

Symmetry Hierarchy of Man-Made Objects

Y. Wang^{1,2}, K. Xu^{1,2}, J. Li², H. Zhang¹, A. Shamir³, L. Liu⁴, Z. Cheng², and Y. Xiong²

¹Simon Fraser University, ²National University of Defense Technology, ³The Interdisciplinary Center, ⁴Zhejiang University

Abstract

We introduce symmetry hierarchy of man-made objects, a high-level structural representation of a 3D model providing a symmetry-induced, hierarchical organization of the model's constituent parts. Given an input mesh, we segment it into primitive parts and build an initial graph which encodes inter-part symmetries and connectivity relations, as well as self-symmetries in individual parts. The symmetry hierarchy is constructed from the initial graph via recursive graph contraction which either groups parts by symmetry or assembles connected sets of parts. The order of graph contraction is dictated by a set of precedence rules designed primarily to respect the law of symmetry in perceptual grouping and the principle of compactness of representation. We show that symmetry hierarchy naturally implies a hierarchical segmentation that is more meaningful than those produced by local geometric considerations. We also develop an application of symmetry hierarchies for structural shape editing.

1. Introduction

Understanding the structure of a 3D shape has been an ongoing research effort in graphics and modeling. High-level or “intelligent” models lend themselves well to many applications such as editing, correspondence, synthesis, and animation. However, most 3D models today are still represented using low-level mesh representations. The definition, as well as construction, of higher-level models still remains a challenge. In this paper, we are interested in developing such models for man-made 3D objects. Man-made artifacts are characterized by high levels of regularity and repetition as reflected by their different forms of symmetry and they are often assembled from parts and sub-parts recursively. We combine these two defining characteristics of man-made shapes: symmetry and hierarchical structures, to define a novel high-level representation of shapes called *symmetry hierarchy*. Specifically, we model a 3D man-made object as a symmetry-induced, hierarchical organization of its constituent parts; see Figure 1 for two such examples.

Symmetry hierarchy does not simply detect and arrange symmetries hierarchically; it is built upon two distinct relations between shape parts: *grouping* of symmetric parts and *assembly* of connected components, where the latter is primarily governed by symmetry. Our algorithm thus operates beyond local geometry analysis for hierarchical shape decomposition, as done in existing works on mesh segmentation, constructed solid geometry (CSG), and scene graphs; it

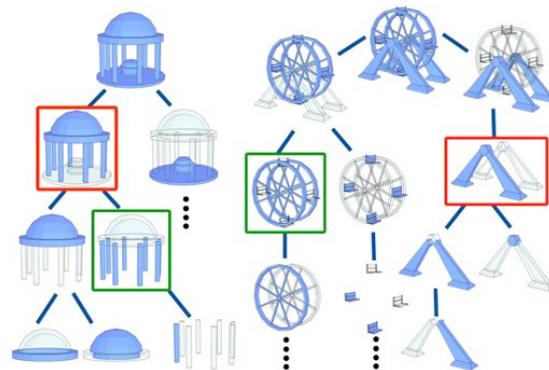


Figure 1: Symmetry hierarchy provides a structural organization of an object's parts. Parts associated with a node are colored blue. Each node represents either a grouping by symmetry (green) or assembly (red) by connectivity.

also goes beyond mere discovery of global or isolated partial symmetries. We aim to provide a natural representation of a shape with a *perceptual* and *functional* organization of its parts. The key challenge then lies in defining a set of rules that support the proper construction and natural prioritization of grouping and assembly relations.

The ultimate goal of symmetry hierarchies is shape understanding. To “understand” a shape, symmetry is expected

to play a crucial role [Ley01]. The Gestalt law of symmetry attributes symmetry as one of the fundamental principles behind *perceptual grouping* [Köh29]. For man-made objects, the ability to infer the functionalities of parts is key to high-level shape analysis and understanding. Symmetry is evidently relevant since symmetric parts tend to perform the same function and should be logically grouped together.

Symmetry analysis of man-made structures is especially appealing yet challenging due to the immensely rich varieties of organizational symmetry structures within them, even for those in the same class (e.g., think of the variety of chairs or tables one may encounter). Inferring a natural and meaningful hierarchical organization of a shape's parts is challenging even with a perfect shape segmentation and all the low-level symmetries detected. Computationally, a modest number of parts already lead to a large search space. For the shape representation to be general, the organizational rules need to handle not only symmetry grouping but also general part assembly that accounts for asymmetries in a shape. Last but not the least, a natural representation inevitably requires a high-level understanding of the shape, which is in general a highly complex cognitive process.

Overview. Our hierarchy construction algorithm takes as input a 3D mesh and returns a tree representation whose leaf nodes are the constituent parts of the mesh and the internal nodes represent *symmetry-grouping* or *part-assembly* operations. The input mesh is first segmented into primitive parts. Next, all inter- and intra-part symmetries are detected and an initial graph of all parts is created where nodes denote parts and edges denote symmetry or adjacency relations between parts. The core of our algorithm builds the hierarchy by *contracting* the initial graph recursively via

- *grouping*: creating a new node grouping two or more existing nodes whose associated shapes are symmetric to each other; the grouped shapes may not be connected.
- *assembly*: creating a new node adjoining two nodes whose associated shapes are connected.

These operations are ordered by a set of *precedence rules* which prioritize contractions of nodes whose corresponding shape structures are more tightly organized perceptually, functionally and whose organization leads to a more *compact* representation. Our implementation is primarily driven by symmetry analysis. The consideration of compactness is inspired by the minimum description length principle. This is a formalization of Occam's Razor in which the best hypothesis for a given set of data is the one that is also the simplest, e.g., compact in size. In addition, secondary consideration is given to geometric criteria, e.g., proximity and connectivity strength, to further enhance symmetry hierarchy as a meaningful organization of shape parts.

In the general setting, the question of "what the most perceptually meaningful organization of a shape's parts is?" will likely yield different answers. People's perception may dif-

fer based on knowledge and context. Our work follows one possible perceptual principle arising from symmetry cues. Even with such a restriction, the set of rules we develop are by no means complete or optimal. We do not guarantee that the symmetry hierarchy obtained is unique, but in all of our experiments, the rules have always led to unique results.

Contributions. Our main contribution is a novel structural representation for man-made shapes, the symmetry hierarchy. Our emphasis on structure rather than geometric measures such as size and angle is particularly suited to man-made objects. However small a structure is, it was designed for a purpose, to serve a function, and its structural significance may outweigh another part larger in size. Although our algorithm does not explicitly model the functionalities of object parts, we often observe the ability of our rules to reflect them in the resulting hierarchies. For example, the wheel and support of the ferris wheel model are correctly revealed in the hierarchy, as shown in Figure 1.

2. Related work

Symmetry-aware processing has received a great deal of attention lately [GPF07]. Most existing works on symmetry detection extract global [KFR04, OSG08] or isolated partial symmetries [MGP06, PSG*06, RBBK07, BBW*09, XZT*09, LCDF10]. The idea of grouping by symmetry has been applied to mesh segmentation but only to the point of grouping mesh faces which form maximal-area regions that possess certain symmetries [PSG*06, XZT*09]. These methods thrive at extracting partial symmetries from a globally asymmetric model, possibly leaving a complex globally symmetry structure unprocessed. When examining a complex object, human perception tends to organize it into a nested hierarchy of structures. Leyton [Ley01] proposes a generative theory of shapes, stipulating that symmetry plays the key role in recovering that hierarchy. Our work can be seen as a preliminary algorithmic realization of this theory.

Some existing methods can detect compound structural regularities in 3D shapes. Most notably, the work of Pauly et al. [PMW*08] detects such regularities from which a one- or two-parameter generative model can be derived. Their approach only considers regular 2D grid structures in the symmetry transformation space without performing a general hierarchical analysis. The thesis work of Martinet [Mar07] includes ideas similar to ours, in particular hierarchy construction based on detected symmetries. The goal there is to discover congruent scene components to improve the efficiency of storage and processing of symmetry-rich scenes; rules for part assembly are not considered. Our goal is shape understanding and we apply symmetry analysis to both part grouping and assembly. With a similar attempt to infer functionality from geometry, the work of Mitra et al. [MY*10] is relevant. Their construction of the interaction graph also considers inter- and intra-part symmetries as in our work,

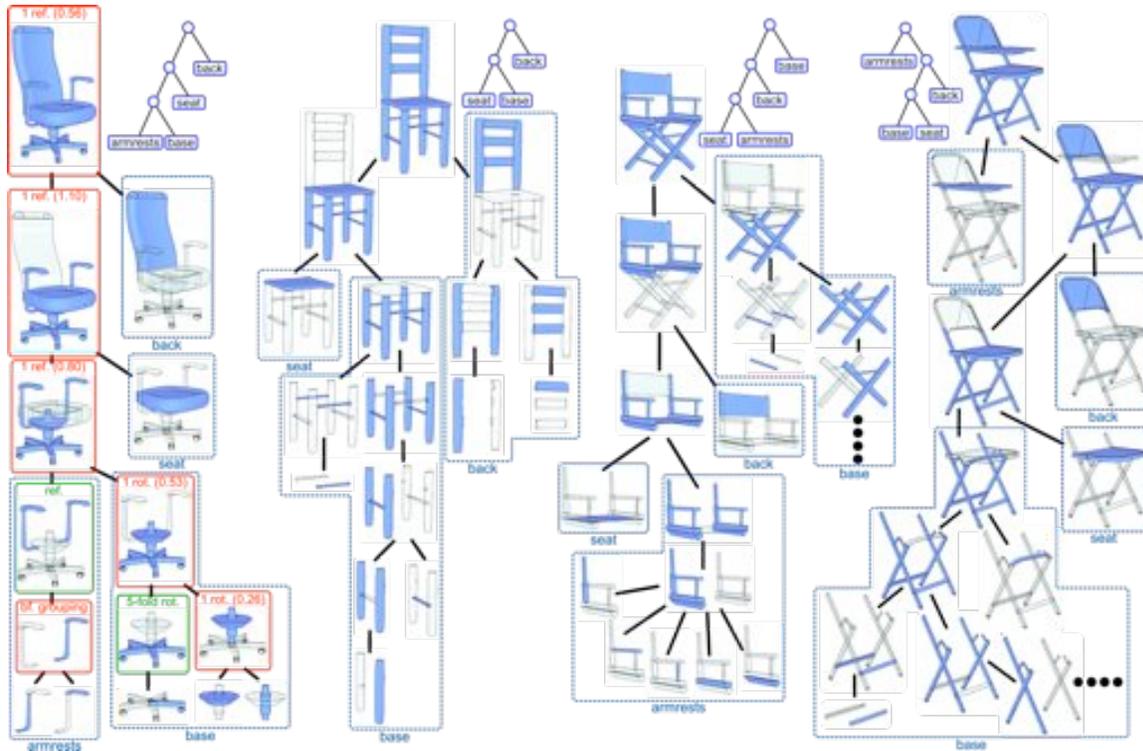


Figure 2: A set of chairs with their symmetry hierarchies. Self-symmetries and connectivity strengths (in brackets) at the nodes are shown for one model (red: assembly nodes; green: grouping nodes; “bf. grouping”: assembly before grouping). The semantic tags are not computed by our algorithm but are shown here to illustrate how functional parts (back, seat, etc.) are grouped together and revealed by our symmetry hierarchies.

but it does not build a hierarchical part organization. Analyses of mechanical part assemblies can be traced back to early works in robotics, e.g., [Liu90], where the theory of symmetry groups was employed to study solid surface contacts.

Inverse procedural modeling also aims to recover structural information from a given shape. Benes et al. [BSM*10] generate an L -system which reproduces a given 2D line art. Bokeloh et al. [BWS10] perform local similarity and symmetry search to decompose a 3D model and synthesize new shapes procedurally via insertion, deletion, and replacement. They do not replicate symmetric parts but look for symmetric areas called *docking sites* so that the non-symmetric parts can be replaced. A common feature shared by these works and ours is the emphasis on structural decomposition. However, they seek procedural modeling rules for shape reproduction and manipulation, while we target a generic shape representation applicable to a wider variety of tasks including segmentation, correspondence, and editing.

Works on hierarchical shape analysis are naturally tied to segmentation. Most methods base their analysis on the geometric properties of *individual* parts such as fitting to primitives [AFS06], shape diameters [SSS*10], convexity and compactness [KJS07], or rely on an intrinsic surface

metric for clustering [KT03, LZ07]. Structural approaches mainly focus on skeleton topology [RT07]. More relevant is the work of Simari et al. [SKS06]. Their folding mesh hierarchy is produced in a top-down fashion by finding recursively the dominant (by area) reflectional symmetry in a connected sub-shape. Our symmetry hierarchy deals with more general symmetries, even between disconnected parts, and it is built bottom-up, allowing for easy handling of models that are largely asymmetric. In addition, our construction follows rules which take into account perceptual cues and the compactness principle, going beyond purely geometric considerations and towards a more functional analysis of shapes.

Gal et al. [GSMCO09] introduce iWires, an intelligent editing paradigm for man-made shapes which incorporates symmetry-based constraints. By working primarily with feature curves, the wires, iWires maintain their individual characteristics and mutual relations while manipulating the object. The editing paradigm is more suited for detail editing. It encodes more global relations between the wires, but only as a discrete set without a higher-level organization. The use of symmetry hierarchies complements iWires by providing such organizational structures to enable part-level structural editing, as we demonstrate in this paper.

Our work is also related to reverse engineering. Automatically inferring geometric aspects of the design intent for solid shapes is well-studied in CAD/CAM and solid modeling. Earlier works such as [SV93] aim to convert boundary representations into CSG representations based on basic solid primitives. Later works try to recover feature trees and geometric constraints [LLM06]. However, the approach in these works concentrates on finding part-level features that would enable dimension changes and construct geometric constraint graphs. Our goal is to recover a more structural representation based primarily on symmetry analysis.

3. Pre-segmentation and initial graph construction

Symmetry-driven segmentation. All the models used in our work come from the Princeton Shape Benchmark, the iWires collection [GSMCO09], and the SHREC'09 databases. Some of them are already composed of isolated manifolds which can be taken as the primitive *parts*. Models defined by a single or a few large mesh pieces need to be segmented. Many existing segmentation algorithms are possible. We follow the well-known minima rule and employ a variant of the hierarchical clustering algorithm using normalized cuts guided by shape concavity [GF08]. However, our goal here is to decompose an object into parts which are the basic entities for analyzing symmetry relations. Hence, it is important that the segmentation reveals symmetric parts as much as possible. The concavity cues can sometimes be weak, especially for scanned models, so that the segmentation results can be imperfect, as shown in Figure 3.

To extract initial parts which better reveal symmetry, we develop a *symmetry-driven enhancement* of the original segmentations. We examine the whole model and detect dominant global or partial reflectional symmetries by extracting the most significant modes in the reflection plane space using the voting approach of [MGP06]. Then, for each part of the original segmentation, we check whether it conforms to the symmetries detected. Whenever we find a part that is symmetrically mapped to two or more different parts, we split the part according to its symmetric counterparts to improve the symmetry of the segmentation; see Figure 3 for some results. The symmetry-enhanced segmentations will assist in constructing more meaningful symmetry hierarchies.

Symmetry detection. Only extrinsic part-level symmetries are considered and they include 1) reflectional or rotational self-symmetries possessed by individual parts, and 2) rotational, translational and reflectional symmetries which relate pairs of parts. These symmetries define the attributes of the initial graph. For both inter- and intra-part symmetry detections, we first re-sample the parts uniformly to achieve independence against their mesh tessellation.

We adopt the voting approach of [MGP06] to detect all the self-symmetries possessed by a part. For symmetry detection between two parts, instead of voting, we utilize a

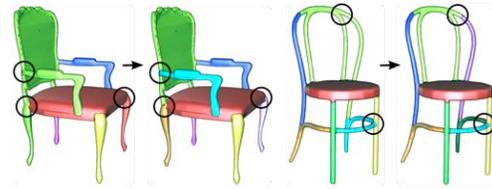


Figure 3: Symmetry enhancement (from left to right in each pair) in mesh segmentation for two synthetic scans. Enhancements and differences are highlighted in circles.

simple registration method which has worked quite effectively since the primitive parts we are dealing with are rather simple. First, dissimilar parts are excluded from consideration quickly by comparing their principal components. For those which have potential symmetry relations, we align one of the parts to the other by matching their centers and PCA axes. Then we use ICP to find a rigid registration between the parts and measure their Hausdorff distance. If the distance falls below a threshold, we regard the shapes to be the same and the extrinsic symmetry between the registered pair is extracted from the transformation matrix of the registration. Translational and rotational symmetries can be derived from the transformation matrix by simple matrix operations. To find a reflectional symmetry, we reflect the first part about the central plane perpendicular to the line connecting the centers of the two parts, and then re-iterate the above rigid registration and thresholding procedure.

Initial graph. The nodes of the initial graph are the model parts and the edges encode all the detected symmetries as well as connectivity between pairs of parts. There is a connectivity edge between two nodes if their corresponding parts are connected and there is a symmetry edge if the parts are symmetric to each other. The connectivity relations between parts in the input mesh are added into the graph. Other connectivity relations between parts are detected by computing the intersection between the bounding boxes of the parts. Each symmetry edge stores the type as well as the necessary parameters defining the associated symmetry. Each node stores the set of self-symmetries it possesses.

We say that two symmetries are *equivalent* if, a) they are both rotational symmetries and share the same rotation axis, but not necessarily the same order of rotations, or b) they are both translational symmetries and their translation vectors are co-linear, or c) they are both reflectional symmetries and share the same reflection plane. A *symmetry clique* in the graph is a subset of nodes in which every pair is connected by a symmetry edge describing an equivalent symmetry, called the *grouping symmetry* of the symmetry clique. The number of nodes in a clique is called its *order*. The order of a rotational or translational symmetry clique is at least two while for a reflectional symmetry, the order is always two.

4. Graph contraction and precedence rules

In a bottom-up fashion starting with nodes in the initial graph, we apply symmetry-guided graph contraction recursively to group or assemble sets of nodes, stopping when there is only one node remaining. This node is the root representing the full input model.

Graph contraction and symmetry inheritance. As we build up the hierarchy, each internal node, attributed by its type, is associated with a set of parts, which we call a *part ensemble*. By definition, each part is also a part ensemble. Each node in the symmetry hierarchy stores a (possibly null) set of extrinsic self-symmetries possessed by the part ensemble at the node. A grouping node is a *unary* (degree-one) node which results from contracting a symmetry clique. In the hierarchy, it points to one (arbitrary) instance in the clique. An assembly node is a *binary* node in the hierarchy pointing to two connected part ensembles; see Figure 2.

The way graph contraction is implemented is fairly standard. It operates on a contraction graph which starts as the initial graph. In each step, the set of nodes contracted is removed from the contraction graph and replaced by a grouping or an assembly node. This is followed by necessary updates on the set of connectivity and symmetry edges. The process then repeats until there is only one node remaining in the contraction graph. Of course, all leaf nodes and new nodes created are stored separately from the contraction graph to populate the resulting symmetry hierarchy.

A key operation after each contraction is to assign to the new node a set of self-symmetries, for the resulting part ensemble. Instead of recomputing these self-symmetries from scratch, we rely on *symmetry inheritance* from the contracted nodes. First, we add to the new node any equivalent self-symmetry possessed by all the contracted nodes. In the case of grouping by rotational or reflectional symmetry, any grouping symmetry of the contracted clique is added as a self-symmetry of the group. The only case where additional symmetry needs to be detected and added is with grouping by translational symmetry, where it is possible that a new reflectional symmetry, with its reflection plane perpendicular to the translation vector, would emerge.

Precedence rules. There are typically multiple symmetry cliques to be grouped and a node may be connected to multiple symmetry or connectivity edges. We now propose precedence rules to order contraction operations. These rules are tested against all the detected symmetry cliques.

Grouping-Assembly Mixing Rules (M1-M2) dictate ordering between grouping and assembly operations.

M1 (Grouping before assembly): Grouping by symmetry takes precedence over assembly operations, with an exception given by the next rule (M2).

Grouping of nodes by contracting a symmetry clique fol-

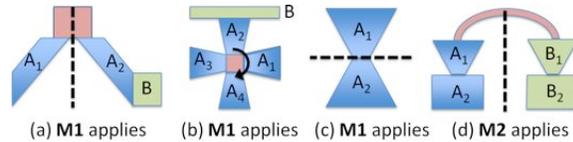


Figure 4: Grouping-Assembly Mixing Rules at work. (a)-(c): A_i 's are grouped by symmetries before assembly with others. (d) A_1 and A_2 are assembled before grouping by symmetry.

lows Gestalt law of symmetry for perceptual grouping, removing redundancy and making the representation compact. For a man-made shape, parts which are symmetric to each other tend to perform the same function. This rule overtakes Gestalt law of proximity [Köh29] for an assembly of connected part ensembles if the assembly invalidates any existing symmetry clique; see Figures 4(a)-(c).

M2 (Assembly before grouping): Assemble before grouping if and only if the assembled nodes belong to symmetry cliques which possess equivalent grouping symmetries.

This is the only instance where the law of proximity takes precedence over the law of symmetry, but only "temporarily". Such an assembly does not invalidate any symmetry clique — it merely *merges* two of them so that the law of symmetry is applied to a larger part ensemble. In a man-made object, it is likely that such an ensemble together perform a certain function. Examples of M2 at work can be found in Figure 4(d) as a 2D illustration and also in Figure 2 (see red assembly nodes marked by "bf. grouping"). Figure 6 highlights the necessity of rule M2 with an unnatural hierarchy as a result of omitting M2 (compare to Figure 2).

Symmetry Grouping Rules (G1-G3) order groupings.

G1 (Clique order): If there are still symmetry cliques of order greater than two in the contraction graph, then higher-order cliques are grouped before lower-order ones.

G2 (Reflectional symmetry): If there are only order-2 cliques in the graph, then group by reflectional symmetry before rotational symmetry and translational symmetries.

G1 follows the compactness principle, since contracting a higher-order clique removes more redundancy; see Figure 5(a). G2 is motivated by the belief that reflectional symmetries tend to trigger a stronger visual response than the other two symmetries [Weg99], e.g., see Figures 5(b-c). Figure 6 highlights the necessity of G2 with an unnatural hierarchy as a result of omitting G2 (compare to Figure 2).

G3 (Proximity in symmetry clique): If G1 and G2 cannot set a precedence, e.g., between rotational and translational symmetries of the same order, then grouping of part ensembles closer in proximity takes precedence.

If G1 and G2, which are related to symmetry order and

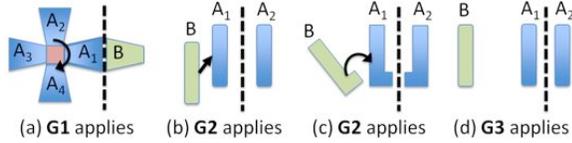


Figure 5: Symmetry Grouping Rules. The A_i 's are grouped before with the B 's in all cases. (a) Rotational symmetry has higher order. (b-c) Reflectional symmetry is stronger than the other two types of symmetries. (d) Ordering by proximity.

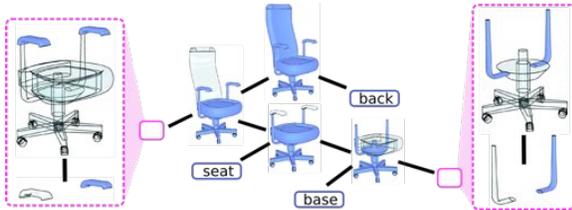


Figure 6: Unnatural hierarchy as a result of omitting certain precedence rules. The same node organization is obtained by omitting $M2$ or $G2$. The zoom-ins reveal parts of the armrest that should be grouped but are separated, in contrast to the more natural result in Figure 2 (first chair).

types, cannot set a precedence, the law of proximity takes over; see Figure 5(d). Any reasonable measure of geometric proximity among part ensembles, e.g., shortest distance between the convex hulls or other bounding volumes of the ensembles, can be applied here. We implemented the latter option with bounding volumes provided by oriented bounding boxes of the part ensembles computed via PCA.

Assembly rules ($A1$ - $A2$) order assembly operations.

A1 (Symmetry preservation): Symmetry-preserving assembly takes precedence over symmetry-breaking assembly.

In other words, we give precedence to assembly operations which preserve the most number of self-symmetries. We examine the number of equivalent self-symmetries shared by two nodes to be assembled and give precedence to an assembly which inherits more self-symmetries from the nodes. This rule is partly motivated by Gestalt law of symmetry which emphasizes the role of symmetry in perceptual grouping. For a man-made object, more shared symmetries likely imply more shared functionalities, as we argue that symmetric parts or part ensembles tend to perform the same function. Finally, the rule is also related to the compactness principle. Indeed, a shared symmetry allows to use a single symmetry to explain a larger set of parts as an aggregate.

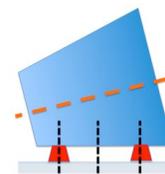
A2 (Connectivity strength): If $A1$ cannot set a precedence, then order assembly operations according to a geometric connectivity strength measure.

Connectivity strength is meant to capture how strongly two part ensembles are connected and the rule serves to enhance a predominantly symmetry-based hierarchical organization with a geometric criterion. Different choices for the strength measure are possible including perceptual criteria such as part salience [HS97]. However, we believe that due to the large variety of connectivity patterns between their parts, defining a general meaningful measure of connectivity strength for man-made objects is by no means an easy task. Such a measure needs to consider both perceptual issues as well as those relating to functionality of the parts, for which additional research is required.

We define a simple and admittedly less than perfect volumetric measure of connectivity strength. Given two part ensembles A and B with oriented bounding boxes (OBB) O_A and O_B computed via PCA, the connectivity strength $Y(A, B) = \frac{volume(O_A) + volume(O_B)}{volume(O_C)}$, where C is the OBB of the union of A and B . A higher Y value implies stronger coupling of the two part ensembles and their assembly would take precedence over those having lower connectivity strength values. In Figure 2, the connectivity strength values are shown in each assembly (red) node.

Algorithm 1 shows the pseudocode for our recursive contraction algorithm focusing on rule application while ignoring much of the details on graph transformation.

Remarks. Our construction is dominated by symmetry-driven rules. Although geometry rules, $G3$ and $A2$, are included, they are always applied with lower priority. As a symmetry-based processing paradigm, our approach distinguishes itself from the others [MGP06, SKS06, PSG*06, XZT*09] by being structural, focusing on factors such as the order of symmetry or the number of shared symmetries instead of geometric measures such as the size of symmetry support. A simple yet illustrative example which highlights the difference is given below. While in



terms of area support, the orange line would be identified as the major line of reflection [PSG*06], our method gives precedence to the small support polygons due to their self- and grouping reflectional symmetries. The two techniques would return rather different upright orientations, for example. It is difficult to assess which one is more meaningful; both have their merits. Indeed, we do not claim to have arrived at an optimal set of rules or that we have fully considered all the symmetry-related or perceptual issues. Our framework can be further enhanced or tuned to suit different purposes.

5. Results and applications

In this section, we show results of symmetry hierarchy construction. We show that hierarchical segmentations derived from symmetry hierarchies are more meaningful than those



Figure 7: Hierarchical segmentations of several chair models. Results marked by boxes produce semantic parts: back, seat, legs, and armrests (if any), e.g., compare to the symbolic hierarchies shown in Figure 2. Note the imperfect result for the last chair as part of the back is not separated from the legs; this is due to the initial segmentation unable to do so (see Figure 3).

Algorithm 1 Graph contraction using the precedence rules.

```

Input:  $G$  = contraction graph = initial graph
while  $|V(G)| > 1$  and  $\exists$  symmetry cliques in  $G$  do
  /* rule M1 applied as we find symmetry cliques first */
   $p$  = symmetry clique (SC) in  $G$  chosen by rules G1-G3
  ContractionByGrouping( $p, G$ )
  /* M2 applied to test for assembly before grouping */
  while  $\exists$  SC  $r$  connected to  $p$  with same symmetry do
    UndoContraction( $p, G$ )
     $p$  = SC after assembling corresponding parts in  $p$  &  $r$ 
  end while
  ContractionByGrouping( $p, G$ )
end while
while  $|V(G)| > 1$  do
   $Q = \{q, \dots\}$  = set of assembly pairs chosen by rule A1
  if  $|Q| > 1$  then
     $q$  = assembly pair chosen from  $Q$  by rule A2
  end if
  ContractionByAssembly( $q, G$ )
end while

```

based on local geometric considerations. We also develop an application of symmetry hierarchy for shape editing.

Symmetry hierarchy results. Figure 2 shows the symmetry hierarchies of a few distinct chair models. We can observe the handling of rotational, translational, and a large variety of combinations of reflectional symmetries by our algorithm. One of the chairs is globally asymmetric due to its armrest and this part is assembled last in the construction, as a result of our symmetry-driven approach. We can also observe that assembly nodes (red) are mostly closer to the root than grouping nodes, reflecting the use of rule M1, with the exception of assembly before grouping (rule M2). The use of other precedence rules is also reflected in the figure. The complete symmetry hierarchies of all the models shown in this paper and more are in the supplementary material.

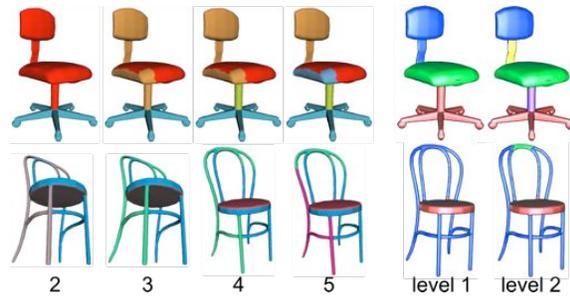


Figure 8: Hierarchical segmentation results from other methods. Left: [AFS06], where the numbers indicate segmentation counts. Right: [SSS*10], with two levels identified by thresholding the SDF values after Gaussian fitting.

Our symmetry hierarchy construction algorithm is quite efficient. The running times depend on the number of parts as well as the vertex count (for computing connectivity strengths). For the most complex model we have tested, the one in the bottom row of Figure 9 composed of 9,299 vertices, 17,686 triangles, and 71 parts, the construction time is about 15 seconds after preprocessing. All other models shown in the paper took no more than 2 seconds to construct the hierarchies. The preprocessing steps rely mostly on existing techniques with the most costly operation being the voting algorithm for detecting self-symmetries; it generally takes a few minutes to process a typical shape.

Hierarchical segmentation. A symmetry hierarchy naturally implies a hierarchical grouping of the parts of an object. In fact, starting from the root of the symmetry hierarchy and reversing the contraction process of the graph provides a unique series of embedded segmentations, each one more detailed than the previous. When traversing down the symmetry hierarchy, a good rule for termination is to not traverse beyond a symmetry grouping node. The rationale is that we treat a part ensemble grouped by symmetry as an in-

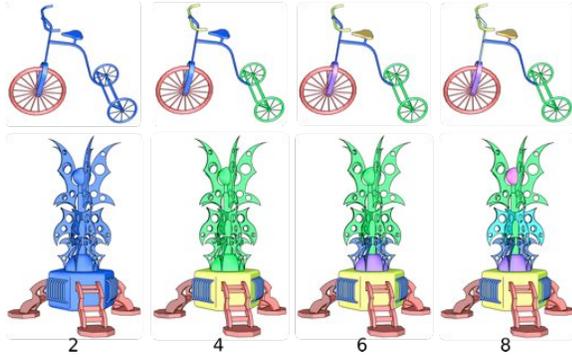


Figure 9: Hierarchical segmentations of two complex models; the numbers indicate segmentation counts.

tegral component of the object. Figure 7 shows some results of the hierarchical segmentations obtained this way for the chairs from Figures 2 and 3. Some of these results can be compared (see Figure 8) to those obtained by other hierarchical segmentation algorithms, e.g., [AFS06] which relies on primitive fitting and [SSS*10] which uses shape diameter function (SDF) values to cluster mesh faces. Note that the latter approach does not give results for all segmentation counts; the actual counts are determined by Gaussian fitting to the set of SDF values. For both compared methods, the results were obtained using software provided by the respective authors. Finally, we show results produced by our method on two complex models in Figure 9.

An important observation is that despite the rather large degrees of geometric and topological variations between the models within the same class, the implied hierarchical segmentations tend to reveal the major functional components consistently for these models. For the chairs, the components are seat, back, base, and armrests. Comparing the symbolic hierarchies shown in Figure 2 and the circled segmentations in Figure 7, we see a match in each case. This strongly hints that the symmetry hierarchies may lead to a consistent segmentation of a set of shapes from the same class.

The work of Golovinskiy and Funkhouser [GF09] deals with the consistent segmentation problem. They may have provided the first nice set of results out of an automatic consistent mesh segmentation algorithm on a set of shapes, including man-made objects such as chairs. However, their analysis relies on a global geometric alignment of the set. Distances between mesh elements from different models after the alignment are then used to drive a clustering scheme. Obviously, geometric alignment can be susceptible to at least part stretching and the use of closest distances for clustering can be misleading in the presence of large geometric and topological variations between the shapes. Our symmetry hierarchy is structural and it is invariant to any symmetry-

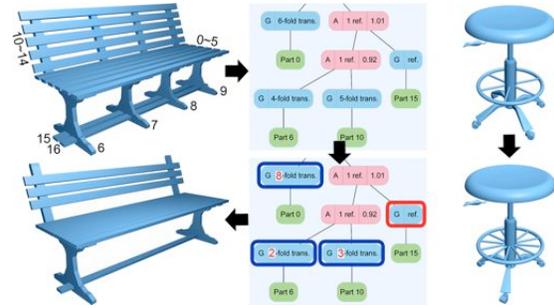


Figure 10: Shape editing by modifying symmetry parameters. Left: changing the orders of translational symmetries (see dark blue nodes). Right: result after changing the orders of rotational symmetries. To save space, we only show the hierarchies using GraphViz (A: assembly; G: grouping).

preserving part stretching; such variations are usually expected of man-made shapes. However, it still remains a challenge to establish semantic correspondence between symmetry hierarchies across a whole set. Recently, Kalogerakis et al. [KHS10] incorporates prior knowledge through semantically labelled training set to solve the consistent segmentation problem. The consistency is implied by shared labels but the segmentations are not hierarchical.

Structural shape editing. Most works on shape editing and manipulation in graphics have so far focused on free-form deformation of natural objects. The deformation and editing of man-made objects present new challenges as the editing method must be acutely aware of the shape semantics. For example, in CAD/CAM design systems, geometric objects such as mechanical parts are augmented with a set of parameters and constraints to define their semantics. Editing is possible by changing the parameters that affect the part geometry. The recent iWires tool of Gal et al. [GSMCO09] uses constraint satisfaction to propagate an editing change to the whole shape. These examples define two basic modes of manipulation that either carries a change from some part structure to the geometry or vice versa.

As a structural representation for man-made shapes, the symmetry hierarchy is naturally suited for both modes of editing. Using a dual representation of objects: one *geometric* and one *structural* using the symmetry hierarchy, we can manipulate either one and carry the effect to the other. This allows the user to manipulate the meaningful shape parts and introduce structural changes to the object at targeted levels or scales. Since these changes are symmetry-aware, they are particularly meaningful for man-made objects.

Using the *structural view* the user can modify the parameters of an existing symmetry by selecting its associated symmetry grouping node from the hierarchy. Addition, deletion, and spatial rearrangement of the part ensemble, i.e.,

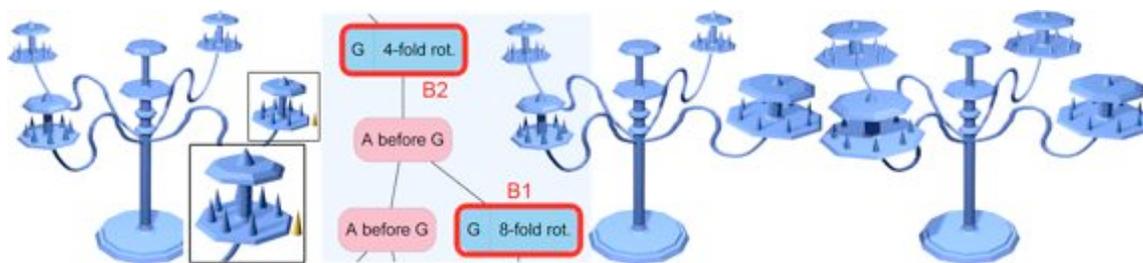


Figure 11: Shape editing with symmetry breaking and restoration at different levels in a hierarchy. Moving a small cone (yellow) breaks rotational symmetries at nodes B1 and B2. Middle: after restoring symmetry at B1. Right: restoration at B2.

the child node of the modified node, are automatically performed based on the new symmetry defined. In Figure 10, we see examples of such editing where the order of translational or rotational symmetries can be changed.

Using the *geometric view* the user can transform (translate, rotate, or and scale), add part(s) to, or delete part(s) from a part ensemble by direct manipulation. However, such actions may *break* existing symmetries. Using the symmetry hierarchy this can be automatically detected at the altered node to trigger proper action. By default, broken symmetries are restored at the *lowest* possible level of the hierarchy to keep changes minimal while *maintaining the contact points* between connected parts. The same editing operation is applied to all parts of the symmetry group and then connecting parts are transformed to keep their connection. Note that this could still leave certain broken symmetries un-restored.

Using the structural view the user can optionally select other nodes higher up in the hierarchy, i.e., at a larger scale, for symmetry restoration. This provides the ability to apply editing in the correct context in different levels of abstraction. In Figure 11, we show the transformation of a small cone feature in a candle stand model having a nested hierarchy of rotational symmetries. By default, the lower-level 8-fold rotational symmetry (node B1) is restored. The broken symmetry at node B2 remains. The user can optionally choose to restore symmetry at B2, which leads to a larger-scale editing effect. To maintain part connectivity, deformation of parts (the curved iron wires in the candle stand model) is necessary with the contacting points associated with the transformed part ensemble serving as deformation handles. The interactive editing sessions for these examples and more are shown in the accompanying video.

6. Conclusion, discussion, and future work

We present symmetry hierarchy, a novel structural shape representation in the form of a symmetry-induced, hierarchical organization of the constituent parts of a shape. We regard our work as a preliminary step towards functional and thus high-level analysis of man-made shapes. Symmetry does not specify what the functionalities are, it merely groups, in a

hierarchical manner, the shape parts which tend to share the same function. It is conceivable that without incorporating any prior knowledge, the kind of geometric and topological variations between functionally equivalent parts in man-made objects may be too much to allow a reliable inference. In this sense, the symmetry hierarchy may serve its best purpose as an effective *intermediate* representation for a truly knowledge-based functional shape analysis.

Limitations. Although the meaningfulness of symmetry hierarchies is evident on many models, the associated analyses may not be suitable for all shapes, particularly those lacking sufficiently rich symmetry structures. In this paper, we only deal with exact extrinsic symmetries of three types and the rather primitive symmetry detection schemes can be less than satisfactory. More robust algorithms which can handle more general forms of symmetries [MGP06, PMW*08] would allow us to analyze a larger set of models.

The meaningfulness of the symmetry hierarchy also depends on the initial segmentation. Our symmetry-based enhancement can turn imperfect segmentations into more meaningful ones in many cases. However, it is also limited by its dependence on symmetry analysis. For example, it is unable to identify the four legs of a chair if part of its back and a leg form an integral part and breaking them apart does not create more symmetries. The last chair of Figure 7 is an example; it also shows an imperfect hierarchy resulting from an imperfect segmentation. More sophisticated schemes which can detect self-similar parts [MGP06] may provide a remedy. Alternatively, supervised learning may help obtain the most meaningful segmentation [KHS10]. Finally, our graph contraction scheme is greedy and its local decisions may not always lead to a globally most meaningful symmetry hierarchy, though a precise definition of what the most meaningful hierarchy is remains elusive and the answer is related to human cognition.

Future work. While the list of limitations above already hint several tasks as future work, there are a few other directions worth pursuing. Our current set of precedence rules are the result of a thought process which take into consideration

various criteria; they are not the result of following a rigorous definition or analysis of an objective function. The latter is likely difficult as it involves proper modeling of a cognitive process but should be attempted, at least in the hope to further enhance the rule set. Any claim on the meaningfulness of the symmetry hierarchies from a perceptual or functional point of view should best be evaluated by a user study. Finally, we believe that the full potential of symmetry hierarchies for analysis of man-made shapes has not been fully explored. For example, a problem of great interest and utility is *consistent symmetry hierarchy* across a set of objects in the same class, e.g., the chairs. The result would be a consistent hierarchical segmentation across the whole set.

Acknowledgments: We thank all the reviewers for their helpful comments. Mesh models shown in the paper were from the Princeton Shape Benchmark, SHREC'09, and Ran Gal. This work was supported in part by an NSERC Grant (No. 611370), the Israel Ministry of Science and Education, the Israel Science Foundation, the National Natural Science Foundation of China (No. 61070071), the Research Fund for the Doctoral Program of Higher Education (No. 20104307110003), and the China Scholarship Council.

References

- [AFS06] ATTENE M., FALCIDIENO B., SPAGNUOLO M.: Hierarchical mesh segmentation based on fitting primitives. *Visual Computer* 22, 3 (2006), 181–193.
- [BBW*09] BOKELOH M., BERNER A., WAND M., SEIDEL H.-P., SCHILLING A.: Symmetry detection using line features. *Computer Graphics Forum (Eurographics)* 28, 2 (2009), 697–706.
- [BSM*10] BENES B., STAVA O., MECH R., ALIGA D., KRISTOF P.: Inverse procedural modeling by automatic generation of L-systems. *Computer Graphics Forum (Eurographics)* 29, 2 (2010), 665–674.
- [BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *ACM Trans. on Graph* 29, 4 (2010).
- [GF08] GOLOVINSKIY A., FUNKHOUSER T.: Randomized cuts for 3D mesh analysis. *ACM Trans. on Graph* 27, 5 (2008), 1–12.
- [GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers & Graphics (Proc. of SMI)* 33, 3 (2009), 262–269.
- [GPF07] GOLOVINSKIY A., PODOLAK J., FUNKHOUSER T.: Symmetry-aware mesh processing. *Princeton University TR-782-07* (2007).
- [GSMCO09] GAL R., SORKINE O., MITRA N., COHEN-OR D.: iWires: An analyze-and-edit approach to shape manipulation. *ACM Trans. on Graph* 28, 3 (2009).
- [HS97] HOFFMAN D. D., SINGH M.: Saliency of visual parts. *Cognition* 63, 1 (1997), 29–78.
- [KFR04] KAZHDAN M., FUNKHOUSER T., RUSINKIEWICZ S.: Symmetry descriptors and 3D shape matching. *Symp. on Geom. Proc.* (2004), 115–123.
- [KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D mesh segmentation and labeling. *ACM Trans. on Graph* 29, 3 (2010), 102:1–12.
- [KJS07] KRAEVOY V., JULIUS D., SHEFFER A.: Shuffler: modeling with interchangeable parts. *Visual Computer* (2007).
- [Köh29] KÖHLER W.: *Gestalt Psychology*. Liveright, 1929.
- [KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graph* 22, 3 (2003), 954–961.
- [LCDF10] LIPMAN Y., CHEN X., DAUBECHIES I., FUNKHOUSER T.: Symmetry factored embedding and distance. *ACM Trans. on Graph* 29, 4 (2010).
- [Ley01] LEYTON M.: *A Generative Theory of Shape*. Lecture Notes in Computer Science, Vol. 2145. Springer, 2001.
- [Liu90] LIU Y.: *Symmetry Groups in Robotic Assembly Planning*. PhD thesis, University of Massachusetts, September 1990.
- [LLM06] LI M., LANGBEIN F. C., MARTIN R. R.: Constructing regularity feature trees for solid models. *Lecture Notes in Computer Science* 4077 (2006), 267–286.
- [LZ07] LIU R., ZHANG H.: Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum (Eurographics)* 26, 3 (2007), 385–394.
- [Mar07] MARTINET A.: *Structuring 3D Geometry based on Symmetry and Instancing Information*. PhD thesis, INP Grenoble, March 2007.
- [MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3D geometry. *ACM Trans. on Graph* 25, 3 (2006), 560–568.
- [MY*10] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating how mechanical assemblies work. *ACM Trans. on Graph* 29 (2010), 58:1–58:12.
- [OSG08] OVSJANIKOV M., SUN J., GUIBAS L.: Global intrinsic symmetries of shapes. *Computer Graphics Forum (SGP)* 27, 5 (2008), 1341–1348.
- [PMW*08] PAULY M., MITRA N. J., WALLNER J., POTTMANN H., GUIBAS L.: Discovering structural regularity in 3D geometry. *ACM Trans. on Graph* 27, 3 (2008), 43:1–11.
- [PSG*06] PODOLAK J., SHILANE P., GOLOVINSKIY A., RUSINKIEWICZ S., FUNKHOUSER T.: A planar-reflective symmetry transform for 3D shapes. *ACM Trans. on Graph* 25, 3 (2006), 549–559.
- [RBBK07] RAVIV D., BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Symmetries of non-rigid shapes. *Proc. Int. Conf. on Comp. Vis.* (2007).
- [RT07] RENIERS D., TELEA A.: Skeleton-based hierarchical shape segmentation. In *Proc. IEEE Int. Conf. on Shape Modeling and Applications* (2007), pp. 179–188.
- [SKS06] SIMARI P., KALOGERAKIS E., SINGH K.: Folding meshes: hierarchical mesh segmentation based on planar symmetry. *Symp. on Geom. Proc.* (2006), 111–119.
- [SSS*10] SHAPIRA L., SHALOM S., SHAMIR A., COHEN-OR D., ZHANG H.: Contextual part analogies in 3D objects. *Int. J. Comp. Vis.* 89, 1-2 (2010), 309–326.
- [SV93] SHAPIRO V., VOSSLER D. L.: Separation for boundary to CSG conversion. *ACM Trans. on Graph* 12, 1 (1993), 35–55.
- [Weg99] WEGEMANS J.: Parallel visual processes in symmetry perception: normality and pathology. *Documenta Ophthalmologica* 95 (1999), 359–370.
- [XZT*09] XU K., ZHANG H., TAGLIASACCHI A., LIU L., LI G., MENG M., XIONG Y.: Partial intrinsic reflectional symmetry of 3D shapes. *ACM Trans. on Graph* 28, 5 (2009).