SymmetryNet: Learning to Predict Reflectional and Rotational Symmetries of 3D Shapes from Single-View RGB-D Images

YIFEI SHI, National University of Defense Technology

JUNWEN HUANG, National University of Defense Technology

HONGJIA ZHANG, National University of Defense Technology

XIN XU, National University of Defense Technology

SZYMON RUSINKIEWICZ, Princeton University

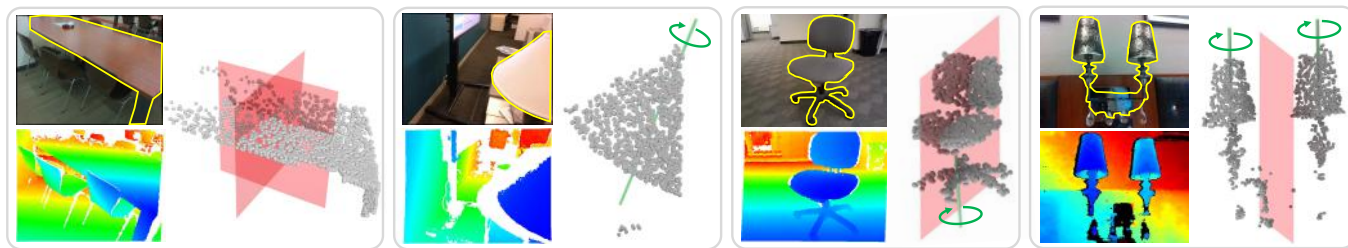KAI XU*, National University of Defense Technology

Fig. 1. We propose an end-to-end deep neural network learned to predict both reflectional and rotational symmetries from single-view RGB-D images. For each example, we show the input RGB-D images with the object of interest segmented out (see the yellow masks) as well as the detection results over the unprojected 3D point clouds. Reflectional symmetries are depicted with red planes (reflection plane) and rotational symmetries with green lines (rotation axis). Note how our method is able to detect the composition of an arbitrary number of symmetries, possibly of different types, present in the same object.

We study the problem of symmetry detection of 3D shapes from single-view RGB-D images, where severely missing data renders geometric detection approach infeasible. We propose an end-to-end deep neural network which is able to predict both reflectional and rotational symmetries of 3D objects present in the input RGB-D image. Directly training a deep model for symmetry prediction, however, can quickly run into the issue of overfitting. We adopt a multi-task learning approach. Aside from symmetry axis prediction, our network is also trained to predict symmetry correspondences. In particular, given the 3D points present in the RGB-D image, our network outputs for each 3D point its symmetric counterpart corresponding to a specific predicted symmetry. In addition, our network is able to detect for a given shape multiple symmetries of different types. We also contribute a benchmark of 3D symmetry detection based on single-view RGB-D images.

Extensive evaluation on the benchmark demonstrates the strong generalization ability of our method, in terms of high accuracy of both symmetry axis prediction and counterpart estimation. In particular, our method is robust in handling unseen object instances with large variation in shape, multi-symmetry composition, as well as novel object categories.

Additional Key Words and Phrases: Symmetry prediction, Neural networks, Counterpart prediction

*Corresponding author: Kai Xu (kevin.kai.xu@gmail.com)

Authors' addresses: Yifei Shi, National University of Defense Technology; Junwen Huang, National University of Defense Technology; Hongjia Zhang, National University of Defense Technology; Xin Xu, National University of Defense Technology; Szymon Rusinkiewicz, Princeton University; Kai Xu, National University of Defense Technology.

1 INTRODUCTION

Symmetry is omnipresent in both nature and the synthetic world. Symmetry detection is therefore a long-standing problem that has attracted substantial attention in both computer graphics and vision [Liu et al. 2010; Mitra et al. 2013]. Symmetry is at heart a purely geometric concept, with a rigorous definition on the basis of transformation invariance and group theory. It might therefore be supposed that symmetry detection can always be solved by a purely

geometric approach. For example, reflectional symmetry in 2D or 3D can be easily parameterized in the transformation space. As such, detection methods such as the Hough transform have historically been utilized to accumulate local cues of symmetry transformations based on detected symmetry point correspondences [Mitra et al. 2006; Podolak et al. 2006; Yip 2000].

If we consider the problem of symmetry detection in the presence of significant missing data, however, it becomes appropriate to abandon purely-geometric approaches and *infer* what symmetries might be present. A common application scenario is estimating symmetries of 3D shapes based on a single-view RGB-D image. Single-view symmetry detection finds various potential applications ranging from object/scene completion, camera tracking, and relocalization, to object pose estimation. Due to partial observations and object occlusion, it also poses special challenges that are beyond the reach of geometric detection. For example, it is difficult, if not impossible, to find local symmetry correspondences and transformations supporting global symmetry analysis. In this situation, symmetry analysis should rely not only on geometric detection but also on statistical inference. The latter necessitates data-driven learning.

In this work, we propose an end-to-end learning approach for symmetry prediction based on a single RGB-D image using deep neural networks. As shown in Figure 1, given an RGB-D image as input, the network is trained to detect two types of 3D symmetries present in the scene, namely (planar) reflectional and (cylindrical) rotational symmetries, and outputs the corresponding symmetry planes and axes, respectively. Directly training a deep model for symmetry prediction, however, can quickly run into the issue of overfitting. This is due to the fact that the network is able to easily "memorize" the symmetry axes of a class of objects in training and will simply perform object recognition at test time. Such an overfitted model cannot generalize well to large shape variation or changes in the symmetries that are present. In fact, symmetry is not a global shape property but rather is supported by local geometric cues: symmetry transformation invariance is defined by local shape correspondences. Straightforward training of symmetry prediction cannot help the network to truly understand the local-to-global support.

To this end, we adopt a multi-task learning approach. Aside from symmetry axis prediction, our network is also trained to predict symmetry correspondences. In particular, given the 3D points present in the RGB-D image, our network outputs for each 3D point its symmetric counterpart corresponding to a specific predicted symmetry. Given any point P_i, its counterpart may lie in the 3D point cloud or be missing due to occlusion or limited field of view. To accommodate both cases, we output the counterpart in two modalities. First, we output a heat map for each point in the point cloud indicating the probability of any other points being the symmetric counterpart of this point. Second, we directly regress the (x, y, z) location of this counterpart. To avoid overfitting, we correlate the two tasks by supervising their learning with a unified supervision signal, i.e., the ground-truth local position of the counterparts. We devise a loss that encourages a point with high counterpart probability to be spatially close to its corresponding ground-truth location.

Through proper parameterization, our network is able to handle reflectional symmetry, as well as continuous and discrete rotational symmetry. Since the number of symmetries present in a 3D shape may vary, a network with single output is not suitable for the symmetry prediction task. To this end, we design our network to produce multiple symmetry outputs. When training the network, however, one needs to know how to match the outputs to different ground-truth symmetries in order to compute proper prediction error for gradient propagation. This is achieved by an optimal assignment process, which keeps the entire network end-to-end trainable.

Through extensive evaluation on three symmetry prediction datasets, we demonstrate the strong generalization ability of our method. It attains high accuracy not only for symmetry axis prediction, but also for counterpart estimation. Therefore, our method is robust in handling unseen object instances with large variation in shape, multi-symmetry composition, as well as novel object categories. In summary, we make the following contributions:

- We propose the problem of reflective and rotational symmetry detection from single-view RGB-D images, and introduce a robust solution based on deep learning.
- We use a series of dedicated tasks (losses) to guide the deep network to learn not only parametrized symmetry axes but also the local symmetry correspondences that support them.
- We realize end-to-end learning of multi-symmetry detection by devising an optimal assignment process for multi-output network training.
- We propose a benchmark for single-view symmetry detection, encompassing a moderately-sized dataset containing both real and synthetic scenes, as well as evaluation metrics.

2 RELATED WORK

Symmetry detection has a large body of literature, which has been comprehensively reviewed by the two excellent surveys of Liu et al. [2010] and Mitra et al. [2013]. Here we focus only on the work that is most related to our specific designs and techniques.

2D symmetry detection. 2D symmetry detection has long been a major topic of interest in computer vision. Many approaches have been proposed, including direct methods [Kuehnle 1991], voting-based methods [Ogawa 1991], and moment-based methods [Marola 1989]. Different types of primitive symmetries such as rotation, translation, and reflection, as well as symmetry groups [Liu and Collins 2000], have been studied. Among all these directions, the detection of bilateral reflectional symmetry and its skewed version [Liu and Collins 2001] from 2D images has received the most attention from the community. Our work is relatively closely related to the detection of skewed bilateral symmetry, since the latter is inherently inferring reflectional symmetry of 3D objects from their 2D projections. In contrast to our work, these works do not output symmetries in 3D space and rely on the presence of a large portion of the symmetric regions.
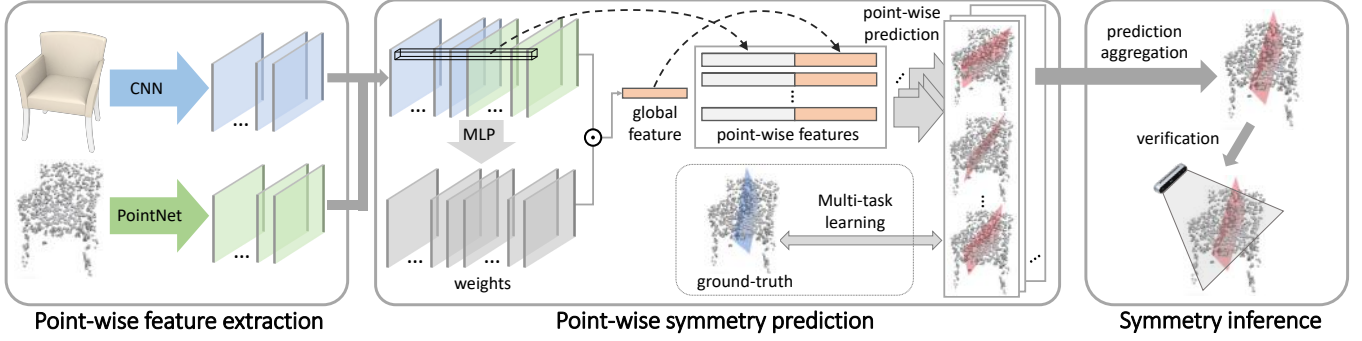
Fig. 2. The pipeline of our proposed symmetry prediction method comprising three major components. Taking an RGB image and a depth image as input, the network first extract point-wise appearance and geometry feature maps. The features are then used for point-wise symmetry prediction via multi-task learning. The final symmetry predictions are made with symmetry aggregation and visibility-based verification.

3D symmetry detection. Since the two seminal works of Mitra et al. [2006] and Podolak et al. [2006], symmetry detection of 3D geometry has attracted much attention in the field of geometry processing. Existing works can be categorized according to different problem settings targeted, such as exact vs. approximate symmetry, local vs. global symmetry, and extrinsic vs. intrinsic symmetry. Different combinations of the settings lead to different problems and approaches, such as the detection of extrinsic global symmetries [Martinet et al. 2006; Podolak et al. 2006], extrinsic partial symmetries [Bokeloh et al. 2009; Lipman et al. 2010; Mitra et al. 2006], intrinsic global symmetries [Ovsjanikov et al. 2008; Raviv et al. 2007], and intrinsic partial symmetries [Raviv et al. 2010; Xu et al. 2009]. Common to these works is the reliance on 3D shape correspondence [Van Kaick et al. 2011], which is regarded as a primary building block of 3D symmetry detection. However, in cases of significant missing data, such as single-view scans, shape correspondence becomes extremely challenging. Continuous rotational and spherical symmetries can be detected locally by slippage analysis [Gelfand and Guibas 2004]. However, it would have to be combined with different methods for other symmetry types.

Learning-based symmetry detection. Early methods for symmetry detection using statistical learning include the use of feed-forward networks to detect and enhance edges that are symmetric in terms of edge orientation [Zielke et al. 1992]. Tsogkas and Kokkinos [2012] employ hand-crafted features and multiple instance learning to detect ribbon-like structures in natural images, which was later extended to detect more general reflectional symmetry [Shen et al. 2016a]. Teo et al. [2015] utilize structured random forests to detect curved reflectional symmetries. Most recently, deep learning has been adopted for the task of 2D symmetry detection [Ke et al. 2017; Shen et al. 2016b], typically detecting reflectional symmetries as 2D skeletons instead of symmetries in 3D space.

Gao et al. [2019] propose PRS-Net, the first deep learning based symmetry detection method for 3D models demonstrating excellent results. They develop a loss function to measure symmetry correspondence that requires the counterpart of any point to lie on the shape surface. This limits their use in handling single-view scans: the reflective counterpart of a point may be far away from

the surface due to missing data, which may lead to high loss and slow convergence.

Learning-based shape correspondence. Deep learning has also been applied to shape correspondence. Existing works mostly focus on learning-based shape descriptors [Huang et al. 2017], which have proven more robust than hand-crafted ones. Wei et al. [2016] learn feature descriptors for each pixel in a depth scan of a human for establishing dense correspondences. Zeng et al. [2017] learn a local patch descriptor for volumetric data, which can be used for aligning depth data for RGB-D reconstruction. Although data-driven local shape descriptors can be used for symmetry detection, it is unclear how to harness them to realize an end-to-end learned symmetry detector. Moreover, severe data incompleteness renders shape correspondence inapplicable.

Learning-based object pose estimation. Our work is also related to 6D object pose estimation based on single-view RGB(D) input, since pose and symmetry usually imply each other. Most existing methods are instance-level [Avetisyan et al. 2019; Choi and Christensen 2012; Georgakis et al. 2018; Hodaň et al. 2015; Konishi et al. 2018; Peng et al. 2019; Wang et al. 2019a] and require a template 3D model, which is unavailable for our single-view symmetry detection task. Recently, Wang et al. [Wang et al. 2019b] achieve category-level 6D pose estimation based on a Normalized Object Coordinate Space (NOCS), a shared canonical representation of object instances within a category. They train a neural network to directly infer the pixel-wise correspondence between an RGB image and the NOCS. 6D object pose is estimated using shape matching. This method finds difficulty in generalizing across different shape categories. Our network, on the other hand, attains satisfying cross-category generality on symmetry prediction making it suited for pose estimation.

3 METHOD

The symmetry of a 3D object is easily measurable when its geometry is fully known. Conventional symmetry detection pipelines for 3D objects normally establish symmetric correspondences within the observed geometrical elements (e.g. points or parts) before aggregating them into meaningful symmetries. However, single-view

observations of real-world objects are usually incomplete due occlusion and limited field of view. Symmetry detection on incomplete geometry is an ill-posed problem which is difficult to solve with existing approaches.

When inferring the underlying symmetries of an incompletely observed object, humans usually resolve ambiguities based on whether the object is familiar. For an object commonly encountered in daily life, a person recognizes its category, estimates its pose, and determines the symmetries, all based on prior knowledge. For a novel, rarely-encountered object, however, she may look for local evidence of symmetry (i.e., establish symmetry correspondences) over the observed geometry and/or *imagined* unseen parts. Clearly, symmetry inference for novel objects is much harder since it involves *simultaneous* shape matching and shape completion. In this work, we propose a unified solution to single-view symmetry detection for both known and novel objects through coupling the predictions of symmetries and symmetry correspondences.

Our solution is to train an end-to-end network for symmetry prediction; see Figure 2. The network consists of three major components. The first module takes an RGB image and a depth image as input and extracts point-wise appearance and geometric features, respectively. These features are subsequently utilized for point-wise symmetry prediction. Finally, the third module performs symmetry aggregation and verification during inference.

3.1 Problem definition

Given an RGB-D image of an 3D object, our goal is to detect its extrinsic reflectional and/or rotational symmetries, if any. In particular, we detect at most M^{ref} reflectional symmetries $\mathcal{S}^{\text{ref}} = \{S_i^{\text{ref}}\}_{i=1,\dots,M^{\text{ref}}}$, which is parameterized as $S_i^{\text{ref}} = \{\mathbf{p}_i^{\text{ref}}, \mathbf{n}_i^{\text{ref}}\}$ with $\mathbf{p}_i^{\text{ref}}$ being a point in the reflection plane and $\mathbf{n}_i^{\text{ref}}$ the plane normal. We also detect at most M^{rot} rotational symmetries $\mathcal{S}^{\text{rot}} = \{S_i^{\text{rot}}\}_{i=1,\dots,M^{\text{rot}}}$, parameterized as $S_i^{\text{rot}} = \{\mathbf{p}_i^{\text{rot}}, \mathbf{n}_i^{\text{rot}}\}$ where $\mathbf{p}^{\text{rot}}$ is a point lying on the rotation axis and $\mathbf{n}^{\text{rot}}$ defines the axis orientation. All symmetries are represented in the camera reference frame.

3.2 Symmetry prediction network

Let us first introduce how the network predicts one symmetry, and then extend it to output multiple symmetries.

Dense-point symmetry prediction. We first extract features for both RGB and depth images and then fuse their feature maps. Following [Wang et al. 2019a], we extract point-wise appearance and geometric features using a fully-convolutional network [Wang et al. 2019a] and a PointNet [Qi et al. 2017], respectively. The two features are then concatenated and used for point-wise prediction tasks. Our network makes individual predictions for each point before aggregating all the predictions to form the final one. The overall prediction loss is $\mathcal{L} = \frac{1}{N} \sum_i^N \mathcal{L}_i$, where $\mathcal{L}_i$ is the prediction loss of point P_i.

Since symmetry is non-local, both local shape properties and global shape structure are crucial for its detection. Therefore, the
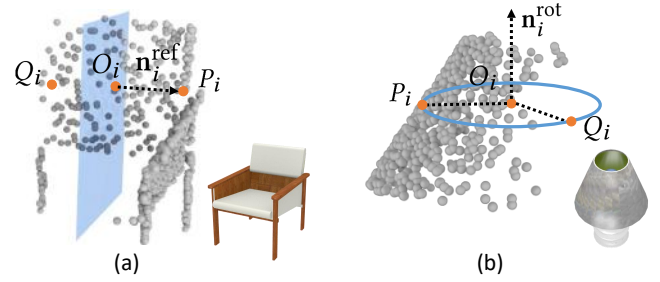

Fig. 3. Demonstration of the relationship between the to-be-predicted point P_i, its counterpart Q_i, and O_i which is the projection of P_i onto (a) the predicted symmetry plane with the plane normal $\mathbf{n}_i^{\text{ref}}$ for reflectional symmetry and (b) the predicted symmetry axis with the unit vector $\mathbf{n}_i^{\text{rot}}$ for rotational symmetry.

point-wise prediction takes both point-wise and global features as input. To compute global features, a straightforward way would be to perform average- or max-pooling over all point features. However, average-pooling over all points is redundant for symmetry detection, which can be determined by features of sparse points [Mitra et al. 2006]. On the other hand, max-pooling may lose too much information. We instead opt for spatially weighted pooling [Hu et al. 2017]. This method measures the significance of the each point by learning a weighted mask for every feature map. We insert a spatially weighted pooling layer after the appearance and geometric feature extraction layers. The resulting global feature is then concatenated with the point-wise features for symmetry prediction.

To improve the prediction accuracy and generality, we train the point-wise symmetry prediction network with a multi-task learning scheme. In particular, the tasks include 1) a classification of *symmetry type* (null if there is no symmetry), 2) a regression predicting the *symmetry parameters*, 3) a regression estimating *the location of the symmetric counterpart* of a given point for the corresponding symmetry, and 4) a classification indicating *whether an input point is the symmetric counterpart of the current point*. To make the point-wise prediction easy to train, all predicted coordinates are represented in a local reference frame centered at the current point, with the same orientation as the camera reference frame.

Although the extra tasks of symmetric counterpart prediction make the point-wise symmetry detection over-constrained, they allow the network to learn the essence of symmetry (i.e., symmetry correspondence) via reinforcing the relation between symmetry parameters and symmetric counterparts.

Given a point P_i, its symmetry prediction loss is defined as

$$\mathcal{L}_i = \mathcal{L}_i^{\text{type}} + \mathcal{L}_i^{\text{sym}}, \tag{1}$$

where $\mathcal{L}_i^{\text{type}}$ is the cross-entropy loss for symmetry type classification (null (0) for no symmetry, 1 for reflectional symmetry, 2 for rotational symmetry). $\mathcal{L}_i^{\text{sym}}$ is the loss for symmetry parameters and symmetric counterparts calculated based on the ground-truth

symmetry type:

$$\mathcal{L}_i^{\text{sym}} = \begin{cases} \mathcal{L}_i^{\text{ref\_reg}} + w^{\text{ref}} \cdot \mathcal{L}_i^{\text{ref\_cp}}, & \text{if ref. sym.} \\ \mathcal{L}_i^{\text{rot\_reg}} + w^{\text{rot}} \cdot \mathcal{L}_i^{\text{rot\_cp}}, & \text{if rot. sym.} \\ 0, & \text{if no sym.} \end{cases} \quad (2)$$

For reflectional symmetry (Figure 3a), the network outputs O_i, the projection of P_i onto the predicted symmetry plane, and measures the distance between this point and its ground-truth location $\hat{O}_i$:

$$\mathcal{L}_i^{\text{ref\_reg}} = d^2(O_i, \hat{O}_i), \quad (3)$$

where d is Euclidean distance. In the local reference frame of P_i, the predicted normal of the symmetry plane $\mathbf{n}_i^{\text{ref}}$ is $\frac{\overrightarrow{P_i O_i}}{|\overrightarrow{P_i O_i}|}$.

The counterpart loss for reflectional symmetry is:

$$\mathcal{L}_i^{\text{ref\_cp}} = \frac{1}{N} \sum_j^N \mathcal{L}^{\text{cls}}(p_{ij}, \hat{p}_{ij}) + d^2(Q_i, \hat{Q}_i), \quad (4)$$

where p_{ij} is the predicted probability that point P_j is the counterpart of point P_i and $\hat{p}_{ij}$ is the ground-truth label (0 for negative and 1 for positive). $\mathcal{L}^{\text{cls}}$ is the cross-entropy loss. Q_i is the predicted symmetric reflection (counterpart) of P_i, and $\hat{Q}_i$ is its corresponding ground-truth. The counterpart loss penalizes when a point with high counterpart probability is spatially distant from the corresponding ground-truth counterpart location. The weight w^{ref} tunes the importance of counterpart prediction.

For rotational symmetry (Figure 3(b)), the network predicts point O_i as the projection of P_i onto the predicted symmetry axis $\tilde{\mathbf{n}}_i^{\text{rot}}$. We define the rotational symmetry prediction loss as:

$$\mathcal{L}_i^{\text{rot\_reg}} = d^2(O_i, \hat{O}_i) + \left|1 - |\mathbf{n}_i^{\text{rot}} \cdot \hat{\mathbf{n}}_i^{\text{rot}}|\right|, \quad (5)$$

where $\hat{O}_i$ and $\hat{\mathbf{n}}_i^{\text{rot}}$ are the corresponding ground-truths.

Unlike reflectional symmetry, in which each point has only one symmetric counterpart, rotational symmetry induces more than one counterpart on the rotational orbit. For discrete rotational symmetry, the number of counterparts is equal to the order of rotational symmetry. For continuous rotational symmetry, on the other hand, the number is infinite. Learning to regress all points on the rotation orbit is extremely difficult if not impossible. We therefore opt to predict the probability for a given input point how likely it is to be in the rotation orbit. In addition, we predict the order r of the rotational symmetry ($r = 0$ for continuous rotational symmetry and $r > 0$ for discrete rotational symmetry) using an MLP and a softmax layer for R-way classification. R is the maximal order of discrete rotational symmetry in the datatset. We set $R = 10$ in our experiment. Note that with this formulation we unify the prediction of continuous and discrete symmetry, leading to reduced model parameters.

The counterpart loss for rotational symmetry is:

$$\mathcal{L}_i^{\text{rot\_cp}} = \frac{1}{N} \sum_j^N \mathcal{L}^{\text{cls}}(p_{ij}^o, \hat{p}_{ij}^o) + \mathcal{L}^{\text{cls}}(r_i, \hat{r}_i), \quad (6)$$

where p_{ij}^o is the predicted probability that a point P_j lies in the ground-truth orbit of P_i, r_i is the predicted order, $\hat{p}_{ij}^o$ and $\hat{r}_i$ are the corresponding ground-truths, and w^{rot} is a trade-off weight.

Handling arbitrary number of symmetries. To accommodate multiple symmetries in our network, one option is to train a recurrent neural network which is able to output an arbitrary number of symmetries sequentially. However, training such a sequential prediction requires a prescribed consistent order for the symmetries, which is obviously infeasible. A more straightforward option is to have an M-way output with M being the maximum number of symmetries per object, which we adopt in our approach. However, training a network with an M-way output still requires predefining the order of different outputs. To circumvent this order dependency, we propose an optimal assignment based approach to train the network for order-independent multi-way output.

In particular, each point produces a maximum of M^{ref} outputs for reflectional symmetry or M^{rot} for rotational symmetry. A classifier is used to determine the presence or absence of each symmetry. For those symmetries verified by the classifier, an optimization is applied to find the maximally-beneficial-matching to the ground-truth symmetries. To be specific, we solve the following optimization during training:

$$\underset{\Pi}{\arg\max} \sum_{m=1}^M \sum_{k=1}^K B_{m,k} \Pi_{m,k},$$

$$\text{s.t.} \sum_{m=1}^M \Pi_{m,k} = 1, k \in \{1 \ldots K\}; \sum_{k=1}^K \Pi_{m,k} \le 1, m \in \{1 \ldots M\}. \quad (7)$$

Π is a permutation matrix with $\Pi_{m,k} \in \{0, 1\}$ indicating whether the k-th ground-truth symmetry matches the m-th predicted symmetries. M is the total number of the output symmetries, and K the total number of the ground-truth symmetries. B is a benefit matrix in which $B_{m,k}$ represents the benefit of matching the k-th ground-truth symmetry to the m-th predicted symmetry. A higher similarity between two symmetries leads to a larger benefit.

We compute the benefit as follows. For reflectional symmetries, given two symmetries S_m^{ref} and S_k^{ref}, and their corresponding reflectional transformations T_m^{ref} and T_k^{ref}, the benefit of matching S_m^{ref} and S_k^{ref} is computed as the Euclidean distance between points that are transformed by the two reflectional transformations respectively:

$$B_{m,k}^{\text{ref}} = \sum_j^N \frac{1}{\left|T_m^{\text{ref}}(P_j) - T_k^{\text{ref}}(P_j)\right| + \epsilon}, \quad (8)$$

where $\epsilon = 0.01$ is a small value used to prevent dividing by zero.

For rotational symmetries, the benefit of matching two symmetries S_m^{rot} and S_k^{rot} is defined as the Euclidean distance between points that are transformed by the two rotational transformations respectively:

$$B_{m,k}^{\text{rot}} = \frac{1}{|\Gamma|} \sum_{\gamma \in \Gamma} \sum_j^N \frac{1}{\left|T_m^{\text{rot},\gamma}(P_j) - T_k^{\text{rot},\gamma}(P_j)\right| + \epsilon}, \quad (9)$$
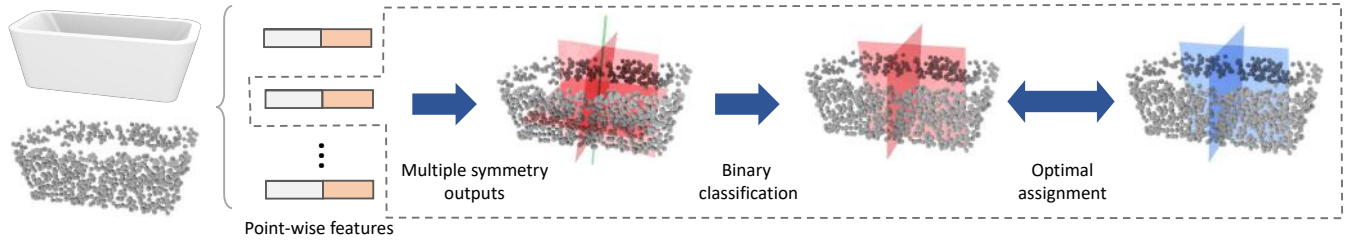
Fig. 4. Illustration of the process of obtaining the present symmetries for each point out of multiple output symmetries via binary classification and optimal assignment.

where $T^{\text{rot},\gamma}$ is the rotational transformation of S^{rot} with a rotation angle of γ. The set of rotation angles is $\Gamma = \{\kappa \cdot \pi/8\}_{\kappa=1,\ldots,16}$. Note that the transformations with different rotation angles are used only for comparing two rotational symmetries; they have nothing to do with the order of the rotational symmetries.

Solving this optimization amounts to finding the assignment that maximizes the total benefit of matching between the predicted and ground-truth symmetries. We use the Hungarian algorithm [Kuhn 1955] to solve the optimization. Figure 4 shows an illustration of the entire process of outputting multiple symmetries per point and finding the optimal assignment.

3.3 Symmetry inference

Prediction aggregation. During inference, we start by extracting point-wise features and making point-wise symmetry predictions. We then aggregate these individual predictions to generate the ultimate prediction. A straightforward method of aggregation is to perform a clustering over the predicted symmetries and select the final predictions as the cluster centers, similar in spirit to [Mitra et al. 2006] and [Podolak et al. 2006]. When clustering, we need to account for the importance of the predictions since they are not equally accurate due to the influence of occlusion and non-uniform lighting. To this end, we introduce a confidence value for each symmetry prediction of each point. In particular, the confidence is evaluated as the probability output by the softmax layer in predicting symmetry type (the $\mathcal{L}_i^{\text{type}}$ in Eq. (1)).

After testing the performance of various clustering algorithms, we found that Density-Based Spatial Clustering (DBSCAN) [Ester et al. 1996] works the best for our task. The dissimilarity between two symmetries is defined in Eq. (8) and Eq. (9). In addition, we use the confidence value of each predicted symmetry as its density weight, thus encouraging the selection of more confident predictions.

Visibility-based verification. Symmetry prediction on complete 3D objects can be conveniently verified by computing the matching error between the original model and the model transformed by the predicted symmetry. Large matching error implies inaccurate/incorrect symmetry prediction. This verification, however, is infeasible when the observation is incomplete because even a correct
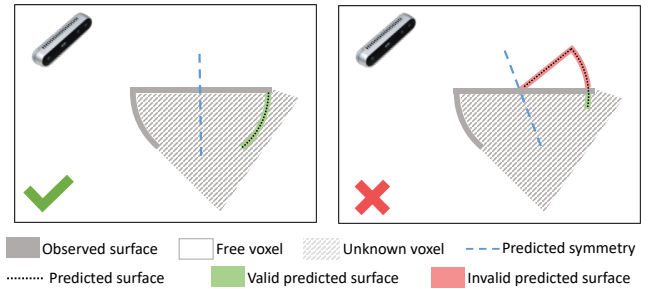

Fig. 5. Verification of the symmetry prediction based on the matching error which is equivalent to the overlap between the transformed points $T(P)$ and the regions of space that are known to be free.

symmetry may have a large matching error due to data incompleteness. We therefore propose a visibility-based verification approach which is suited to our data, i.e., single-view RGB-D images.

As depicted in Figure 5, we first compute a volumetric representation of the space observed by the depth image. Based on visibility w.r.t. the camera pose, the volumetric map contains three types of voxels: observed, free, and unknown. Unknown voxels represent those which are either occluded or outside the FOV. The verification then computes the matching error as the overlap between the transformed surface points and the known free regions. A large overlap means a large confirmed mismatch. We then filter out the predicted symmetries with a large mismatch.

We have also tested using this visibility-based verification as an extra constraint (loss) in the network training. However, we found that it leads to slow convergence while resulting in little improvement in prediction accuracy.

4 IMPLEMENTATION DETAILS

Network architecture. To extract point-wise color features, we use a fully-convolutional network consisting of five convolutional layers, each of which is followed by a Batch Normalization (BN) layer. It encodes an input RGB image of size $H \times W \times 3$ into a $H \times W \times 256$ feature space. We use PointNet to extract geometric features. The architecture of our PointNet implementation is the same as the point cloud segmentation network described in [Qi et al. 2017]. It encodes a point cloud with N points into a $N \times 256$ feature

matrix. The size of the global feature is 512. After concatenating local and global features, the size of per-point feature is 1024. The symmetry predictor is a three-layer Multi-Layer Perceptron (MLP) which takes the per-point features as input and outputs symmetries. Each MLP is followed by a BN layer. The weights w^{ref} and w^{rot} are both set to 0.5. M^{ref} and M^{rot} in the multi-way prediction of reflectional and rotational symmetries are 9 and 3, respectively.

Training and inference. We implement the prediction network in PyTorch [Paszke et al. 2019]. The Adam optimizer [Kingma and Ba 2015] is used with a base learning rate of 0.0001. We use the default hyper-parameters of $\beta_1 = 0.9$, $\beta_2 = 0.99$, and a weight decay of 0.9. The batch size is 32. For DBSCAN, the maximum neighborhood distance eps is 0.2. The minimal number of neighbours for a point to be considered as a core point is 500. We filter the symmetry predictions whose confidence value is less than 0.2. For the visibility-based verification, we filter out the predictions with more than 50 counterpart points located in the known-empty region.

5 RESULTS AND APPLICATIONS

5.1 Benchmark

In order to train and evaluate the proposed network, we have constructed a 3D symmetry detection benchmark for single-view RGB-D images. The benchmark is built upon ShapeNet [Chang et al. 2015], YCB [Calli et al. 2015], and ScanNet [Dai et al. 2017]. For each of the three datasets, we automatically compute symmetries on the 3D models using existing methods. The symmetry labels are then meticulously verified by experienced workers. Finally, we transfer the symmetries of the 3D models to each RGB-D image, transforming by each object's pose. The details of collecting symmetry detection annotations for these datasets are as follows:

ShapeNet consists of 3D CAD models with category labels. We first use an optimization-based symmetry detection method to find the ground-truth symmetries in each model, then perform RGB-D virtual scans and transfer the ground-truth symmetry annotations to the local camera coordinates of each RGB-D image. We split this dataset into four subsets: rendered RGB-D images of training models (*Train*), rendered RGB-D images from novel views of training models (*Holdout view*), rendered RGB-D images of testing models (*Holdout instance*), and rendered RGB-D images of models in untrained categories (*Holdout category*). The details of the train and holdout categories are provided in the supplemental material.

YCB is a dataset originally built for robotic manipulation and 6D pose estimation. It contains RGB-D videos of table-top objects with different sizes, shapes, and textures. High quality 3D reconstructions are provided for each object. We manually annotate ground-truth symmetries for these reconstructed 3D models, and transfer them to the local camera coordinates of each RGB-D image by using the ground-truth 6D pose of each object. We follow the original train/test split established in Calli et al. [2015].

ScanNet is a dataset containing RGB-D videos of indoor scenes, annotated with 3D camera poses, surface reconstructions, and semantic segmentations. The recent work Scan2CAD [Avetisyan et al. 2019]

Table 1. Statistics of the benchmark.

Dataset	Subset	#View	#Object	#Scene
ShapeNet	Train	300 000	30 000	-
	Holdout view	7 200	2 400	-
	Holdout instance	7 200	2 400	-
	Holdout category	4 800	1 600	-
YCB	Train	16 189	18	80
	Test	2 949	18	12
ScanNet	Train	13 126	1 642	400
	Holdout view	4 723	1 642	400
	Holdout scene	1 121	425	100

provides individual alignment between 3D CAD models and the objects present in the reconstructed surfaces. To obtain the ground-truth symmetries, we first perform an optimization-based symmetry detection on the 3D CAD models, then transfer the detected symmetries to each RGB-D frame. We split the original train/test split into three subsets: RGB-D images of the training scenes (*Train*), holdout RGB-D images of the training scenes (*Holdout view*), and RGB-D images of the testing scenes (*Holdout scene*).

The statistics of the benchmark datasets are reported in Table 1.

5.2 Evaluation metric

To evaluate and compare the proposed method, we show precision-recall curves [Funk et al. 2017] produced by altering the threshold of the confidence value of the prediction. To determine whether a predicted symmetry is a true positive or a false positive, we compute a dense symmetry error from the difference between the predicted symmetry and the ground-truth symmetry. Specifically, for a reflectional symmetry, the dense symmetry error of the predicted symmetry S^{ref} and the ground-truth symmetry $\hat{S}^{ref}$ of an object with points $\mathcal{P} = \{P_i\}, i \in [1, N]$ is computed as:

$$\mathcal{E}^{ref} = \frac{1}{N} \sum_i^N \frac{\left\| T^{ref}(P_i) - \hat{T}^{ref}(P_i) \right\|_2}{\rho}, \qquad (10)$$

where T^{ref} and $\hat{T}^{ref}$ are the symmetric transformations of S^{ref} and $\hat{S}^{ref}$, respectively, and ρ is the max distance from the points in $\mathcal{P}$ to the symmetric plane of $\hat{S}^{ref}$.

For rotational symmetries, the dense symmetry error between a predicted symmetry S^{rot} and the ground-truth symmetry $\hat{S}^{rot}$ is:

$$\mathcal{E}^{rot} = \frac{1}{|\Gamma|} \frac{1}{N} \sum_{\gamma \in \Gamma} \sum_i^N \frac{\left\| T^{rot,\gamma}(P_i) - \hat{T}^{rot,\gamma}(P_i) \right\|_2}{\rho}, \qquad (11)$$

where $T^{rot,\gamma}$ is the rotational transformation of S^{rot} with a rotation angle of γ. The set of rotation angles is $\Gamma = \{\kappa \cdot \pi/8\}_{\kappa=1,\dots,16}$, and ρ is the max distance from the points in $\mathcal{P}$ to the rotational axis of $\hat{S}^{rot}$.

In all experiments, we set the dense symmetry error threshold to be 0.25 for both reflectional and rotational symmetries. The confidence value of a predicted symmetry S is computed by the number of input

points and the number of samples (symmetries predicted by each point) belonging to the same cluster as symmetry S.

5.3 Ablation studies

To study the importance of each component of our method, we compare our full method against several variants. A specific part of the pipeline is taken out for each variant, as follows:

- **No RGB Input**: without the input RGB image channels (see the first component in Figure 2). The network can only learn knowledge about symmetries based on geometry.
- **No Counterpart Predictions**: without multi-task learning in the form of counterpart predictions $\mathcal{L}^{\mathrm{ref\_cp}}$ or $\mathcal{L}^{\mathrm{rot\_cp}}$ during training (see the second component in Figure 2).
- **No Verification**: without the visibility-based filtering of false positives during inference (see the third component in Figure 2).

Figure 6 shows the results of our ablation studies, for reflectional (left column) and rotational (right column) symmetry detection. The full method outperforms the simpler variants in almost all cases. Omitting counterpart prediction degrades the results the most, especially for reflectional symmetry detection. This demonstrates that the multi-task learning scheme is crucial to our approach. An interesting observation is that the baseline without RGB input achieves comparable or even better results on subsets containing novel objects (see Figure 6 d, e, and f). This demonstrates that generalization to unknown objects requires geometry, and confirms the intuition presented in Section 3.

5.4 Comparison to baselines

We evaluate our method against three baseline symmetry detection methods for objects based on RGB-D images:

- **Geometric Fitting** [Ecins et al. 2018]: a state-of-the-art symmetry detection approach for point clouds. It first generates a set of symmetry candidates, and then performs symmetry refinement based on geometric constraints. Since their focus is to detect reflectional symmetries, we only compare with it on the reflectional symmetry prediction task.
- **RGB-D Retrieval**: an intuitive approach for symmetry detection that finds, for each object in an RGB-D image, the most similar object present in the training data. The precomputed symmetries are then transferred from the training data to be the symmetry predictions. To achieve this, we train a FoldingNet [Yang et al. 2018] to extract the feature vectors of all objects in the training data. During testing, $L2$ distance in the feature space is used to retrieve the most similar RGB-D image.
- **Shape Completion**: a two-step approach which first performs a shape completion [Liu et al. 2019] on the input point cloud and then detects symmetries on the completed shape by a geometric symmetry detection method [Li et al. 2014]. We compare to it on the reflectional symmetry prediction task of ShapeNet.
- **DenseFusion [Wang et al. 2019a]**: a cutting-edge approach to estimate the 6D pose of the known objects. We transform the ground-truth symmetries of the known objects by using the

(a) Ref. sym., holdout view (b) Rot. sym., holdout view

(c) Ref. sym., holdout instance (d) Rot. sym., holdout instance

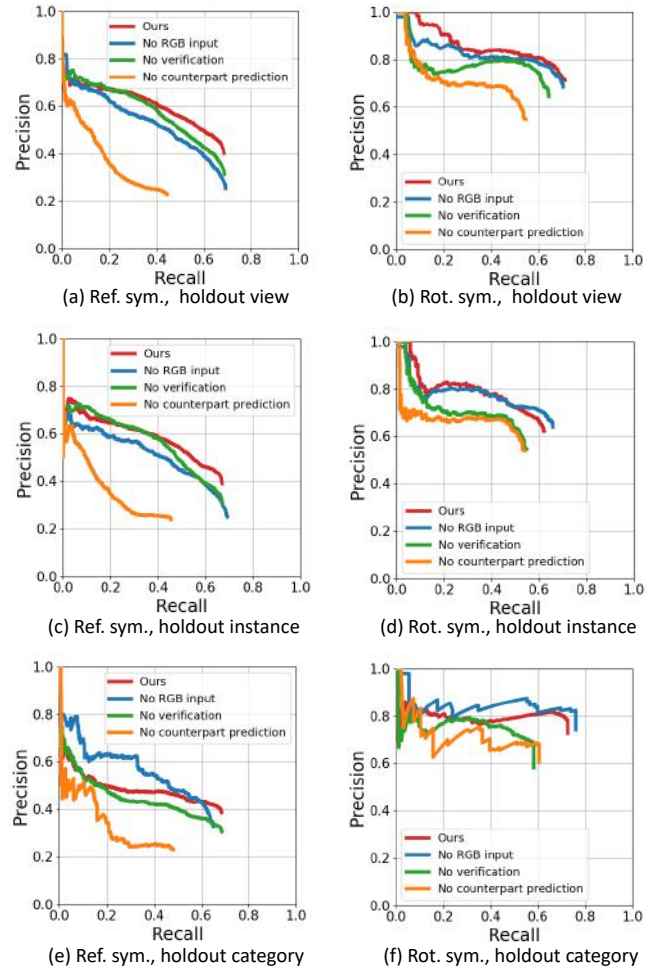(e) Ref. sym., holdout category (f) Rot. sym., holdout category

Fig. 6. Ablation studies: comparison of symmetry prediction performance on three subsets of ShapeNet between our full proposed method (red) and its several variants (blue: without RGB input; green: without verification procedure; orange: without counterpart prediction, i.e. without multi-task learning). The left and right columns show predictions of reflectional and rotational symmetry, respectively. Note that in most cases using RGB input helps prediction, but the variant without RGB input stands out at predicting the reflectional symmetry for the held-out category (e).

predicted 6D pose produced by DenseFusion, thus obtaining the predicted symmetries for each object in the RGB-D images. Since DenseFusion only works on scenarios where the geometries of the target objects are known, we compare to it on the YCB dataset.

The comparisons are plotted in Figure 7 (reflectional symmetry) and Figure 8 (rotational symmetry). They show that our method outperforms the baselines by a large margin, for both the reflectional and rotational symmetry detection tasks, and over all the data subsets. Crucially, our method achieves relatively high performance on subsets (ShapeNet *holdout category* and ScanNet *holdout scene*) that include very different objects from those present in the training data.
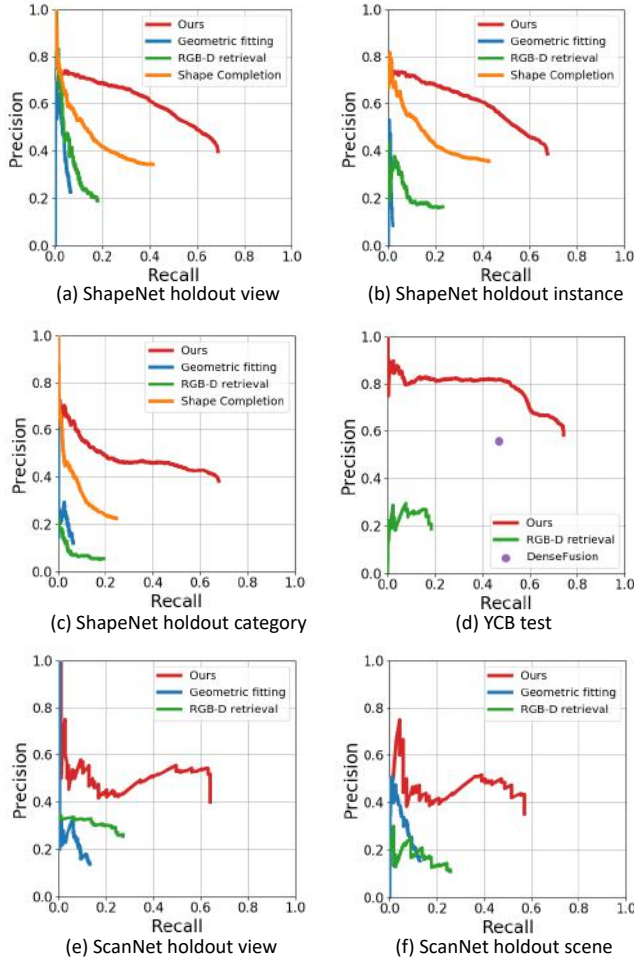
Fig. 7. Comparison between our proposed approach and the baseline methods on the performance of predicting reflectional symmetry for three datasets divided into six subsets: (a) ShapeNet holdout view; (b) ShapeNet holdout instance; (c) ShapeNet holdout category; (d) YCB test; (e) ScanNet holdout view; and (f) ScanNet holdout scene. Note that in (c), our approach is significantly more precise at symmetry prediction for categories that it has never encountered before (holdout category), demonstrating high generalization ability.
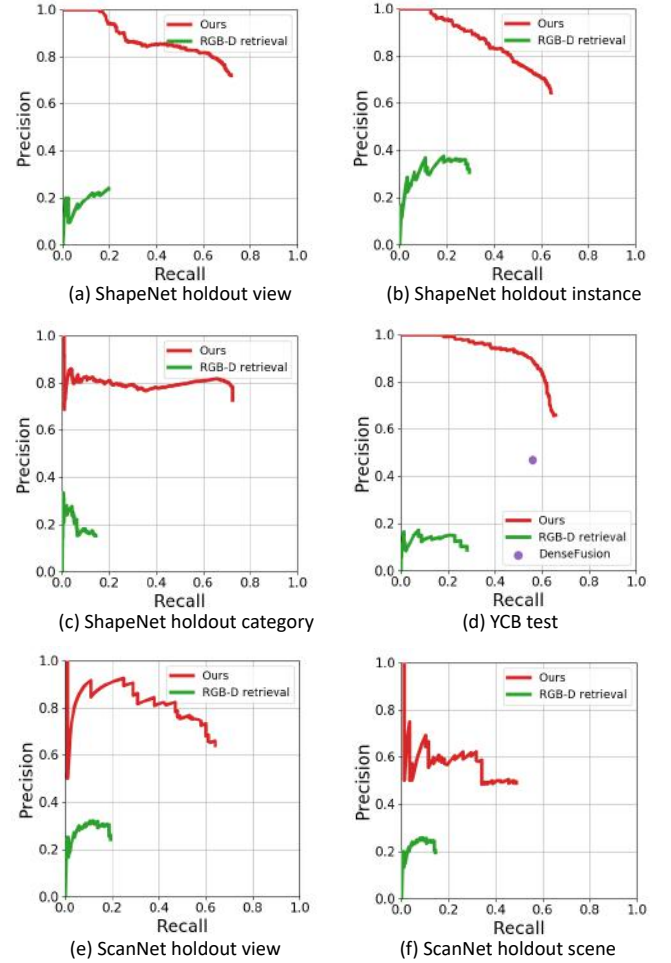

Fig. 8. Comparison between our proposed approach and the baseline methods on the performance of predicting rotational symmetry for three datasets divided into six subsets: (a) ShapeNet holdout view; (b) ShapeNet holdout instance; (c) ShapeNet holdout category; (d) YCB test; (e) ScanNet holdout view; and (f) ScanNet holdout scene. Similar to Figure 7(c), the generalization ability of our approach is illustrated in (c).

The Shape Completion baseline is inferior to our method, especially on the ShapeNet *holdout Category* subset, due to its poor generality in completing the shape on the untrained categories. Even though the Geometric Fitting baseline has high precision for the symmetries it detects, it fails to detect most of the symmetries since it can be easily influenced by incomplete observations. The RGB-D Retrieval baseline shows worse performance than our method, especially on the ShapeNet *holdout Category* subset and the ScanNet dataset, due to its weaker generalization ability. The DenseFusion baseline achieves a relatively high precision and recall on the YCB dataset. However, it cannot be extended to datasets containing objects that have not been seen during training.

5.5 Qualitative results

Figure 9 visualizes the symmetry prediction results for both synthetic and real data. Our approach detects both reflectional and rotational symmetries in challenging cases, such as novel objects, objects with multiple symmetries, and objects with heavy occlusion. We also show the qualitative comparison with baselines in Figure 17. We see that partial observations and occlusion interfere with the ability of Geometric Fitting and PRS-Net to establish correspondences on the observed points. RGB-D Retrieval provides poor features for untrained objects, leading to an inability to predict accurate symmetries. Shape Completion can generate reasonably good geometry for common objects, but it is less capable in cases where the objects are novel or heavily occluded. Our method, in

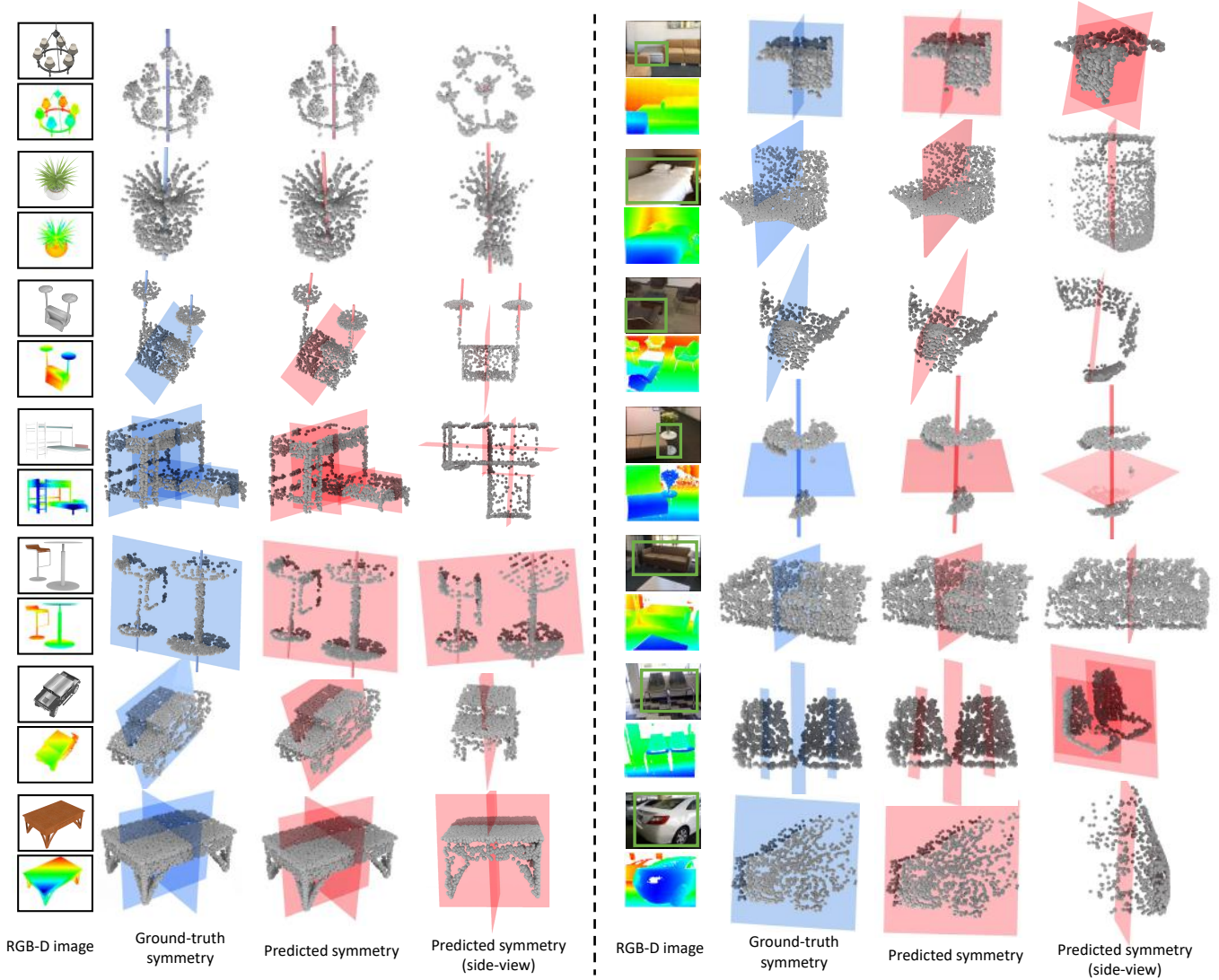| RGB-D image | Ground-truth symmetry | Predicted symmetry | Predicted symmetry (side-view) | RGB-D image | Ground-truth symmetry | Predicted symmetry | Predicted symmetry (side-view) |

Fig. 9. Qualitative symmetry prediction results on ScanNet [Dai et al. 2017] and ShapeNet [Chang et al. 2015]. The first column has two images of the object, i.e. the input RGB image with the target object marked in green rectangle (upper) and the input depth image (lower). The second column shows the ground-truth symmetries of the objects. The last two columns visualize the predicted symmetries by our method. Our method is able to handle objects with all forms of symmetry compositions (reflectional symmetry only, rotational symmetry only and multi-symmetry).

contrast, successfully predicts symmetries with high accuracy for all of these examples.

5.6 Sensitivity to occlusion

To evaluate the capability of our approach when it comes to occluded objects, we create a dataset based on ShapeNet by grouping the data according to the occlusion ratio. In order to generate data with mutual occlusion, we randomly add a foreground mask in the rendered RGB-D images. The occlusion ratio of the object is computed by dividing the area of the occluded region by the area of the whole surface. Examples of the occluded data are provided

in the supplemental material. Note that both self-occlusion and mutual-occlusion are included.

Figure 10 compares our approach with the Shape Completion, Geometric Fitting, and RGB-D Retrieval baselines, all three of which are capable of finding symmetries for occluded objects. It is evident that our method performs better than the baseline methods for all the experiments. While the overall performance is generally affected as the occlusion ratio increases, ours outperforms the baseline methods and shows a relatively smaller decreasing rate in all cases. Qualitative results on the real data with occlusions are shown in Figure 1, Figure 9 and Figure 17.
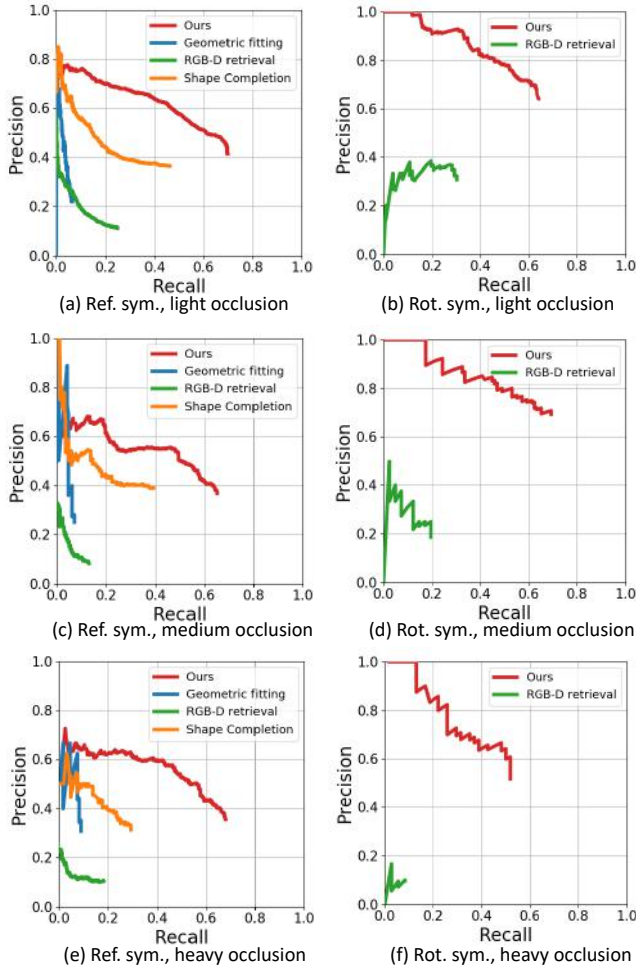
Fig. 10. Occlusion ratio sensitivity evaluation of our proposed approach (red), compared to the Geometric Fitting method (blue) and RGB-D Retrieval (green). The three rows include different occlusion ratios: light (50-60%) occlusion for (a) and (b); medium (60-70%) occlusion for (c) and (d); and heavy (70-80%) occlusion for (e) and (f). The left and right columns illustrate reflectional and rotational symmetry detection, respectively. While the performance of all methods decreases with increasing occlusion, our approach outperforms the baselines in all scenarios, and demonstrates relatively modest decrease in performance with increasing occlusion.

5.7 Evaluation of counterpart prediction

To evaluate the quality of the predicted counterparts, we compute and plot the distribution of the Euclidean distance of each predicted counterpart to its ground-truth counterpart, similar to [Kim et al. 2011]. Figure 11 shows the plots on the three ShapeNet subsets for reflectional symmetry. The x-axis of the plots represents a varying Euclidean distance (error) threshold. The y-axis shows the percentage of counterpart correspondences whose Euclidean distance are within the threshold. Larger AUC (Area Under the Curve) represents better performance. Compared to the baselines, our method is
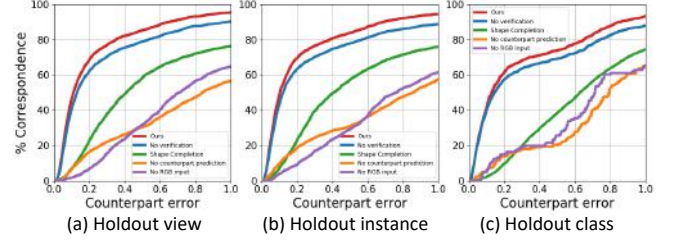

Fig. 11. Comparison between our proposed approach and the baseline methods on the performance of predicting counterparts of the reflectional symmetries on ShapeNet dataset. Each curve depicts the percentage of counterpart correspondences (y-axis) whose Euclidean distance (error) are within the thresholds (x-axis).
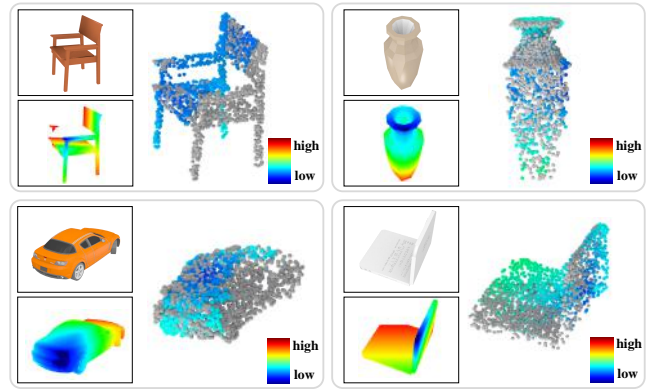

Fig. 12. Visualization of predicted counterparts. The grey points represent the input points. The colored points are the predicted counterparts, where color encodes the Euclidean distance error of the predictions.

more accurate on the counterpart prediction task over all ShapeNet subsets. Figure 12 shows a visualization of predicted counterparts.

5.8 Prediction of discrete rotational symmetry

To evaluate the performance of our method on predicting discrete rotational symmetry, we create a dataset of shapes from ShapeNet, with ground-truth rotational symmetries of various orders. We randomly add foreground occlusions to the rendered RGB-D images to verify the sensitivity to occlusion. Details of the dataset can be found in the supplemental material. We use two metrics to evaluate the results. First, since the order prediction is a classification task, classification accuracy is used. Second, the error of rotational angle is reported. Note that the error of rotational angle is computed as $|2\pi/r - 2\pi/\hat{r}|$, where r and $\hat{r}$ are the predicted and the ground-truth order of the rotational symmetry, respectively. Figure 13 shows the results. We see that our proposed method can accurately predict the order of discrete rotational symmetries when the occlusion is not present or light. The performance drops as the occlusion increases.
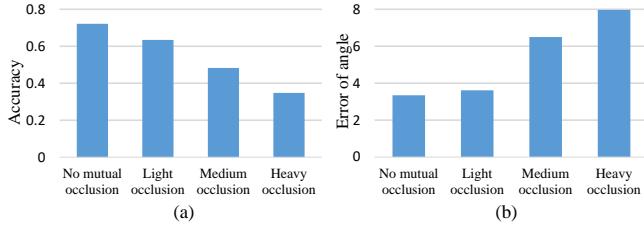
Fig. 13. Evaluation of discrete rotational symmetry prediction of our method. The plots show the classification accuracy (left) and the error of angle (right) on input with different occlusion ratios (the ratio of occluded area over the whole bounding box of the target object): no mutual occlusion, light (50-60%) occlusion, medium (60-70%) occlusion and heavy (70-80%) occlusion. The error of angle is in degrees.

Table 2. Training and runtime performance statistics of each component of the proposed SymmetryNet.

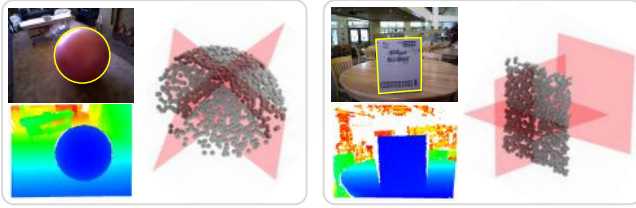| Dataset	Network train	Network inference	Aggregation	Verification
ShapeNet	64 h	50 ms	50 ms	40 ms
YCB	20 h	50 ms	50 ms	40 ms
ScanNet	26 h	50 ms	50 ms	40 ms

Fig. 14. Two failure cases. Left: Our method cannot deal with spherical symmetry characterized by the center of a sphere; It detects two reflectional symmetries wrongly. Right: When a cuboid object is viewed orthogonally from only one face, our method would fail to infer the reflectional symmetry in the depth direction.

5.9 Runtime analysis

Table 2 reports the timing of each component in our approach on a server with an Intel® Xeon® CPU E5-2678 v3 @ 2.50GHz × 48, 128GB RAM, and an Nvidia TITAN V graphics card. Note that our method is dramatically faster compared to the state-of-the-art method in [Ecins et al. 2018] which takes about 15 seconds to detect symmetries for an object.

5.10 Failure cases

Figure 14 shows two typical failure cases found in our experiments. The first case is that our method is unable to deal with other symmetry types than it was trained for. In the example, it detects two reflectional symmetries for a spherical symmetry. Another case is when a cuboid object is viewed orthogonally from only one face, our method would fail to infer the location of the reflection plane along the depth direction since the shape information is completely missing along that direction.
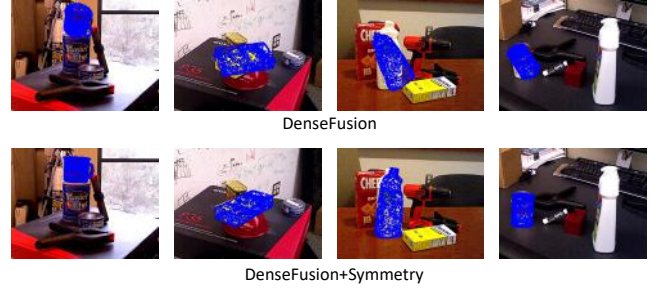
DenseFusion

DenseFusion+Symmetry

Fig. 15. By incorporating the feature of our predicted symmetries, Dense-Fusion+Symmetry achieves more accurate pose estimations, compared to the original DenseFusion approach [Wang et al. 2019a].
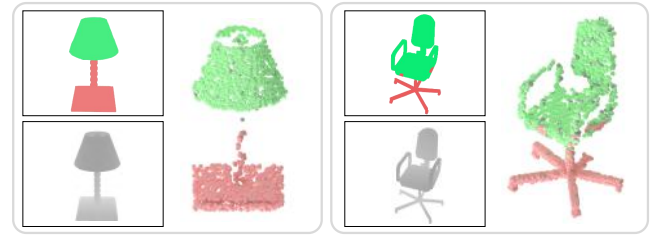

Fig. 16. Symmetry-induced RGB-D segmentation. The input RGB-D images are segmented through projecting the points segmented and labeled with different predicted symmetries. The segmentations are visualized by distinct colors both on the 3D point clouds and the input RGB images. Note that the outlier points are colored grey.

5.11 Applications

Various applications can potentially benefit from symmetry prediction. A straightforward application is symmetry-based object completion as commonly demonstrated in many symmetry detection works (e.g., [Bokeloh et al. 2009]). Here, we focus on a more unique application, i.e., how to apply our predicted symmetries to assist 6D object pose estimation from single-view RGB-D images.

To demonstrate the effectiveness of the predicted symmetries on the improvement of 6D pose estimation, we combine the predicted symmetry information of our approach to the state-of-the-art 6D pose estimation approach DenseFusion [Wang et al. 2019a]. To be specific, we feed the parameters of predicted symmetries to DenseFusion as extra features. These features are then processed by an MLP and concatenated to the point cloud feature in DenseFusion. We train the network using the same data as described in [Wang et al. 2019a]. Figure 15 demonstrates how the predicted symmetries boost the performance of DenseFusion.

Another interesting application is symmetry-induced segmentation of RGB-D images. Thanks to our symmetry prediction, the input RGB-D images can be segmented into parts presenting different symmetries, although there was no such segmentation label for RGB-D images available during training. This can be realized naturally by projecting the point-wise symmetry labels to the input RGB-D images. Figure 16 demonstrates two examples of this application.
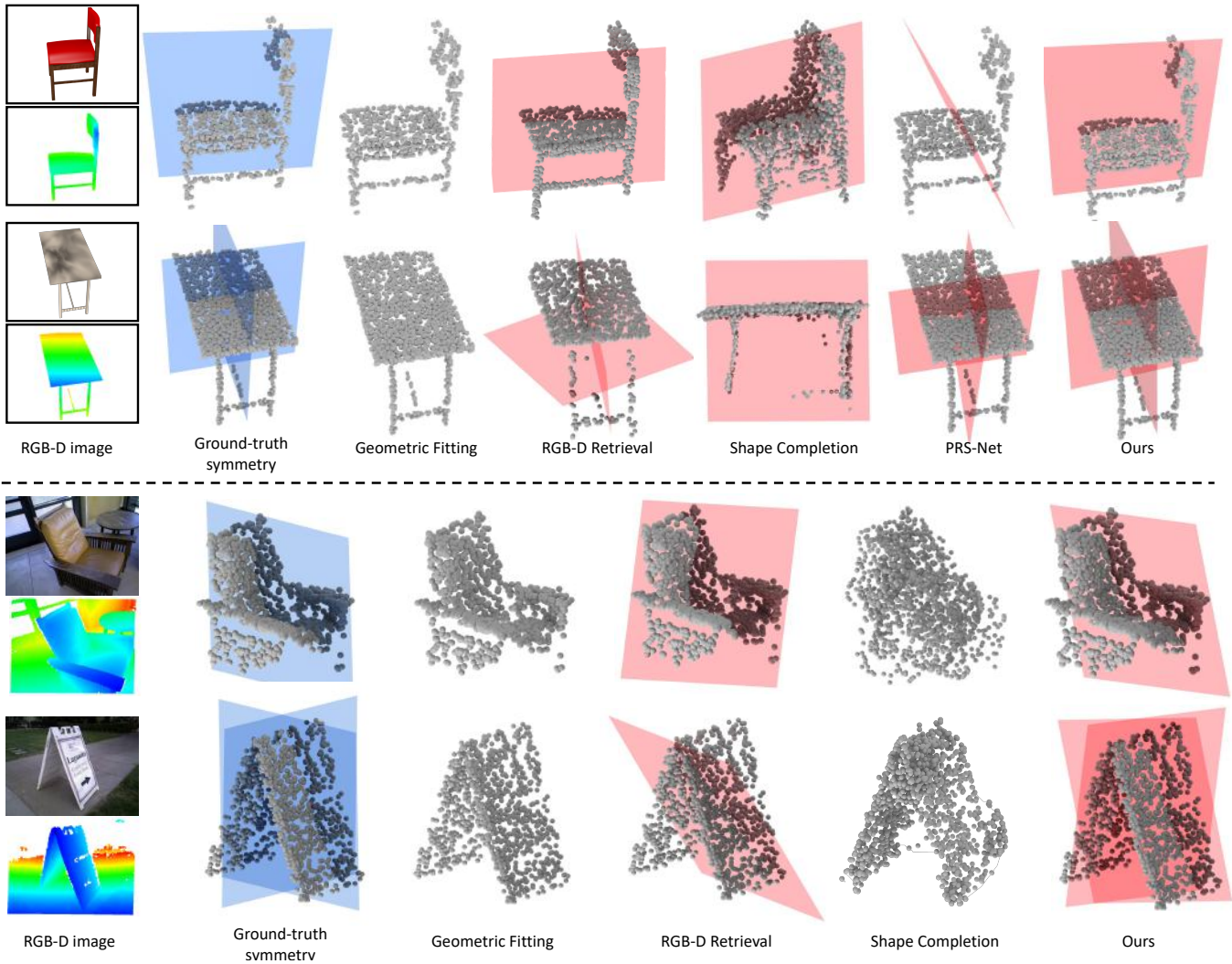
Fig. 17. Qualitative comparisons to previous works (Geometric Fitting [Ecins et al. 2018], RGB-D Retrieval, PRS-Net [Gao et al. 2019] and Shape Completion [Li et al. 2014; Liu et al. 2019].) on both synthetic data and real data. The Geometric Fitting baseline fails to detect any symmetry. The PRS-Net baseline correctly predicts the number of symmetries but fails to regress the parameters of the symmetries accurately. The RGB-D Retrieval baseline could not predict the parameters of symmetries correctly. The Shape Completion baseline predicts accurate symmetries for objects with simple geometry, but fails on cases where the objects are novel or occluded. Our method achieves the best performance on the four examples.

6 DISCUSSION AND CONCLUSIONS

We have proposed a novel problem of detecting 3D symmetries from single-view RGB-D images. Due to partial observation and object occlusion, the problem is challenging, to the point of being beyond the reach of purely geometric detection methods. Instead, we have proposed an end-to-end deep neural network that predicts both reflectional and rotational symmetries for 3D objects based on a single RGB-D image. Several dedicated designs make our method general and robust. *First*, our network is trained on multiple tightly coupled tasks to achieve outstanding generalizability for both types of symmetries. *Second*, we devise an optimal assignment module in our network to enable it to output an arbitrary number of symmetries.

Our current method has certain limitations, which we believe will inspire future research:

- Our method relies on a good object-level segmentation. If the segmentation mask of an object of interest contains other objects, the symmetry detection will be affected. Although there have been many powerful deep models trained for RGB-D segmentation, it would still be interesting to integrate object detection/segmentation and symmetry detection into one unified deep learning model.
- Our current network can only deal with reflectional and rotational symmetries. Extending it to other types of symmetry should not be difficult, although it may make the network harder

to train. In general, finding a suited parameterization/representation of symmetry for end-to-end learning is a fundamental and interesting future direction to pursue.

- Our method cannot handle hierarchical (nested) symmetries such as those considered in [Wang et al. 2011]. We expect that recursive neural networks (RvNN) could be utilized for this case, following the series of works on using RvNNs for 3D structure encoding/decoding [Li et al. 2017; Yu et al. 2019].
- Our network relies on strong supervision. Annotating symmetries for RGB-D data is a non-trivial endeavor. Therefore, it would be interesting to look into unsupervised or self-supervised approaches to symmetry detection, through exploiting rich geometric constraints.

ACKNOWLEDGMENTS

REFERENCES

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2614–2623, 2019.

Martin Bokeloh, Alexander Berner, Michael Wand, H-P Seidel, and Andreas Schilling. Symmetry detection using feature lines. In *Computer Graphics Forum*, volume 28, pages 697–706. Wiley Online Library, 2009.

Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015.

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.

Changhyun Choi and Henrik I Christensen. 3d pose estimation of daily objects using an rgb-d camera. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3342–3349. IEEE, 2012.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

Aleksandrs Ecins, Cornelia Fermüller, and Yiannis Aloimonos. Seeing behind the scene: Using symmetry to reason about objects in cluttered environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7193–7200. IEEE, 2018.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1692–1701, 2017.

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. *arXiv preprint arXiv:1910.06511*, 2019.

Natasha Gelfand and Leonidas J Guibas. Shape segmentation using local slippage analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 214–223, 2004.

Georgios Georgakis, Srikrishna Karanam, Ziyan Wu, and Jana Kosecka. Matching rgb images to cad models for object pose estimation. *arXiv preprint arXiv:1811.07249*, 2018.

Tomáš Hodaň, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. Detection and fine 3d pose estimation of texture-less objects in rgb-d images. In *Proc. IROS*, pages 4421–4428. IEEE, 2015.

Qichang Hu, Huibing Wang, Teng Li, and Chunhua Shen. Deep cnns with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3147–3156, 2017.

Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Transactions on Graphics (TOG)*, 37(1):1–14, 2017.

Wei Ke, Jie Chen, Jianbin Jiao, Guoying Zhao, and Qixiang Ye. Srn: side-output residual network for object symmetry detection in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1068–1076, 2017.

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics (TOG)*, 30(4):1–12, 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. ICLR*, 2015.

Yoshinori Konishi, Kosuke Hattori, and Manabu Hashimoto. Real-time 6d object pose estimation on cpu. *arXiv preprint arXiv:1811.08588*, 2018.

Andreas Kuehnle. Symmetry-based recognition of vehicle rears. *Pattern recognition letters*, 12(4):249–258, 1991.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Bo Li, Henry Johan, Yuxiang Ye, and Yijuan Lu. Efficient view-based 3d reflection symmetry detection. In *SIGGRAPH Asia 2014 Creative Shape Modeling and Design*, pages 1–8. 2014.

Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. Grass: Generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017.

Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, and Thomas Funkhouser. Symmetry factored embedding and distance. In *ACM SIGGRAPH 2010 papers*, pages 1–12. 2010.

Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. *arXiv preprint arXiv:1912.00280*, 2019.

Yanxi Liu and Robert T Collins. A computational model for repeated pattern perception using frieze and wallpaper groups. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 537–544. IEEE, 2000.

Yanxi Liu and Robert T Collins. Skewed symmetry groups. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

Yanxi Liu, Hagit Hel-Or, Craig S Kaplan, Luc Van Gool, et al. Computational symmetry in computer vision and computer graphics. *Foundations and Trends® in Computer Graphics and Vision*, 5(1–2):1–195, 2010.

Giovanni Marola. On the detection of the axes of symmetry of symmetric and almost symmetric planar images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):104–108, 1989.

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.

Niloy J Mitra, Leonidas J Guibas, and Mark Pauly. Partial and approximate symmetry detection for 3d geometry. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 560–568. ACM, 2006.

Niloy J Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan. Symmetry in 3d geometry: Extraction and applications. In *Computer Graphics Forum*, volume 32, pages 1–23. Wiley Online Library, 2013.

Hideo Ogawa. Symmetry analysis of line drawings using the hough transform. *Pattern Recognition Letters*, 12(1):9–12, 1991.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Sida Peng, Yuan Liu, Qixing Huang, Xiaowei Zhou, and Hujun Bao. PVNet: Pixel-wise voting network for 6dof pose estimation. In *Proc. CVPR*, pages 4561–4570, 2019.

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.

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.

D. Raviv, A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Symmetries of non-rigid shapes. In *Proc. ICCV*, pages 1–7. IEEE, 2007.

Dan Raviv, Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Full and partial symmetries of non-rigid shapes. *International journal of computer vision*, 89 (1):18–39, 2010.

Wei Shen, Xiang Bai, Zihao Hu, and Zhijiang Zhang. Multiple instance subspace learning via partial random projection tree for local reflection symmetry in natural images. *Pattern Recognition*, 52:306–316, 2016a.

Wei Shen, Kai Zhao, Yuan Jiang, Yan Wang, Zhijiang Zhang, and Xiang Bai. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 222–230, 2016b.

Ching L Teo, Cornelia Fermuller, and Yiannis Aloimonos. Detection and segmentation of 2d curved reflection symmetric structures. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1644–1652, 2015.

Stavros Tsogkas and Iasonas Kokkinos. Learning-based symmetry detection in natural images. In *European Conference on Computer Vision*, pages 41–54. Springer, 2012.

Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3343–3352, 2019a.

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 *Proc. CVPR*, pages 2642–2651, 2019b.

Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiquan Cheng, and Yueshan Xiong. Symmetry hierarchy of man-made objects. In *Computer graphics forum*, volume 30, pages 287–296. Wiley Online Library, 2011.

Lingyu Wei, Qixing Huang, Duygu Ceylan, Etienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016.

Kai Xu, Hao Zhang, Andrea Tagliasacchi, Ligang Liu, Guo Li, Min Meng, and Yueshan Xiong. Partial intrinsic reflectional symmetry of 3d shapes. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009.

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

Raymond KK Yip. A hough transform technique for the detection of reflectional symmetry and skew-symmetry. *Pattern Recognition Letters*, 21(2):117–130, 2000.

Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9491–9500, 2019.

Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

Thomas Zielke, Michael Brauckmann, and Werner von Seelen. Intensity and edge-based symmetry detection applied to car-following. In *European Conference on Computer Vision*, pages 865–873. Springer, 1992.