Data-Driven Structural Priors for Shape Completion

Minhyuk Sung[1] Vladimir G. Kim[1,2] Roland Angst[1,3] Leonidas Guibas[1]
[1]Stanford University [2]Adobe Research [3]Max Planck Institute for Informatics

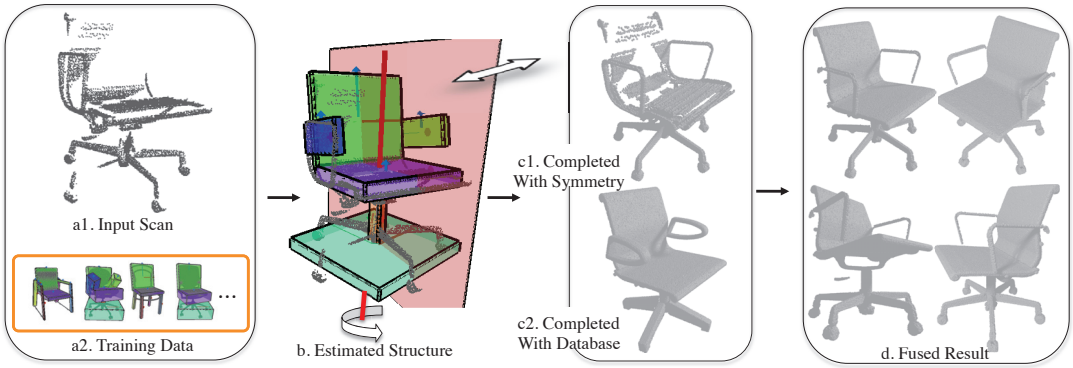

Figure 1: *Given an incomplete point scan with occlusions (a1), our method leverages training data (a2) to estimate the structure of the underlying shape, including parts and symmetries (b). Our inference algorithm is capable of discovering the major symmetries, despite the occlusion of symmetric counterparts. The estimated structure is used to augment the point cloud with additional points from symmetry (c1) and database (c2) priors, which are further fused to produce the final completed point cloud (d). Note how the final result leverages symmetry whenever possible for self-completion (e.g., stem, armrests), and falls back to database information transfer in case all symmetric parts are occluded (e.g., back).*

Abstract

Acquiring 3D geometry of an object is a tedious and time-consuming task, typically requiring scanning the surface from multiple viewpoints. In this work we focus on reconstructing complete geometry from a single scan acquired with a low-quality consumer-level scanning device. Our method uses a collection of example 3D shapes to build structural part-based priors that are necessary to complete the shape. In our representation, we associate a local coordinate system to each part and learn the distribution of positions and orientations of all the other parts from the database, which implicitly also defines positions of symmetry planes and symmetry axes. At the inference stage, this knowledge enables us to analyze incomplete point clouds with substantial occlusions, because observing only a few regions is still sufficient to infer the global structure. Once the parts and the symmetries are estimated, both data sources, symmetry and database, are fused to complete the point cloud. We evaluate our technique on a synthetic dataset containing 481 shapes, and on real scans acquired with a Kinect scanner. Our method demonstrates high accuracy for the estimated part structure and detected symmetries, enabling higher quality shape completions in comparison to alternative techniques.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: shape analysis, shape completion, shape collections

1 Introduction

Despite the recent advances in 3D scanning technology, rapid acquisition of complete 3D geometry still remains one of the key challenges in robotics and graphics. A typical scan obtained with a consumer-level device may have large missing regions due to limitations in the depth range of the scanner, lighting conditions, and reflective properties of the underlying materials. Even without these hardware limitations, to generate a complete scan, the user or an autonomous agent have to acquire the geometry from multiple viewpoints, which is time-consuming and can be infeasible if the environment poses limitations on the choice of viewpoints. Incomplete geometry leads to challenges for autonomous agents in planning an interaction with an object, limits capabilities of geometry analysis algorithms (e.g., in inferring semantic parts), and produces content that has little use in virtual reality applications. To address this problem, we propose a method for completing a point cloud from a single-view scan by introducing structural priors including expected symmetries and geometries of parts. In a preprocessing step, we leverage a collection of segmented 3D shapes to learn a structural prior which captures positions and orientations of parts and global and partial symmetries between parts that are expected in a given class of shapes. This data-driven approach enables us to estimate the global part structure and to detect symmetries in partial and seemingly asymmetric scans. Having access to this global part structure, the input scan can then in turn be completed, by exploiting geometry from both the observed partial scan itself (via symmetries) as well as the shape collection.

Although symmetric objects can be completed by copying observed regions to occluded counter-parts, existing symmetry detection algorithms (e.g., [Pauly et al. 2008; Mitra et al. 2006]) infer symmetry from the input shape, which means that they need to observe at least some fraction of the occluded counter-part to detect the symmetry. Moreover, even if symmetries are successfully detected, not all occluded regions are guaranteed to have visible counterparts. In these cases, the shape can be completed by copying the occluded regions from the input database. To do so, our method predicts locations, scales, and orientations of occluded parts in the partial input scan. Unlike previous work that leverages global rigid shape alignment to make these predictions [Shen et al. 2012; Kim et al. 2013a], we treat each part as a potential reference coordinate system, which enables detecting a global structure even if only a fraction of the surface is

visible. To produce the final completed shape, our method efficiently combines both data sources by preferentially using self-symmetries and reverting to copying database patches only if symmetric counterparts are not available. By combining both data sources, the partial input scan itself and a shape collection, we take advantage of complementary cues, resulting in higher accuracy and completeness of the reconstructed shape.

The main technical contribution of our paper is an algorithm to estimate the structure of a partially scanned shape. In particular, we compute a segmentation and labeling of the input point cloud, estimate the position, size, and orientation of visible and occluded parts, and detect partial and global symmetries. The estimated structure for the teaser shape is depicted in Figure 1.b, where parts are shown as oriented boxes and symmetries are the plane of reflection (for all parts) and the axis of rotation (for the stem). The key advantage of our method is robustness to severe occlusions. We achieve it by placing each part in a local coordinate system and aggregating local cues in an optimization scheme to predict the shape structure.

In offline processing, given a collection of segmented shapes, we compute per-point local shape features and train point-to-part classifiers. Also, for all pairs of parts we learn distributions for their relative position, scale, and orientation. During the inference, we define a global optimization problem that favors a part arrangement that is consistent with local observations (e.g., feature-based classification and locally-estimated coordinate systems). We optimize our objective function in several steps. First, we label each input point based on local shape descriptors. Second, we use these predictions to produce multiple candidate part representations for each shape segment, and then select among the predictions by defining a joint optimization problem that finds optimal representations for all parts. Third, we re-segment the point cloud based on proximity to estimated parts. Fourth, we improve continuous parameters such as position, scale, and orientation of parts and estimate symmetry axes to provide additional constraints on part relations. Finally, we inject multiple candidate part representations for missing parts, and repeat the procedure starting with the second step. Once the structure is inferred, we use the symmetries and database parts to produce the complete point cloud.

We run experiments with 5 shape collections containing 481 segmented shapes, and attempt to complete the shapes obtained via simulated scans as well as real Kinect depth images. To evaluate our system we devise a novel benchmark for data-driven shape completion. We found that fusing symmetry and database priors produces better completions than relying on only one of the two data sources. We also found that our data-driven symmetry detection technique produces more accurate symmetry parameters for partial scans than state-of-the-art methods that assume that the input shape is complete. Similarly, our approach to detecting part structure estimates more accurate part parameters on partial scans, and produces completions that are closer to the true surface.

Key Contributions:

- a probabilistic shape model that can be applied to analyze partial scans with severe occlusions;
- a data-driven structure estimation algorithm that simultaneously segments the partial input scan, estimates positions, scales, and orientations of visible and occluded parts, and detects global and local symmetries;
- a shape completion technique that combines a database of parts and symmetries to produce a more accurate result;
- a benchmark for evaluating shape completion methods.

2 Related Work

There is a large body of work on analyzing single-view scans that enables segmenting [Silberman et al. 2012] and recognizing [Lai et al. 2011; Janoch et al. 2011] salient objects. The main objective of this work is to complete a shape from a partial single-view scan with missing or occluded regions. In this problem, we assume that the object has been successfully segmented and correctly classified. Our objective is also different from the goal pursued by surface reconstruction algorithms that typically assume that most of the surface was captured by the scanner (e.g., see [Kazhdan and Hoppe 2013; Choi et al. 2015]). In the presence of significant occlusions the shape completion problem becomes ill-posed and one has to rely on additional priors such as physics, symmetry, and exemplar geometries. Leveraging physics (e.g., gravity and stability) only enables coarse completions that do not produce plausible geometry [Shao et al. 2014; Zheng et al. 2013]. Thus, in this overview of related work, we only discuss techniques that use symmetry or a database of similar exemplars to complete a shape with significant missing regions. In addition, since it is crucial to estimate part structure for effectively using both of these priors, we also discuss related advances in part-based models.

Symmetry-driven shape completion. Several existing approaches leverage symmetry analysis to fill the occluded regions from their visible counter-parts. Thrun et al. [2005] use reflective symmetries and continuous rotational symmetries. Pauly et al. [2008] replicate regular structures that form a lattice with discrete rotational, translational, and scaling symmetries. Zheng et al. [2010] detect repeating structures to consolidate and densify LiDAR scans of buildings. All of these methods rely on the assumption that symmetries can be detected in the scanned data itself. This narrows the use cases to the scans where many repeating parts are visible (in order to detect the symmetries), with some repeating elements being occluded (in order for the symmetry to be useful in shape completion). While available symmetry detection algorithms assume a nearly complete shape [Mitra et al. 2013], Sipiran et al. [2014] focus on leveraging local features to handle larger occluded regions, yet, their method still requires some repeated structures to be visible. Unlike these techniques we treat symmetry detection in a data-driven approach that is trained on a collection of complete 3D shapes, and where we leverage visible parts to obtain hints for positions of symmetry planes and axes.

Data-driven shape completion. Another alternative solution for completing a shape is copying example patches or parts from the database. Pauly et al. [2005] retrieve and align shapes to do the completion. Other approaches take advantage of efficient feature-based descriptors and simply use the most similar object to complete the 3D scan [Nan et al. 2012; Kim et al. 2013b; Li et al. 2015]. These techniques always rely on the assumption that the database includes a very similar shape, and thus, their value is limited to cases when a database captures all shape variations expected in the scans. Shen et al. [2012] expand the space of variations by aligning parts from multiple objects to complete the point clouds. In some cases they reconstruct parts that are not visible in the 3D scan (due to hardware resolution), but they use RGB image in these cases. Their method also heavily relies on the assumption that it is relatively easy to rigidly align each shape in a database to the point cloud. In contrast, our method does not use RGB data and does not assume that a single rigid transformation is sufficient to bring input scan and the training dataset to the same coordinate system, and thus, detects part structure even if only a small fraction of the surface is visible.

Part-based models. Our part representation is motivated by probabilistic deformable part models that found many successful applications in computer vision [Felzenszwalb et al. 2008]. Going beyond a single tree-structured part model, hierarchical part gram-

mars have been developed for parsing images [Han and Zhu 2009], 3D scenes [Wu et al. 2014], and facades [Martinovic and Van Gool 2013]. Even though these models typically handle occlusions, they focus on classifying visible regions instead of predicting the occluded parts. Since the goal of our work is to complete detailed geometry from a partial scan of a 3D object, our part representation is equipped with additional geometric information, such as self-symmetries, positions, orientation, and scales of parts.

Probabilistic part models have been developed for 3D shapes to estimate part compatibility [Chaudhuri et al. 2011; Kalogerakis et al. 2012; Xu et al. 2012], represent relations between parts [Fish et al. 2014], and their co-variations [Yumer and Kara 2014]. These methods, however, focus on shape modeling and editing, and cannot be directly used to analyze unlabeled point clouds. Several methods have been developed for detecting part structure of unlabeled shapes without any supervision [Golovinskiy and Funkhouser 2009; Sidi et al. 2011; Hu et al. 2012; Huang et al. 2011; Huang et al. 2014] or with some user input [Kalogerakis et al. 2010; Xie et al. 2014; Makadia and Yumer 2014; Yumer et al. 2014]. All of these previous techniques assume that the entire object is visible and thus cannot be applied to predict parts in occluded regions. The method that is the most similar to ours, Kim et al. [2013a], represents part structure as an arrangement of deformable boxes that can further be used to analyze novel shapes. To handle significant occlusions we extend their approach to include relationships between parts (i.e., relative orientations, positions, scales, and symmetries). The key advantage of our representation over methods that use global reference frame (e.g., [Kim et al. 2013a; Shen et al. 2012]) is that any visible part can serve as a reference to guide the prediction of occluded parts.

3 Overview

We tackle the shape completion problem with a data-driven technique that uses a database of segmented objects to build priors on part-level shape variations. We assume that the scanned shape has been classified and segmented from the surrounding geometry — thus our priors are trained independently for each shape category. We formulate an optimization problem which explores the space of possible part arrangements that are compatible with the provided partial scan. In a training phase, our system extracts the priors from a shape database, and then uses them during the inference to guide the search towards more plausible solutions. Crucially, those priors capture the *relations* between parts so that the observed parts can inform about the presence and location of the unobserved parts. Moreover, this framework naturally enables the exploitation of partial symmetries contained within a part, or between multiple parts.

We use a simple part model and associate a small number of continuous parameters for each part to increase their expressiveness. Specifically, the representation of a part includes a local coordinate system (i.e. a pose in $\mathrm{SE}(3)$ including an offset and an orientation) and an axis-aligned anisotropic scaling. Our priors on part relations capture the covariance of those continuous parameters between all possible pairs of parts. Those pairwise relations enable us to hallucinate unobserved parts in occluded regions. Given a partial scan we use the learned priors to jointly infer the structure:

i) Assignments of input points to parts.

ii) Presence, label, and continuous parameters of parts.

iii) Parameters of partial symmetries, such as a reflection planes and rotation axes.

Once a proxy structure of parts and partial symmetry relations is estimated, the shape can be completed by copying observed points to symmetric areas. If all of the repeating structures are occluded, the system relies on the recovered part structure to retrieve possible substitute parts from the part-annotated database, and copies points from the retrieved parts to complete the point cloud.

The following sections describes in detail our structure estimation process (Sec. 4) and the shape completion process that uses the inferred structure to fill missing regions (Sec. 5).

4 Structure Estimation

The first and most critical step of our method is the estimation of the shape structure for a given point cloud. In particular, given a point cloud P, the goal of this step is to detect a set of parts B, part labels L_B, part parameters Θ_B, symmetry relations A, their parameters Θ_A, and an assignment of input points to parts: $m : P \rightarrow B$. Figure 2, shows an example structure, where the position, size, and orientation of a part is depicted by an oriented box, and symmetries are depicted by planes of reflections and axes of rotation. To estimate these parameters we define an energy function that balances between the likelihood of the estimated structure in the input point cloud vs. the prior structures observed in training examples.

Several of the terms in the energy function are learned from the patterns discovered in a collection of training examples. In a training step, our system takes as input a collection of co-aligned 3D meshes with ground-truth part labels L fom the given category. We assume that our input shapes are consistently segmented and each segment has a unique label (e.g., legs of a chair have four unique labels), while not all labels have to present in every instance (e.g., only swivel chairs would have a stem part). Our system uniformly samples points on each example mesh and the resulting labeled point clouds are used as the training data, as explained in more detail in the next subsections. The following subsections define the structure parameters (Sec. 4.1), the energy function (Sec. 4.2), and the inference procedure to find the optimal set of parameters for a given input (Sec. 4.3).

4.1 Structure Parameters

Our structure is represented by a set of parts B and symmetries A (see Figure 2). We now provide details for part and symmetry parameters.

Part parameters. To ensure that our estimated part structure is consistent with the training examples, each part $b \in B$ has a unique label $l_b \in L_B$. To use our part representation for partial scans, we attach a local coordinate system to the part, which enables us to represent part relations without a global reference frame. More specifically, a part b is parametrized with a rotation $\mathbf{R}_b \in \mathrm{SO}(3)$, translation $\mathbf{t}_b \in \mathbb{R}^3$, and an anisotropic axis-aligned scaling $\mathbf{s}_b \in \mathbb{R}^3$. Thus our part parameters $\theta_b \in \Theta_B$ are defined by $\theta_b = (\mathbf{R}_b, \mathbf{t}_b, \mathbf{s}_b)$. As seen in Figure 2 (and everywhere else in the paper), we depict θ_b with a unit cube that is scaled, rotated, and translated according to part parameters.

Partial symmetry parameters. Symmetries provide additional regularization on the estimated part structure, and are also critical for the shape completion step. Symmetric counter-parts (i.e., the labels of symmetric parts) are typically provided with the training data for a given class of objects. We found that rotational and reflectional symmetries are sufficient for all datasets presented in this paper (e.g., see Figure 2). Symmetries are defined on either a single part and/or pairs of parts. For example, the back of a chair (a single part) and armrests (a pair of parts) share the same reflection plane. A symmetry relation $a \in A$ is therefore defined by its parameters $\theta_a \in \Theta_A$ and a tuple of symmetric counter-parts $(b_a, b'_a) \in B \times B$. Our inference procedure optimizes for the parameters θ_a so that

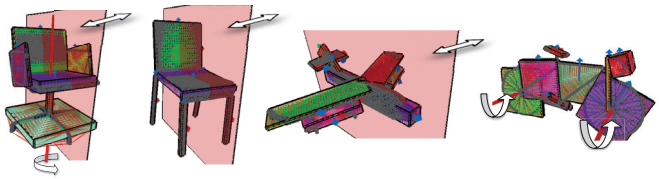

Figure 2: *This figure shows parts and symmetry relations provided in training examples. Oriented boxes depict part positions, scales, and orientations, the red plane indicates reflective symmetry, and the red line depicts the axis of rotation.*

the induced linear transformation $\mathbf{T}_a$ maps part b_a to b'_a. For self-symmetric parts, we have $b_a = b'_a$, and symmetries involving more than two parts can be modeled with several pairwise symmetries by constraining the symmetry parameters to be equal. Reflectional symmetries are uniquely described by a plane of reflection (i.e., a plane normal $\mathbf{n}_a$ and a scalar offset d_a), whereas rotational symmetries are given by a rotations axis and a rotation angle. Minimal parameterizations of 3D lines are slightly cumbersome (e.g., with Plücker coordinates). We therefore use an over-parameterization in terms of axis direction $\mathbf{d}_a$ and translation $\mathbf{t}_a$ for defining the rotation axis $\{\mathbf{X}|\mathbf{X} = \mathbf{t}_a + \lambda\mathbf{d}_a, \lambda \in \mathbb{R}\}$. These parameters define θ_a.

4.2 Structure Fitting Energy

In this section we define an objective function that is designed to capture how well the input point cloud P is represented by the structure parameters $(L_B, \Theta_B, \Theta_A)$, and a point-part assignment m. Our energy combines low-level geometric cues that come from the partial scan and mid-level regularization terms learned from the training data. The low-level terms include per-point feature-based classification scores and segmentation smoothness, while the mid-level terms penalize based on distances between scan points and parts, pairwise part relations, and partial symmetries.

Point classification. We extract local geometric features of the observed points and feed them into point classifiers whose classification scores w are used as input evidence

$$E_{\text{pnt}}(L_B, m) = \sum_{b\in B} \frac{1}{n_b} \sum_{p\in P} \delta(m(p) = b)(-\log w(p, l_b)), \qquad (1)$$

where $n_b = |\{p \in P|m(p) = b\}|$ is the set of points assigned to part b, $\delta(.)$ is the indicator function. We use the training data to learn the point classifier $w(p, l) \in [0, 1]$ which captures how likely it is that an observed point p belongs to part $l \in L$. In particular, for a given object category, we use local geometric features to build a random classification forest [Breiman 2001] for the parts provided in the training data. Since we use local features our per-point classifiers assume that all symmetric parts belong to the same category (e.g., all chair legs are assigned to the same class). To ensure that our approach can handle outliers in the input point cloud, we define a null part b_{null} and set $w(p, l_{b_{\text{null}}}) = 0.1$.

Our geometric features are computed from eigenvalues λ and eigenvectors $\mathbf{v}$ of a covariance matrix capturing the distribution of points within a local neighborhood: λ_1/λ_0, λ_2/λ_0, $\mathbf{v}_0^T\mathbf{g}$, $\mathbf{v}_2^T\mathbf{g}$ where $\mathbf{g}$ denotes the up- or gravity direction. We also use height, absolute curvature and shape diameter function (SDF). Features are estimated over neighborhoods of sizes $0.1, 0.2, 0.3, 0.4, 0.5$ of mesh radius. We compute the features with the publicly available implementation of [Kim et al. 2014].

Segmentation smoothness. We add a Potts-like regularization term to favor that nearby points belong to a similar segment

$$E_{\text{smooth}}(m) = \sum_{p_i\in P} \sum_{p_j\in N(p_i)} \delta(m(p_i) \neq m(p_j)) \left(d - \|p_i - p_j\|_2\right)^2, \qquad (2)$$

where $N(p)$ is the set of neighboring points within distance threshold $d = 0.05$ of mesh radius.

Parts-points distances. Our intuition is that scanned points should be close to the surface of the part they are assigned to and also that the part should fit tightly around the points. For simplicity, we assume that the surface of a part should roughly align with the surface of an oriented box defined by θ_b. We found this assumption to be sufficient to analyze datasets presented in this paper.

Thus, given a point cloud and a surface that corresponds to θ_b, we sample points Q on the surface, and use a variation of the sum of minimum distances (SMD) [Eiter and Mannila 1997] that factors in visibility to quantify our intuition:

$$E_{\text{SMD}}(L_B, \Theta_B, m) = \sum_{b\in B} E_{P\to Q}(l_b, \theta_b, m) + E_{Q\to P}(\theta_b), \qquad (3)$$

$$E_{P\to Q}(l_b, \theta_b, m) = \frac{1}{n_b} \sum_{p\in P} \delta(m(p) = l_b) \min_{q\in Q} \|\mathbf{p} - \mathbf{T}_b\mathbf{q}\|_2^2, \qquad (4)$$

$$E_{Q\to P}(\theta_b, m) = \frac{1}{v_b} \sum_{q\in Q} v(\mathbf{T}_b\mathbf{q}) \min_{p\in P: m(p)=b} \|\mathbf{p} - \mathbf{T}_b\mathbf{q}\|_2^2, \qquad (5)$$

where $v : Q \to \{0, 1\}$ is a visibility indicator function which evaluates to $v(q) = 1$ if point q is visible and 0 if it is occluded, $v_b = \sum_{q\in Q} v(\mathbf{T}_b\mathbf{q})$, and $\mathbf{T}_b\mathbf{q} = \mathbf{R}_b \operatorname{diag}(\mathbf{s}_b)\,\mathbf{q} + \mathbf{t}_b$.

Considering visibility is important as otherwise occluded regions could lead to truncated parts. We compute visibility by rendering input points as small spheres (0.01 of mesh radius) and use depth-buffer lookup to test visibility for a given point in space.

The points Q are sampled on the surface of an axis-aligned unit cube, with 1000 points in total. The sampling is chosen adaptively per part such that a spatially uniform sampling density results after applying the anisotropic scaling.

Part relations. The part-level regularization term captures the overall geometric layout of all the parts, based on their pairwise relations. A major challenge in defining part relations for analyzing partial scans is that those relations have to be expressed without making any assumption about a global reference frame. Our solution is to use the local coordinate system of each part to represent the distribution for parameters of another part. In particular, we design a pairwise part feature $x(\theta_b, \theta_{b'}) \in \mathbb{R}^N$ that captures the geometric relation between parts b, and b', where the pose and the size of a part b in the coordinate system of part b' is captured by

$$\mathbf{T}_{b\to b'} = \begin{bmatrix}\mathbf{R}_{b'}^T & -\mathbf{R}_{b'}^T\mathbf{t}_{b'}\end{bmatrix} \begin{bmatrix}\mathbf{R}_b & \mathbf{t}_b \\ \mathbf{0} & 1\end{bmatrix} \begin{bmatrix}s_{b,1} & & & \\ & s_{b,2} & & \\ & & s_{b,3} & \\ & & & 1\end{bmatrix} \in \mathbb{R}^{3\times 4}. \qquad (6)$$

Note that we define the part pose such that a point $\mathbf{p} \in \mathbb{R}^3$ expressed in part b's local coordinate system is mapped to a global reference frame by $\mathbf{R}_b\mathbf{p} + \mathbf{t}_b$. Therefore, the first three columns of $T_{b\to b'}$ capture the orientation and scale of part b as seen from part b''s local coordinate system, whereas the last column denotes the centroid of part b again seen from part b'. Since we assume that training shapes are consistently oriented upwards with respect to consistent gravity direction $\mathbf{g}$, we also include the absolute height as a feature:

$$\mathbf{h}_b^T = \mathbf{g}^T \begin{bmatrix}\mathbf{R}_b & \mathbf{t}_b\end{bmatrix} \begin{bmatrix}s_{b,1} & & & \\ & s_{b,2} & & \\ & & s_{b,3} & \\ & & & 1\end{bmatrix} \in \mathbb{R}^4. \qquad (7)$$

The complete feature is then the concatenation:

$$\mathbf{x}(\theta_b, \theta_{b'}) = \left(\operatorname{vec}(\mathbf{T}_{b\to b'})^T \quad \mathbf{h}_b^T \quad \operatorname{vec}(\mathbf{T}_{b'\to b}) \quad \mathbf{h}_{b'}^T\right)^T, \qquad (8)$$

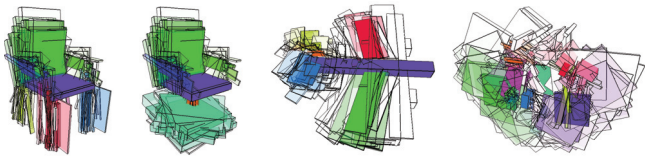

Figure 3: *This figure visualizes the variance of part structure for a single reference part (chair seat, airplane fuselage, and bike front frame). High color saturation indicates low pairwise part relation energy E_{pair} between the reference and a colored part.*

where $\mathrm{vec}(\mathbf{T})$ denotes the column-wise vectorization of matrix $\mathbf{T}$.

We learn a prior on the feature described in Equation 8 using the training data. In particular, we use PCA on each segment to find an initial guess for an oriented box, and run an ICP procedure to anisotropically deform the box to the underlying segment and estimate position and scale. We sort the PCA-based axes to have the same direction and ordering as their closest global coordinate axes to ensure that part parameters are compatible across all shapes (this becomes feasible since all the training shapes are co-aligned). Then, for every pair of part labels (l, l'), the distribution of the feature vector for parts with those labels is modeled with a multivariate normal distribution. We then extract the L^2 mean vectors and covariance matrices of feature vectors $\mu(l,l') \in \mathbb{R}^N, \Sigma(l,l') \in \mathbb{R}^{N\times N}$, and these define our probabilistic part model. We now define the pairwise part relation penalty using the Mahalanobis distance:

$$E_{\text{pair}}(L_B, \Theta_B) = \sum_{(b,b')\in B\times B} \|x(\theta_b, \theta_{b'}) - \mu(l_b, l_{b'})\|^2_{\Sigma(l_b,l_{b'})}. \quad (9)$$

Figure 3 shows examples how the pairwise relation energy varies according to the poses and scales of parts for a single reference part. This pairwise part relation term is crucial to reason about the overall structure of the shape. For example, this term can resolve noisy point classifier scores, where the point-to-part assignment is ambiguous or where the correct part-assignment is not ranked at the top by the point classifier. Moreover, the input to a point classifier is generally restricted to a local neighborhood around a point, so they cannot discern symmetric parts. Lastly, pairwise relations between observed and an unobserved parts can be aggregated to generate a prediction for the pose of the unobserved part. We will make use of that in the inference stage, as will be described in the next section.

Partial and global symmetries. Symmetry is another important mid-level cue used in our energy function. It is defined by the symmetry parameters Θ_A, where A is the set of expected symmetry relations that were observed in the training data. We define the symmetry term as:

$$E_{\text{symm}}(\Theta_A, m) = \sum_{a\in A} \sum_{p\in P: m(p)=b_a} \min_{p'\in P: m(p')=b'_a} \rho\left(\left\|p' - \mathbf{T}_a p\right\|_2^2\right), \quad (10)$$

which roughly captures for each symmetry a how well a point p on a part $b_a = m(p)$ maps to its symmetric counter-part b'_a under the transformation $\mathbf{T}_a$. Here, $\rho(d) = \min(d, d_{\max})$ is a truncated robust cost to handle partially observed symmetric parts.

Constraints. In our optimization we represent the rotation of a part as a 3×3 matrix $\mathbf{R}_b \in \mathbb{R}^{3\times 3}$ with an additional orthogonality constraint $C(\Theta_B) : \forall b \in B : \mathbf{R}_b^T\mathbf{R}_b = \mathbf{I}_3$. The determinant of $\mathbf{R}_b$ is checked during initialization s.t. $|\mathbf{R}_b| = 1$ to ensure valid rotation matrices (during continuous optimization, the update step would need to be quite large to reach a state with $|\mathbf{R}_b| = -1$ which we in fact never observed). Unit length constraints $C(\Theta_A)$ are added for plane normals $\mathbf{n}_a$ and direction vectors $\mathbf{d}_a$ of rotational symmetries.

We also include symmetry relation constraints between part coordinate systems θ_b that ensure that part-to-part symmetries are preserved at the scale of boxes. For parts b_a and b'_a related by a partial symmetry a we define the following constraint:

$$\begin{bmatrix}\mathbf{R}_{b'_a} & \mathbf{t}_{b'_a}\\ \mathbf{0} & 1\end{bmatrix}\mathrm{diag}\left([\mathbf{s}_{b'_a}, 1]\right) = \mathbf{T}_a\begin{bmatrix}\mathbf{R}_{b_a} & \mathbf{t}_{b_a}\\ \mathbf{0} & 1\end{bmatrix}\mathrm{diag}\left([\mathbf{s}_{b_a}, 1]\right). \quad (11)$$

Let us call the set of those part-symmetry constraints by $C(\Theta_A, \Theta_B)$.

Final energy. In the inference procedure we optimize the following final objective function:

$$E = w_1E_{\text{pnt}} + w_2E_{\text{smooth}} + w_3E_{\text{SMD}} + w_4E_{\text{pair}} + w_5E_{\text{symm}}$$
$$\text{s.t. } C(\Theta_A), C(\Theta_B), C(\Theta_A, \Theta_B), \quad (12)$$

where w_i are weight parameters, we empirically set $w_1 = 0.0025$, $w_2 = w_3 = 1, w_4 = 10^{-4}, w_5 = 10^2$ for all experiments presented in this paper.

4.3 Structure Inference

In this section we present an optimization procedure for Equation 12 with respect to the parts B (parameters Θ_B and labels L_B), symmetries A (parameters Θ_A), and point-to-part assignments M_P. This is a mixed discrete-continuous optimization problem, which is difficult to optimize over all variables jointly. Thus, we provide an alternating optimization scheme that selects an appropriate approach to update a subset of variables, holding the remaining ones fixed. We initialize our optimization with feature-based per-point classification, and then alternate between optimizing for part labels L_B and part pose parameters Θ_B (as a global discrete optimization), point-to-part assignments M_P, part pose parameters Θ_B and symmetry parameters Θ_A (as local continuous optimization), and generation of additional candidate parts for occluded regions. The global discrete optimization provides us with a rough estimate of low-energy pose parameters Θ_B and the continuous optimization further refines that estimate. To keep our exposition simple, we initially assume that all shapes in the database have the same number and labels of parts (i.e., $|L| = |B|$), and then we explain how this is generalized to cases when not all parts exist in all shapes. Figure 4 illustrates our optimization pipeline.

Initialization. Our method is bootstrapped by estimating candidate part parameters based on the per-point classification scores $w(p, l)$. Several candidate part representations Θ^{cand} are generated for each part label by decomposing the input scan into multiple segments. In particular, we consider all points p that are assigned to label l with high confidence $w(p, l) > \tau_{\text{class}}$ as vertices of a graph (all our experiments set $\tau_{\text{class}} = 0.7$). Edges are introduced if the distance between two points is below $\tau_{\text{dist}} = 0.2$ of the scan radius. Each connected component of the resulting graph yields a hypothesis $\theta^{\text{cand}} \in \Theta^{\text{cand}}$ by fitting an oriented box (the same procedure as in training parameter estimation is used). Each hypothesis also stores which points have been used for its estimation.

However, unlike the training set, we cannot expect the partial input scan to be co-aligned with the rest of the database. PCA-based axes are therefore ambiguous in their direction (and often, their ordering). Thus, instead of adding a single hypothesis θ^{cand} per connected component, we in fact add 24 of them: each possible choice of assigning PCA-axes to a right-handed coordinate system provides one hypothesis. The next optimization step then picks the most suitable hypothesis for each part.

Part labels and orientations prediction. We start our iterative algorithm with a discrete optimization step which selects the best part representations Θ_B and labels L_B from a set of hypothesis,

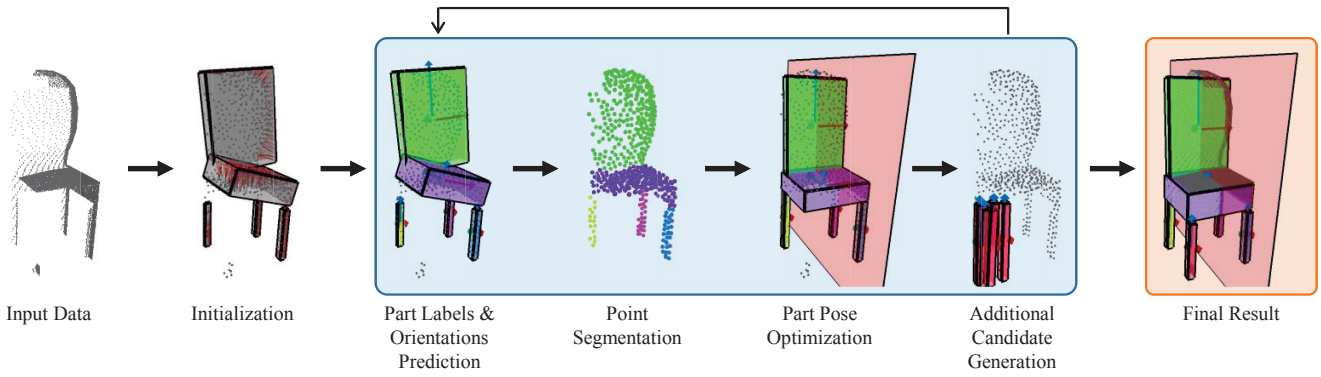

Figure 4: *This figure illustrates our inference pipeline. First, we detect initial part representations, and then iteratively solve a global MRF to find labels and orientations of parts, re-segment the point cloud, optimize for continuous part pose parameters and symmetries, and generate additional suggestions for novel parts. Our method iterates until part parameters remain fixed which leads to the final structure prediction (on the right).*

while keeping the remaining variables fixed. We formulate this step as a Markov random field, where nodes are candidate representations Θ^{cand}, pairwise potentials are defined by part relations E_{pair}, and unary potentials are defined with per-point energies E_{pnt} (based on points assigned to a given representation). The node labels are defined by part labels with an extra null part $L \cup \{l_{\text{null}}\}$. The null part exists because only $|B|$ of $|\Theta^{\text{cand}}|$ can correspond to real parts. A unary cost of $\tau_{\text{null}} = -\log(w(p, l_{b_{\text{null}}}))$ must be paid to assign a node to the null part. To ensure that there is at most one part per label, the pairwise cost for two representations to have the same label is set to $\tau_{\text{same}} = \infty$. This multi-label assignment problem is solved using TRW-S algorithm [Kolmogorov 2006], which effectively minimizes

$$\min_{\Theta_B \subset \Theta^{\text{cand}}, L_B \cup \{l_{\text{null}}\}} w_1 E_{\text{pnt}} + w_4 E_{\text{pair}}, \tag{13}$$

where the energy terms are understood to be augmented with the unary cost for label l_{null} and pairwise mutual exclusion.

Point segmentation. After estimating part labels and parameters, we re-estimate the point-to-part assignments M_P while keeping all terms not affected by these assignments fixed. This is naturally treated as a segmentation problem. We again construct a MRF where this time, nodes are points and the labels are point assignments to parts B. We define unary potentials based on the point-to-part distances $E_{P \to Q}$ and classifier scores E_{pnt} and pairwise terms based on the Potts-model E_{smooth}. In this step we ignore part-to-point distances $E_{Q \to P}$ and symmetric distances E_{symm} since it is computationally impractical to estimate the involved correspondences without knowing all point-to-part assignments. The resulting MRF is solved with TRW-S algorithm, thereby minimizing

$$\min_{M_P} w_3 E_{P \to Q} + w_1 E_{\text{pnt}} + w_2 E_{\text{smooth}}. \tag{14}$$

Part pose optimization. Now that the coarse part structure of the shape is estimated and points are segmented, we refine the continuous part and symmetry parameters Θ_B, Θ_A. As part parameters change the correspondences between points and parts (E_{SMD}) and the notion of symmetric points (E_{symm}) change as well. Thus, in this step we leverage an ICP-inspired procedure, where we alternate between estimating corresponding points ($P \to Q$, $Q \to P$, and $P \to_a P$), and parameters Θ_B, Θ_A. The first step is a trivial nearest neighbor lookup. To update the continuous parameters we use an interior point algorithm [Wächter and Biegler 2006] using numerically estimated gradients, optimizing the following:

$$\begin{aligned} \min_{\Theta_B, \Theta_A} \; & w_3 E_{\text{SMD}} + w_4 E_{\text{pair}} + w_5 E_{\text{symm}} \\ \text{s.t.} \quad & C(\Theta_A), C(\Theta_B), C(\Theta_A, \Theta_B) \end{aligned} \tag{15}$$

The algorithm terminates when the point-to-part and symmetric point-to-point correspondences do not change.

Additional candidate generation. All previous iterative steps are likely to produce locally optimal solutions that are close to the initial part candidates Θ^{cand}. In our initialization step we produced these candidates based on the observed point cloud, but now that we estimated a global part structure for the observed points we have an opportunity to augment the candidate set with parameters for occluded parts, that can only be inferred via part relations and symmetry. Similarly to Fish et al. [2014], for each part b in the initial candidate set, we generate a hypotheses for a missing part parameter $\theta_{b'}$ using the conditional mean of the pairwise relation between b and b' given θ_b. Note that we reject parameters $\theta_{b'}$ that suggest that there is a part in an un-occluded area without points support. In particular, we reject such hypothesis if more than 80% of the surface is visible. We then repeat the iterative procedure starting with part labels and orientation prediction with the new candidate representations Θ^{cand} that include best solutions from the previous iteration as well as newly generated candidates. This interplay between refining poses, sampling new hypothesis based on non-local evidence, and solving an assignment problem leads to a powerful framework for recovering a proxy part structure.

Handling variance in part topology. Most of the previous discussion focused on cases when one expects all parts to exist in all shapes, i.e., $|B| = |L|$. This does not hold in practice when object instances can contain different alternatives for parts (e.g., swivel chair vs four-legged chair), or can have optional parts (e.g., arm-rests). In our training dataset we record all possible permutations of parts and run an independent optimization for each part configuration. We then pick the lowest-energy solution.

5 Shape Completion

We leverage the estimated part structure to complete the shape, using part information Θ_B and symmetry information Θ_A. In this final step of our pipeline our goal is to densify the point cloud with additional points that correspond to the geometry in the occluded regions.

The symmetry is always a preferred source of data since the scanned data itself is likely to have qualities of the original attributes (if it is indeed symmetric). Thus, we copy all observed points that belong to symmetric parts to their symmetric counter-parts with symmetry transforms $\mathbf{T}_a$. Let us denote these reflected points as P_{symm}.

To complete regions that do not have visible symmetric parts, we next retrieve candidate parts from the database. If our part includes visible points we use the method of Shen et al. [2012] for retrieval.

Specifically, for each part b, we define a voxel occupancy grid V_b with voxel size 0.01 for each axis. We compute a similar grid for training parts $V_{b'}$, and measure part-to-part similarity as in previous work: $(1-\alpha)\frac{<V_b,V_{b'}>}{N_b}+\alpha\frac{<V_b,V_{b'}>}{N_{b'}}$, where N_b and $N_{b'}$ is the number of occupied voxels in grids V_b and $V_{b'}$, and $\alpha = 0.7$. If a part is fully occluded we simply compare the dimensions of the bounding box, defining similarity as: $\exp(-\|s_b - s_{b'}\|^2)$. We denote those database-driven points as P_{data}.

We use our visibility grid as defined by the term $v(q)$ to decide whether to use any of the two point sets. Specifically, for each occluded grid cell, we check if there exist points in P_{symm} to complete the shape in that cell. If this is not the case, points from P_{data} are used. This produces a densified point cloud P_{complete}.

6 Results

In this section, we demonstrate experimental results of the proposed algorithm. First, we define a novel synthetic benchmark for evaluating shape completion techniques. We then evaluate our method on the benchmark, as well as symmetry-only and database-only completions where structure is still estimated with our algorithm. We also compare to previous work on shape completion that used alternative structure estimation algorithms that were more prone to partiality in the input scans. Finally, we show qualitative results on Kinect scans.

Shape Completion Benchmark. We use several public shape collections to generate our quantitative benchmark. Our benchmark includes 58 airplanes, 64 chairs, and 35 bicycles from Part Assembly dataset [Shen et al. 2012], 37 tables from ShapeNet[1], 287 chairs from COSEG dataset [Wang et al. 2012]. All models in these datasets have been manually decomposed into 6-9 parts, annotated with symmetry planes and axes (see Figure 2), co-aligned and approximately scaled to the size of real-life objects.

In our experiments, we follow a leave-one-out approach where training is performed on all but one left-out shape. We depict one such experiment in Figure 5. For the hold-out shape we sample 100,000 points P_{true}, randomly select a viewpoint, and remove the occluded points to generate a partial point cloud P (Figure 5.a). We then run our method to produce a completed shape P_{complete} (Figure 5.b), and measure two types of errors: accuracy and completeness. The accuracy records the fraction of points in P_{complete} that are within $\tau_{\text{eval}} = 0.02$ of a point in P_{true}. Figure 5.c depicts accuracy errors for all points on P_{complete} where blue indicates that error is below τ_{eval} (i.e., non-blue were incorrectly added by our method). The completeness records the fraction of points in P_{true} that are within τ_{eval} of a point within P_{complete}, and Figure 5.d depicts completeness errors for all points in P_{true} (i.e., non-blue points were not completed by our method). The supplemental material also provides statistics for the accuracy of per-point labels produced by our structure prediction algorithm.

Shape completion results. We evaluate our method primarily by completing synthetic scans. We demonstrate some of the challenging cases successfully handled by our method in Figure 6. Our inference procedure detects structure despite significant missing regions, and then takes advantage of symmetries and exemplar parts to complete the missing data. We also refer to the supplemental material for the results on the entire benchmark.

Comparison to database and symmetry priors. In this experiment, we evaluate the importance of fusing database and symmetry priors. In particular, we compare P_{complete}, $P_{\text{data}} \cup P$, and $P_{\text{symm}} \cup P$. Figure 7 demonstrates some qualitative results. Unsurprisingly,

[1] http://shapenet.cs.stanford.edu

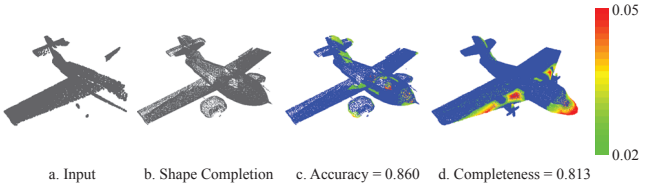

Figure 5: *This figure demonstrates our evaluation technique. Given the input point cloud (a), we run our algorithm to complete the shape (b), and then evaluate accuracy and completeness. The accuracy image (c) colors the points of the completed shape based on the distance to the nearest point on the true shape (see error bar on the right). The completeness image (d) colors the points of the true shape based on the distance to the nearest point on the completed shape. See supplemental material for the results on the entire benchmark.*

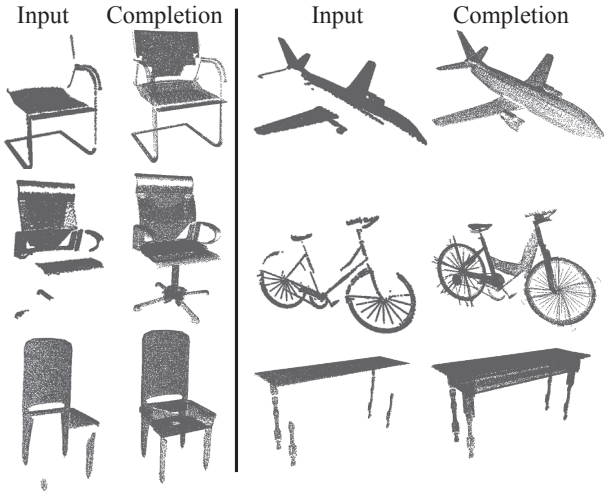

Figure 6: *This figure demonstrates some challenging partial scans successfully completed by our algorithm.*

database-only completions are often not accurate since the database might not include shapes with similar parts. While symmetry-only completions produce more accurate results, they typically have large incomplete areas. Figure 11 supports these qualitative observations quantitatively: in this figure the centre of a box in a 2D plane corresponds to its accuracy (x-axis) and completeness (y-axis), and the size of the box corresponds to standard deviations. We observe that our method provides the most complete surface, at expense of being slightly less accurate than symmetry-only completions.

Comparison to Shen et al. [2012]. The completion results we discussed so far all benefit from our structure estimation algorithm. In this experiment we compare our work to alignment-based structure estimated in Shen et al. [2012]. Since their implementation is not publicly available and relies on a slightly different input (they use color images in addition to depth), we re-implemented a version of their algorithm that only leverages geometric cues. In particular, we first try every rotation around the up axis to align every training 3D shape to the input point cloud. Since our training data is co-aligned we pick the best rotation that minimizes the sum of average distances between input point cloud and training examples. We then select parts to complete the surface in occluded regions using Shen et al.'s part-to-part similarity metric. Note that we do not implement the final part conjoining step, but we believe that this does not affect the comparison, since our method would also benefit from enforcing contacts after parts are retrieved. We present some qualitative results in Figure 8 (for fairness, we do not use symmetry cues for our results in that figure). Note that the Shen et al.'s method fails to correctly estimate front orientations when large missing regions are present, and also fails to align parts if the size of the input object is not typical for the objects in the database. This leads to an inferior quantitative performance, as can be seen in Figure 11.

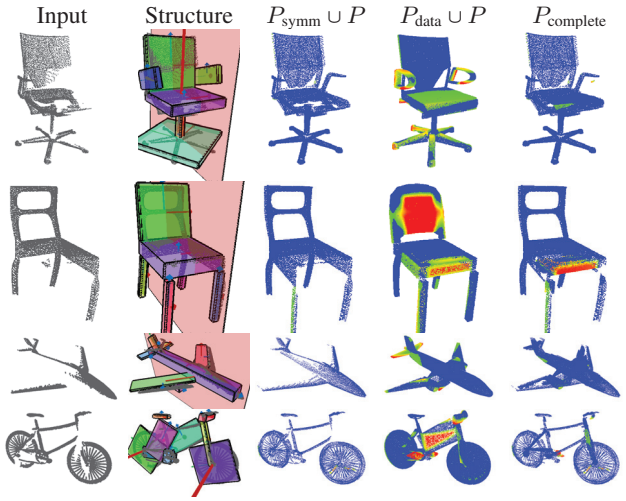

Figure 7: *This figure demonstrates our shape completion results: input point cloud, estimated structure, symmetry-only completion accuracy, data-base only completion accuracy, and the final combined result. Note how symmetry-only completion consistently leaves large blank regions, and database-driven approach hallucinates pieces of geometry that should not be there (see red regions). In contrast, our method effectively combines both priors to produce accurate and complete shapes.*

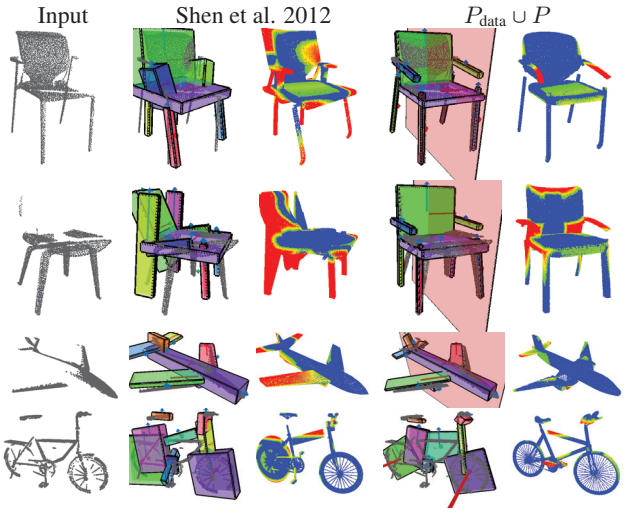

Figure 8: *In this figure we compare our method to our implementation of Shen et al. [2012]. In their work the structure is estimated by rigidly aligning a database shape to a partial scan, which is prone to error since both shapes must have relatively compatible global coordinates. In our work, each part provides a coordinate system, so we estimate part structure more reliably, which leads to more accurate completions (for fairness we do not use symmetry cues for our completions in this figure).*

Comparison to symmetry detection. Existing symmetry detection techniques only handle small amount of occlusions since they infer the symmetric relations from the data itself, and thus, by nature, have to be able to observe some repetitions before detecting a symmetry. Our symmetry detection, however, is data-driven, and it allows handling more severe occlusions. In particular we compared our method to a state-of-the art symmetry detection algorithm — a method based on Planar Reflective Symmetry Transform [Podolak et al. 2006]. Figure 9 demonstrates shape completions with detected symmetries. We observe that missing regions can mislead Podolak et al.'s method to pick an approximate symmetry (e.g., if one wing of an airplane is missing top-to-bottom symmetry becomes stronger than left-to-right symmetry). This problem is quantitatively evaluated in Figure 10 showing that Podolak et al.'s approach gets confused by top-to-bottom symmetry of an airplane in about 20% of the cases while our method does not make this mistake (see supplemental material for the result over all categories). On the other hand, our

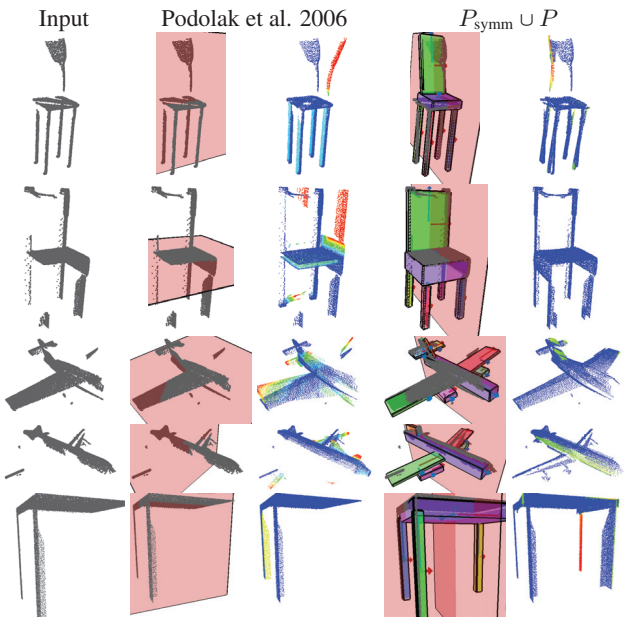

Figure 9: *In this figure we compare our method to the symmetry detection technique of Podolak et al. [2006]. Detection of a symmetry plane under severe occlusions is difficult since some of the repeating elements might be occluded. Our data-driven approach is more reliable in such cases because it exploits part priors to estimate symmetries. For fairness, in this figure, we show symmetry-only completions without copying points from the database.*

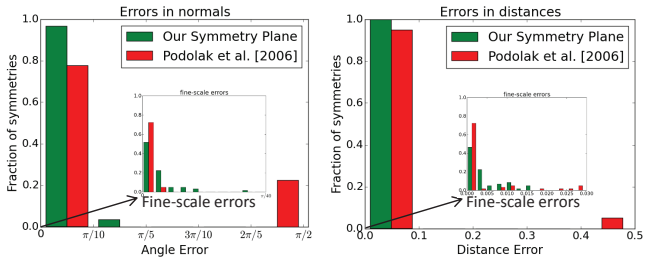

Figure 10: *This figure depicts the performance of our method and Podolak et al. [2006] for symmetry detection in the airplane dataset by evaluating the error of the predicted symmetry planes. The left histogram shows errors in plane normals (as the angle between the normal of the true symmetry plane and the predicted normal), and the right histogram shows errors in distances to the origin (as a difference between true and predicted distances). Note that our method (green bars) does not have large-scale errors in normals in comparison to Podolak et al. [2006] (red bars) which fails in about 20% of the cases (see Figure 9 for some example failures). The inset shows that our approach suffers from smaller-scale errors, which explains why both symmetry-only completions have comparable average quantitative performance on the benchmark.*

method suffers from more fine-scale errors in cases where [Podolak et al. 2006] succeeds. This can be explained by the part priors that introduce a small bias. We believe that these small errors can be eliminated if needed, e.g. by gradually increasing the weight w_5 of the symmetry energy term E_{symm} throughout the optimization. Although quantitative performance of both techniques for symmetry-based shape completion is comparable with self-occlusions (see Figure 11), our symmetry-based completion becomes more reliable under stronger occlusions (see Figure 12).

Effect of stronger occlusions. In this experiment we evaluate the effect of stronger occlusions. In addition to self-occlusions we also generate a random square that occludes a surface from a given viewpoint, and tweak the area until only about 50% of the surface is visible (effectively decreasing the completeness of the baseline result). We evaluated the results on airplanes and bicycle datasets and observed that our symmetry-only reconstructions produce more complete results whereas our database-only reconstructions improve more significantly over alignment-based methods. As before the

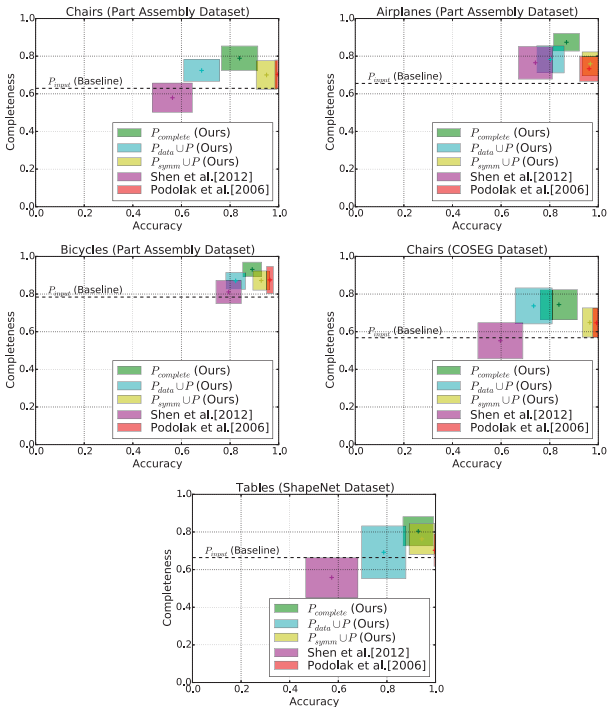

Figure 11: *This figure summarizes quantitative results on our shape completion benchmark. Each box corresponds to performance of a different method, where the centre of a box corresponds to mean accuracy (x-axis) and mean completeness (y-axis), and width and height are set to standard deviations. The dashed line provides a baseline of using the input itself as the output (the accuracy of this method is always 1). Not surprisingly, symmetry-based completions provide the highest accuracy, but typically lack completeness, while database-driven completions are less accurate. Our method provides the most complete shapes in all datasets with relatively small drop in accuracy.*

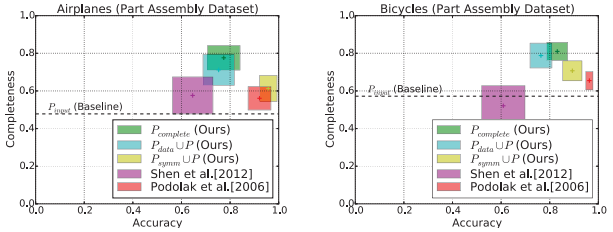

Figure 12: *We evaluate the performance of our and alternative methods in the presence of stronger occlusions (simulated by a square between the surface and the viewpoint, which effectively shifts the completeness of the baseline). The gap in performance between our approach and alternative techniques increases as occlusions get stronger. See supplemental material for all results.*

fused result produces the most complete shapes (see Figure 12).

Effect of individual energy terms. In this experiment we evaluate the effect of each individual energy term used in the structure estimation algorithm (Section 4.2). In particular, we run our pipeline on part assembly chair dataset with the following modifications:

(a) we remove per-point classifiers everywhere except initialization ($w_1 = 0$ and the points are labeled based on their distances to the oriented part boxes),

(b) we remove the smoothness energy ($w_2 = 0$),

(c) we remove the part-related energy terms ($w_4 = w_5 = 0$, the local coordinate system for parts is based on the initial guess),

(d) we remove the symmetry terms and constraints ($w_5 = 0$),

To evaluate the quality of the predicted structure we measure the Hausdorff distance between estimated and ground truth part abstractions (i.e., the oriented boxes). As shown in Figure 13, the pairwise terms in (d) and per-point classifier terms in (b) greatly affect to the overall performance. Although the symmetry terms and constraints

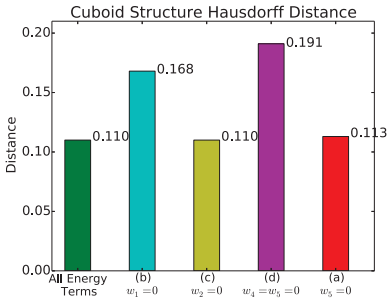

Figure 13: *This plot shows the Hausdorff distance between estimated part structure and ground truth when disabling some of the energy terms. For (a)-(d) cases, please see the text.*

(a) do not significantly affect Hausdorff distance, the accuracy of symmetry planes and rotation axis is important for the quality of the reconstruction. While smoothness term (c) does not play a significant role for benchmark examples, it is a useful regularization for noisier Kinect scans.

Analyzing Kinect scans. We tested our method with the scan data obtained with a Kinect 2. We manually generated a background segmentation mask — however, one could use the heuristic provided in [Shen et al. 2012] to find the foreground object. Note that unlike simulated scans, the data also exhibits high-frequency surface noise due to sensor limitations. Despite the noise, our method estimated the correct structure, and produced reasonable shape completions as shown in Figure 14. These preliminary results suggest that our system can be a useful tool for autonomous agents interacting with environments. First, our estimated part structure could be directly used to understand the orientation of the object and semantics of its regions (e.g., with the provided analysis the agent might avoid putting items on the seat to keep the area open for a human). Second, the completed point cloud can be used in other geometry analysis tasks that rely on global cues, such as predicting affordances [Kim et al. 2014] and understanding style [Xu et al. 2010]. Third, the predicted completions can guide the autonomous agent (or a human user) to scan the regions that are likely to yield additional geometric details. While it is possible for us to also directly deform meshes from the input database to the completed point clouds, we opted out to presenting just the densified output, aiming to preserve as much of the original features of the scanned shape as possible. This leads to results that are less amenable to photorealistic image synthesis, yet, might be more useful in other virtual reality applications. For example, one might use the densified points to quickly visualize how a for-sale product might fit in pre-defined indoor environment or use them as a coarse reference in a modeling interface. For these applications, the results can be further color-coded depending on whether a point is real or was completed with a prior.

Timing. We implemented our algorithm in a single-threaded CPU-only C++ program, and execute it on a 2.6 GHz Intel processor. In the training stage the longest step is computing local geometric features, which takes about 3min per shape. After that the classifier is built in about 3min (as measured for a dataset of 60 shapes). Other steps of the training stage are negligible. At the inference stage, a model is analyzed in about 4 minutes, where about 0.6min is spent on global label and orientation prediction, about 3.4min are spent on continuous pose optimization, and the other steps are negligible.

Limitations. Our method has some limitations. First, our model does not take part connections into account. We believe that one can use optimization proposed by Shen et al. [2012], to ensure that the final model is connected. Second, we can improve our results for Kinect scans by leveraging RGB image information as in Shen et al. Third, we also assume that the shapes do not have significant

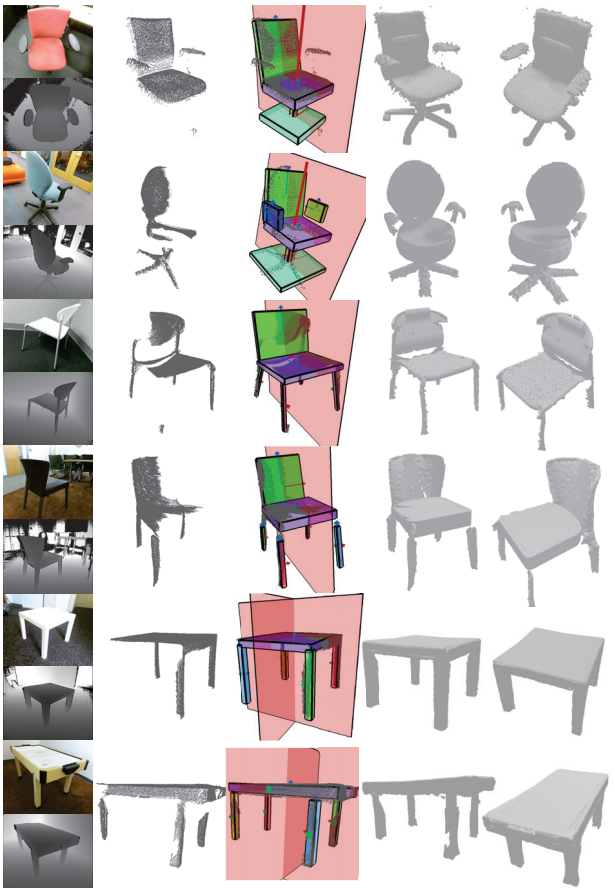

Figure 14: *We evaluate the performance of our method on single-view scans acquired with Kinect 2. We show the input RGBD data (note that the color image is not used by our method), a foreground point cloud, our estimated part structure, and the completed shapes.*

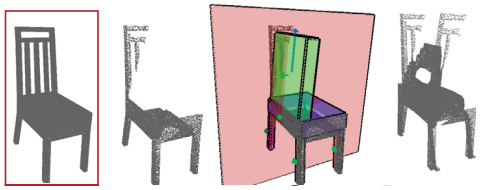

Figure 15: *Failure example due to partial symmetry of the input point cloud. The visible part of the back is interpreted as a side of a back, so the estimated structure is incorrect.*

variations in which parts form an object, and thus it is more suitable for analyzing individual man-made objects with relatively consistent part structure. However, we believe that our method can be extended to 3D scenes with two critical changes: redefining part relations to include a notion of hierarchy (i.e., groupings of objects and parts reduce complexity of the representation [Liu et al. 2014]), and generalizing the symmetry term to a regularization term that favors repetitions on different levels of the hierarchy (i.e., objects often repeat in a scene, but transformation typically cannot be learned [Kim et al. 2012]). Finally, our structure estimation fails in some cases, for example Figure 15 demonstrates that partial symmetry of an object can lead our system to an incorrect interpretation of the observed data.

7 Conclusion and Future Work

In summary, we provide the first system that completes a shape using both symmetric relations and database parts. In addition to leveraging this high-level idea, our main technical contribution is a data-driven technique for estimating shape structure from incomplete point clouds. The key difference from previous approaches is that our method does not rely on a global coordinate system, instead every part defines local coordinates, and then all parts are jointly optimized to find the most plausible arrangement. This enables the prediction of parts in occluded regions, and the estimation of symmetries even if the input partial scan is asymmetric due to occlusions. Finally, we devise a quantitative benchmark for data-driven shape completion, which will be a valuable asset for the community. We demonstrate that our method outperforms all alternative techniques on the benchmark.

In the future, we plan to extend our framework to complete occlusions in 3D scenes with a hierarchical representation for part relations. Another interesting research direction is to enable user interaction to fix problems with shape completions. For example, the input in Figure 16 is ambiguous since there are multiple interpretations for the support structure of the chair. We modified our optimization to return multiple local minima in the energy function and it prompted two interpretations for the user to choose from. We will continue developing interfaces for exploring variations in plausible shape completions. The main technical challenge for that direction is the generation of sufficiently diverse, but still plausible solutions.

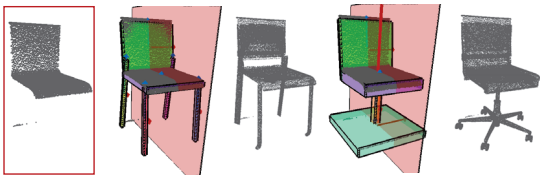

Figure 16: *The input points cloud is ambiguous so we show several local minima in the solution space to let the user choose from plausible configurations.*

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions. This project was supported by NSF grants CCF 1011228, DMS 1228304, AFOSR grant FA9550-12-1-0372, ONR MURI grant N00014-13-1-0341, a Google Focused Research Award, the Korea Foundation and Advanced Studies, and the Max Planck Center for Visual Computing and Communications.

References

BREIMAN, L. 2001. Random forests. *Machine Learning 45*, 1, 5–32.

CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3d modeling. *ACM TOG 30*, 4 (July), 35:1–35:10.

CHOI, S., ZHOU, Q.-Y., AND KOLTUN, V. 2015. Robust reconstruction of indoor scenes. *CVPR*.

EITER, T., AND MANNILA, H. 1997. Distance measures for point sets and their computation. *Acta Informatica 34*, 2, 109–133.

FELZENSZWALB, P., MCALLESTER, D., AND RAMANAN, D. 2008. A discriminatively trained, multiscale, deformable part model. In *IEEE CVPR*, 1–8.

FISH, N., AVERKIOU, M., VAN KAICK, O., SORKINE-HORNUNG, O., COHEN-OR, D., AND MITRA, N. J. 2014. Meta-representation of shape families. *ACM TOG 33*, 4, 34:1–34:11.

GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. Consistent segmentation of 3D models. *Proc. SMI 33*, 3, 262–269.

HAN, F., AND ZHU, S.-C. 2009. Bottom-up/top-down image parsing with attribute grammar. *IEEE PAMI 31*, 1, 59–73.

HU, R., FAN, L., , AND LIU, L. 2012. Co-segmentation of 3d shapes via subspace clustering. *SGP 31*, 5, 1703–1713.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. In *SIGGRAPH Asia*.

HUANG, Q., WANG, F., AND GUIBAS, L. 2014. Functional map networks for analyzing and exploring large shape collections. *ACM TOG 33*, 4 (July), 36:1–36:11.

JANOCH, A., KARAYEV, S., JIA, Y., BARRON, J., FRITZ, M., SAENKO, K., AND DARRELL, T. 2011. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshop on Consumer Depth Cameras in Computer Vision*, 1168–1174.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. In *SIGGRAPH*, 102:1–102:12.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM TOG 31*, 4 (July), 55:1–55:11.

KAZHDAN, M., AND HOPPE, H. 2013. Screened poisson surface reconstruction. *Transactions on Graphics 32*, 3.

KIM, Y. M., MITRA, N. J., YAN, D.-M., AND GUIBAS, L. 2012. Acquiring 3d indoor environments with variability and repetition. *ACM TOG 31*, 6 (Nov.), 138:1–138:11.

KIM, V. G., LI, W., MITRA, N. J., CHAUDHURI, S., DIVERDI, S., AND FUNKHOUSER, T. 2013. Learning part-based templates from large collections of 3d shapes. *ACM TOG 32*, 4, 70:1–70:12.

KIM, Y. M., MITRA, N. J., HUANG, Q., AND GUIBAS, L. 2013. Guided real-time scanning of indoor objects. *Computer Graphics Forum 32*, 7, 177–186.

KIM, V. G., CHAUDHURI, S., GUIBAS, L., AND FUNKHOUSER, T. 2014. Shape2pose: Human-centric shape analysis. *ACM TOG 33*, 4 (July), 120:1–120:12.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE PAMI 28*, 10, 1568–1583.

LAI, K., BO, L., REN, X., AND FOX, D. 2011. A large-scale hierarchical multi-view rgb-d object dataset. In *IEEE ICRA*, 1817–1824.

LI, Y., DAI, A., GUIBAS, L., AND NIESSNER, M. 2015. Database-assisted object retrieval for real-time 3d reconstruction. *Eurographics*.

LIU, T., CHAUDHURI, S., KIM, V. G., HUANG, Q., MITRA, N. J., AND FUNKHOUSER, T. 2014. Creating consistent scene graphs using a probabilistic grammar. *ACM TOG 33*, 6, 211:1–211:12.

MAKADIA, A., AND YUMER, M. E. 2014. Learning 3d part detection from sparsely labeled data. In *3DV*, IEEE.

MARTINOVIC, A., AND VAN GOOL, L. 2013. Bayesian grammar learning for inverse procedural modeling. In *IEEE CVPR*, 201–208.

MITRA, N. J., GUIBAS, L., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. *SIGGRAPH*.

MITRA, N. J., PAULY, M., WAND, M., AND CEYLAN, D. 2013. Symmetry in 3d geometry: Extraction and applications. *CGF*.

NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *SIGGRAPH Asia 31*, 6.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3d scan completion. *SGP*.

PAULY, M., MITRA, N. J., WALLNER, J., POTTMANN, H., AND GUIBAS, L. J. 2008. Discovering structural regularity in 3d geometry. *ACM TOG 27*, 3 (Aug.), 43:1–43:11.

PODOLAK, J., SHILANE, P., GOLOVINSKIY, A., RUSINKIEWICZ, S., AND FUNKHOUSER, T. 2006. A planar-reflective symmetry transform for 3D shapes. *ACM TOG 25*, 3.

SHAO, T., MONSZPART, A., ZHENG, Y., KOO, B., XU, W., ZHOU, K., AND MITRA, N. J. 2014. Imagining the unseen: Stability-based cuboid arrangements for scene understanding. *ACM TOG 33*, 6 (Nov.), 209:1–209:11.

SHEN, C.-H., FU, H., CHEN, K., AND HU, S.-M. 2012. Structure recovery by part assembly. *ACM TOG 31*, 6 (Nov.), 180:1–180:11.

SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *SIGGRAPH Asia 30*, 6, 126:1–126:9.

SILBERMAN, N., HOIEM, D., KOHLI, P., AND FERGUS, R. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*, 746–760.

SIPIRAN, I., GREGOR, R., AND SCHRECK, T. 2014. Approximate symmetry detection in partial 3d meshes. *CGF 33*, 7, 131–140.

THRUN, S., AND WEGBREIT, B. 2005. Shape from symmetry. In *ICCV*, 1824–1831.

WÄCHTER, A., AND BIEGLER, L. T. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming 106*, 1.

WANG, Y., ASAFI, S., VAN KAICK, O., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Active co-analysis of a set of shapes. *ACM TOG 31*, 6 (Nov.), 165:1–165:10.

WU, C., LENZ, I., AND SAXENA, A. 2014. Hierarchical semantic labeling for task-relevant rgb-d perception. *RSS*.

XIE, Z., XU, K., LIU, L., AND XIONG, Y. 2014. 3d shape segmentation and labeling via extreme learning machine. *SGP*.

XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z. 2010. Style-content separation by anisotropic part scales. *SIGGRAPH Asia 29*, 5.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3d shape galleries. 57:1–57:10.

YUMER, M. E., AND KARA, L. B. 2014. Co-constrained handles for deformation in shape collections. *ACM TOG 33*, 6.

YUMER, M., CHUN, W., AND MAKADIA, A. 2014. Co-segmentation of textured 3d shapes with sparse annotations. In *IEEE CVPR*, 240–247.

ZHENG, Q., SHARF, A., WAN, G., LI, Y., MITRA, N. J., COHEN-OR, D., AND CHEN, B. 2010. Non-local scan consolidation for 3d urban scenes. *ACM TOG 29*, 4 (July), 94:1–94:9.

ZHENG, Y., COHEN-OR, D., AND MITRA, N. J. 2013. Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum 32*, 195–204.