetries are present in the challenging shape. The baseline methods detect symmetries either wrongly or inaccurately. Since PRS-Net [11] restricts the number of predictions, 2 valid and 1 invalid symmetries are detected, and our method is able to detect all the valid symmetries. This is also evidenced by our highest F-score when compared with existing methods.

4.4. Robustness on Incomplete Shapes

Although it is challenging, our method can also reveal reasonable planar symmetries from incomplete shapes. To evaluate the robustness on incomplete shapes, we conduct experiments on Partial ShapeNet and MVP that are composed of incomplete shapes, and the results are produced by the model pretrained on ShapeNet in Section 4.3. In order to validate the detected symmetry planes from incomplete shapes, we use *inlier ratio* introduced in Section 3.3. The *inlier ratio* should be lower (resp. higher) if the shape is with larger (resp. smaller) parts missing. We choose the values to be 0.9 and 0.8 for Partial ShapeNet and MVP respectively.

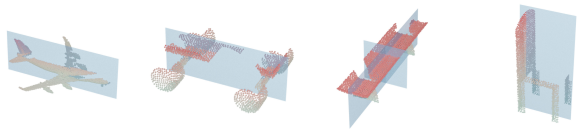


(a) OBB (b) RSD [16] (c) ADS [24] (d) PRS-Net [11]

(e) PRST [35] (f) PAS [20] (g) PASD [27] (h) Ours

Figure 4: Qualitative reflective symmetry detection results compared with different methods.

Evaluation on Partial ShapeNet. In order to further test the robustness for incomplete shapes, following the setting in PRS-Net [11], we evaluate our method on Partial ShapeNet, in which a randomly chosen contiguous region is removed for each shape in ShapeNet. Quantitative results are illustrated in Table 2, in which our method achieves the lowest SDE, which illustrates that our method can handle incomplete shapes well.



(a) airplane (b) skateboard (c) table (d) chair

Figure 5: Qualitative results for reflective symmetry detection on MVP. Though the input shape is incomplete, our method can still detect symmetries robustly and accurately.

Evaluation on MVP. In Partial ShapeNet from PRS-Net, a randomly selected contiguous region is removed from the surface, which typically has a limited impact on the primary geometric structure. However, the challenging MVP point clouds are generated from single-view depth maps, which are often sparse, with large portions of missing data. Consequently, the MVP dataset poses greater difficulty and presents more significant challenges. As shown in Figure 5, although the input shapes are incomplete to a varying degree, multiple symmetry planes are detected accurately and robustly by our method, which verifies that the proposed method can deal well with incomplete shapes with large parts missing. Because the corresponding complete shapes in MVP are available, symmetry distance errors (SDE) are computed based on them to compare with the baseline methods. Since PSD [16], ADS [24] and PASD [27] are not applicable to point clouds, we do not compare with them on

Method	PCA	OBB [3]	ADS [24]	PASD [27]	PAS [20]	PRST [35]	PRST with GEDT [35]	PRS-Net [11]	Ours
SDE ($\times 10^{-4}$)	6.67	2.40	7.58	28.72	4.34	4.61	3.77	1.54	**1.16**

Table 2: The symmetry distance error ($\times 10^{-4}$) measured with different methods on partial shape set of ShapeNet [2].

Method	PCA	OBB [3]	PAS [20]	PRST [35]	PRS-Net [11]	Ours
SDE	57.21	49.05	12.06	46.30	34.10	**8.50**
SDE*	57.21	49.05	32.56	89.61	46.13	**9.50**

Table 3: The SDE ($\times 10^{-4}$) measured with different methods on MVP [33]. SDE* measures the performance when the incomplete shape is randomly rotated.

MVP. The numerical results are listed in Table 3, the SDE measures the performance for aligned shapes and SDE* for the randomly rotated version. It reveals that the proposed method is robust with incomplete shapes and outperforms other methods. PCA and OBB [3] are independent to rotation transforms and get the same SDEs for aligned and rotated shapes. PAS [20] and PRST [35] detect symmetries for a shape by discretizing it to a grid, which is sensitive to rotation transforms. Because PRS-Net [11] uses the rotation augmentation for training, it is more robust than [20, 35] even though it takes voxels as input.

4.5. Ablation Studies and Visualization

Ablation study of SymGroup. To demonstrate the efficacy of our proposed SymGroup, we conduct experiments that replace our SymGroup with DBSCAN and Mean-Shift, which are implemented similar to SymGroup to make the gradients able to be back-propagated. Additionally, we also perform robustness testing to see the effects of parameters of the three clustering methods in Figure 6. For the clustering algorithm SymGroup, we set τ in the range of [0.02, 0.07] and ξ in the range of [8, 96], we evaluate the performance by SDE and F-score on ShapeNet and the results are visualized in Figure 6, from which (72, 0.05) reach the highest F-score and low SDE, while the difference for settings close to it is minimal, which proves the robustness of our proposed SymGroup. For DBSCAN and Mean-Shift, we apply similar settings that give them different parameters. We can clearly see that Mean-Shift is sensitive to the parameters and have significantly inferior performance compared with SymGroup. DBSCAN reaches higher F-score and lower SDE compared with Mean-Shift, and is more robust against parameters. Nonetheless, our proposed SymGroup algorithm surpasses DBSCAN in terms of both performance and robustness.

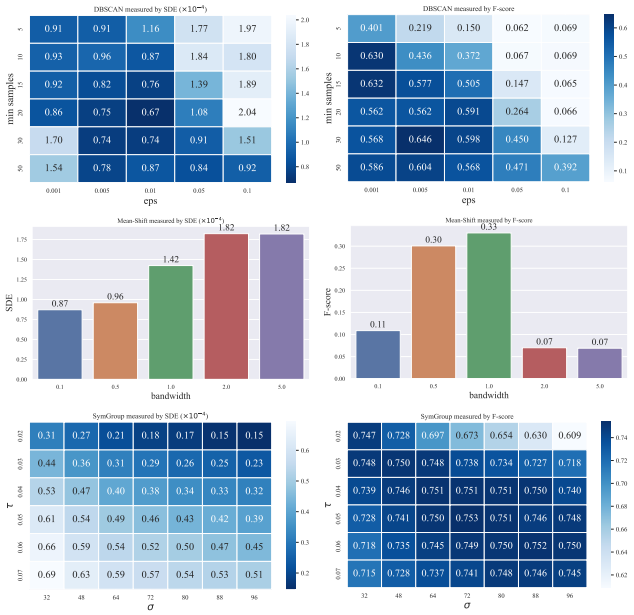


Figure 6: Robustness of different clustering methods evaluated by both SDE($\times 10^{-4}$) and F-score. Please zoom in for better visualization.

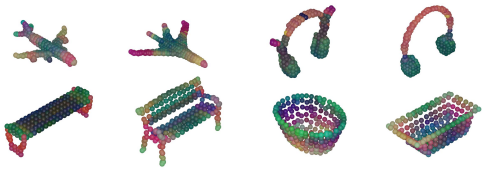


Figure 7: E(3) invariant feature visualization. Points lying in various positions and orientations but share similar geometry structures have a high degree of similarity.

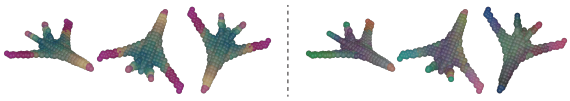


Figure 8: Left half: feature with frame averaging; right half: feature without frame averaing.

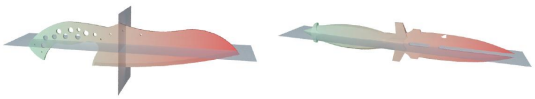
E(3) invariant feature visualization. In order to demonstrate the effectiveness of the introduced E(3) invariant features more intuitively, as displayed in Figure 7, we visualize

the E(3) invariant features extracted by φ. From left to right and top to bottom: airplane, earphone, bench and basket are shown respectively. The color of each point is reduced from the feature to 3D by PCA and linearly mapped to $[0, 1]$ (Min-Max Scaling). It can be clearly seen that, for points lying in various positions and orientations but share similar geometry structures, they have a high degree of similarity, implying symmetric correspondences can be reasoned from the extracted E(3) invariant features. In Figure 8, we randomly rotate a typical shape and present the E(3) invariant feature visualization results with and without frame averaging from 3 different viewpoints, which clearly shows the effects of frame averaging.

4.6. Computation Efficiency

We evaluated the proposed method's computational efficiency against several baseline methods using the same settings as PRS-Net [11]. Specifically, we evaluated the performance of RSD [16], ADS [24], PASD [27], PRST [35], PRST with GEDT [35], and PAS [20] on a typical model consisting of 1052 vertices and 4532 faces. We observed that the aforementioned methods take 0.51, 2.82, 3.40, 5.00, 0.97, and 0.42 seconds, respectively. OBB and PCA methods, which use simple geometry processing, are much faster at 0.02 seconds and 1.9 ms, respectively, but less robust. In contrast, our proposed method takes 0.027 seconds. As a method that also utilizes neural networks, PRS-Net [11] exhibits a low latency of 1.81 ms due to its simple network architecture composed of several convolutional layers and lightweight fully connected layers that can be efficiently parallelized. Despite being slower than the previous state of the art, PRS-Net [11], our approach is fast enough for real-time processing (27 ms, 37 FPS) and achieves the best accuracy compared to other baselines. Notably, our model contains significantly fewer trainable parameters (2.25k) than PRS-Net (79.8k), which provides further evidence for the efficiency of our proposed approach.

5. Conclusion



(a) (b)

Figure 9: Two examples of failure cases. Our method fails to handle shapes with extremely thin and flat structures.

We present E3Sym, a novel approach to detect 3D global planar symmetry that is end-to-end unsupervised and does not require any labeled data. By extracting E(3) invariant features, the symmetric correspondences can be established efficiently and robustly, from which the dense global planar symmetries are determined. By applying SymGroup which produces clustered symmetry planes, the network can be trained using symmetry distance loss in an unsupervised manner. However, as shown in Figure 9, the correspondences for shapes like knives with very thin and flat structures are extremely hard to handle, and the reflective symmetry planes parallel to the paper are not successfully detected. We will address such difficult cases in future work.

Acknowledgement

This work was supported in part by the National Key Research and Development Program of China (No. 2018AAA0103002), in part by the Beijing Municipal Natural Science Foundation for Distinguished Young Scholars (No. JQ21013), in part by the National Natural Science Foundation of China (No. 62061136007), in part by the Youth Innovation Promotion Association, CAS.

References

[1] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L. Crowley. Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2209–2218, 2017. 1

[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. 5, 6, 7, 8

[3] Chia-Tche Chang, Bastien Gorissen, and Samuel Melchior. Fast oriented bounding box optimization on the rotation group SO (3, r). *ACM Transactions on Graphics (TOG)*, 30(5):1–16, 2011. 6, 7, 8

[4] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3D point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021. 3

[5] Jean-Baptiste Cordonnier, Aravindh Mahendran, Alexey Dosovitskiy, Dirk Weissenborn, Jakob Uszkoreit, and Thomas Unterthiner. Differentiable patch selection for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2351–2360, 2021. 3, 4

[6] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J. Guibas. Vector neurons: A general framework for so(3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12200–12209, October 2021. 3

[7] Aleksandrs Ecins, Cornelia Fermuller, and Yiannis Aloimonos. Detecting reflectional symmetries in 3D data through symmetrical fitting. In *Proceedings of the IEEE Interna-*

tional Conference on Computer Vision Workshops, pages 1779–1783, 2017. 1

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

[9] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020. 3

[10] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh. *ACM Transactions on Graphics (TOG)*, 38(6):1–15, 2019. 1

[11] Lin Gao, Ling-Xiao Zhang, Hsien-Yu Meng, Yi-Hui Ren, Yu-Kun Lai, and Leif Kobbelt. PRS-Net: Planar reflective symmetry detection net for 3D models. *IEEE Transactions on Visualization and Computer Graphics*, 27(6):3007–3018, 2020. 1, 2, 5, 6, 7, 8, 9

[12] Lukáš Hruda, Ivana Kolingerová, and Libor Váša. Robust, fast and flexible symmetry plane detection based on differentiable symmetry measure. *The Visual Computer*, 38(2):555–571, 2022. 2

[13] Shi-Min Hu, Dun Liang, Guo-Ye Yang, Guo-Wei Yang, and Wen-Yang Zhou. Jittor: a novel deep learning framework with meta-operators and unified graph execution. *Science China Information Sciences*, 63(222103):1–21, 2020. 5

[14] Weihua Hu, Muhammed Shuaibi, Abhishek Das, Siddharth Goyal, Anuroop Sriram, Jure Leskovec, Devi Parikh, and C Lawrence Zitnick. ForceNet: A graph neural network for large-scale quantum calculations. *arXiv preprint arXiv:2103.01436*, 2021. 3

[15] Penglei Ji and Xinguo Liu. A fast and efficient 3D reflection symmetry detector based on neural networks. *Multimedia Tools and Applications*, 78(24):35471–35492, 2019. 1, 2

[16] Michael Kazhdan, Bernard Chazelle, David Dobkin, Adam Finkelstein, and Thomas Funkhouser. A reflective symmetry descriptor. In *European Conference on Computer Vision*, pages 642–656. Springer, 2002. 1, 2, 6, 7, 9

[17] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. Symmetry descriptors and 3D shape matching. In *Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry processing*, pages 115–123, 2004. 1

[18] Vladimir G Kim, Yaron Lipman, Xiaobai Chen, and Thomas Funkhouser. Möbius transformations for global intrinsic symmetry analysis. In *Computer Graphics Forum*, volume 29, pages 1689–1700. Wiley Online Library, 2010. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Simon Korman, Roee Litman, Shai Avidan, and Alex Bronstein. Probably approximately symmetric: Fast rigid symmetry detection with global guarantees. In *Computer Graphics Forum*, volume 34, pages 2–13. Wiley Online Library, 2015. 1, 2, 6, 7, 8, 9

[21] Feiran Li, Kent Fujiwara, Fumio Okura, and Yasuyuki Matsushita. A closer look at rotation-invariant deep point cloud analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16218–16227, 2021. 3

[22] Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, and Thomas Funkhouser. Symmetry factored embedding and distance. *ACM Transactions on Graphics (SIGGRAPH 2010)*, July 2010. 2

[23] Yaron Lipman and Thomas Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):1–12, 2009. 2

[24] Aurélien Martinet, Cyril Soler, Nicolas Holzschuch, and François X Sillion. Accurate detection of symmetries in 3D shapes. *ACM Transactions on Graphics (TOG)*, 25(2):439–464, 2006. 2, 6, 7, 8, 9

[25] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. VV-Net: Voxel VAE net with group convolutions for point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8500–8508, 2019. 3

[26] Yongwei Miao, Feixia Hu, Xudong Zhang, Jiazhou Chen, and Renato Pajarola. SymmSketch: Creating symmetric 3D free-form shapes from 2D sketches. *Computational Visual Media*, 1:3–16, 2015. 1

[27] Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3D geometry. *ACM Transactions on Graphics (TOG)*, 25(3):560–568, 2006. 1, 2, 6, 7, 8, 9

[28] Ryan L. Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. In *International Conference on Learning Representations*, 2019. 3

[29] Rajendra Nagar and Shanmuganathan Raman. Fast and accurate intrinsic symmetry detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 417–434, 2018. 2

[30] Rajendra Nagar and Shanmuganathan Raman. Detecting approximate reflection symmetry in a point set using optimization on manifold. *IEEE Transactions on Signal Processing*, 67(6):1582–1595, 2019. 2

[31] Rajendra Nagar and Shanmuganathan Raman. 3DSymm: robust and accurate 3D reflection symmetry detection. *Pattern Recognition*, 107:107483, 2020. 1

[32] Maks Ovsjanikov, Jian Sun, and Leonidas Guibas. Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library, 2008. 2

[33] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8524–8533, 2021. 5, 8

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An

imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[35] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graph.*, 25(3):549–559, 2006. 2, 6, 7, 8, 9

[36] Omri Puny, Matan Atzmon, Edward J. Smith, Ishan Misra, Aditya Grover, Heli Ben-Hamu, and Yaron Lipman. Frame averaging for invariant and equivariant network design. In *International Conference on Learning Representations*, 2022. 3, 4

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2, 3, 4

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 3

[39] Yi-Ling Qiao, Lin Gao, Shu-Zhi Liu, Ligang Liu, Yu-Kun Lai, and Xilin Chen. Learning-based intrinsic reflectional symmetry detection. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2022. 2

[40] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pages 9323–9332. PMLR, 2021. 3

[41] Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 3D-rotation-equivariant quaternion neural networks. In *European Conference on Computer Vision*, pages 531–547. Springer, 2020. 3

[42] 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 (TOG)*, 39(6):1–14, 2020. 1, 2

[43] Zeyun Shi, Pierre Alliez, Mathieu Desbrun, Hujun Bao, and Jin Huang. Symmetry and orbit detection via Lie-algebra voting. In *Computer Graphics Forum*, volume 35, pages 217–227. Wiley Online Library, 2016. 1

[44] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, and C Lawrence Zitnick. Rotation invariant graph neural networks using spin convolutions. *arXiv preprint arXiv:2106.09575*, 2021. 3

[45] Patricio Simari, Evangelos Kalogerakis, and Karan Singh. Folding meshes: hierarchical mesh segmentation based on planar symmetry. In *Symposium on Geometry Processing*, pages 111–119, 2006. 1

[46] Ivan Sipiran, Robert Gregor, and Tobias Schreck. Approximate symmetry detection in partial 3D meshes. In *Computer Graphics Forum*, volume 33, pages 131–140. Wiley Online Library, 2014. 1, 2

[47] Pablo Speciale, Martin R Oswald, Andrea Cohen, and Marc Pollefeys. A symmetry prior for convex variational 3D reconstruction. In *European Conference on Computer Vision*, pages 313–328. Springer, 2016. 1

[48] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6410–6419, 2019. 3

[49] Hui Wang and Hui Huang. Group representation of global intrinsic symmetries. In *Computer Graphics Forum*, volume 36, pages 51–61. Wiley Online Library, 2017. 1

[50] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3523–3532, 2019. 3

[51] 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. 3

[52] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D steerable cnns: Learning rotationally equivariant features in volumetric data. *Advances in Neural Information Processing Systems*, 31, 2018. 3

[53] Kai Xu, Hao Zhang, Wei Jiang, Ramsay Dyer, Zhiquan Cheng, Ligang Liu, and Baoquan Chen. Multi-scale partial intrinsic symmetry detection. *ACM Transactions on Graphics (TOG)*, 31(6):1–11, 2012. 2

[54] Kai Xu, Hao Zhang, Andrea Tagliasacchi, Ligang Liu, Guo Li, Min Meng, and Yueshan Xiong. Partial intrinsic reflectional symmetry of 3D shapes. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. 2009. 2

[55] Dmitry Yarotsky. Universal approximations of invariant maps by neural networks. *Constructive Approximation*, pages 1–68, 2021. 3

[56] Hagit Zabrodsky, Shmuel Peleg, and David Avnir. Symmetry as a continuous feature. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(12):1154–1166, 1995. 2