Recurring Part Arrangements in Shape Collections

Youyi Zheng[1]   Daniel Cohen-Or[2]   Melinos Averkiou[3]   Niloy J. Mitra[3,1]

[1]KAUST   [2]Tel Aviv University   [3]University College London

Abstract

Extracting semantically related parts across models remains challenging, especially without supervision. The common approach is to co-analyze a model collection, while assuming the existence of descriptive geometric features that can directly identify related parts. In the presence of large shape variations, common geometric features, however, are no longer sufficiently descriptive. In this paper, we explore an indirect top-down approach, where instead of part geometry, part arrangements extracted from each model are compared. The key observation is that while a direct comparison of part geometry can be ambiguous, part arrangements, being higher level structures, remain consistent, and hence can be used to discover latent commonalities among semantically related shapes. We show that our indirect analysis leads to the detection of recurring arrangements of parts, which are otherwise difficult to discover in a direct unsupervised setting. We evaluate our algorithm on ground truth datasets and report advantages over geometric similarity-based bottom-up co-segmentation algorithms.

1. Introduction

> *... that form ever follows function. This is the law.*
> — L. Sullivan (1896)

Large geometric differences often hide semantic similarity across many related objects. Discovering such latent commonalities reveals interesting object characteristics along with their unifying semantic connections, and can greatly benefit model exploration and content creation. Consistent cross-model part level decompositions are rarely explicitly encoded in model collections, necessitating automatic analysis to reveal such connections.

In recent years, various approaches have been developed to collectively analyze model sets. Such co-analysis algorithms either assume access to labeled training sets to facilitate supervised learning [KHS10,LMS13], or heavily rely on geometric similarity across the models for successful analysis [HFL12, HZG*12, KLM*12, KLM*13]. In the presence of significant geometric and topological variations, we still lack appropriate algorithms to automatically analyze collections of semantically related objects. For example, we want to link consistent parts across the chairs in Figure 1-top, which is still beyond the realm of state-of-the-art purely geometry-based methods (see Table 1).

We observe that certain patterns in the form of *arrangements* among object parts are intrinsic to object characteristics and *recur* across models belonging to semantically related shape collections (e.g., legs support chair seats in typical part configurations). Discovering such patterns requires identifying what are the parts, how they are arranged, and how they correspond across the models. Our main observation is that although part geometries can significantly vary across related models, their spatial arrangements (which we refer to as part arrangements) remain consistent, and hence can be used to establish part correspondence.

In particular, given a set of multi-component models in a shape collection, our goal is to group the input components to form object *parts*, whose consistency links to interesting and meaningful semantics across model collections. For example, in Figure 2, the relative arrangement of wheels, frames, handles, seats is discovered, eventually leading to consistent part correspondence across the models. Note that regions of the models can go unclaimed (shown in gray).

We focus on man-made objects as commonly found in online 3D repositories. Such models are typically non-manifold, come in multi-components, and contain large ge-

Figure 1: *We present a top-down indirect analysis to discover corresponding parts (bottom) across objects, even with significant geometric and topological differences (top).*
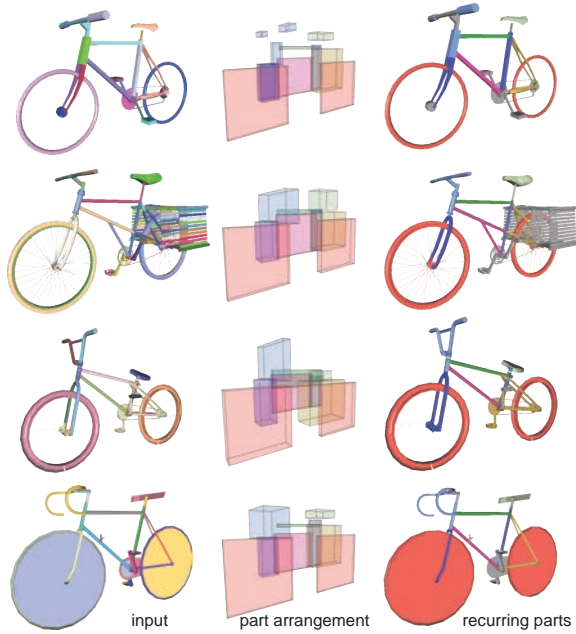
Figure 2: *Starting from unlabeled models (left), we discover recurring part arrangements (middle) and in turn extract corresponding parts, even in presence of significant geometric and topological variations (right). Unassigned components are indicated in gray.*

ometric differences. A direct bottom-up approach would be to create a set of possible parts for each model, and then, 'cluster' the parts across the different models. However, due to large geometric differences, typical geometric shape descriptors are not sufficiently descriptive to facilitate reliable clustering or grouping (see Table 1). Alternatively, one could try to directly solve for point-to-point correspondence, which can be ambiguous for models with large geometric differences (see [vKZHCO11]).

Instead, we propose an indirect analysis. First, we group different components from each model to form candidate *parts*. Parts from the *same* model are then combined to form different spatial arrangements. We focus on *pairs* of parts or simply *pair arrangements* (PA) as the basic unit of arrangement. Although we cannot compare parts from different models; we can, however, compare PA-s from different models, *without* establishing explicit part correspondence (see Figure 3). The information is then propagated down to compare segments, both within and across models. Multiple such PA-s, in turn, reveal larger consistent *part arrangements* (see Figure 2). Finally, in a gather stage, the information is accumulated to reveal recurring part arrangements and establish part correspondence.

We tested our algorithm on a range of man-made object collections with significant geometric differences. To the best of our knowledge, we are the first to demonstrate that an indirect, unsupervised analysis working on abstracted part

arrangements can reveal interesting substructures in model collections. We created a benchmark consisting of models with semantic parts manually marked, and used it to compare our algorithm with state-of-the-art alternatives. In summary, we formulate the problem of co-analysis of a shape collection as discovering recurring part arrangements; design an unsupervised algorithm based on a novel indirect top-down analysis of part arrangements to reveal part correspondence; and evaluate the algorithm on a range of publicly available shape collections of man-made objects.

2. Related Work

Shape analysis. In order to create novel yet useful shapes from existing models, it is often desirable to have a semantic understanding of the source objects. Researchers have long aimed at inferring such information from *geometry alone*. For example, Fu et al. [FCODS08] infer upright orientation from a given model; iWires [GSMCO09] and followup efforts [XZCOC12, ZCOM13] analyze inter- and intra-part geometric relations for smart object manipulation and creation; Bokeloh et al. [BWS10] extract partial symmetry information towards inverse procedural modeling; encoding a hierarchy of symmetrically related parts in individual models [WXL*11] or in model collections [vKXZ*13]; Kalogerakis et al. [KCKK12] learn a probabilistic distribution over a part-based model encoding multiple object styles, part cardinalities, and part placements, and use it for shape synthesis. Such methods either work on isolated models (hence, no variations across models), or assume input/training models to have consistent partitioning and part annotations, or rely on geometric features for correspondences.

As humans, we categorize objects into semantically meaningful parts, and classify their parts based on their use [FSH11, LMS13]. Can we do the same computationally? Grabner et al. [GGVG11] investigate the intriguing question of 'what makes a chair a chair'. Our goal is also to extract semantic structure, without access to tagging or semantic groups, purely based on input geometry.

Co-analysis of model collections. Compared to traditional analysis of individual shapes, multiple shapes offer richer information regarding what changes and what remains invariant across related shapes. Attene et al. [ARSF09] proposed the "ShapeAnnotator" to support interactive segmentation and annotation of 3D models, organizing them in a shape ontology to demonstrate applications in both virtual modeling and physical prototyping. Later, Golovinskiy and Funkhouser [GF09] presented an automatic approach that uses pairwise rigid alignment between models to establish correspondence and extract consistent segmentation. Subsequent improvements continue to rely on pairwise rigid alignments and then diffuse the reliable alignment information regularized by loop constraints to obtain improved correspondence assignment [NBCW*11, HZG*12, KLM*12].
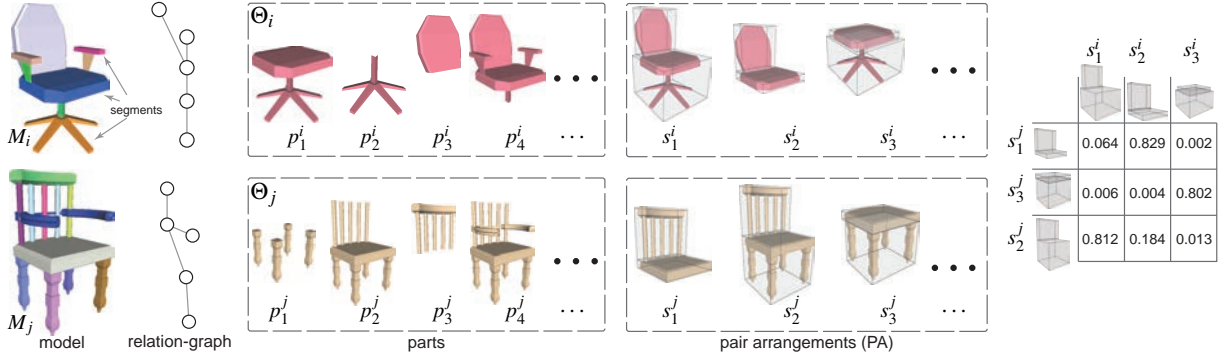
Figure 3: *Starting from multi-component models, we construct their relation graphs (see Section 4.1), which are then used to group components to form candidate parts. Non-conflicting pairs of such parts form pair arrangements (PA) (e.g., s_1^i is formed by (p_1^i, p_3^i)). Our main observation is as follows: while parts across different models are difficult to compare in the absence of any suitable correspondence (e.g., how to compare p_1^i to p_1^j?); pairs of parts, i.e., PA-s can be directly compared across models, and in turn be used to reveal similarity between parts across different models. We compare PA-s in a descriptor space; e.g., see the similarity matrix on the right; elements with higher values indicating higher similarity. Finally, recurring parts and their arrangements (see Figure 11(e)) are detected, often revealing semantic connections.*

Many model collections exhibit strong geometric similarity across the input shapes among their corresponding parts. Hence, researchers have proposed geometric feature-based clustering to define consistency among parts [KHS10, XLZ*10, HKG11, SvKK*11, vKTS*11, HFL12, WAvK*12, MXLH13]; extract low degree of freedom deformation linking different models for novel model exploration [OLGM11]; perform co-abstraction [YK12]; detect curve style [LZW*13] for co-analysis of shapes; and have employed geometric features and context to learn semantic correspondence using an SVM classifier [LMS13]. Recently, Kim et al. [KLM*13] learn a part-based deformable model by simultaneously optimizing for part decomposition, part correspondence, and a low complexity deformation model that best encodes input models.

Since the methods rely on part-level geometry to be consistent (and hence corresponding descriptors being consistent), they fail to reveal interesting relations across models under significant geometric variations. Specifically, the methods expect sufficient model pairs to be very similar geometrically in terms of size, orientation, etc. in order to reliably 'propagate' correspondence (see Section 5).

3. Overview

Starting from models from shape collections (e.g., Princeton Shape Benchmark, Trimble 3D warehouse, etc.), our goal is to group input components of models to reveal recurring parts and their arrangements. Since the models have large geometric variations, we look at relations (e.g., symmetry, contact) to reveal semantic consistency and not geometric consistency of the parts. (In the rest of the text, we refer to the different components from a model as *segments*.) For each model, based on their relations, we construct a set of possible groupings of segments to form candidate *parts*,

where each part is a connected subgraph of the relation-graph obtained from the initial multi-component models (see Section 4.1). If we can now compare parts coming from different models, we can discover a consistent pattern of part arrangements. However, in the absence of part-level correspondence, directly comparing them across different models is difficult (see Figures 1 and 3). We have an interleaved problem: in order to compare *part arrangements* we have to first determine what are the *parts*; while, relevance of parts is determined by consistency among their arrangements.

We take an indirect approach. Candidate parts from *within* models are grouped as model substructures to encode relative geometric arrangement. We consider pairs of parts (i.e., a PA) as the smallest non-trivial arrangement that reveals shape structure. As a key enabler, PA-s can be efficiently encoded and compared *across* different models without requiring any correspondence information. In Section 4.2, we describe how we create, encode, and compare PA-s from different models to form the PA-similarity matrix $\mathbb{M}_1$. Then, by accumulating information from $\mathbb{M}_1$, we define a part-similarity matrix $\mathbb{M}_2$ to compare parts across different models; $\mathbb{M}_2$ in turn is used to compare segments to form a segment-similarity matrix $\mathbb{M}_3$. Thus, the cascade $\mathbb{M}_1 \Rightarrow \mathbb{M}_2 \Rightarrow \mathbb{M}_3$ reveals segment-level similarity across models.

Finally, in Section 4.3, we use spectral clustering on $\mathbb{M}_3$ to group the segments and then extract recurring parts and their arrangements using an MRF formulation.

4. Algorithm

Given a set of models $\{M_1, M_2, \dots\}$, we first create candidate parts (denoted by set Θ_i) by grouping initial components for each model M_i. The key challenge is to reliably
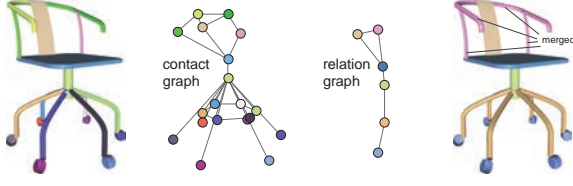
Figure 4: *Given a multi-component model (left) we create a contact-graph to capture pairwise component interactions. Then, based on partial symmetry information, we group components, forming a simplified relation-graph.*

compare two parts $p_a \in \Theta_i$ and $p_b \in \Theta_j$ coming from *different* models (i.e., $M_i \neq M_j$), without access to point- or part-level correspondence.

4.1. Generating Initial Candidate Parts

We encode contact and symmetry information in input models as relation graphs and generate possible part candidates. For each model M_i, we create a graph where each node represents a component or segment in M_i. Two nodes are connected by an edge if they are in contact (see also [MYY*10]). Specifically, for each pair of segments, we compute the smallest distance between them. If this distance is less than a threshold α, we consider the segments to be in contact ($\alpha = 0.01$ of the model's bounding box diagonal).

We assume that the input models have a consistent upright orientation (as commonly found in Trimble 3D warehouse models). Further, we pre-align the models using their PCA axes as proposed in Zheng et al. [ZCOM13]. Since the input models are assumed to have upright orientations, we simply have to decide between a PCA direction flip on the xy-plane (alternatively [KLM*12] can also be used).

In order to handle models with different part cardinalities (e.g., chairs with symmetrically arranged four vs. five legs), we simplify the graph based on symmetry information. We perform a graph simplification by grouping/merging segments by progressively collapsing the biggest symmetry groups [MGP06] if their common symmetry plane/axis is aligned to the upright direction (see Figure 4). Any resulting duplicate edges are removed (see also [KCKK12]).

Man-made models coming from public repositories typically have many small and spurious segments (e.g., a car model can have many small bolts, each as a separate component). This leads to an unnecessarily large number of segments. Using a simple heuristic, we address the issue by removing the small nodes (by volume), while updating the graph connectivity. Specifically, if nodes n_i and n_j both share a direct connection with a small node n_k; we introduce an edge e_{ij} (unless already present), while removing the node n_k and edges e_{ik}, e_{jk}. We remove such small nodes one at a time, removing the smallest node in the current iteration, and stopping when each of the remaining nodes' volume is more than 1-2% of the model's bounding box volume.
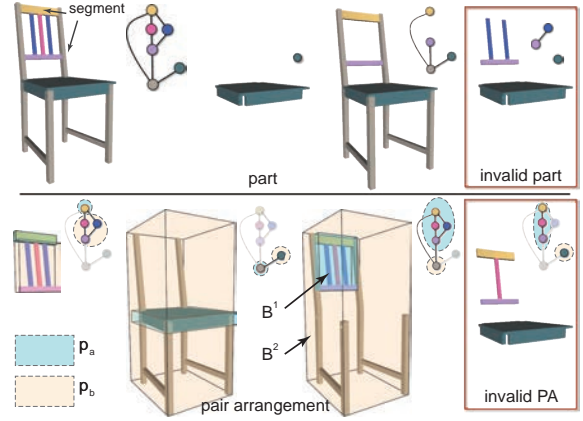

Figure 5: *Starting from a model, we first generate a set of valid candidate parts, which are then grouped together in pairs to form pair arrangements (PA).*

We call the simplified graph the *relation-graph* with each of its connected subgraphs (i.e., set of segments) forming a valid *part*. Figure 5-top shows a few examples; the highlighted part is invalid as the corresponding subgraphs are not connected by any direct edge in the relation-graph. In the end, for each model M_i, we have its relation-graph and a set of candidate parts.

4.2. Comparing Candidate Parts

Creating pair arrangements (PA-s). First, we extract arrangements of parts within a model. We use *pairs* of parts as the smallest unit of arrangement. Given a set of candidate parts Θ_i for a model M_i, not all pairs of parts, however, are useful (see Figure 5). Specifically, we form *pair arrangements* (PA) using a pair of parts $s_{ab} := (p_a, p_b)$ such that $p_a, p_b \in \Theta_i$ and (i) the subgraphs corresponding to the pair of parts in the original relation-graph of M_i are connected by (at least) a direct edge; and (ii) the set of segments of the parts are disjoint. For example, in Figure 5 the bottom-right figure shows an invalid PA that violates condition #(i). Note that the PA-s are only comprised of candidate parts from the same model and thus trivially have the same coordinate frame. We next define a descriptor to characterize the spatial arrangement of parts in such PA-s.

PA similarity matrix $\mathbb{M}_1$. Our motivation is to effectively differentiate between PA-s with a different arrangement of parts. One simple solution would be to compute a signature based on the geometry of the underlying parts. For example, we can compute a shape distribution [OFCD02] corresponding to the union of the parts in a PA. However, since our focus is to characterize part arrangements rather than part geometries, such an approach is not suitable (see Figure 6). The key is to ignore small-scale geometric variations and instead focus on how the parts are arranged. We propose a simple descriptor to do so.

Given PA $s_{ab} := (p_a, p_b)$, we compute the bounding boxes of its component parts. Note that we compute axis-aligned bounding boxes instead of oriented bounding boxes since at this stage the models are already axis-aligned. Based on the relative positions (along upright direction) of the centroids of p_a and p_b, we name their bounding boxes as B^1 and B^2, with B^1 denoting the upper box. We define the descriptor for PA s_{ab} using the set of pairwise distances between vertices of B^1 and B^2. Specifically,

$$\mathcal{S}(s_{ab}) := \{\|\mathbf{v}_i(B^1) - \mathbf{v}_j(B^2)\|\} \tag{1}$$

for $i, j \in [0, 7]$ going over the eight vertices $\mathbf{v}_i$ of each bounding box. (Note that we can select any consistent but fixed ordering among the vertices.) Finally, to account for scale variations, we normalize the descriptor using its Euclidean norm as: $\mathcal{S}(s_{ab}) \leftarrow \mathcal{S}(s_{ab})/\|\mathcal{S}(s_{ab})\|$.

In Figure 6, we compare the proposed spatial arrangement descriptor to shape distribution (computed using a uniform set of 10K points per PA with 64 histogram bins and averaged over 5 runs) on a small example. In each case, we computed descriptors for a set of 69 PA-s extracted from a set of three airplane models, computed their pairwise distances, and embedded the distances to 2D using multi-dimensional scaling (MDS). Our proposed descriptor by focusing on PA-s over the geometry of the parts can more reliably cluster similar arrangements (arrangement of orange/blue boxes in this example), while differentiating between different ones.

We construct a PA similarity matrix $\mathbb{M}_1$ to store the pairwise similarity between all pairs of PA-s coming from different models (i.e., $M_i \neq M_j$). Note that the number of columns/rows in $\mathbb{M}_1$ is equal to the total number of PA-s across all the input models. Specifically, given two PA-s $s_{ab} \in M_i$ and $s_{cd} \in M_j$, we define the corresponding similarity matrix entry in $\mathbb{M}_1$ as $\exp(-\|\mathcal{S}(s_{ab}) - \mathcal{S}(s_{cd})\|^2/2\sigma^2)$,
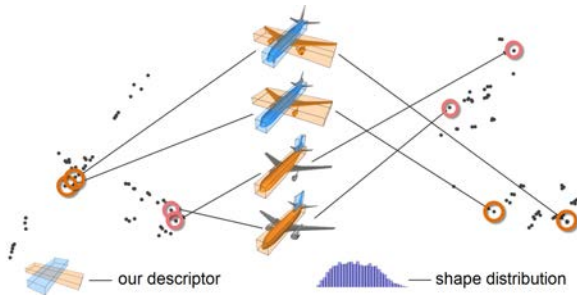

Figure 6: *We design a simple descriptor for PA-s, based on the relative arrangement of the bounding boxes of their individual parts. A candidate part (e.g., fuselage) can participate in different PA-s and thus appear in multiple regions of the descriptor space. The sub-figures show MDS mapping to 2D applied to the pairwise PA distances computed using the proposed descriptor (left) and shape distribution (right). Focusing on part arrangements over part geometries reveals high-level similarity.*

i.e., higher distances getting lower similarity values. We set $\sigma = 0.1$.

Alternative descriptors can also be used, e.g., we experimented with a descriptor that computes the translation/rotation/scaling ($\in \mathbb{R}^9$) necessary to map $B^1 \rightarrow B^2$, but found the performance comparable for the repository models that were mostly axis-aligned. The challenge then was to relatively weigh distance, angle, and scale.

Part similarity matrix $\mathbb{M}_2$. We use the PA similarity matrix $\mathbb{M}_1$ to vote for similarity among corresponding candidate parts coming from different models. We encode the similarity as the part-similarity matrix $\mathbb{M}_2$ with number of rows/columns equal to the total number of parts across all the models, i.e., $\sum_i |\Theta_i|$. Starting with $\mathbb{M}_2 = 0$, we take the appropriate similarity values from $\mathbb{M}_1$ and accumulate them to form $\mathbb{M}_2$. Intuitively, a high entry in $\mathbb{M}_2$ indicates a pair of parts that can be swapped across models, while preserving their respective PA descriptors, i.e., without disturbing the resultant arrangement.

For any valid PA-s $s_{ab} \in M_i$ and $s_{cd} \in M_j$ with $M_i \neq M_j$, we have a corresponding entry $\mathbb{M}_1(s_{ab}, s_{cd})$, capturing the similarity between $s_{ab} := (p_a, p_b)$ and $s_{cd} := (p_c, p_d)$. Now, in a key enabling step, we are ready to assess the similarity between $p_a \leftrightarrow p_c$ and $p_b \leftrightarrow p_d$. Intuitively, if parts can be swapped without affecting the corresponding PA descriptors, we expect them to be 'similar' and accumulate the evidence. Note that now similarities are between a pair of parts, but coming from *different* models M_i and M_j. We simply assign $\mathbb{M}_2(p_a, p_c) \leftarrow \mathbb{M}_2(p_a, p_c) + \mathbb{M}_1(s_{ab}, s_{cd})$ and similarly for $\mathbb{M}_2(p_b, p_d)$. The main observation is that we arrived at these part similarity values simply relying on pair arrangements. (Note that we assumed (p_a, p_b) and (p_c, p_d) to be relatively ordered based on corresponding bounding boxes B^1, B^2 as defined earlier. If not, we switch the assignments to $p_a \leftrightarrow p_d$ and $p_b \leftrightarrow p_c$.)

Segment similarity matrix $\mathbb{M}_3$. Finally, we use the information gathered in $\mathbb{M}_2$ to construct a similarity matrix $\mathbb{M}_3$ (with number of rows/columns equal to the total number of nodes across all relation-graphs) between pairs of segments (i.e., initial components) coming from different models, starting with $\mathbb{M}_3 = 0$. However, we cannot directly distribute the entries from $\mathbb{M}_2$ to $\mathbb{M}_3$ since different parts can be made of a different number of segments. We observe that splitting a part into any two subparts results in one of the following: (i) at least one invalid subpart, i.e., the nodes are disconnected in the corresponding relation-graph; (ii) two subparts where each subpart is a single node (i.e., segment); (iii) two subparts where one subpart is a part and another is a segment; or (iv) two smaller parts.

Based on these observations, we propose a recursive algorithm to accumulate part similarity ($\mathbb{M}_2$) into $\mathbb{M}_3$. For any pair of parts $p_a \in M_i$ and $p_b \in M_j$, we look at the corresponding similarity entry $\mathbb{M}_2(p_a, p_b)$. Further, say part p_a

has n_a segments and part p_b has n_b segments. Then, there are $\binom{n_a}{2}$ and $\binom{n_b}{2}$ different ways of splitting each of the respective parts into two subparts. We now handle the four cases listed above.

Say part p_a is split into a^1 and a^2; and similarly part p_b into b^1 and b^2. Since they have a relative ordering as defined before, we discuss the groupings of the form (a^1, b^1) (the process is repeated for the groupings of the form (a^2, b^2)).
Type (i) split: We discard splits resulting in any invalid part.
Type (ii) split: If a^1 and b^1 are both individual segments, we assign $\mathbb{M}_3(a^1, b^1) \leftarrow \mathbb{M}_3(a^1, b^1) + \mathbb{M}_2(p_a, p_b)$.
Type (iii) split: If only a^1 is a single segment and b^1 is a part, we distribute $\mathbb{M}_2(p_a, p_b)$ to the corresponding elements of $\mathbb{M}_3$ using a^1 and the single segments constituting b^1. We distribute the score based on the relative volume of the segment in b^1 normalized by the total volume of all the segments in b^1. We similarly distribute scores if b^1 is a single segment with a^1 being a part.
Type (iv) split: If a^1 and b^1 are both parts, we select the best matched splits (based on $\mathbb{M}_1$ entries), and then recursively continue the procedure. Note that in this setting there is another entry $\mathbb{M}_2(a, b)$ that may also be distributed to the matrix $\mathbb{M}_3$, further voting for segment-level similarity.

4.3. Extracting Recurring Parts

Given segment-similarity matrix $\mathbb{M}_3$, we directly use spectral clustering to group the segments, both within and across the models. Thus, for a grouping with k clusters, we get k groups across all the models, although each model can have k or fewer such members. Figure 7 shows spectral clusters for a small set of airplane models yielding groups of fuselage, wings, tail, front wheel, etc.

Our goal now is to select parts, one from each model M_i, such that the set of selected parts have maximum *consistency* across the model collection. Note that since similarity is inherited from matrix $\mathbb{M}_1$, consistency among parts is based on consistency among arrangements. Essentially, we have a labeling problem: For each model M_i, our goal is to select only *one* of its parts, i.e., select a label l_i from the set of all its parts labeled as $\{l_i^0, l_i^1, \dots\}$. The selected parts should be consistent across the model collection. We define unary and binary terms to quantify this consistency.

The unary term is to impose a hard constraint that a model selects a part only from its own label set, and not from another model. Thus, $E(M_i \rightarrow l_j^k)$ is given equal weight (set to 1) if $i = j$, and high penalty (set to ∞) for $i \neq j$, for any k.

The joint assignment likelihood term is defined as: $E(M_i \rightarrow l_i^k, M_j \rightarrow l_j^l) := \exp(-sim(p_a, p_b)^2)$ where, $p_a \in \Theta_i$ denotes the part from M_i corresponding to the label l_i^k and similarly for $p_b \in \Theta_j$ from M_j. We now describe how to estimate $sim(p_a, p_b)$.

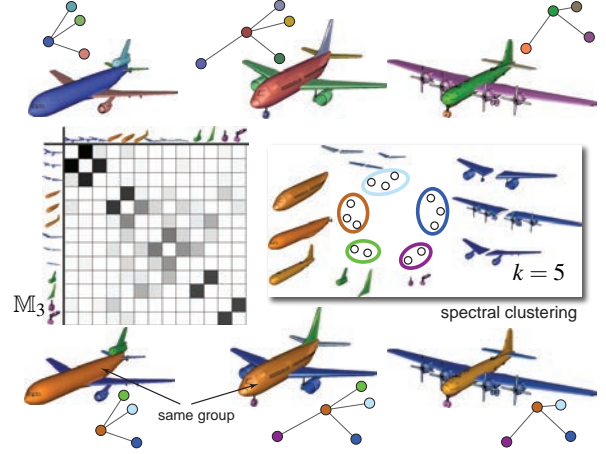Intuitively, we look at the graphs of p_a and p_b, and sum

Figure 7: *We use standard spectral clustering on $\mathbb{M}_3$ to group the segments, both within and across the different models. The figure shows oversegmented input models (top) and the same models after regrouping segments (bottom), with nodes from the same group colored similarly.*

up their edge-to-edge similarity, while matching edges based on their node colors. We break p_a (and p_b) into smaller segments (at least with two nodes), and compare these segments if their group colors match. These smaller segments are simply a subset of PA-s of M_i (and M_j). For each p_a, we select $\{s_a^1, s_a^2, \dots\}$ from the original PA set of M_i (similarly for p_b). Any selection of the form s_a (similarly s_b) satisfies two properties: (i) the segments in s_a are contained in p_a, i.e., $s_a \subset p_a$; and (ii) if $s_a = (p_1, p_2)$ then the segments in p_1 and p_2 should all have a single (group) color, and segments in p_2 should all have a second color.

Now based on group index/color between nodes in p_a and p_b, for each s_a^k we have one (or more) corresponding s_b^k. Hence, we simply define: $sim(p_a, p_b) := \sum_k sim(s_a^k, s_b^k) = \sum_k \mathbb{M}_1(s_a^k, s_b^k)$ using the observation that the respective entries are already in $\mathbb{M}_1$. Our goal is to solve for the best labeling such that:

$$\{l_i\}^\star := \underset{\{l_i\}}{\operatorname{argmin}} \sum E(M_i \rightarrow l_j^k) + \sum_{i,j} E(M_i \rightarrow l_i^k, M_j \rightarrow l_j^l) \tag{2}$$

using α-β swap [BK04].

We select the number of clusters by looping through $k = 2 : n$, with n being the number of the nodes in the relation-graph. For example, in Figure 7, $k = 5$ yields the lowest cost (MRF costs are: $5.7, 5.2, 5.1, 4.5$ for $k = 2, 3, 4, 5$, resp.).

5. Evaluation

Datasets. We tested our algorithm on various datasets of man-made objects obtained from public repositories (e.g., Trimble 3D Warehouse, Princeton Shape Benchmark, and [XZCOC12]). The raw models came in multi-components
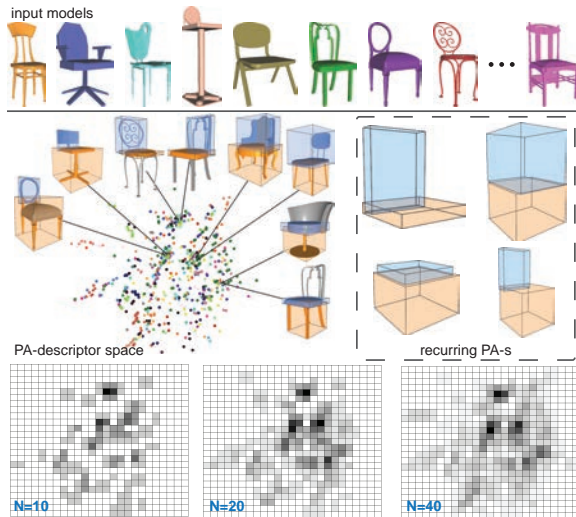
Figure 8: *PA-s generated from different models get embedded in a descriptor space. PA-s arising from consistent parts across models share similar location in this space. For visualization, here we map the space to 2D using MDS. As the number of input models grows (N=10, 20, 40), the recurring PA-s become increasingly prominent (in darker color) as the probability of unrelated PA-s coming from different models ending up at the same descriptor location decreases sharply.*

and often contain a large number of spurious components. We ignore the small components before creating candidate parts. We assumed the models to have consistent up-vector and pre-aligned them (see Section 4.1).

Our datasets (see supplementary material) comprised of different model collections including chairs (39 models), tables (21 models), battleships (8 models), airplanes (19 models), infant-beds (9 models), etc. Note that for models with less geometric variations, one should append our arrangement-based descriptor with a per-part (i.e., geometry inside bounding boxes) shape descriptor. This will, for example, help differentiate between a hollow versus an occupied box. However, in this paper, we investigate the core arrangement-based approach, and hence we selected models with significant geometric and topological variations.

Results. Figures 2, 10, and 11 show consistent part arrangements and parts for various model collections. (Please see supplementary material for all the results). Even for the same model collection, different models can have a different number of parts (e.g., airplanes with/without landing gear), or certain segments can stay unassigned (shown in gray). Note the diversity of the datasets, both in terms of geometric and topological variations. Our algorithm, by mainly focusing on the arrangement of parts, can detect consistency, and thereby extract parts and reveal their correspondence across models. In each case, the part arrangements reveal non-trivial parts.

Figure 9: *(Top-to-bottom) Starting from models across different shape collections, our method can still reveal consistent part arrangements and parts.*

For example, in the case of the chairs, we discover parts for back, seat, legs; for the tables, we get top and supporting legs; for the bicycles, we get wheels, frame, handle, seat; for the beds, we get frame, back, mattress, pillow; etc.

Since our method ignores low-level geometric details by abstracting candidate parts by their bounding boxes, our approach detects consistent part arrangements even among a mixture of different collections. For example, in Figure 9, we detect consistent leg and support arrangements among chairs, tables, sofas, etc. This result can enable new content creation possibilities by mixing model parts based on their part arrangement consistency.

Some semantically inconsistent parts as erroneously detected by our algorithm are highlighted in Figure 11. For example, when the windscreen and side mirrors are wrongly labeled as corresponding parts; or the top of a chair back stays unclaimed; etc.
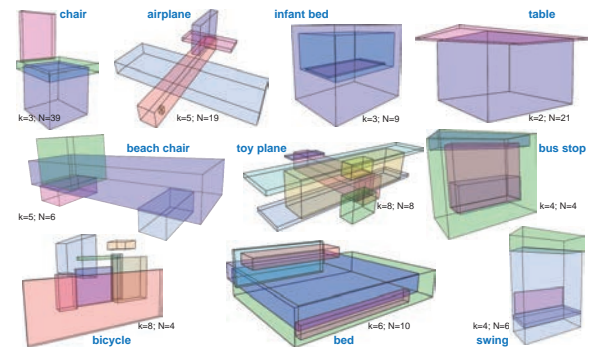

Figure 10: *Different representative part arrangements extracted for various model collections. Note that our unsupervised analysis relies only on available geometry information and has no access to part names or additional tagging information (see also Figure 11). The symbols k and N denote the number of part clusters and the number of models in the shape collections, respectively.*
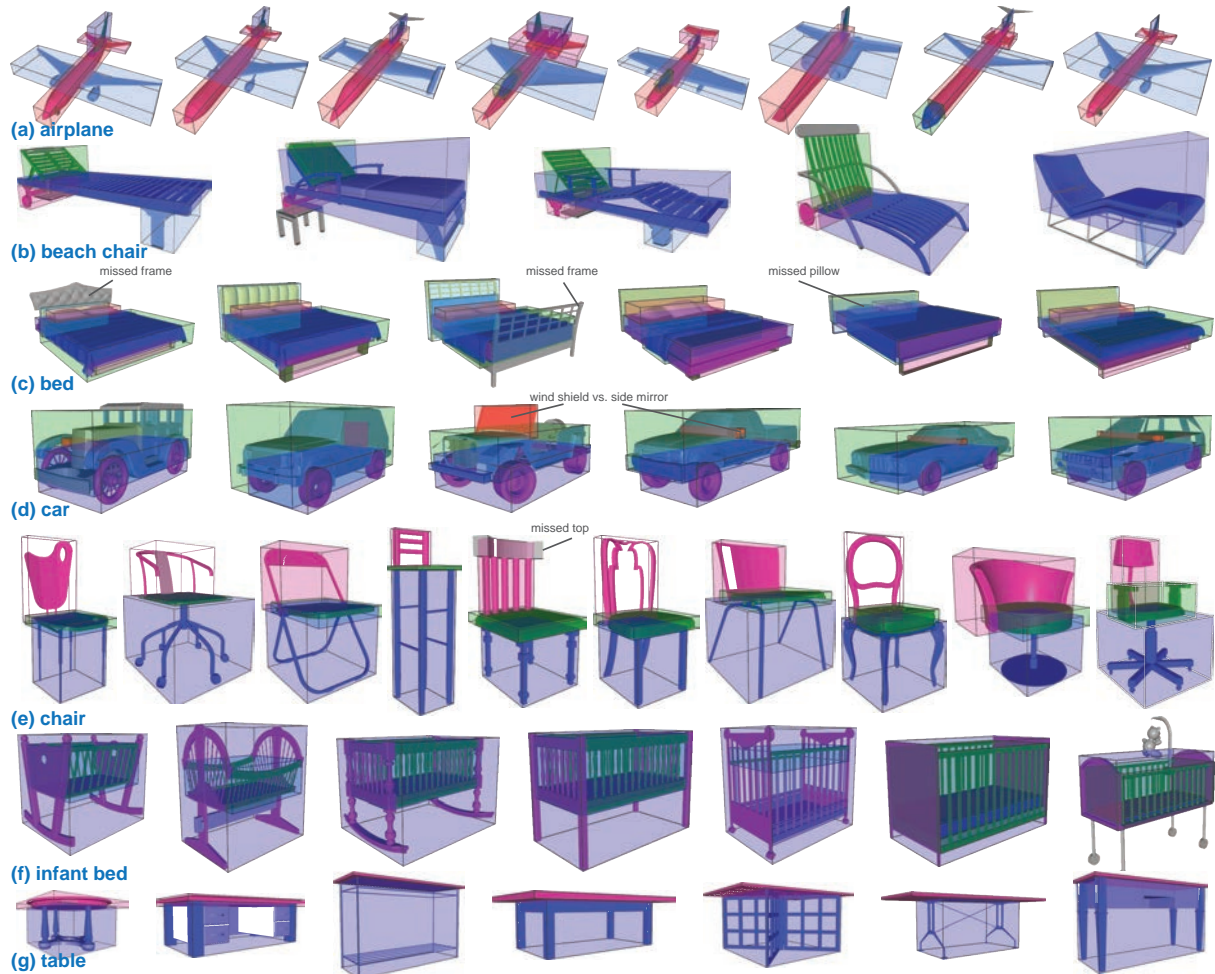
Figure 11: *Consistent part arrangements extracted using our unsupervised algorithm. For each model collection, we show only a subset of the results showing both the extracted part arrangements and parts. Unclaimed components are in gray (see also supplementary material).*

Comparison. Our goal is to discover consistency in the space of part arrangements, as well as establish part-level *semantic* correspondence. In order to quantitatively assess such semantic part correspondence, we first setup a manually annotated ground truth dataset. For four model collections, we manually grouped the segments in each model into parts and tagged them with an available part label. Part labels were selected from Wikipedia and were: leg, arm, back, seat for chair; supporter and bed for infant bed; wings, horizontal tail, vertical tail, landing gear, and body for airplane; sheet, pillow, front legs, back legs, back frame, front frame for bed. Note that the raw models came as multi-component inputs, which satisfy our assumption. For polygon soups, our method will be less suited.

We compared our algorithm against several state-of-the-art co-segmentation algorithms on the same dataset (using

the authors' implementations). Note that although the input requirements are quite different, we did this comparison mainly to validate our output. With the exception of Kim et al. [KLM*13], the rest of the algorithms assume the input to be manifold meshes. Hence, for those algorithms, in a preprocessing step, we first manually repaired the models and converted them to manifold meshes (using [LA13]). Note that most of these methods rely on various geometric features for bottom-up clustering; or geometric similarity between certain model pairs, which is then diffused to less similar models. Instead, we establish part correspondence in a top-down fashion via extracted repeated part arrangements, and can ignore fine-level geometric dissimilarity.

For each algorithm output, we manually tagged the final parts using the available part labels. We used inter-model part-correspondence to propagate the labels. Finally,
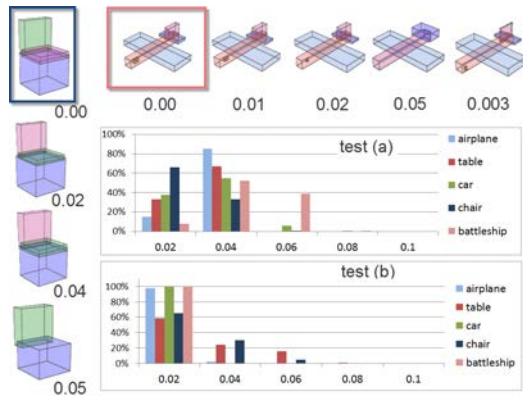
Figure 12: *Stability of results under different input segmentations. The percentage histograms show the distances between the discovered and the original part arrangements for 5 datasets, across 100 runs of the two tests. Distance is measured as the average distance between the corresponding parts' bounding box vertices. Notice the examples of chair and plane where distance is negligible.*

we counted the fraction of correct part tags (each segment gets one tag) to measure accuracy. For the preprocessed manifold models, we projected the face tags to their parent input models. Table 1 presents a summary of the comparison. In many cases, even on manifold meshes, clustering in the geometric feature space simply failed – we mark them by '×'.

Table 1: *Semantic segmentation accuracy against manually annotated groundtruth labeling across different algorithms.*

Class	Kim'13	Wang'12	Hu'12	Huang'12	ours
chair	0.59	0.69	0.30	0.83	0.96
bed	0.25	0.27	×	0.65	0.91
inf. bed	0.53	0.36	0.61	0.37	0.71
airplane	0.52	0.36	×	0.58	0.94

Robustness. Our algorithm relies on the input oversegmentations. We performed two tests to evaluate the stability of the algorithm on varying input segmentations. (Note that we cannot handle input as polygon soups due to computational complexity, and also due to degenerate configurations.) In the first test (#a), we selected 50% of the input models at random and ran our algorithm on such selections. For each model family, we compared the results across 100 random runs and quantitatively evaluated the closeness of the recovered parts with the base results. In the second test (#b), we selected 70% of the input models at random and independently split their segments (10-20%) also at random, and compared the stability of the results across 100 runs. In both tests, we found the discovered parts to be stable with respect to (moderate) variations in initial partitioning (see Figure 12). In most cases, the variations in the extracted part arrangements are marginal.

User study. We hypothesize that certain arrangements of parts are intrinsic to a given type of object, and these are extracted by our algorithm. To test this, we conducted a user study using a selection of shapes from our datasets (airplane, bed, car, chair, table). First, bounding boxes of the corresponding (extracted) part arrangements were shown in random order. Users were asked to *recognize* what type of shape these abstract boxes represent. Second, the same part arrangements were shown in a different random order, but this time the user had eleven options to choose from, drawn from the names of all datasets used in this paper.

136 users (age range 18-55 years) participated in the user study, with 74% being male and 62% having some computer graphics experience. Even in the first step, users were able to correctly identify airplane (93%), bed (100%), chair (99%) and table (99%), while they failed to recognize car reliably (35%). In the second stage, not surprisingly, the recognition success was slightly higher.

Timing. The complexity of our algorithm is $O(N^3)$, where N is the size of $\mathbb{M}_{j=1,2,3}$, which strongly depends on the number of nodes in the relation-graph as the size of matrix $\mathbb{M}_1$ depends on it. In our examples, the typical number of nodes in the relation-graphs is less than 15, e.g., for the airplanes and the cars it is around 10. While the typical number of PA-s was a few dozens for sparse graphs, it went up to a few hundreds for complex graphs with loops (e.g., bikes, toy-planes). The corresponding running times were a few seconds to 2-5 minutes (see also demo) for a single core implementation. Most example model collections took less than two minutes to process, as their relation-graphs are sparse. However, for complex datasets such as bike, car, and bed, it took about 5 minutes.

Limitations. Our algorithm suffers from the following limitations: (i) while our method assumes access to man-made models with multiple components, we can not handle polygon soups or a single connected manifold mesh or raw scanned point clouds; and (ii) since we search for all possible part arrangements, we run into scalability challenges for models with many spurious components or model collections with many models.

6. Conclusions and Future Work

We presented an unsupervised algorithm to extract consistent parts and their arrangements in model collections of semantically related shapes. In a top-down approach, arrangements of parts are compared across different models and the findings are then diffused to recover consistent parts across the different models. We evaluated our approach on a range of models with large geometric and topological variations, and compared our results with state-of-the-art co-segmentation methods, using manually-annotated benchmark datasets.

In the future, we would explore higher order part groupings involving triplets or quadruplets of parts. Further, the extracted parts along with their correspondence can be used for model synthesis while preserving discovered part arrangements to realize large geometric/topological changes.

References

[ARSF09] ATTENE M., ROBBIANO F., SPAGNUOLO M., FALCIDIENO B.: Characterization of 3D shape parts for semantic annotation. *Computer-aided Design 41*, 10 (2009), 756–763. 2

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI 26*, 9 (2004), 359–374. 6

[BWS10] BOKELOH M., WAND M., SEIDEL H.-P.: A connection between partial symmetry and inverse procedural modeling. *TOG (SIGGRAPH) 29*, 4 (2010), 104:1–104:10. 2

[FCODS08] FU H., COHEN-OR D., DROR G., SHEFFER A.: Upright orientation of man-made objects. *TOG (SIGGRAPH) 27*, 3 (2008), 42:1–42:7. 2

[FSH11] FISHER M., SAVVA M., HANRAHAN P.: Characterizing structural relationships in scenes using graph kernels. In *SIGGRAPH Asia* (2011), pp. 34:1–34:12. 2

[GF09] GOLOVINSKIY A., FUNKHOUSER T.: Consistent segmentation of 3D models. *Computers & Graphics (SMI) 33*, 3 (2009), 262–269. 2

[GGVG11] GRABNER H., GALL J., VAN GOOL L.: What makes a chair a chair? In *CVPR* (2011), pp. 1529–1536. 2

[GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iwires: an analyze-and-edit approach to shape manipulation. *TOG (SIGGRAPH) 28*, 3 (2009), 33:1–33:10. 2

[HFL12] HU R., FAN L., LIU L.: Co-Segmentation of 3D Shapes via Subspace Clustering. *CGF (SGP) 31*, 5 (2012), 1703–1713. 1, 3

[HKG11] HUANG Q., KOLTUN V., GUIBAS L.: Joint shape segmentation with linear programming. *SIGGRAPH Asia 30*, 6 (2011), 125:1–125:12. 3

[HZG*12] HUANG Q.-X., ZHANG G.-X., GAO L., HU S.-M., BUTSCHER A., GUIBAS L.: An optimization approach for extracting and encoding consistent maps in a shape collection. *SIGGRAPH Asia 31*, 6 (2012), 167:1–167:11. 1, 2

[KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model for component-based shape synthesis. *TOG (SIGGRAPH) 31*, 4 (2012), 55:1–55:11. 2, 4

[KHS10] KALOGERAKIS E., HERTZMANN A., SINGH K.: Learning 3D mesh segmentation and labeling. *TOG (SIGGRAPH) 29*, 4 (2010), 102:1–102:12. 1, 3

[KLM*12] KIM V. G., LI W., MITRA N. J., DIVERDI S., FUNKHOUSER T.: Exploring collections of 3d models using fuzzy correspondences. *TOG (SIGGRAPH) 31*, 4 (2012), 54:1–54:11. 1, 2, 4

[KLM*13] KIM V. G., LI W., MITRA N. J., CHAUDHURI S., DIVERDI S., FUNKHOUSER T.: Learning part-based templates from large collections of 3d shapes. *TOG (SIGGRAPH) 32*, 4 (2013), 54:1–54:11. 1, 3, 8

[LA13] LAFARGE F., ALLIEZ P.: Surface reconstruction through point set structuring. *CGF (EG) 32*, 2pt2 (2013), 225–234. 8

[LMS13] LAGA H., MORTARA M., SPAGNUOLO M.: Geometry and context for semantic correspondence and functionality recognition in manmade 3d shapes. *ACM TOG* (2013). 1, 2, 3

[LZW*13] LI H., ZHANG H., WANG Y., CAO J., SHAMIR A., COHEN-OR D.: Curve style analysis in a set of shapes. *CGF* (2013), to appear. 3

[MGP06] MITRA N. J., GUIBAS L., PAULY M.: Partial and approximate symmetry detection for 3d geometry. *TOG (SIGGRAPH) 25*, 3 (2006), 560–568. 4

[MXLH13] MENG M., XIA J., LUO J., HE Y.: Unsupervised co-segmentation for 3d shapes using iterative multi-label optimization. *Comput. Aided Des. 45*, 2 (2013), 312–320. 3

[MYY*10] MITRA N. J., YANG Y.-L., YAN D.-M., LI W., AGRAWALA M.: Illustrating How Mechanical Assemblies Work. *TOG (SIGGRAPH) 29*, 3 (2010), 58:1–58:12. 4

[NBCW*11] NGUYEN A., BEN-CHEN M., WELNICKA K., YE Y., GUIBAS L.: An optimization approach to improving collections of shape maps. *CGF (SGP) 30*, 5 (2011), 1481–1491. 2

[OFCD02] OSADA R., FUNKHOUSER T., CHAZELLE B., DOBKIN D.: Shape distributions. *ACM TOG 21*, 4 (2002), 807–832. 4

[OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3d shapes. *TOG (SIGGRAPH) 30*, 4 (2011), 33:1–33:10. 3

[SvKK*11] SIDI O., VAN KAICK O., KLEIMAN Y., ZHANG H., COHEN-OR D.: Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *SIGGRAPH Asia 30*, 6 (2011), 126:1–126:10. 3

[vKTS*11] VAN KAICK O., TAGLIASACCHI A., SIDI O., ZHANG H., COHEN-OR D., WOLF L., HAMARNEH G.: Prior knowledge for part correspondence. *CGF (EG) 30*, 2 (2011), 553–562. 3

[vKXZ*13] VAN KAICK O., XU K., ZHANG H., WANG Y., SUN S., SHAMIR A., COHEN-OR D.: Co-hierarchical analysis of shape structures. *TOG (SIGGRAPH) 32*, 4 (2013), 69:1–69:10. 2

[vKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *CGF 30*, 6 (2011), 1681–1707. 2

[WAvK*12] WANG Y., ASAFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *SIGGRAPH Asia 31*, 6 (2012), 165:1–165:10. 3

[WXL*11] WANG Y., XU K., LI J., ZHANG H., SHAMIR A., LIU L., CHENG Z., XIONG Y.: Symmetry hierarchy of man-made objects. *CGF (EG) 30*, 2 (2011), 287–296. 2

[XLZ*10] XU K., LI H., ZHANG H., COHEN-OR D., XIONG Y., CHENG Z.-Q.: Style-content separation by anisotropic part scales. *SIGGRAPH Asia 29*, 6 (2010), 184:1–184:10. 3

[XZCOC12] XU K., ZHANG H., COHEN-OR D., CHEN B.: Fit and diverse: set evolution for inspiring 3d shape galleries. *TOG (SIGGRAPH) 31*, 4 (2012), 57:1–57:10. 2, 6

[YK12] YUMER M. E., KARA L. B.: Co-abstraction of shape collections. *SIGGRAPH Asia 31*, 6 (2012), 166:1–166:11. 3

[ZCOM13] ZHENG Y., COHEN-OR D., MITRA N. J.: Smart Variations: Functional Substructures for Part Compatibility. *CGF (EG) 32*, 2pt2 (2013), 195–204. 2, 4