Co-Hierarchical Analysis of Shape Structures

Oliver van Kaick* Kai Xu† Hao Zhang* Yanzhen Wang† Shuyang Sun* Ariel Shamir‡ Daniel Cohen-Or§
*Simon Fraser Univ. †HPCL, Nat. Univ. of Defense Tech. ‡Interdisciplinary Center §Tel Aviv Univ.

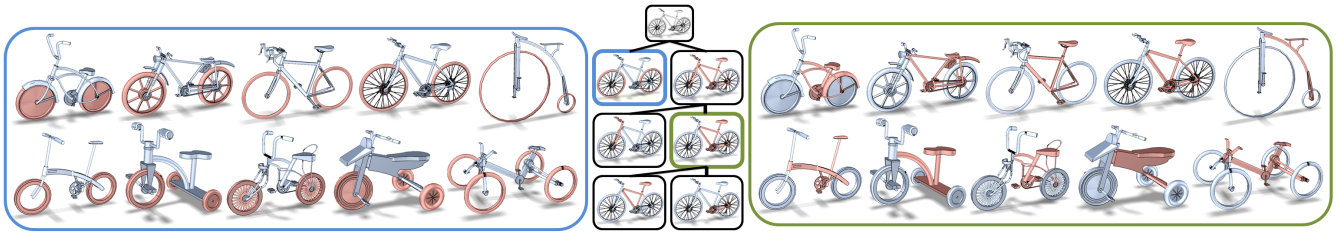

Figure 1: *Structural co-hierarchical analysis of a set of velocipedes (bicycles, tricycles and four-cycles). The resulting co-hierarchy (center) is illustrated by a single sample shape from the set, where each node represents a part assembly. Two of the nodes (highlighted in blue and green) are expanded to show the insight gained by the analysis which relates parts with rather different geometries but similar functions.*

Abstract

We introduce an unsupervised *co-hierarchical* analysis of a *set* of shapes, aimed at discovering their hierarchical part structures and revealing relations between geometrically dissimilar yet functionally equivalent shape parts across the set. The core problem is that of *representative co-selection*. For each shape in the set, one representative hierarchy (tree) is selected from among many possible interpretations of the hierarchical structure of the shape. Collectively, the selected tree representatives maximize the *within-cluster* structural similarity among them. We develop an iterative algorithm for representative co-selection. At each step, a novel *cluster-and-select* scheme is applied to a set of candidate trees for all the shapes. The tree-to-tree distance for clustering caters to structural shape analysis by focusing on spatial arrangement of shape parts, rather than their geometric details. The final set of representative trees are unified to form a structural co-hierarchy. We demonstrate co-hierarchical analysis on families of man-made shapes exhibiting high degrees of geometric and finer-scale structural variabilities.

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

Keywords: Structural shape analysis, co-hierarchical analysis, part correspondence, representative co-selection

Links: DL PDF

1 Introduction

One of the most fundamental shape analysis problems is to infer the part structure of a shape. Shape understanding, especially one at the structural or functional level, goes beyond decomposition of a shape into its constituent parts [Shamir 2008]. A higher-level organization of the parts, in particular, a *hierarchy*, is often more structure-revealing [Shapira et al. 2010; Wang et al. 2011; Jain et al. 2012]. The coarse-to-fine organization provided by a hierarchy expresses the functional structure or semantics of a shape more informatively than a mere enumeration of the shape's parts, while placing less emphasis on geometric details.

It is generally believed that human perception of shapes and structures is hierarchical [Palmer 1977; Hoffman and Richards 1984]. Moreover, a top-down part organization of an object tends to better reflect its one or many functions [Carlson-Radvansky et al. 1999]. The expressive power of hierarchical models becomes more critical when a related but diverse *set* of shapes is analyzed together. These shapes loosely belong to the same family, implying a consistence in the composition of their major functional components. However, the shapes may exhibit a high degree of geometric variability as well as variation in their finer-scale structures, e.g., consider sets of chairs or velocipedes (Figure 1). A unified explanation of the part structures within such a set is necessarily coarse-to-fine, which again suggests that a hierarchical representation is vital.

In this paper, we study structural hierarchy extraction from 3D shapes. We argue that shape understanding cannot be reliably acquired from a single example. Our approach is based on the key observation that objects designed to serve similar functions are often structurally similar; this assumption tends to hold at least at the coarse scale and with respect to the major functional components of the objects, although functionality often involves semantic aspects that cannot be derived from the geometry alone. The structural semantics of such a set of objects can be learned via a *co-analysis* of the set which exploits the underlying structural similarity.

The analysis algorithm we develop performs *unsupervised* analysis on a given set of shapes to learn the similarity as well as variability among the part structures of the shapes in the set. The result is a new structural representation, the *structural co-hierarchy*, which is a *binary* tree structure providing a unified representation of the learned shape structures. In particular, it provides the correspondences between geometrically dissimilar yet functionally equivalent shape parts across the set; see Figure 1. Structural co-hierarchies enable applications such as hierarchical segmentation, attribute transfer, and shape-aware editing [Wang et al. 2011], and any application of co-analysis is also applicable to co-hierarchies.

ACM Reference Format
van Kaick, O., Xu, K., Zhang, H., Wang, Y., Sun, S., Shamir, A., Cohen-Or, D. 2013.
Co-Hierarchical Analysis of Shape Structures. ACM Trans. Graph. 32, 4, Article 69 (July 2013), 10 pages.
DOI = 10.1145/2461912.2461924 http://doi.acm.org/10.1145/2461912.2461924.

DOI: http://doi.acm.org/10.1145/2461912.2461924

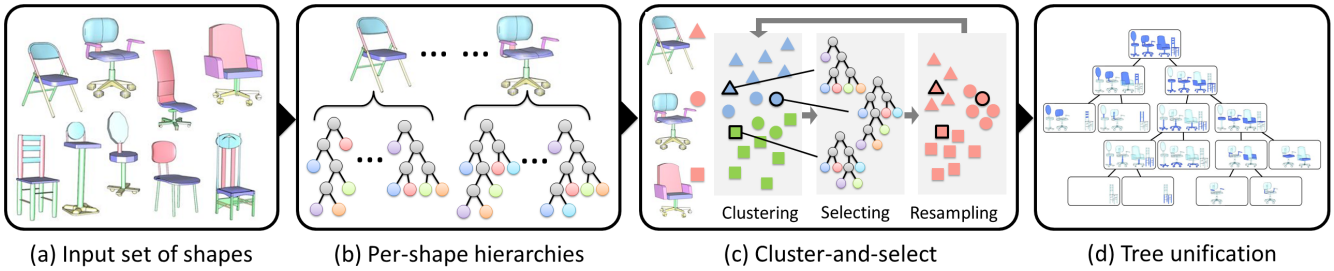

Figure 2: *Major steps of co-hierarchical analysis. First, initial per-shape candidate trees are obtained by sampling. Then, iteratively, the shapes are clustered based on distances between candidate trees, one representative tree per shape is selected, and new sets of candidate trees are obtained by resampling. At the end, the final tree representatives are unified to form the structural co-hierarchy.*

Representative co-selection. The main challenge in our analysis is that there may be many possible interpretations of the hierarchical structure of any shape in the set. We realize that there is not a unique structural hierarchy, a binary tree, that is the best representation. For example, it is not clear whether the seat of a chair must be grouped with the back or the legs.

Co-analysis provides a means to resolve the ambiguity in choosing an appropriate structural representation for each shape. In this context, we define our criterion as selecting one *representative* tree or hierarchy per shape to maximize the structural similarity among the chosen trees. To account for structural variations, the criterion should not be imposed on the whole set. This naturally leads to a clustering problem and the goal is shifted to maximizing *within-cluster* similarity with each cluster reflecting a *mode of structural variation* in the set of shapes. However, our unique problem setting makes the clustering problem unconventional:

- The entities to be clustered are not pre-determined and their selection is part of the computational process.
- Classical clustering often maximizes *between-cluster dissimilarities*. Co-analysis of a whole set on the other hand, is also concerned with exposing similarities between clusters, at least at the coarser levels of the hierarchies.

Hence, our core analysis involves a new problem on *representative co-selection* with clustering analysis offering a means to solve it.

Algorithm overview. We solve the representative co-selection problem by performing an *iterative cluster-and-select* optimization. Starting with the input set of 3D shapes, we first obtain an initial set of candidate structural trees per shape. Then iteratively,

1. The shapes are clustered, based on a structural tree-to-tree distance, to maximize within-cluster structural similarity.
2. One representative is selected per shape while accounting for both within-cluster and between-cluster structural similarity.
3. New sets of candidate trees are resampled around the selected tree representatives.

The final selected tree representatives are unified based on node-to-node correspondence to form a compound tree structure, which constitutes the structural co-hierarchy; see Figure 3. Figure 2 outlines the major steps of the algorithm.

Contributions. The main contributions of our work include:

- A *structure-driven* shape analysis enabling *co-analysis* of a set of shapes exhibiting significant geometric dissimilarity.
- A *hierarchical* analysis which allows the co-analysis to deal with structural variability in the set.
- A novel *unsupervised cluster-and-select* scheme which learns both the structural similarity and variability in the set.
- A new representation, the *structural co-hierarchy*, which unifies the learned shape structures.

By analyzing a set of shapes collectively, we gain knowledge about the shapes that is not deductible by observing the individuals in isolation. In this context, we believe that our works offers the first *co-analysis* that results in a *hierarchical* structural representation of a set of shapes. We show results from co-hierarchical analysis on families of man-made shapes exhibiting high degrees of geometric and finer-scale structural variability, and the structural understanding that the analysis reveals.

2 Related work

Much work on shape segmentation has been devoted to extracting the primitive parts of a shape based on low-level geometric cues [Shamir 2008]. A higher-level question is how the parts relate to each other. Attributed graph representations have been developed to encode pairwise part relations or contextual part descriptions [Shapira et al. 2010; Mitra et al. 2010; Chaudhuri et al. 2011; Fisher et al. 2011; Ovsjanikov et al. 2011]. However, representations focusing only on the low-level primitive parts and their relations are often too flat and sensitive to local geometry or structural variations to reveal the true underlying structure of a shape.

High-level shape analysis. Despite the challenges that still exist in fully solving low-level analysis tasks such as shape segmentation, there have been recent works that focus on high-level shape analysis by assuming that low-level segmentations are available [Martinet 2007; Mitra et al. 2010; Shapira et al. 2010; Xu et al. 2010; Chaud-

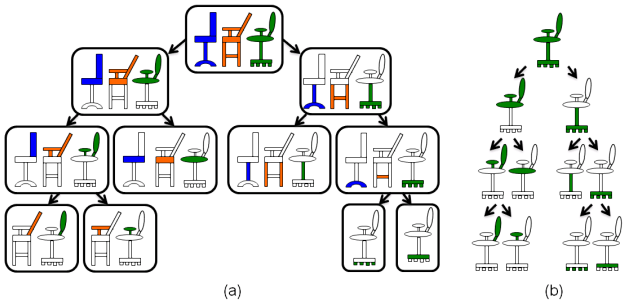

Figure 3: *Illustration of a structural co-hierarchy (a) and one of the selected per-shape candidate trees (b). The co-hierarchy unifies three such trees by combining corresponding nodes from them.*

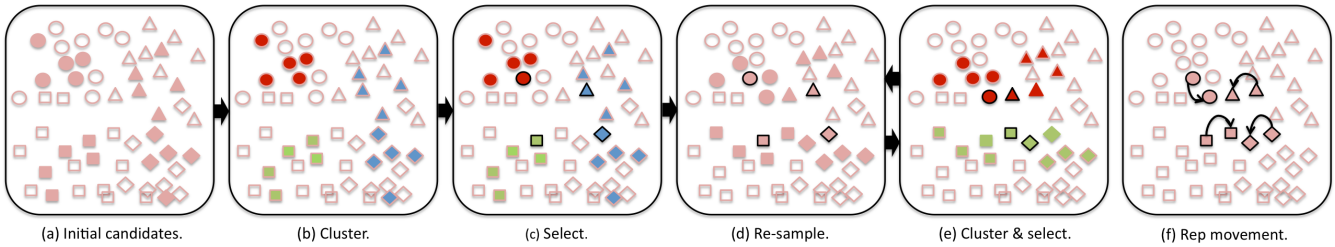

Figure 4: *An illustration of the iterative cluster-and-select process. The input contains four shapes, each associated with a set of trees (where circles, squares, diamonds, and triangles correspond to the four shapes). Euclidean distance roughly reflects tree-to-tree distance (small distance = higher structural similarity). (a) An initial set of candidate trees, shown as the filled elements, is selected. Hollow elements represent unsampled trees. (b) The shapes are clustered based on structural similarity among their associated candidate trees, resulting in three clusters (three colors). (c) One representative tree (black border) is selected for each shape to maximize their mutual similarity. (d) New candidates are sampled around the tree representatives. (e) New clusters (two in this iteration) are formed with the current set of candidates. (f) Note how the representatives evolve to maximize the commonality among the trees. The process then iterates between (e) and (d).*

huri et al. 2011; Fisher et al. 2011; Wang et al. 2011; Kalogerakis et al. 2012; Kim et al. 2012; Xu et al. 2012; Zheng et al. 2013] or by circumventing the segmentation problem [Ovsjanikov et al. 2011]. These developments are driven by a demand for higher-level analysis tools and facilitated by the increasing availability of large repositories of pre-segmented, even semantically tagged, shapes, e.g., the Trimble 3D Warehouse, the segmentation benchmark [Chen et al. 2009], and the COSEG [Wang et al. 2012] dataset for co-segmentation. Despite having individual tagged segmentations and common class labels, these shape repositories still lack an organization of the shape parts into high-level structures.

Hierarchical analysis. Early work on cognition [Hoffman and Richards 1984] stipulates that for the sake of object recognition, the visual system decomposes a given shape into a hierarchy of parts. Most works on hierarchical segmentation analyze individual shapes based on geometric properties of shape parts such as shape diameters [Shapira et al. 2010], convexity, compactness [Kraevoy et al. 2007], or other measures for primitive fitting [Attene et al. 2006]; other methods rely on surface metrics to perform clustering [Katz and Tal 2003; Liu and Zhang 2007; de Goes et al. 2008]. There have been recent works that construct structural hierarchies, primarily based on symmetry and connectivity information [Martinet 2007; Wang et al. 2011; Jain et al. 2012]. All of these works analyze individual or pairs of shapes, with hard-coded rules or objectives to guide the analysis to suit the intended applications; there is no utilization of knowledge latent in a larger shape set.

Co-analysis. Works on co-analysis extract necessary knowledge present in a set of shapes to facilitate shape analysis. Golovinskiy and Funkhouser [2009] obtain a consistent segmentation of a set by aligning all the shapes and then clustering their primitives. Xu et al. [2010] cluster a set of shapes based on a particular shape style in order to synthesize new shapes via style transfer. Huang et al. [2011] explore the knowledge latent in a set of shapes through a joint segmentation approach. Sidi et al. [2011] co-analyze a set of shapes by unsupervised clustering in a descriptor space and Wang et al. [2012] extends the approach to the semi-supervised setting. However, none of these methods compute a hierarchical structural organization. In contrast, our work constructs a unified hierarchy from a given set at the structural level. Also, the hierarchical representation provides a richer characterization of the set of shape structures beyond a coarse template or part-level correspondence.

Co-abstraction. Recently, Yumer and Kara [2012] introduce a co-abstraction method, where the models in a set are abstracted as much as possible while still preserving the unique geometric characteristics that distinguish one model from another. A hierarchy of abstractions is created for each model, and then the best hierarchical level is chosen for each shape so that the abstraction goal is satisfied for the entire set. Our co-hierarchy extraction method shares similarities with this approach, however, our hierarchies represent structural information, and we do not simply choose the best hierarchy that represents each shape, but rather go one step further by building a co-hierarchy that represents all the shapes in the set.

Clustering and distance. Many algorithms have been developed for cluster analysis [Everitt et al. 2011]. Note that we cluster hierarchical representations of shapes instead of hierarchically clustering the shapes. In our analysis, we need to take into account the different levels of the trees during the clustering, though we do not perform the clustering separately at each level of the trees. Instead, we compare the hierarchies by means of a *tree-to-tree* distance. For binary trees, which is the case of our hierarchies, polynomial metrics can be devised [Torsello et al. 2005]. Towards this end, we develop a novel structure-driven tree-to-tree distance and a clustering algorithm to extract a unified structural hierarchy.

Multi-instance clustering. The clustering step in our cluster-and-select is similar to *multi-instance* clustering (MIC). MIC finds clusters of *bags*, each represented by multiple *instances*. The most straightforward application setting is where a data entity can be represented by multiple features or examples. Our clustering step involves clustering shapes (the bags) according to their candidate trees (the instances). Our clustering solution is inspired by the algorithm of Zhang and Zhou [2009] in which the clustering is performed according to a *bag distance*: commonly a Hausdorff-type distance that involves measuring the distances between dynamically chosen instances in the two bags. Instead of using a k-medoids clustering, we rely on diffusion maps and hierarchical clustering in the embedding space to obtain more stable results.

3 Overview

The input to our algorithm is a set $\mathbf{M} = \{M_1, \ldots, M_n\}$ of meshes representing n shapes belonging to the same family, i.e., they serve similar functions and possess varying degrees of similarity in their part structures. We focus on man-made shapes due to the rich structural variability commonly found in them. We assume that each shape has been upright-oriented, with proper alignment, and partitioned into a set of primitive parts (Section 5). Our co-hierarchical analysis proceeds as shown in Figure 2.

Per-shape hierarchy. In our approach, a structural hierarchy h is a *binary* tree representing a hierarchical decomposition of the constituent parts of a shape. In particular, the root is the entire shape and each node of h represents a subset of parts, as illustrated in Figure 3(b). In this paper, the term *hierarchy* is used interchangeably with *tree* and always refers to a per-shape structural hierarchy. We equate a node in a structural tree with the parts it represents.

Initial candidate tree generation. The initial set of candidate trees for each shape is obtained by sampling from the population of all possible structural trees for the shape. To reduce the sample space, we identify groups of shape parts that share common symmetries. These symmetry groups, along with the remaining shape parts, form atomic entities to be further grouped and they are leaves of the trees. We obtain multiple hierarchies by farthest point sampling on the space of top-down binary decompositions of the set of atomic entities; see Section 5 for details.

Cluster-and-select. In each iteration, the shapes are clustered by diffusion maps and an agglomerative hierarchical clustering in the embedded space. The shape-to-shape distance employed is based on the minimum tree-to-tree distance measured between candidates which belong to the shapes. The tree-to-tree distance is defined hierarchically and based on a structural node-to-node distance which emphasizes more on spatial arrangement of shape parts than their detailed geometry. Next, we select a representative tree per shape to account for both within-cluster and between-cluster structural similarity. Finally, we resample a subset of the candidate hierarchies to create new trees that are more similar to the cluster representatives. We then repeat the cluster-and-select process, and stop the iterations when there are no significant changes to the clustering.

Figure 4 provides an important illustration of the iterative cluster-and-select process. It shows each step, as well as how the sets of candidate trees evolve to maximize within-cluster similarity.

Tree unification and co-hierarchy. The tree representatives selected are unified to form the structural co-hierarchy. We obtain a correspondence between the nodes of each selected tree and a representative tree. Next, these correspondences are used to perform a *union* of the tree nodes and connectivity to define the co-hierarchy, as shown in Figure 3. The co-hierarchy can be more formally defined as follows. Let $h_1, \ldots, h_n$ be the trees selected by the iterative cluster-and-select step, one per shape from $\mathbf{M}$. The corresponding structural co-hierarchy is generally a set of binary tree structures, each unifying a subset of the h_i's.

Let $\mathcal{H}^*$ be one of these trees corresponding to $M' \subset M$. Each node of $\mathcal{H}^*$ represents a set of part assemblies taken from the h_i's. These sets are defined inductively as follows. First, the root of $\mathcal{H}^*$ contains all the shapes from $\mathbf{M}'$ in their entirety. Now let $\mathbf{n}$ be a node in $\mathcal{H}^*$ whose associated set of part assemblies is A. Then $\mathbf{n_1}$ and $\mathbf{n_2}$ with associated sets of part assemblies A_1 and A_2, respectively, are the child nodes of $\mathbf{n}$ in $\mathcal{H}^*$ if all the members of A_1 and A_2 are defined as follows. For any $a \in A$ which is also a node in the hierarchy h_i, if a_1 and a_2 are the children of a in h_i, then either $a_1 \in A_1$ and $a_2 \in A_2$, or $a_1 \in A_2$ and $a_2 \in A_1$.

4 Node-to-node and tree-to-tree distance

The tree-to-tree distance measures the structural similarity between two hierarchies. It plays an essential role in quantifying whether two hierarchies reflect similar part structures in two shapes. Given two hierarchies, both of which are binary trees, we define the tree-to-tree distance *recursively*. The key component of the tree-to-tree distance is a *node-to-node* distance which measures the similarity between two binary decompositions of the part assemblies involved. Since the shapes that we analyze can have a high degree of variation in the geometry and part composition at the finer scale, we develop a distance measure that focuses more on the coarse structure of the shapes rather than on their geometric details.

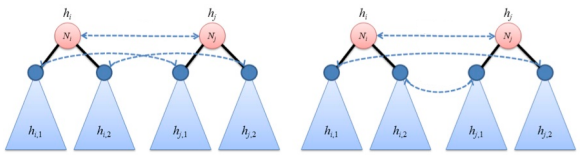

Figure 5: *Two cases for defining the tree-to-tree distance (1).*

Recursive tree-to-tree distance. Given two hierarchies h_i and h_j, the tree-to-tree distance D_t is defined as

$$\begin{aligned} D_t(h_i, h_j) = \min[&D_n(N_i, N_j) + \\ &\omega(D_t(h_{i1}, h_{j1}) + D_t(h_{i2}, h_{j2})), \\ &D'_n(N_i, N_j) + \\ &\omega(D_t(h_{i2}, h_{j1}) + D_t(h_{i1}, h_{j2}))], \end{aligned} \tag{1}$$

where N_i and N_j are the roots of h_i and h_j, respectively, and D_n or D'_n is the node-to-node distance. As illustrated in Figure 5, h_{i1} and h_{i2} are the left and right subtrees of N_i, respectively; h_{j1} and h_{j2} are defined similarly. In summary, we measure the similarity of two nodes and then apply the distance recursively to their children. We try the two possible assignments (left-to-left and right-to-right, as well as left-to-right and right-to-left) of children and select the one that gives the shortest distance, as shown in Figure 5. Thus, D'_n is the node distance for the switched case.

The weight $\omega \in (0, 1]$ is used to downplay the importance of the nodes deeper in a hierarchy. Since we want to emphasize the high-level structural similarity between hierarchies, while ignoring some detailed geometrical and structural differences, we empirically use $\omega = 0.8$ throughout all experiments in this paper. In order for our distance measure to behave as a metric, we normalize it by the recursive sum of weights $1 + \omega(2 + \omega(\ldots))$, similarly to dividing by the number of nodes as proposed in [Torsello et al. 2005].

Structural node-to-node distance. Let N_i and N_j be two nodes (part assemblies) belonging to two candidate trees. Let the two children of N_i be N_{i1} and N_{i2} and the two children of N_j be N_{j1} and N_{j2}. The node-to-node distance $D_n(N_i, N_j)$ measures the similarity between the binary decompositions implied by the two nodes and their children. We define this distance as

$$\begin{aligned} D_n(N_i, N_j) &= D_b(N_{i1}, N_{j1}) + D_b(N_{i2}, N_{j2}), \\ D'_n(N_i, N_j) &= D_b(N_{i1}, N_{j2}) + D_b(N_{i2}, N_{j1}), \end{aligned} \tag{2}$$

where D_b is the distance between the *axis-aligned* bounding boxes of the part assemblies represented by the two children. In other words, D_b measures the "effort" needed to transform one bounding box into the other. Thus, the node distance D_n is defined as the effort needed to transform the binary decomposition of one node to that of another node. If either N_i or N_j does not have children, we define the node distance as $|\mathbf{l}_i - \mathbf{l}_j|$, where $\mathbf{l}_i$ and $\mathbf{l}_j$ are vectors containing the lengths of the three dimensions of the bounding boxes of N_i and N_j, respectively.

Bounding box distance. We define the bounding box distance $D_b(B_1, B_2)$ between two axis-aligned boxes B_1 and B_2 as the amount of scaling and translation needed to transform one box into the other. Given two tetrahedra, we can define a unique affine

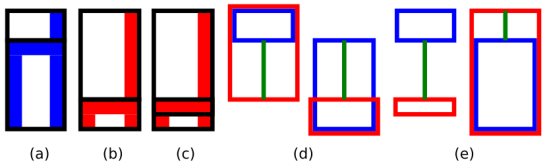

Figure 6: *The Hausdorff distance can be problematic when matching two pairs of boxes, exemplified here for two stretched chairs. Consider the binary decomposition of the blue chair in (a) and the two decompositions of the red chair in (b) and (c). (d) The sum of the Hausdorff distance (in green) is larger when matching the decompositions (a) and (b) that intuitively correspond to each other, than when when matching the less meaningful (a) and (c) in (e).*

transformation A between them, as long as their volume is non-degenerate. Thus, we take the affine transformation between the tetrahedra contained in each box and further decompose the transformation into scaling (s_x, s_y, s_z) and translation (t_x, t_y, t_z) components [Goldman 1992]. Rotation, shear or reflections are not present since we are dealing with axis-aligned boxes. Finally, we define $D_b = |t_x| + |t_y| + |t_z| + |1 - s_x| + |1 - s_y| + |1 - s_z|$, which captures any non-zero translation or any scaling deviating from unity. Before this computation, we define the bounding boxes of the children in relation to their parents, so that the coordinates of the children are normalized to $[0, 1]$ along each dimension.

It may seem inappropriate at first to incorporate scaling and the amount of translation into a structural measure. However, scaling is important in order to capture the structural notion that two parts with similar function on two shapes should have relatively the same size. This advantage is illustrated in Figure 6, where we show a problematic example that occurs when the Hausdorff distance between two boxes, which does not incorporate scaling, is used instead. Moreover, the sum of scaling and translation provides us with a smooth measure, which is a better choice for the clustering task. Finally, defining the boxes always in relation to their parent nodes guarantees that the relative positioning of parts is taken into consideration, which is more appropriate when dealing with shapes of varied geometry where the structure is more important.

Although our node distance is based on the simple idea of comparing the configurations of axis-aligned boxes, the distance is able to capture the general positioning of shape parts, which leads to the successful extraction of co-hierarchies as shown in Section 8. Nevertheless, as our work concentrates more on the co-hierarchical analysis than on the definition of the tree distance, we believe that more sophisticated definitions of the tree and node distances can definitely lead to improvements in the analysis.

5 Candidate hierarchy generation

For each shape $M_i \in \mathbf{M}$, we generate a set $\mathbf{H_i}$ of candidate hierarchies. $\mathbf{H_i}$ is generated by sampling multiple partitions of a graph that describes the relations between shape parts.

Preprocessing. We assume that the input shapes are already segmented into primitives. Many models (e.g., those collected from Trimble 3D Warehouse) are already composed of disconnected components that we use as segments. For shapes represented by a single connected mesh or a few large components with complex shapes, we use the symmetry-enhanced mesh segmentation method in [Wang et al. 2011] to further decompose the shapes into smaller segments. Nevertheless, any state-of-the-art segmentation method can be used to individually segment each input shape. Moreover, all the shapes in the input set are aligned to a canonical frame. Specifically, we obtain the upright orientation of the models, and then align the two remaining axes with a standard registration method.

For each shape, we analyze the shape parts and create a *part relation graph* similarly to [Wang et al. 2011], where each node represents a part and each edge encodes either a symmetry or connectivity relation between two nodes (obtained by measuring the proximity between the parts, that is, two parts are connected if they are close-by according to a threshold). The candidate hierarchies of the shape are obtained by sampling binary decompositions of this graph.

Candidate sampling. In principle, any sampling method that generates reasonable hierarchies of the shapes can be used for this step. However, given the diversity of shapes in our sets, it is difficult to define a generic criterion to determine whether a sampled hierarchy is reasonable or meaningful. Thus, we resort to a random sampling approach. Our assumption is that even if unreasonable samples are generated, we do not expect that they will find similar counterparts in the sample sets of the other shapes. On the other hand, there is a higher probability that the coherent hierarchies will be clustered together due to their similarity.

In our approach, given an input graph, we sample different binary decompositions in a top-down manner. This is accomplished in two stages. In the first stage, we start with the full set of nodes and enumerate all the binary partitions of connected components. Since the graphs are sparse and typically have less than 30 nodes, this step is efficient in practice. Next, we select a subset (typically 20) of these binary partitions according to farthest point sampling (FPS), to ensure that our sample set has good variability. The distances between partitions are calculated with our structural node-to-node distance (Section 4). We then apply the same sampling recursively on each partition. In the second stage, we take all possible combinations of these recursive partitions to build valid candidate trees, yielding a large number of trees. We again apply FPS to select a subset of trees with variations (also around 20 samples). The result is a set of candidate hierarchies for each shape, where we are able to directly control the size and variability of this set. An example of a set of candidates trees is shown in Figure 7.

We also experimented with a bottom-up sampling process which is more efficient. However, note that in a bottom-up process we have less control over the variability of the partitions at the higher levels of the hierarchy, which are the most important levels for our analysis. Thus, a top-down scheme is more appropriate.

6 Representative co-selection

We perform the representative co-selection on the set of candidate trees sampled from the input set of shapes, in order to construct the co-hierarchy (Section 7). The tree-to-tree distance for the co-selection, defined in Section 4, plays a critical role in our analysis.

6.1 Clustering of shapes

Given the multiple candidate trees for each shape, we cluster the shapes into groups and select a single hierarchy for each shape (the cluster-and-select). The clustering is achieved with a multi-instance algorithm described in this section, while the hierarchy selection is carried out with the algorithm described in Section 6.2. The goal in multi-instance clustering is to find clusters of bags, where each bag is represented by multiple instances [Zhang and Zhou 2009]. In our setting, the main analysis involves clustering shapes (bags) according to their multiple hierarchies (instances).

The key to multi-instance clustering is to define a bag-to-bag distance based on the multiple instances associated to each bag. This is

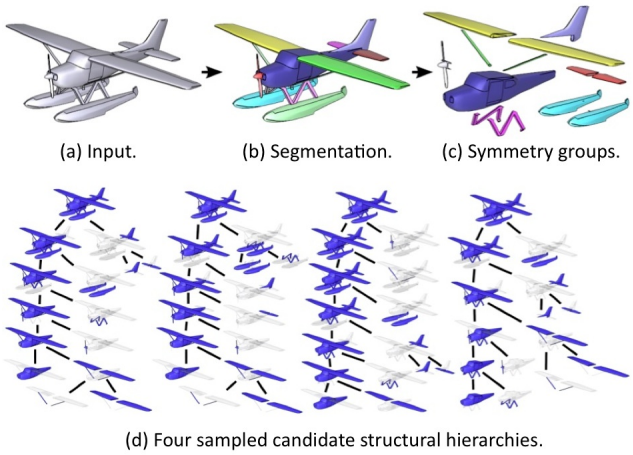

Figure 7: *Input, segmentation, symmetry groups, and a few sampled candidate hierarchies for an airplane model.*

commonly achieved with a Hausdorff-type set distance. The most common choice compares all the pairs of instances of two bags and selects the minimum distance. Similarly, in our setting, we define the distance between two shapes as

$$D_{\text{bag}}(M_i, M_j) = \min_{h_i \in \mathbf{H_i}, h_j \in \mathbf{H_j}} D_t(h_i, h_j), \tag{3}$$

where D_t is the tree-to-tree distance defined in Section 4.

Next, we form a bag-to-bag distance matrix by computing the distance between all pairs of shapes. Then, any clustering algorithm can be applied based on this matrix. The specific algorithm proposed by Zhang and Zhou [2009] is a variant of the k-medoids algorithm. In order to avoid the stochastic component of the k-medoids, we opt instead for a solution leading to more stable results. We first embed the shapes into a normalized space by computing the diffusion map of the distance matrix [Nadler et al. 2005]. We set the diffusion map parameter $t = 5$ and use 3 eigenvectors for the embedding, since that approximates well the diffusion distance [Nadler et al. 2005]. Next, we perform an agglomerative hierarchical clustering in the embedded space [Everitt et al. 2011].

To determine the proper number of clusters, we compute groupings with different numbers of clusters and use the change in the *silhouette index* of the groupings to guide our choice, as commonly done in cluster analysis [Everitt et al. 2011]. More specifically, if $sil(k)$ is the silhouette index of the grouping with k clusters, we choose the grouping k where $|sil(k) - sil(k-1)|$ is maximized.

6.2 Tree selection

After the multi-instance clustering, the shapes are grouped so that the similarity among their hierarchies is maximized. However, since the hierarchies for two different clusters may also share some similarity at the higher levels, we select a single tree per shape in an attempt to maximize the similarity *across* clusters as well.

We first extract for each cluster C_c a representative tree m_c with

$$m_c = \arg\min_{h_i \in C_c} \sum_{h_j \in C_c} D_t(h_i, h_j), \tag{4}$$

where we consider all the hierarchies h_i or h_j in cluster C_c. This can be seen as selecting the *medoid* of each cluster, although note that differently from the original algorithm of Zhang and Zhou [2009], a medoid here is a single instance and not a bag.

Next, for each shape, we select from its candidate set the tree that is closer to the medoid of its cluster, again according to D_t. This gives us a simple mechanism to select a single hierarchy per shape where the similarity *within* the cluster is maximized. We denote the set of hierarchies selected for cluster C_c as $\mathbb{H}_c$. See Figure 4(e) for an illustration of two such sets, each with a cardinality of two.

In order to also maximize the similarity *across* clusters, we perform this selection in a constructive manner. We start by considering the *most central* cluster C_1, defined as the cluster whose medoid is the most central to all the other medoids, similarly as in (4). We call the medoid of this cluster the dominant medoid m^+. Next, we proceed to another cluster C_2 and perform a new selection of one hierarchy per shape so that the similarity to m^+ is maximized. We call the newly selected hierarchies $\mathbb{H}^+$. We compare $\mathbb{H}^+$ to $\mathbb{H}_2$ (the selected trees that maximize the similarity to m_2), and keep $\mathbb{H}^+$ as the selected hierarchies for this cluster if their *cost* is not much higher than the cost of $\mathbb{H}_c$, according to a threshold. Otherwise, we keep $\mathbb{H}_c$ as the selected hierarchies. The cost of a selection is defined as the within-distance of the hierarchies:

$$\text{cost}(\mathbb{H}) = \sum_{h_i \in \mathbb{H}} \sum_{h_j \in \mathbb{H}, h_i \neq h_j} D_t(h_i, h_j). \tag{5}$$

Intuitively, we are keeping the new selection $\mathbb{H}^+$ that has a better between-cluster similarity only if it does not violate the within-cluster similarity. We apply the same procedure to all the remainder clusters, which gives us the set of hierarchies $h_1, \ldots, h_n$ selected for shapes $M_1, \ldots, M_n$. The hierarchies maximize the similarity within their clusters and also across the clusters when this does not compromise the former.

6.3 Candidate resampling

Although the previous steps ensure a selection that maximizes the similarity among the hierarchies, there is no guarantee that the initial hierarchies generated individually for each shape possess sufficient similarity. Thus, in this step we resample candidate hierarchies so that they are more similar to the selected medoids. For each shape, we sort all the candidate hierarchies according to their distance to the medoid of the shape's cluster. We eliminate the farthest hierarchies and replace them with resampled hierarchies. The resampled hierarchies are constructed with a top-down procedure similar to that described in Section 5. However, for each new sample, the binary decompositions are chosen so that the similarity of the hierarchy to the medoid of each cluster is maximized. The similarity is again measured with the node-to-node distance.

We perform the cluster-and-select followed by resampling (conceptually regarded as being part of the "select") for a number of iterations, and stop when there is no noticeable change in the clustering of the shapes and the selected hierarchies.

7 Tree unification

The goal of tree unification is to take the set of representative trees selected for each shape and build the co-hierarchy. To accomplish this step, we compute a correspondence from each selected hierarchy h_i to the dominant hierarchy m^+. Next, the union of all the nodes that are in correspondence defines a node in the co-hierarchy $\mathcal{H}^*$. The connectivity of $\mathcal{H}^*$ is also directly derived from the union of the connectivity of all the trees h_i.

To compute a correspondence between binary trees, we proceed recursively, similarly as in our definition of the tree-to-tree distance in Section 4. We match the two root nodes of the trees and then we only need to decide which of the two children are assigned to each

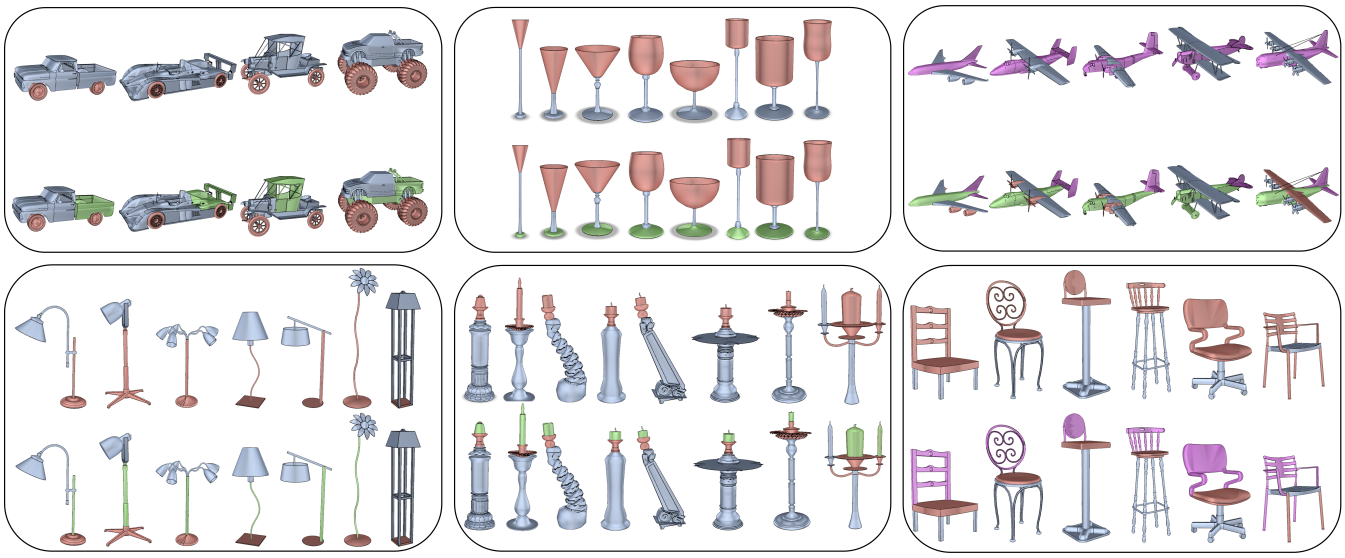

Figure 8: *Consistent hierarchical segmentation results corresponding to structural co-hierarchies obtained for various sets.*

other. This is done by finding the best assignment according to the tree-to-tree distance D_t. Next, we apply the same correspondence computation recursively on the matched children.

8 Results

We applied our co-hierarchical analysis to shapes belonging to eight object categories: chairs (40), airplanes (20), candles (20), lamps (20), velocipedes (20; including bicycles, tricycles, and four-wheeled vehicles), automobiles (10), and goblets (10). The shapes all come from the Trimble 3D Warehouse or Princeton Shape Benchmark. The eight categories are quite diverse in terms of the intended functionalities of objects. Except for the goblet set, which is a set for sanity check, each set of shapes exhibits a high degree of geometric variability and varying degrees of structural variability.

Co-hierarchical analysis. Figure 1 shows a co-hierarchy obtained for the velocipede set in its tree form, where we observe our algorithm's ability to find a correspondence between geometrically dissimilar yet functionally equivalent parts across the shape set. In Figure 8, we show similar results for other object categories. Note however that rather than showing the co-hierarchies as trees, which consume much space, we show hierarchical segmentation results.

Consistent hierarchical segmentation of a set of shapes is an immediate application of co-hierarchical analysis and helps to visualize the results of the analysis in a more compact form. In the figures, each row of a matrix of shapes shows one level of the co-hierarchy. The segmentation is obtained by collecting all the nodes of the co-hierarchy at the given level, and labeling with the same color all the shape parts corresponding to the collected nodes and their children. Thus, parts in correspondence across different shapes have the same color. The coloring is kept consistent across levels so that the binary split of each node can be followed, i.e., the first two nodes of the hierarchy are colored red and blue; the blue node is then split into blue and green at the next level, and so on. The segmentations for all the sets can be found in the supplementary material.

As shown in the figures, the structural similarities of the sets are revealed by the higher-level nodes in the co-hierarchy, despite of the significant structural variability, especially at the finer scales, among the individual shapes. We observe that the most common functional parts are represented by higher-level nodes, while less common parts, at finer scales of the objects, are represented by lower-level nodes. For example, the airplane wings and rudder are common structures and appear at the first two levels, while the engines are less common and appear at deeper levels. Similarly happens for velocipede wheels (common) and handlebars (less similar across the shapes, appearing at deeper levels). Moreover, since we emphasize more on spatial arrangement of the shape parts than their individual geometry, our method is almost oblivious to anisotropic part scaling, e.g., lamp shades and chair legs of different heights.

We observe from the results that our hierarchies are typically composed of two or three levels at most, with up to seven binary splits, although the shapes can be further decomposed. This is the case since on one hand, symmetry groups are not broken by our tree decompositions, implying shallow trees for most shapes. More importantly, the algorithm finds representatives that are similar to most of the other trees in a cluster. This similarity typically occurs at the coarser levels of the trees, and therefore the constructed co-hierarchies also reflect this level of similarity.

Figure 9 shows a result from co-analyzing a small chair set where more levels in the co-hierarchy were obtained. In this case, we allow symmetry groups to be broken and the chairs share the common, deeper hierarchical part structure. This small example demonstrates our algorithm's ability to perform fine-grained analysis as long as the data supports the result.

Figure 8 also shows that failure cases are present: the last model in the airplane, chair, candle, and lamp sets. These failure cases can be attributed in part to inadequate sampling (chair and lamp) and in part to our node-to-node distance (plane and candle) being imperfect in handling the many different functional structures that appear in the sets. Designing a general structural distance is a difficult problem, especially when the goal is to apply it to the understanding of the functionality of highly diverse shape collections across different categories. Moreover, the problem is intrinsically challenging as we perform such an analysis without any supervision.

Diverse sets. Although our tree distance is based on the simple concept of examining the configurations of bounding boxes, we

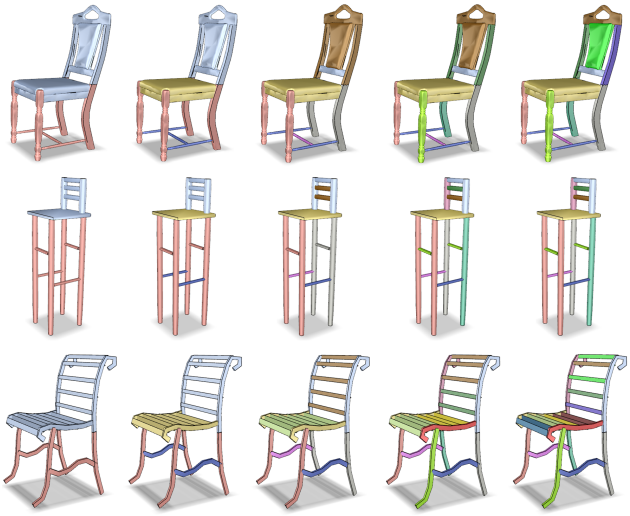

Figure 9: *Five-level hierarchical segmentations of a small set of chair models, resulting from co-hierarchical analysis.*

observe that the co-hierarchical analysis performs well on diverse datasets. Figure 10 shows an example of the embedding and clustering obtained by our method on the highly varied set of velocipedes. We observe that shapes that are structurally similar tend to become part of the same cluster. Moreover, in the supplementary material, we show that the cluster-and-select also has the potential of handling mixed sets composed of more than one category of shapes (e.g., chairs + lamps + airplanes + goblets), which is beyond the capability of existing co-segmentation algorithms.

Iterative improvement. Now we look closely at each step of our analysis algorithm and show that the structural similarity between the chosen representatives improves via the iterative approach. Before that however, we first show that maximizing structural similarity over the whole set (i.e., with only a single cluster) produces

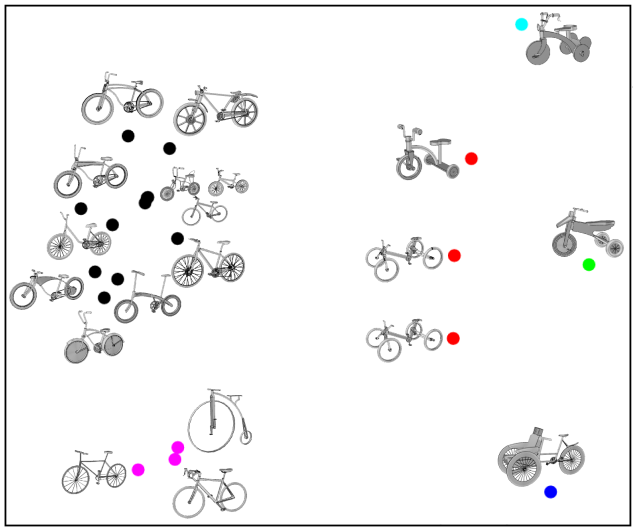

Figure 10: *Embedding and clustering of the velocipedes obtained with the cluster-and-select scheme. Shapes that are structurally similar tend to be in the same cluster.*

undesirable results. This alternative objective can be stated as

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{|\mathbf{H}_i|} \sum_{l=1}^{|\mathbf{H}_j|} x_{ik} x_{jl} D_t(h_{ik}, h_{jl}), \text{ s. t. } \sum_{k=1}^{|\mathbf{H}_i|} x_{ik} = 1, \forall i,$$

where x_{ik} indicates whether hierarchy k for shape i is selected. The constraint to which the optimization is subject ensures that we select a single hierarchy per shape, while the objective is to minimize the pairwise tree-to-tree distances among selected trees. This is a binary quadratic optimization program since for a selection we require $x_{ik} \in \{0, 1\}$. To optimize this objective, we solve the continuous case and then discretize the solution.

In Figure 11, we plot the mutual similarity score obtained for different representative selection results. It is evident that the iterative approach improves the score and having a single cluster is undesirable when there is variability in the set. The mutual similarity score is defined as the inverse of the sum of pairwise tree-to-tree distances between all the selected trees.

Co-analysis vs. per-shape hierarchy. We also confirm the advantage of analyzing a set of shapes by comparing the hierarchies selected by our co-analysis for a set of velocipedes to the hierarchies computed individually for each shape with the symmetry hierarchy construction of Wang et al. [2011]. As shown in Figure 12, there is clearly not much commonality among the individually-generated trees. Thus, building a co-hierarchy from these trees would result in a structure without meaningful correspondences across shape parts, as opposed to our co-hierarchical analysis which has been developed to address this task.

Statistics and timing. The number of parts per shape in our dataset ranges from 3 (e.g., goblets) to 20 (e.g., velocipedes). We always take 20 samples via FPS to form the initial set of candidates per shape; note that there can be repetition in the samples if the shapes have a simple structure, e.g., goblets. The iterative algorithm typically terminates after 3 to 5 iterations. The most time consuming part of the algorithm is the initial sampling. It can take from a few minutes for sets with a few parts (e.g., lamps) to a couple of hours on sets with richer part composition (e.g., velocipedes). The iterative representative co-selection takes on the order of minutes, since the main task involves the computation of all pairwise tree-to-tree distances to perform the clustering. The resampling, although also based on a top-down sampling, takes on the order of minutes since only a single tree needs to be constructed per sample.

9 Conclusion, discussion, and future work

We present an algorithm for co-hierarchical analysis of a set of shapes. The key contribution lies in combining a hierarchical shape representation and co-analysis to counter the significant geometric and finer-level structural variability present in the set of shapes. Hierarchical shape representations emphasize more on higher-level structures than finer-scale geometric or structural details. By exploiting knowledge gained from a set, our algorithm offers an effective means to resolve the ambiguity in choosing the right structural representation for each shape. We have shown that it allows to establish correspondences between geometrically dissimilar yet functionally equivalent part assemblies across the set.

High-level vs. low-level analysis. Like many recent works, we focus on the challenge of high-level shape analysis that is built on top of shape segmentations computed by state-of-the-art methods. While results from the latter are not yet perfect, works on low-level shape segmentations, e.g., [Shamir 2008; Golovinskiy and

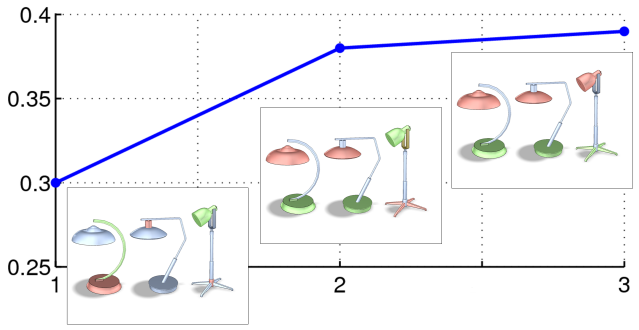

Figure 11: *Improvements on mutual structural similarity (vertical axis), as well as consistency of the tree representatives (insets), over iterative cluster-and-select. Each inset shows a subset of tree representatives selected, but for compactness, only the second level of the trees, in the form of a segmentation, is depicted. The first data point corresponds to the result by maximization over the whole set, i.e., without cluster analysis. The second data point is after applying the cluster-and-select, while the third data point is after resampling and then the second iteration of cluster-and-select.*

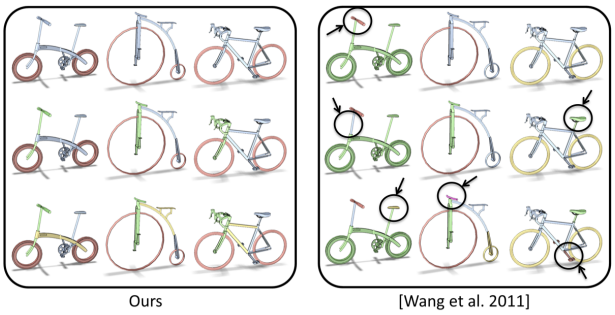

Figure 12: *Comparing our co-hierarchical analysis (left) to the symmetry hierarchy extraction of Wang et al. [2011] (right). The results from the individual hierarchy construction are obviously inconsistent across shapes in the set (small parts resulting from the construction are highlighted to draw attention).*

Funkhouser 2008; Kalogerakis et al. 2010; Huang et al. 2011], have matured over the years. The two lines of research complement each other. Instead of starting with fixed segmentations, we could work directly on over-segmentations of the input shapes and jointly optimize low-level shape segmentations and structural hierarchies. The joint optimization amounts to a search for hierarchical decompositions into more primitive shape parts (the over-segmentation). While the current algorithmic framework still remains intact, the search complexity would increase dramatically.

Binary decomposition. Our work is limited to producing recursive *binary* shape decompositions, which may be unnatural in some cases. Expanding the search to multi-way decompositions also increases the search complexity. By performing a co-analysis however, the restrictive nature of binary tree representations is compensated by having consistency of the hierarchies across the set — functionally meaningful part correspondence is maintained.

Ground truths for hierarchical analysis. We believe that human perception of hierarchical structural organizations is potentially less agreeable than that of (flat) shape segmentations. For example, there is probably little disagreement on how a human model should be segmented into its parts, but people's interpretations may very well vary on how the parts should be organized into a tree, especially a binary tree. Hence care needs to be taken when collecting ground truth data and executing a supervised approach; we leave such investigations to future work. Our unsupervised co-analysis does not claim to produce the best structural hierarchy when judged subjectively. Instead, it returns one good structural interpretation, while ensuring consistency across a set of shapes.

Tree sampling. The major performance bottleneck turns out to be the sampling step. With significant diversity in the functional properties of the different classes of shapes, it is difficult to find a generic set of criteria for defining a "good" structural decomposition. Balanced volume partitioning, compactness of the decomposed components, or objectives for graph cut optimization were all tested but being purely geometric, none performed satisfactorily over all object categories. We eventually resorted to FPS, which was shown to be effective, but at the expense of having to consider a very large sample space. Random sampling works since the set of shapes are expected to share structural similarity, which is reflected on the more meaningful hierarchies. Less meaningful hierarchies, even if produced by FPS, are unlikely to be selected by the algorithm either at the cluster-and-select or the resampling step.

Other limitations. There are a few other limitations to add to the list discussed so far. While the algorithmic framework outlined in our work is sufficiently general to work with different forms of per-shape hierarchical trees, currently, our structural trees are sampled only in terms of how the part assemblies are organized. To allow both symmetry groupings and part assemblies to be sampled would require an extended node-to-node distance. Our current node-to-node distance is defined using axis-aligned bounding boxes, which is somewhat restrictive. It is most appropriate for man-made shapes without part articulation. Designing more general node-to-node distances and criteria for tree sampling both point to functional shape analysis. The general question is how to design *functional* descriptors, rather than shape or geometry descriptors, which most effectively characterize the functionality of shape parts; this deserves an independent investigation.

Future work. We plan to explore various ways to extend the current optimization framework, including joint optimization of low-level segmentations and structural co-hierarchies, as well as interactive solutions such as active learning [Wang et al. 2012]. Also, we have only explored the use of basic cluster analysis tools. More advanced options for clustering or determining the number of clusters may yield improved results. Finally, we would like to establish a large dataset with hierarchically defined semantic labels. The dataset will allow more comprehensive evaluation of hierarchical shape analysis schemes and serve as training data for supervised learning of hierarchical shape models.

We regard our work as only an initial step in the direction of hierarchical structural analysis. The general goal is to provide a bridge between low-level geometric details of a shape and a deeper understanding of the shape's functionality and semantics. We believe that research on shape analysis in computer graphics has much to gain towards learning and utilizing hierarchical models. This would parallel the recent renewed interests in hierarchical models in computer vision [Sudderth et al. 2005; Wolf et al. 2006] and machine learning research [Rose and Karnowski 2010].

Acknowledgements We would like to thank all the reviewers for their comments and suggestions. This work is supported in part by the Natural Science and Engineering Research Council of Canada (grant no. 611370), NSFC (61202333), CPSF (2012M520392), and the Israel Science Foundation (grant no. 324/11).

References

ATTENE, M., FALCIDIENO, B., AND SPAGNUOLO, M. 2006. Hierarchical mesh segmentation based on fitting primitives. *Visual Computer 22*, 3, 181–193.

CARLSON-RADVANSKY, L., COVEY, E., AND LATTANZI, K. 1999. "What" effects on "where": Functional influence on spatial relations. *Psychological Science 10*, 6, 519–521.

CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. on Graph (SIGGRAPH) 30*, 4.

CHEN, X., GOLOVINSKIY, A., AND FUNKHOUSER, T. 2009. A benchmark for 3D mesh segmentation. *ACM Trans. on Graph (SIGGRAPH) 28*, 3, 73:1–12.

DE GOES, F., GOLDENSTEIN, S., AND VELHO, L. 2008. A hierarchical segmentation of articulated bodies. *Computer Graphics Forum (SGP) 27*, 5, 1349–1356.

EVERITT, B. S., LANDAU, S., LEESE, M., AND STAHL, D. 2011. *Cluster Analysis*. Series in Probability and Statistics. Wiley.

FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. on Graph (SIGGRAPH) 30*, 4.

GOLDMAN, R. N. 1992. Decomposing linear and affine transformations. In *Graphics Gems III*. Academic Press, 108–116.

GOLOVINSKIY, A., AND FUNKHOUSER, T. 2008. Randomized cuts for 3D mesh analysis. *ACM Trans. on Graph 27*, 5, 1–12.

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

HOFFMAN, D. D., AND RICHARDS, W. A. 1984. Parts of recognition. *Cognition*, 65–96.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. *ACM Trans. on Graph (SIGGRAPH Asia) 30*, 6.

JAIN, A., THORMÄHLEN, T., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Exploring shape variations by 3D-model decomposition and part-based recombination. *Computer Graphics Forum (Eurographics) 31*, 2.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. on Graph (SIGGRAPH) 29*, 3, 102:1–12.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model of component-based shape synthesis. *ACM Trans. on Graph (SIGGRAPH) 31*, 4.

KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. on Graph (SIGGRAPH) 22*, 3, 954–961.

KIM, Y. M., MITRA, N., YAN, D., AND GUIBAS, L. 2012. Acquiring 3D indoor environments with variability and repetition. *ACM Trans. on Graph (SIGGRAPH Asia) 31*, 6.

KRAEVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. In *Proc. Pacific Graphics*, 129–138.

LIU, R., AND ZHANG, H. 2007. Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum (Eurographics) 26*, 3, 385–394.

MARTINET, A. 2007. *Structuring 3D Geometry based on Symmetry and Instancing Information*. PhD thesis, INP Grenoble.

MITRA, N. J., YANG, Y.-L., YAN, D.-M., LI, W., AND AGRAWALA, M. 2010. Illustrating how mechanical assemblies work. *ACM Trans. on Graph (SIGGRAPH) 29*, 58:1–58:12.

NADLER, B., LAFON, S., COIFMAN, R. R., AND KEVREKIDIS, I. G. 2005. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In *NIPS*, 1–8.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. on Graph (SIGGRAPH) 30*, 4.

PALMER, S. E. 1977. Hierarchical structure in perceptual representation. *Cognitive Psychology 9*, 4, 441–474.

ROSE, R. C., AND KARNOWSKI, T. P. 2010. Deep machine learning - a new frontier in artificial intelligence research. *IEEE Computational Intelligence Magazine 5*, 4, 13–18.

SHAMIR, A. 2008. A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6, 1539–1556.

SHAPIRA, L., SHALOM, S., SHAMIR, A., COHEN-OR, D., AND ZHANG, H. 2010. Contextual part analogies in 3D objects. *Int. J. Comp. Vis. 89*, 1-2, 309–326.

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. *ACM Trans. on Graph (SIGGRAPH Asia) 30*, 6.

SUDDERTH, E. B., TORRALBA, A., FREEMAN, W. T., , AND WILLSK, A. S. 2005. Learning hierarchical models of scenes, objects, and parts. In *Proc. Int. Conf. on Comp. Vis.*, 1331–1338.

TORSELLO, A., HIDOVIC-ROWE, D., AND PELILLO, M. 2005. Polynomial-time metrics for attributed trees. *IEEE Trans. Pat. Ana. & Mach. Int. 27*, 7, 1087–1099.

WANG, Y., XU, K., LI, J., ZHANG, H., SHAMIR, A., LIU, L., CHENG, Z., AND XIONG, Y. 2011. Symmetry hierarchy of man-made objects. *Computer Graphics Forum (Eurographics) 30*, 2, 287–296.

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 Trans. on Graph (SIGGRAPH Asia) 31*, 6.

WOLF, L., BILESCHI, S., AND MEYERS, E. 2006. Perception strategies in hierarchical vision systems. In *Proc. IEEE Conf. on Comp. Vis. and Pat. Rec.*, 2153–2160.

XU, K., LI, H., ZHANG, H., COHEN-OR, D., XIONG, Y., AND CHENG, Z. 2010. Style-content separation by anisotropic part scales. *ACM Trans. on Graph (SIGGRAPH Asia) 29*, 6, 184:1–9.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Trans. on Graph (SIGGRAPH) 31*, 4.

YUMER, M., AND KARA, L. 2012. Co-abstraction of shape collections. *ACM Trans. on Graph (SIGGRAPH Asia) 31*, 6.

ZHANG, M.-L., AND ZHOU, Z.-H. 2009. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence 31*, 47–68.

ZHENG, Y., COHEN-OR, D., AND MITRA, N. J. 2013. Functional substructures for part compatibility. *Computer Graphics Forum (Eurographics) 32*, 2.