Meta-representation of Shape Families

Noa Fish[1]* Melinos Averkiou[2]* Oliver van Kaick[1] Olga Sorkine-Hornung[3] Daniel Cohen-Or[1] Niloy J. Mitra[2]
[1]Tel Aviv University [2]University College London [3]ETH Zurich

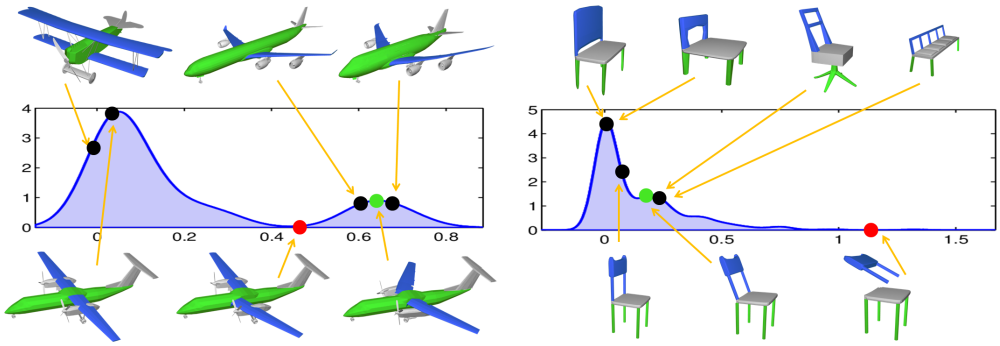


Figure 1: *Meta-representations of two families of shapes, where we show one selected probability distribution from each representation. Here, we see the distribution for the angle between the main axes of airplane wings and fuselage, and the angle between the main axes of chair backs and legs. Note that the main axes are oriented to the same direction, implying that two orthogonal parts form a zero angle. There are two major modes in each distribution, where examples of shapes corresponding to the black dots are shown. Besides such exploration, the meta-representation can also be used for applications like guided editing: the user deforms selected shapes, taking them to lower probability states (red dots), and then the system, guided by the meta-representation, returns the shapes to higher probability states (green dots).*

Abstract

We introduce a meta-representation that represents the *essence* of a family of shapes. The meta-representation learns the configurations of shape parts that are common across the family, and encapsulates this knowledge with a system of geometric distributions that encode relative arrangements of parts. Thus, instead of predefined priors, what characterizes a shape family is directly learned from the set of input shapes. The meta-representation is constructed from a set of co-segmented shapes with known correspondence. It can then be used in several applications where we seek to preserve the identity of the shapes as members of the family. We demonstrate applications of the meta-representation in exploration of shape repositories, where interesting shape configurations can be examined in the set; guided editing, where models can be edited while maintaining their familial traits; and coupled editing, where several shapes can be collectively deformed by directly manipulating the distributions in the meta-representation. We evaluate the efficacy of the proposed representation on a variety of shape collections.

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

Keywords: shape collections, consensus relations, model editing

Links: DL PDF WEB DATA CODE

*Joint first authors.

1 Introduction

High-level shape analysis goes beyond low-level analysis of the geometric properties of shapes and attempts to extract higher-level semantic information to aid in shape manipulation [Mitra et al. 2013]. As part of this effort, single shapes, particularly man-made objects, have been analyzed to extract semantic relations that can be made useful in various applications. One example is in applying constraints on a shape editing process, thereby achieving an intelligent edit where the prescribed geometric characteristics of the manipulated shape are preserved [Gal et al. 2009; Xu et al. 2009; Li et al. 2010; Zheng et al. 2011]. Ideally, the aim is to understand the *essence* of the family of shapes directly from their object geometry, in order to use this knowledge in the applications to determine the geometric shape and configuration of shape parts.

Analyzing an individual shape, however, is inherently limited as we are looking at a single example from a specific class of shapes. It is difficult to understand what really characterizes the family of the shape without additional information or other examples from the same class. Humans typically rely on their prior knowledge to infer this information. Hence, with the growth of shape repositories, researchers have focused on co-analyzing sets of shapes in order to benefit from the collective information in the group [Kalogerakis et al. 2010; Huang et al. 2011; Sidi et al. 2011; Wang et al. 2012; Kim et al. 2013]. However, the typical outcome of these methods, a segmentation and/or correspondence, does not really summarize properties that define the shapes as elements of a set.

Starting from a set of co-segmented shapes, we create a representation that captures the *essence* of the shape family, towards the goal of quantifying validity of the shapes. We call this representation a *meta-representation*. Specifically, we learn a system of geometric distributions to encode relative arrangements of parts across the family. Note that our representation is complementary to the one proposed by Kalogerakis et al. [2012] who learn a Bayesian network to capture co-occurrence statistics of parts for the purpose of shape synthesis. Instead, we focus on the distribution of part arrangements as learned from a collection of shapes. For example, in the case of a family of planes, we discover that there are two main

modes for the angle between the wings and fuselage (see Figure 1).

The meta-representation can then be used in a wide array of applications where one seeks to preserve the familial identity of a manipulated shape. Essentially, instead of assuming generic priors on allowed deformations (e.g., as-rigid-as-possible deformation [Sorkine and Alexa 2007]) or semantic relations (e.g., parallel or orthogonal part configurations being preferred [Gal et al. 2009; Zheng et al. 2011]), we directly use the representation to determine likely arrangements of parts. As our main application, we use the meta-representation to guide the editing of a shape to ensure that it remains a *valid* element of the set by keeping its main geometric characteristics. Thus, a desirable deformation amounts to refining part geometries and positions to increase validity as quantified by the meta-representation. The meta-representation then provides such guidance. In practice, this is realized by an editing tool that constrains parts to a valid space as the shape is edited.

Furthermore, by summarizing the input shape family, the meta-representation can also be used for exploring the set of shapes. In addition, it provides a collective handle to simultaneously refine a collection of shapes. For example, the user can directly edit the system of distributions, while our system adjusts the input shapes according to the prescribed meta-representation. This leads to a novel and intuitive *coupled editing* tool for sets of shapes.

We evaluate the proposed representation on various datasets and demonstrate the advantage of having such a meta-representation both for analyzing and manipulating shape families.

2 Related Work

In this section, we first discuss the works directly related to the concept of a meta-representation of a family of shapes. Then, we examine the relevant works in shape editing and synthesis.

Analysis of families of shapes. With the growing abundance of 3D shape collections, many approaches have been proposed to analyze families of shapes to benefit from the collective information. The majority of these works focus on consistently segmenting sets of shapes using a range of strategies including spectral clustering [Sidi et al. 2011], linear programming [Huang et al. 2011], active learning [Wang et al. 2012], subspace clustering [Hu et al. 2012], multi-label optimization [Meng et al. 2013], template fitting [Kim et al. 2013], etc. A few others have focused on extracting a consistent hierarchy for the set [van Kaick et al. 2013], or consistent part arrangements [Zheng et al. 2014] from an input family of shapes. Other works have focused on the exploration of shape sets, for example, by using a template that when deformed allows to indirectly navigate the shape space [Ovsjanikov et al. 2011], or by directly parameterizing the template space [Averkiou et al. 2014]. Although describing shapes with templates has similarities to our work, the models used in these works do not capture the relative orientation of parts and are not applied to interactive editing.

The question of how to exactly extract and encode a representation that captures the essence of a shape family, however, has not yet been thoroughly investigated. The works more closely related to this idea are those of Chaudhuri et al. [2011] and Kalogerakis et al. [2012]. Inspired by earlier efforts to represent shapes as parts and their connections [Funkhouser et al. 2004], they used a part-based representation as the basis for learning probabilistic models to describe shape families. Specifically, the models represent topological information such as the likelihood of the presence of certain parts and the cardinality of these parts, and the existence of certain shape styles according to the topology and geometry of the shapes. The models can be used to synthesize new shapes, but parts are placed with an automatic procedure. While our model is also based on a part-representation, we instead focus on the geometry of the part configurations. We learn the relative positioning and appearance of shape parts that is typical for a family of shapes. Thus, our work complements the more topological models of previous work.

Shape editing. In the context of organic shapes, several methods have been proposed to facilitate 3D shape editing (see [Botsch and Sorkine 2008] for a survey). Both surface-based and volume-based methods have been proposed to preserve local geometric details in the course of shape deformation.

In the context of man-made shapes, high level shape editing has been pursued. Realizing this goal amounts to detecting relationships between shape parts or features, and using these relationships to constrain the shapes during editing, referred to as the *analyze-and-edit* approach. For example, Xu et al. [2009] use slippable motion analysis to detect joints on the shapes, which can then be used as articulations to deform the shapes; Gal et al. [2009] detect feature curves (*wires*) on the objects, whose spatial relationships are preserved when deforming the shape; Li et al. [2010] introduce *arterial snakes* as feature curves used to deform 3D shapes that are inherently 1D, while Zheng et al. [2011] propose instead to use spatial controllers (similar to cages) placed on shape parts as the primitives for editing. In a different context, Lin et al. [2011] propose an analyze-and-edit approach for retargeting of 3D architecture, where the input models are analyzed and decomposed into a set of 1D structures that are easier to retarget.

These works can be seen as part of the larger group of structure-aware approaches, as the relationships that are captured typically seek to preserve the structural properties of the shapes (c.f., [Mitra et al. 2013]). However, in the shape editing approaches discussed above, structure is typically learned from a *single* shape, and not inferred from a set of shapes as is our goal in this work. Detecting structural relationships between multiple components is also important in other domains: Fisher et al. [2011] detect relationships between objects in scenes to perform scene or object retrieval; Yumer and Kara [2012] produce identity preserving mutually consistent coabstractions for shape collections; while Shtof et al. [2013] apply the structure-aware principle to sketch-based modeling. Differently from these works, Sumner et al. [2005] deform a mesh with inverse kinematics according to example deformations. Although the mesh is deformed based on external input, the user is responsible for providing the appropriate examples that guide the deformation.

Shape synthesis. There has also been work on automatically synthesizing novel shapes by creating variations from a set of base shapes, so that the user does not have to directly edit the shapes. These approaches typically combine parts from different shapes [Funkhouser et al. 2004], e.g., according to probabilistic inference [Kalogerakis et al. 2012], by exploiting partial symmetry [Bokeloh et al. 2010], through an evolutionary approach [Xu et al. 2012], by capturing the functionality of part structures [Zheng et al. 2013], or by directly navigating shape space manifolds implicitly defined by constrained surfaces [Yang et al. 2011]. Nevertheless, as discussed before, these approaches rely on probabilistic or symmetry-based models that mainly capture topological and style information about the shapes. We go beyond such models to also capture geometric relationships between parts.

3 The Meta-representation

In this section, we describe the meta-representation at a high-level. In the subsequent sections, we give details on its construction and how it can be used for different applications.

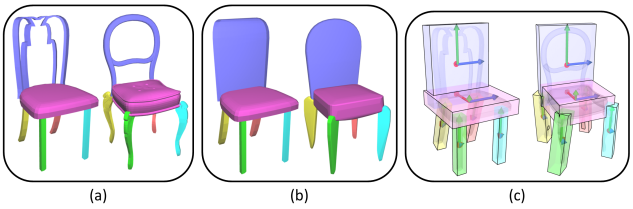


Figure 2: *Part abstraction: given the segmented and labeled shapes in (a), we compute the convex hull of each part (b), and then use the hulls to extract an OBB for each part (c), while also consistently ordering the OBB axes across different shapes.*

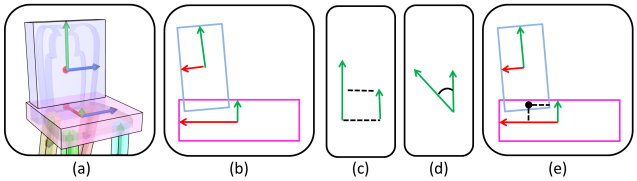


Figure 3: *(a) Given a pair of parts represented as two OBBs with their axes colored in red, green, and blue, illustrated in 2D in (b), we compute a set of binary relations that describe their relative arrangement. In this work, we consider: (c)* SCALE *relations, (d)* ANGLE *relations, and (e)* CONTACT *relations.*

Assumptions about the input. We assume that the input shapes are coming from the same family and are pre-segmented and consistently labeled. That is, the label of each shape segment is taken from a pre-defined set of labels that are relevant to the particular family. We follow this assumption both for the training set that defines the meta-representation and those shapes that are handled by the applications (which may not be part of the training set). Several unsupervised algorithms exist to automatically obtain such a labeled segmentation from an input set of shapes [Sidi et al. 2011; Huang et al. 2011; Hu et al. 2012; Meng et al. 2013; Kim et al. 2013; Laga et al. 2013], as well as semi-supervised algorithms [Wang et al. 2012]. Note that most of these algorithms automatically segment the shapes and assign generic labels. The user can then assign semantic names to the labels.

Next, each shape part is represented as an oriented bounding box, and we compute a set of *relations* for the boxes. The relations are functions that capture geometric configurations of the boxes and they can be *unary*, capturing the appearance of a single box in relation to the entire shape, and *binary*, capturing the relative positioning and appearance of a pair of boxes. The purpose of the relations is to capture any consistency of geometric configuration between the parts across the input shapes. The box representation and the relations that we compute are described in detail in Section 4.

The meta-representation. The goal of the meta-representation is to capture the essence of a specific family of shapes in terms of the geometric relationships between their shape parts. The meta-representation can then be used to estimate whether the parts of an unknown shape are in a typical arrangement for the family of shapes in question. Thus, it is natural to encode the meta-representation as a probabilistic model of the relations. In our work, we encode it as a probability density function (PDF) *independently* for each relation, and associate the PDFs to the part labels. Then, given parts with respective labels, we can query the PDFs to infer the probability of the part configuration. Figure 1 shows an example.

More formally, given a set of labels $\mathcal{L} := \{l_1, \ldots, l_m\}$, and a superset of relations $\mathcal{R} := \{R_1, \ldots, R_n\}$, divided into sets of unary relations $\mathcal{U} := \{U_i\}$ and binary relations $\mathcal{B} := \{B_j\}$, the meta-representation encodes a PDF for label l_i and unary relation U_k:

$$\text{PDF}_{l_i,U_k}(r) : \mathbb{R} \to \mathbb{R}, \tag{1}$$

and a PDF for every pair of labels $\{l_i, l_j\}$ and binary relation B_k:

$$\text{PDF}_{l_i,l_j,B_k}(r) : \mathbb{R} \to \mathbb{R}. \tag{2}$$

The collection of PDFs can be used to estimate the probability of a specific value $r \in \mathbb{R}$ for any relation R_k. They can be learned from the observed relation values extracted from a training set, as explained in Section 4, and used for different applications, as discussed in Section 5.

Complexity of learning the model. Our main assumption when designing the meta-representation is that the relations and labels are statistically independent, implying that we learn a PDF separately for each relation and individual or pair of labels. Such a simplified model is unable to capture any correlation that may exist between the relations. For example, the angle between the front wing and fuselage of an aircraft may be strongly tied to the angle between the fuselage and stabilizer of the aircraft, characterizing the style of the aircraft as a fighter jet or a passenger airplane. However, learning such a model involving correlations between all relations would require a great amount of training data (on the order of thousands of shapes), since otherwise outliers can easily bias the learning.

Nevertheless, our simplified model, which encodes only unary and pairwise relations while ignoring their correlation, makes it easy to learn a more robust model from much smaller training sets (only hundreds to dozens of shapes, which is common in many repositories). It is also more space- and time-efficient, allowing us to obtain a quick assessment of the likelihood of a part configuration. However, this comes at a cost: the meta-representation can sometimes lead to conflicts and contradictory constraints. We handle such errors by looking for a solution where we recover correlation by consistency of the pairwise relations (see Section 5). In summary, our design choice makes the representation and leaning simpler, albeit at the cost of a more involved framework for applications.

4 Learning the Meta-representation

The meta-representation is learned from a training set of shapes from the same family $\mathcal{S} := \{S^1, \ldots, S^n\}$. As a pre-processing step, the shapes are all normalized to the unit sphere, consistently segmented and aligned. First, an abstracted representation is computed for each shape part (Figure 2). Next, a set of relations is computed for every individual part and between every pair of parts (Figure 3). Finally, a statistical model that describes the relations is learned. We now discuss these steps in detail.

Part abstraction. Each input shape S^i is pre-segmented into a set of parts $\mathcal{P}^i := \{P^i_1, \ldots, P^i_m\}$. Note that some shapes may not have all the parts (e.g., a chair may not have arms). We represent each shape part P^i_j of a shape S^i as an oriented bounding box (OBB). The OBB for part P^i_j is described by its center $\mathbf{c}^i_j$, three axes $(\mathbf{a}^i_{j,1}, \mathbf{a}^i_{j,2}, \mathbf{a}^i_{j,3})$, and the extents $(e^i_{j,1}, e^i_{j,2}, e^i_{j,3})$ of the axes. In order to extract an OBB for a part, we build a set of candidate OBBs and select the one that better captures the symmetries of the part. This method yields more meaningful results than other approaches we experimented with, such as principal component analysis (PCA), or selecting minimum volume boxes.

The construction works as follows: We first compute the convex hull of the part's vertices. Each of the faces of the convex hull defines a plane onto which we project all the vertices of the hull.

Next, we compute the 2D bounding box (with minimum area) of the projected vertices [Schneider and Eberly 2003], and extrude the bounding box by following the direction of the plane's normal until we reach the most distant vertex of the hull. This defines a candidate OBB. Finally, we choose the candidate with the maximum number of reflective symmetry planes.

To test whether any of the three planes defined by the center $\mathbf{c}_j^i$ and the axes $(\mathbf{a}_{j,1}^i, \mathbf{a}_{j,2}^i, \mathbf{a}_{j,3}^i)$ possesses a reflective symmetry, we uniformly sample points on the surface of the part and reflect them across the potential symmetry plane. We then measure the distance of the reflected points to the surface. If a sufficient fraction (> 0.9) of the reflected points is closer than a threshold (0.0001), we mark the corresponding plane as a reflective symmetry plane. If there is more than one box which maximizes the number of symmetry planes, we break tie by selecting the one with the minimum volume. If none of the candidate boxes have symmetry planes, the smallest volume box is selected. Note also that, similar to [Zheng et al. 2011], we detect parts with rotational symmetry and mark them as special primitives with only one meaningful (rotation) axis.

Consistent axes ordering. To ensure that the axes of part boxes are consistently ordered across different shapes, we assume that all the shapes have the same upright orientation and face the same direction. We sort the OBB axes so that they best align with the global shape axes. Specifically, the first axis is set as the one that best aligns with the global x-axis, and the second axis as the one that best aligns with the y-axis and is orthogonal to the first chosen axis. This procedure ensures a consistent ordering for the majority of shapes. We manually override the automatic fix when the orientation is not consistent. However, due to the regularity of part arrangements in the selected sets, we only needed to fix the orientation for two shapes. An example of the part abstraction and axes ordering is shown in Figure 2. The consistent ordering leads to a meaningful representation of relations, as described next.

Inter-part symmetry. A shape may contain multiple parts with the same label and in many cases these parts are reflectively symmetric. In order to detect reflective symmetry between two parts with the same label, we employ a variant of the reflective symmetry detection described above. Here, the candidate plane is simply that given by the vector connecting the two centers of the parts along with the half-way point between the centers.

Part relations. Given a shape S^i, we compute a set of unary relations for every part P_j and a set of binary relations between every pair of parts (P_j^i, P_k^i). We define a set of relations to describe the geometric configuration of the shape parts. In our work, we choose unary relations that capture mainly the extent of each part axis relative to the scale of the entire shape:

$$\text{EXTENTS}(P_j^i) := \{e_{j,t}^i / d_i\},\ \forall t = 1 \dots 3, \tag{3}$$

where d_i is the diagonal of the bounding box of shape S^i.

The binary relations capture relative rotations, translations and scales between parts (illustrated in Figure 3):

$$\begin{aligned}
&\text{SCALES}(P_j^i, P_k^i) := \{e_{j,t}^i / e_{k,u}^i\},\ \forall t = 1 \dots 3; u = 1 \dots 3,\\
&\text{ANGLES}(P_j^i, P_k^i) := \{\angle(\mathbf{a}_{j,t}^i, \mathbf{a}_{k,u}^i)\},\ \forall t = 1 \dots 3; u = 1 \dots 3,\\
&\text{CONTACTS}(P_j^i, P_k^i) := \{t_{j,1}^i, t_{j,2}^i, t_{j,3}^i, t_{k,1}^i, t_{k,2}^i, t_{k,3}^i\},\\
&\qquad \text{where} \quad t_{j,m}^i = 2\|\mathbf{v}_j\| \cos(\angle(\mathbf{v}_j, \mathbf{a}_{j,m}^i))/e_{j,m}^i\\
&\qquad\qquad \text{with} \quad \mathbf{v}_j = \mathbf{p}_{\text{int}}(P_j^i, P_k^i) - \mathbf{c}_j^i.
\end{aligned} \tag{4}$$

Essentially, the contact relation between parts (P_j^i, P_k^i) is represented as the relative placement of the intersection point between their two boxes ($\mathbf{p}_{\text{int}}$), in the scaled coordinate system of P_j^i and of P_k^i. The intersection point, if it exists, is found by forming a grid of points on the frame of each box and testing containment of each grid point within the other box. We set $\mathbf{p}_{\text{int}}$ to be the average of the set of points which are found to be contained.

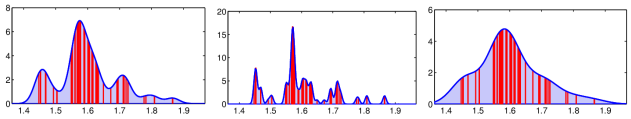


Figure 4: *Bandwidth selection to create the kernel density estimator (KDE): (a) Automatic selection with our criterion (red bars are training values). (b) Small bandwidth: note how there are many modes and gaps. (c) Large bandwidth: a single mode is created.*

Note that the chosen set of relations is redundant and may over-constrain the configuration between two boxes. The redundancy, however, helps towards more robust estimation.

Probability density function. As outlined in Section 3, we collect the relations for individual boxes and between pairs of boxes for all the shapes in the set, and then build the PDFs for unary (PDF_{l_i,R_k}) and binary (PDF_{l_i,l_j,R_k}) relations. Each PDF is effectively represented by a 1D kernel density estimator (KDE) [Silverman 1986]. Kernel density estimation is a standard non-parametric technique for estimating the PDF of a random variable, and represents the density as a sum of kernels g, each centered at one training sample (one relation value) $x_l \in X$:

$$\text{KDE}(r) := \sum_{l=1}^{|X|} g(r - x_l, h)/|X|, \tag{5}$$

where X is the entire set of training samples, and h is the *bandwidth* of the kernel. For our model, we use the common Gaussian kernel:

$$g(t, h) := \exp(-t^2/2h^2)/\sqrt{2\pi h^2} \quad \text{with } t \in (-\infty, \infty).$$

Selecting an appropriate kernel bandwidth is an important problem, illustrated in Figure 4. If the chosen bandwidth is too small, as shown in (b), the distribution does no generalize well, and several modes and gaps exist in the density function. If a large bandwidth is selected, as in (c), then important low probability regions are smoothed out in the distribution. Finally, with the correct bandwidth, as in (a), a more meaningful distribution with three large modes is created.

In our work, the bandwidth h is set based on a fixed scale parameter relative to the range of data in the distribution:

$$h := \sigma \cdot (\text{perc}_{95} X - \text{perc}_5 X)\,, \tag{6}$$

where perc_5 denotes the 5th percentile of X, and the scale $\sigma = 0.05$ was determined experimentally. Using the percentiles instead of the full data range makes the selection more robust by ignoring outliers. This criterion works well in practice when compared to other well-known alternatives such as cross-validation with the mean integrated squared error or rules of thumb [Chiu 1996]. These criteria are more suitable for finding a cluster structure in the data, and tend to separate the distribution into several modes. On the other hand, our criterion based on a scale parameter allows us to select the appropriate level of detail so that the distributions generalize well.

In general, note that the PDFs capture the commonality of the data in terms of the frequency of values. In regions of the KDE with a large number of training samples, the sum of Gaussians will create

peaks with high function values, spread according to the variance of the samples, while regions with only a few samples will have lower function values. Thus, since the KDE represents a continuous density function, the area under the curve corresponds to the probabilities. In practice, to extract a probability $p(r)$ for a specific value r, we integrate the function around a small ϵ-interval of r:

$$p(r) := \int_{r-\epsilon}^{r+\epsilon} \text{KDE}(r)\ dr. \tag{7}$$

In our implementation, we set $\epsilon = 0.01$ of the range of values in the PDF. Note that the probabilities cannot be easily used in an absolute sense, as they will be influenced by the structure of modes in the distribution. However, they can be used in a comparative manner, to compute the probability gain after changes to the relations.

5 Using the Meta-representation

In this section, we discuss different usages of the meta-representation. The representation, which summarizes the input shape collections, can be directly used for finding interesting shape configurations in the collection, reshaping any input model guided by the meta-representation, or for collectively editing all the input shapes by directly manipulating the meta-representation.

5.1 Exploration of shape families

Given the set of PDFs that define the shape meta-representation, we developed a tool for exploring interesting shape clusters closely tied to the configurations of certain parts. The motivation behind this is that any relation's PDF can exhibit multiple areas where probability density is higher, which in turn means that several shapes posses similar values for that relation. For example, by looking at the angle between the fuselage and wing of an airplane, we might observe a cluster of planes with orthogonal wings, and a cluster of planes with angled wings (see Figure 1).

In our exploration interface, the user starts by loading a family of shapes along with their extracted meta-representation. The user then selects any shape as a guidance, and picks one or two shape parts that she wants to use for exploring configurations inside the shape family. Then, any of the unary and binary relations can be selected, and the corresponding PDF can be inspected. Clicking anywhere on the PDF causes the set of loaded shapes to be sorted according to their distance from the clicked value. Then the user can browse through the nearest shapes around the clicked value, in increasing distance. This immediately defines an ordering of the shapes that allows the user to explore the shapes which are most similar in terms of exhibiting that specified value for the relation and the parts in question (see supplementary video).

5.2 Guided shape editing

Editing a shape can be a difficult task as various geometric and semantic aspects of the shape need to be considered to ensure a *valid* result. We take advantage of the meta-representation as a facilitator for this task, and use it as a guidance tool when editing shapes, to create a shape where the part configuration is similar to that observed in the shape family.

We propose an interactive shape editing tool where a user can manipulate the parts of a shape to create a variation of that shape. The user can scale, rotate, and translate one or more parts of the shape. Once an editing action has been carried out, the tool consults the meta-representation to restore the validity of the deformed shape. We can think of the user edit as taking the shape to a certain point in the space of all relations (i.e., points on the different PDF curves), which is associated with a probability given by the meta-representation. Next, our goal is to achieve a part arrangement so that the shape parts move to a (nearby) configuration with higher probability, corresponding to a valid shape. In this process, we constrain the deformation to take the shape to the closest valid state, so that the current part configuration is preserved as much as possible. We first formulate the problem using the meta-representation, and then propose an optimization to enable interactive applications.

Guided editing formulation. We pose the problem of taking an edited shape to a valid state as an optimization where we seek to increase the probability of the part configurations. This can be accomplished by modifying the part configurations so that the probability of their relations is locally maximized. Thus, assuming that the configuration of the shape parts $\mathcal{P}$ is described by a matrix C, we can pose the deformation goal as

$$\text{Def}(\mathcal{P}) := \arg\max_{C} \text{Obj}(C, \mathcal{P}), \tag{8}$$

where the objective function is given by

$$\begin{aligned}\text{Obj}(C, \mathcal{P}) := \quad & \exp\left(-\lambda \|C - C^0\|\right) + \prod_{\forall P_i} p_{P_i}(C_i) \times \\ & \prod_{\forall \{P_i, P_j\}} p_{\{P_i, P_j\}}(C_i, C_j).\end{aligned} \tag{9}$$

Here, C^0 denotes the initial configuration of the parts, C_i is the entry of C corresponding to the configuration of part P_i, p_{P_i} is the unary probability of part P_i, given by

$$p_{P_i}(C_i) = \prod_k p_{l_i, U_k}(f_{U_k}(C_i)), \tag{10}$$

and $p_{\{P_i, P_j\}}$ is the pairwise probability of parts P_i and P_j:

$$p_{\{P_i, P_j\}}(C_i, C_j) = \prod_k p_{l_i, l_j, B_k}(f_{B_k}(C_i, C_j)), \tag{11}$$

where l_i and l_j are the labels of P_i and P_j, respectively, p_{li, U_k} is the probability according to Equation (7) applied on PDF_{l_i, U_k}, p_{l_i, l_j, B_k} is the probability from $\text{PDF}_{l_i, l_j, B_k}$, f_{U_k} is a function that computes the value of relation U_k according to the configuration C_i of part P_i, and similarly f_{B_k} is a function that computes the value of relation B_k according to the configurations C_i and C_j of parts P_i and P_j. We used a scale factor $\lambda = 0.1$.

Thus, the first term of the objective function ensures that the solution does not deviate far from the initial part configuration, while the second term captures the probability of the entire part configuration as a combination of probabilities for individual parts and pairs of parts, computed according to the meta-representation. We now describe how to find such a solution.

Global optimization. We can directly search for a solution to the objective function in (9) with a non-linear optimizer, such as the BFGS quasi-Newton method. Note that the representation for the part configurations in the objective function is independent of the relation set. The relations $\mathcal{R}$ are designed to capture the properties that the configuration should satisfy to ensure shape validity, while the configurations C give the actual part positioning. In our implementation, we encode each C_i in terms of the position of the OBB center, the scales of its three main axes, and three Euler angles that describe the rotation of the OBB. Thus, the optimizer can directly search for the part configurations that satisfy the objective function.

However, this problem is a non-linear optimization of size $|C|$, which can take a considerable amount of time to solve when the input shapes consist of several parts (e.g., several minutes for a shape

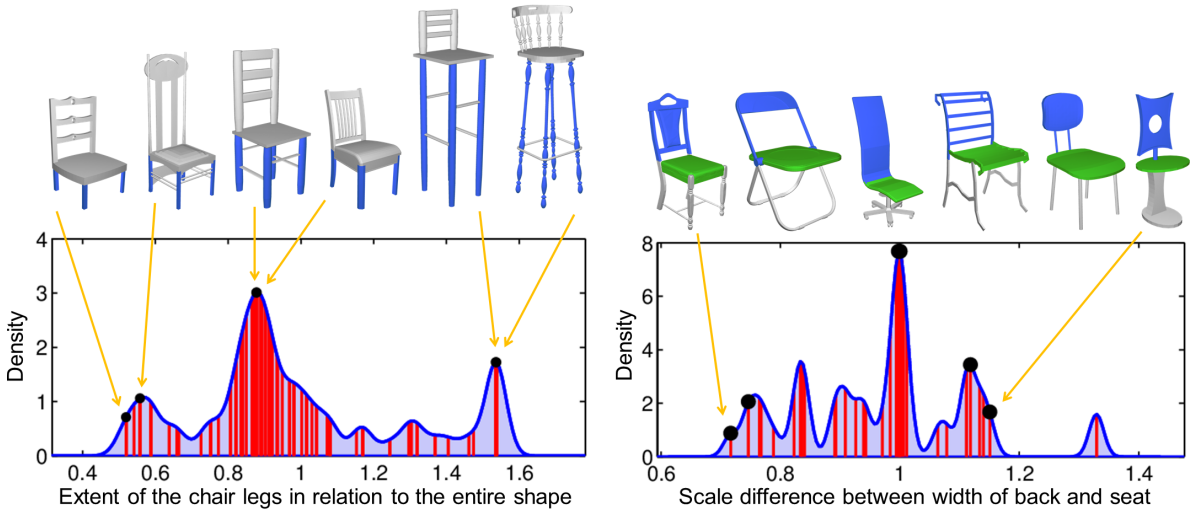


Figure 5: *The meta-representation enables the exploration of shape repositories: when clicking on different locations of the distributions, the exploration tool presents models with the selected relation values. (a) shows a unary relation for the blue parts, while (b) shows a binary relation between the green and blue parts. The shapes are ordered according to an increase of the selected relations values (black dots). Note that the 3rd and 4th chair both correspond to the highest peak. The red bars are all the training samples used to build the distributions.*

even with a few parts). Since our goal is an interactive tool, we propose a heuristic solution to speed up the computation of this objective, which we describe next.

Progressive solution. We break the global problem into a series of local optimization iterations. In each iteration, we pick one part $P^\star$ and solve for its position according to a set of parts that were already fixed. Next, $P^\star$ is added to the set of fixed parts, and we continue with the remaining part(s). We first describe how to determine a good propagation order (i.e., which part to fix next), and then how to position (i.e., fix) the selected part.

Determining a propagation order: Given a set of fixed parts $\mathcal{F} := \{P_1, \ldots, P_m\}$, our goal is to select part $P^\star$ from the set of remaining parts $\mathcal{M} := \{P_{m+1}, \ldots, P_n\}$ that still need to be moved to a fixed position. For each pair of parts (P_k, P_l) with $P_k \in \mathcal{F}, P_l \in \mathcal{M}$, let ${B_i}^c(P_k, P_l)$ be the current value of relation B_i between parts P_k and P_l, and ${B_i}^{opt}(P_k, P_l)$ its value after searching for a higher probability state of P_l according to the fixed part P_k. Let, $p({B_i}^c(P_k, P_l))$ and $p({B_i}^{opt}(P_k, P_l))$ be the corresponding probabilities. We define the probability gain as,

$$\text{PG}[B_i(P_k, P_l)] := \delta \cdot [p({B_i}^{opt}(P_k, P_l)) - p({B_i}^c(P_k, P_l))] \quad (12)$$

where,

$$\begin{aligned} \delta &= |{B_i}^{opt}(P_k, P_l) - {B_i}^c(P_k, P_l)| / R_i(k, l), \\ R_i(k, l) &= \max_c\{{B_i}^c(P_k, P_l)\} - \min_c\{{B_i}^c(P_k, P_l)\}. \end{aligned}$$

We select the part with the maximum gain as the part to be fixed next, i.e., $P^\star \leftarrow \arg\max_{l,i} \text{PG}[B_i(P_k, P_l)]$.

Given a relation value ${B_i}^c(P_k, P_l)$, we follow the ascending gradient direction from this value to a local maximizer in the distribution, according to the PDF corresponding to B_i. This provides a target value ${B_i}^{opt}(P_k, P_l)$. Note that if ${B_i}^c(P_k, P_l)$ happens to be in a very low probability region, we jump to the closest mode of significant probability. We also bypass modes that have a probability that is lower than a threshold (0.01 in our experiments). Figure 1 shows examples of such movements.

Solving for a part position: We now position the selected part $P^\star$ given a set of already fixed parts $\mathcal{F}$. One option is to directly use a modified formulation of Equation (8) wherein configuration C only considers the parts in $\mathcal{F}$ and $P^\star$; and only relations involving $(P^\star, P_k)$ for all $P_k \in \mathcal{F}$ are considered.

However, in practice, we found an approximate method to be much faster and more suitable for interaction. In this approximate method, each of the fixed parts independently suggests a new part position for $P^\star$; these positions are then combined together for the final position. Specifically, each part $P_k \in \mathcal{F}$ proposes a position $P_k^\star$ based on the relations $\{B_i(P_k, P^\star)\}$. For an even smoother interactive experience, we also experimented with defining the influence of a fixed part based on its adjacency. In this setting, we only consider suggestions for $P^\star$ given by its neighboring parts. We found this approach to be approximately three times faster with little to no difference in the resulting configuration. Since the relations that we consider over-constrain the position of the part, for each part $P_k \in \mathcal{F}$, we sort the relations according to their probability gain and select the subset of relations $\mathcal{B}_k$ that both maximizes the gain and determines a unique position for $P_k^\star$. This subset also determines a confidence weight $w_k \leftarrow \sum_{j \in \mathcal{B}_k} PG[B_j(P_k, P^\star)]$ for this proposed position. The final position for $P^\star$ is then taken as the weighted average of the candidates $P_k^\star$.

Inter-part symmetry handling. Inter-part symmetries within a shape, which are detected in the analysis phase (see Section 4), are utilized in our heuristic with the purpose of symmetry preservation, since this is a natural constraint for man-made objects. First, if the user deforms a part that has symmetric counterparts, the system begins by applying a similar deformation to all its symmetric parts, so that they are consistently configured prior to optimization. Next, throughout the optimization, symmetric parts are deformed as a group: only a single part of the group is directly deformed and then the deformation for the remaining parts is automatically inferred from that. Note that same-label parts that are not symmetric also exhibit a measure of consistency in their configurations, but to a lesser extent than symmetric parts. Therefore, such parts are also handled as a group, i.e., their configurations are determined by the same set of fixed parts, although they are optimized separately.

Global optimization. In comparison with an expensive global approach, we still achieve two goals. First, we maximize the probability of the relations by moving each part to increase the overall probability gain. Although we do not ensure a global maximum, we reach a local maximum of relation probabilities. This is demonstrated by the following experiment: if we apply the global optimization to the result of the progressive solution in Figure 7(b), the part configuration does not change significantly, showing that the probabilities are indeed located at a local maximum. Second, by starting the progressive solution from the user edit and following a local optimizer path, we ensure that we do not deviate much from the initial solution as only deformations that increase the overall probability are allowed.

In terms of complexity, an iteration of the propagation is much faster to solve (in the order of seconds), as there is only a single part to be optimized at each step. Thus, the aggregated time of all propagation iterations is also much less than the time needed to perform the global optimization, making this heuristic well suited to an interactive tool. For example, the first deformation in Figure 7(b) takes 0.33 seconds with the progressive solution, while the global optimization takes 1.8 hours.

5.3 Coupled shape editing

Editing multiple shapes at the same time in a coupled manner can be faster and more productive in situations where a modeler would like to perform the same type of edits to a family of shapes. The meta-representation directly allows such coupled guided edits through the relation PDFs. We have developed a coupled shape editing tool to illustrate this concept.

In the coupled editing tool, the user starts by loading a family of shapes, as before, pre-analyzed to extract the meta-representation. By selecting one or two parts from any shape in the family, and specifying a relation, the user can view the corresponding PDF. Recall that the PDF is built from a set of training samples of relation values coming from all the shapes in the family. The user can then edit the curve using a range of different manipulations. She can click anywhere on the curve to select a point, and then scale the training samples to the left and to the right of the selected point. This can be used to set all training samples for that relation to a specific value that is desired (see Figure 10 for an example where the angle between chair back and chair seat is set to a very small range of around 90 degrees).

This manipulation of the curve immediately defines a mapping between the initial training sample values and their newly specified values. We then iterate over all the shapes and set the value for that specified relation to the new value coming from the mapping induced by the new curve. This is done by rotating, translating and scaling the parts in question so that the value for the specified relation between them is set to the new value. This edit can break the relation between the rest of the parts for each shape, therefore we use the original contact relations to rotate and translate these parts until the contact relations are restored. Finally, since the meta-representation is no longer valid after the training shapes have been edited, in the last step we recompute the meta-representation based on the new part configurations for each shape in the family.

6 Experiments and results

In this section, we describe the experiments performed to evaluate the meta-representation. We designed three tools that illustrate the different areas where the meta-representation may be of use.

For all our experiments, we used four datasets of man-made shapes. The largest dataset contains around 400 chairs, taken from the COSEG benchmark database [Wang et al. 2012]. We also used a smaller dataset containing around 40 chairs, and two datasets with around 20 bicycles and 20 planes [van Kaick et al. 2013]. Smaller datasets were chosen to demonstrate that even with a small number of training samples, the meta-representation can capture important and useful relations between shape parts. All the datasets were pre-processed, starting from segmentation and consistent labeling, to establish correspondence between parts, abstraction of shape parts into boxes, consistent ordering of box axes, building the set of relations, and learning the model PDFs.

6.1 Exploration of shape repositories

Figures 1 and 5 show examples of exploration enabled by the meta-representation. In Figure 1, the user selected the wings and fuselage of airplanes, and navigated thorough the relations until selecting the angle between the first axes of the two parts. Note that these axes are aligned to point to the same general direction. The distribution shows that there are two main modes in the set: one mode located at around 0 radians (corresponding to wings orthogonal to the fuselage) and another mode centered at 0.7 radians (wings bent by 40 degrees). By clicking on locations of the two modes, the system presents the corresponding shapes. In Figure 5 (a), the user selected a unary relation for the chair legs, corresponding to the extent of the vertical axis of the legs in relation to the size of the entire shape. With the tool, the user can identify the three modes that exist and display a few representative shapes. In (b), the scale difference between the second axis of the seat and back is selected (corresponding to the chair width). We observe that from the point of view of this relation, the chairs exhibit larger variation. The user can form an understanding of the groups by inspecting a few shapes for each mode. Additional examples of exploration are shown in the supplementary video.

These examples show an advantage of the per-relation exploration: instead of pre-selecting a single relation and clustering the shapes into a few representative groups, the user is able to explore the set according to the relation that is relevant for a given task (e.g., finding bar chairs with tall legs or jet airplanes with bent wings). This allows the user to learn about the different shape variations that exist in the set (regarding the specific relation selected), as well as how prevalent they are (as implied by the shape of the distribution). Additionally, nearest neighbors can be retrieved according to a specific relation, providing models in similar geometric configurations.

Representative abstraction: The meta-representation can also be utilized to enhance exploration by automatically creating a repre-

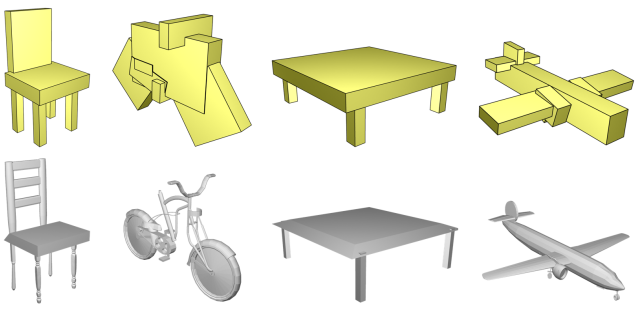


Figure 6: *Top row: representative abstract configurations automatically obtained from the meta-representations of families of chairs, bikes, tables, and planes. Bottom row: corresponding representative shapes synthesized for the abstract configurations.*

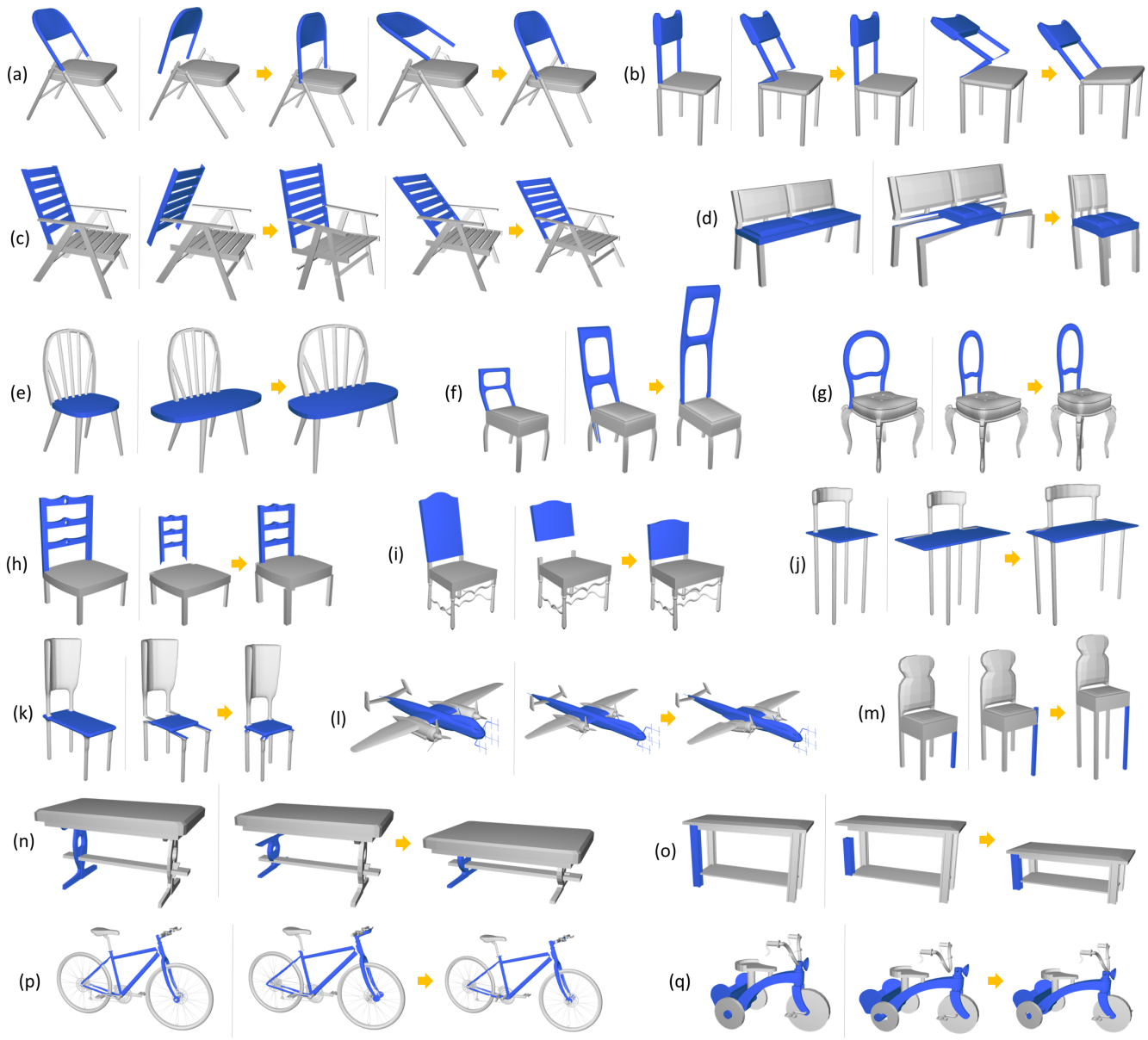


Figure 7: *Gallery of editing results guided by the meta-representation: each example shows the original shape and one or more edits where the user rotated or scaled one part. The shapes optimized according to the meta-representation are shown after the arrows.*

sentative box configuration and shape for a given shape family. The abstraction is created as follows: given a set of shapes, we first analyze the existence and number of instances of each semantic part. Since each shape differs in this aspect, we only retain semantic parts that exist in σ or more of the shapes (we use $\sigma = 0.5$). The number of instances of a semantic part is based on what is most common across the set (e.g., common number of chair legs). We determine the size properties of each part by querying the PDF of each extent relation for the highest probability mode. In order to infer the relative rotations between the parts, we employ an approach similar to the propagation method described in Section 5.2. Here, however, as we aim to create an abstraction that is most indicative of the set, we consider pure probability in place of probability gain. Using the PDFs for all angle relations, we compute the highest probability mode for each relation, and each pair of parts, and use it to guide the propagation. Finally, we connect the parts based on an example shape for which the contact relations are maximized. This provides a configuration of boxes that represents a family of shapes. In order to create representative shapes, we find the parts that are closest in size and configuration to each box in the abstraction and combine them to form a shape, while deforming the parts to fit the abstract configuration. See Figure 6 for representative abstractions and shapes computed for four sets. Note that the representative abstraction is derived directly from the meta-representation and does not assume any seeding model.

6.2 Guided shape editing

Figure 7 displays results of guided editing obtained with the progressive solution described in Section 5.2, where the user deformed selected parts of the shapes and the guided editing tool then returned shapes optimized according to the meta-representation. Note that

the examples in (a), (c), (e), (g), (h), and (i), were optimized with the meta-representation extracted from the small set of chairs, while the other edits were guided by the large set of 400 chairs.

We identify two improvements conveyed by the progressive optimization in these examples. First, if the configuration of parts created by the user is less common for the set, the system deforms the shape to a configuration with higher probability. This can be seen in the first example in (b), where although the user bent the back of the chair, the system rotated it back to a vertical position, as not many chairs have that specific inclined angle in the set. On the other hand, if the back is deformed to a larger angle, the system keeps the shape in this state, as there are more chairs with this configuration (also see Figure 1). Secondly, although the edited part becomes disconnected from the rest of the shape after the user edit, it is again connected to the shape as enforced by the contact point relations.

More importantly, if the editing constraints were derived from a single shape, we could incorrectly assume certain semantics, e.g., that the back and seat or wing and fuselage in Figure 1 need to remain orthogonal. In this regard, the meta-representation adds guidance with flexibility, by allowing more deformation freedom where the family supports it, or constraining the shape if it does not. Figure 8 further exemplifies this fact: although a chair is severely deformed, having lost its defining characteristics, the guided editing system is able to correct it, showing the effect of the *chair prior* that is present in the meta-representation.

Figure 8: *Correcting a chair with a severe deformation using the meta-representation.*

Figure 9 shows a sequence of three editing operations applied on the same shape to illustrate the function of the guided editing tool. We can see how after each edit, multiple relation values are taken to lower probability states. These are then restored to higher probability states by the optimization.

6.3 Coupled shape editing

An example of coupled editing is shown in Figure 10. The user directly manipulates the distribution of angles between the first axes of chair seat and legs, to obtain a simplified curve with less variation in the angles. Next, all the shapes are automatically deformed to conform the set to the new distribution. Thus, all the inclined legs become straight (forming an angle of 90 degrees with the seat), as angles that deviate too much from 90 degrees have a probability of zero in the new distribution.

As discussed in Section 5.3, this is a useful application when the goal is to collectively edit a set so that it satisfies specific geometric configurations. The overall variation that exists in a relation can be reduced by manipulating the distributions, or outlier models can be forced to conform to a specific range of values by removing their data points from the PDFs. Thus, the distributions act as a higher-level representation of the entire set, and the user is able to impact the geometry of several shapes by manipulating this representation.

7 Conclusions

We introduced a meta-representation to capture the essence of part configurations in a family of shapes. This is accomplished with a system of distributions of unary and binary relations of shape parts, which can then be used for applications such as repository exploration, guided shape editing, and coupled editing of a set.

Limitations and future work. With the system of distributions learned from a family of shapes, we effectively model shape validity in terms of probability, where the probabilities are derived from the frequency of part configurations in the set. This is appropriate for an exploration tool, as we are interested in exploring common styles in the set. The distributions can also capture some of the shape semantics if the set contains a representative sample of shapes from the family. In this context, there are some directions for introducing additional semantics into the representation. One possibility is to allow the user to add *forbidden regions* to the distributions, implying that part configurations with the corresponding relation values should not exist in the set, potentially allowing asymmetric distributions. In this way, the user can provide additional semantic information. Ultimately, the user could also design the distributions by manually drawing the PDFs to imply the semantics of the set.

Regarding our current model, we introduced a criterion for selecting a satisfactory bandwidth for each distribution, however, user supervision could also help in determining the optimal bandwidth, yielding a more accurate system of distributions. Moreover, if a shape possesses $|\mathcal{P}|$ parts, the representation consists of $\binom{|\mathcal{P}|}{2} \times |\mathcal{R}|$ curves. Thus, another direction for future work is to design a more compact representation. This has to be balanced with the fact that,

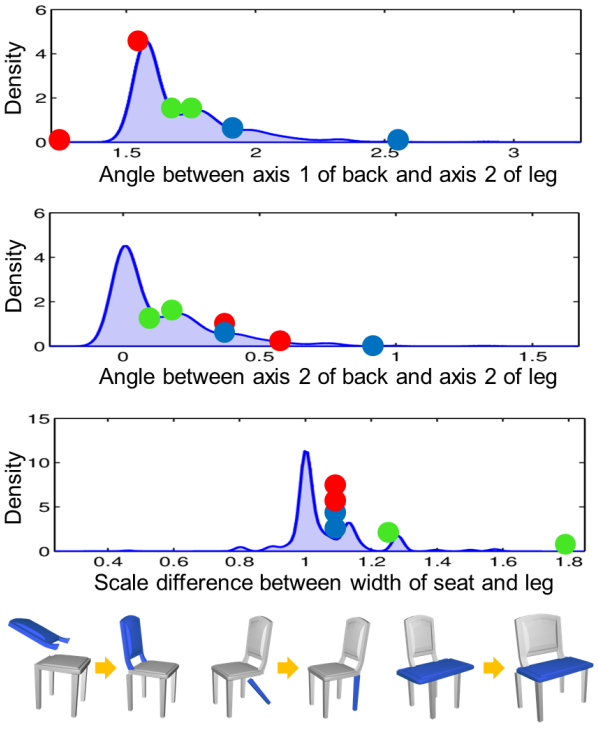


Figure 9: *Guided editing tool. Bottom: a sequence of three edits. Top: three relations and the values corresponding to the parts involved in the edits are shown before and after optimization (blue is the first edit, red is the second, and green is the third).*

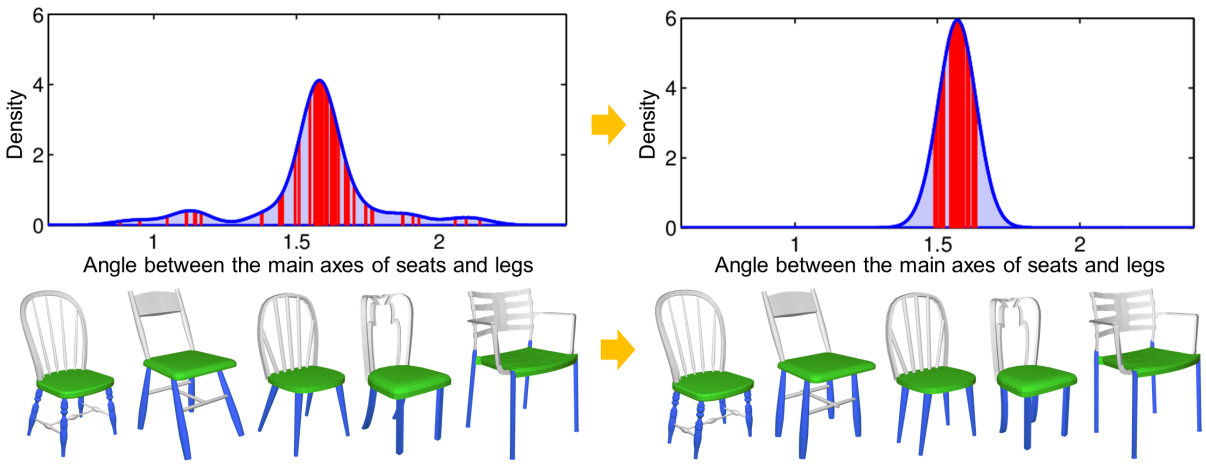
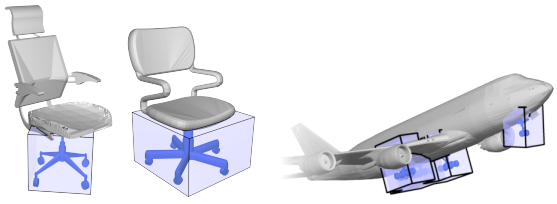


Figure 10: *Coupled editing of a family of shapes obtained with the meta-representation: the distribution on the left (angle in radians between the main axes of seats and legs) is directly manipulated by a user, who changes the curve to acquire the more compact profile on the right. As a result, all the models in the set are automatically deformed to conform to the new distribution (bottom row).*

as we are able to obtain more data, we would also seek to learn the correlations in the sets, possibly requiring a more complex model.

Furthermore, although the OBB construction described in Section 4 is stable across many shapes, it does not always yield optimal boxes, leading to inconsistent results in the presence of ambiguities. For example, the legs of the swivel chairs in Figure 11(a) cannot be oriented consistently based only on their symmetries, while the boxes computed for the landing gears in Figure 11(b) are also incorrectly aligned due to the geometry of the parts' convex hulls. One possibility for circumventing these problems could be to incorporate an algorithm that learns to predict the consistent box orientation from the orientation of other parts in the same shape. Another alternative is to replace the part primitives with other types of proxies, such as curves or sheets that do not need alignment.



(a) (b)

Figure 11: *Inconsistencies in the alignment of different OBBs.*

In terms of our implementation choices, different sets of relations can be used with the meta-representation, as we experimented only with one specific set. For shape editing, currently the ordering in which parts are deformed is determined by a probability gain criterion, not by a semantic ordering of parts. For example, the stabilizer of a plane can have more impact on wings than the fuselage. Thus, more sophisticated criteria can be developed to determine such ordering. Also, the unary relations are currently not taken into account when optimizing the shapes with the progressive solution, as they are not derived from the positions of other parts. Thus, these may also be added to the heuristic solution. Finally, one more direction for future work is to use the meta-representation directly for shape synthesis. This could be accomplished by appropriately introducing variation in the relation values of the shapes, according to an existing set of distributions or user-designed PDFs.

Acknowledgements. We thank the reviewers for their comments and suggestions for improving the paper. This work was supported in part by the Israeli Science Foundation (grant no. 1790/12), the U.S.-Israel Binational Science Foundation (grant no. 2012376), the Marie Curie Career Integration Grant 303541, the ERC Starting Grants iModel (StG-2012-306877) and SmartGeometry (StG-2013-335373), and gifts from Adobe Research. Oliver van Kaick is grateful to the Azrieli Foundation for the award of an Azrieli Fellowship. Melinos Averkiou is grateful to the Rabin Ezra Scholarship Trust for the award of a post-graduate bursary.

References

AVERKIOU, M., KIM, V. G., ZHENG, Y., AND MITRA, N. J. 2014. Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. *Computer Graphics Forum (Eurographics) 33*.

BOKELOH, M., WAND, M., AND SEIDEL, H.-P. 2010. A connection between partial symmetry and inverse procedural modeling. *ACM Trans. Graph (SIGGRAPH) 29*, 4, 104:1–10.

BOTSCH, M., AND SORKINE, O. 2008. On linear variational surface deformation methods. *IEEE Trans. Vis. & Comp. Graphics 14*, 1, 213–230.

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

CHIU, S.-T. 1996. A comparative review of bandwidth selection for kernel density estimation. *Statistica Sinica 6*, 1, 129–145.

FISHER, M., SAVVA, M., AND HANRAHAN, P. 2011. Characterizing structural relationships in scenes using graph kernels. *ACM Trans. Graph (SIGGRAPH) 30*, 4, 34:1–12.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph (SIGGRAPH) 23*, 3, 652–663.

GAL, R., SORKINE, O., MITRA, N. J., AND COHEN-OR, D. 2009. iWIRES: An analyze-and-edit approach to shape manipulation. *ACM Trans. Graph (SIGGRAPH) 28*, 3, 33:1–10.

HU, R., FAN, L., AND LIU, L. 2012. Co-segmentation of 3D shapes via subspace clustering. *Computer Graphics Forum (SGP) 31*, 5, 1703–1713.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. *ACM Trans. Graph (SIGGRAPH Asia) 30*, 6, 125:1–12.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. 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. Graph (SIGGRAPH) 31*, 4, 55:1–11.

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

LAGA, H., MORTARA, M., AND SPAGNUOLO, M. 2013. Geometry and context for semantic correspondences and functionality recognition in man-made 3D shapes. *ACM Trans. Graph 32*, 5, 150:1–16.

LI, G., LIU, L., ZHENG, H., AND MITRA, N. J. 2010. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. Graph (SIGGRAPH Asia) 29*, 6, 152:1–10.

LIN, J., COHEN-OR, D., ZHANG, H. R., LIANG, C., SHARF, A., DEUSSEN, O., AND CHEN, B. 2011. Structure-preserving retargeting of irregular 3D architecture. *ACM Trans. Graph (SIGGRAPH Asia) 30*, 6, 183:1–10.

MENG, M., XIA, J., LUO, J., AND HE, Y. 2013. Unsupervised co-segmentation for 3D shapes using iterative multi-label optimization. *Computer-Aided Design 45*, 2, 312–320.

MITRA, N. J., WAND, M., ZHANG, H., COHEN-OR, D., AND BOKELOH, M. 2013. Structure-aware shape processing. In *Proc. Eurographics State-of-the-art Reports*.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph (SIGGRAPH) 30*, 4, 33:1–10.

SCHNEIDER, P. J., AND EBERLY, D. H. 2003. *Geometric Tools for Computer Graphics*. Morgan Kaufmann, San Francisco.

SHTOF, A., AGATHOS, A., GINGOLD, Y., SHAMIR, A., AND COHEN-OR, D. 2013. Geosemantic snapping for sketch-based modeling. *Computer Graphics Forum (Eurographics) 32*, 2, 245–253.

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. Graph (SIGGRAPH Asia) 30*, 6, 126:1–10.

SILVERMAN, B. W. 1986. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Symp. Geometry Processing*, 109–116.

SUMNER, R. W., ZWICKER, M., GOTSMAN, C., AND POPOVIĆ, J. 2005. Mesh-based inverse kinematics. *ACM Trans. Graph (SIGGRAPH) 24*, 3, 488–495.

VAN KAICK, O., XU, K., ZHANG, H., WANG, Y., SUN, S., SHAMIR, A., AND COHEN-OR, D. 2013. Co-hierarchical analysis of shape structures. *ACM Trans. Graph (SIGGRAPH) 32*, 4, 69:1–10.

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. Graph (SIGGRAPH Asia) 31*, 6, 157:1–10.

XU, W., WANG, J., YIN, K., ZHOU, K., VAN DE PANNE, M., CHEN, F., AND GUO, B. 2009. Joint-aware manipulation of deformable models. *ACM Trans. Graph (SIGGRAPH) 28*, 3, 35:1–9.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Trans. Graph (SIGGRAPH) 31*, 4, 57:1–10.

YANG, Y.-L., YANG, Y.-J., POTTMANN, H., AND MITRA, N. J. 2011. Shape space exploration of constrained meshes. *ACM Trans. Graph (SIGGRAPH Asia) 30*, 6, 124:1–124:12.

YUMER, M. E., AND KARA, L. B. 2012. Co-abstraction of shape collections. *ACM Trans. Graph (SIGGRAPH Asia) 31*, 6, 166:1–11.

ZHENG, Y., FU, H., COHEN-OR, D., AU, O. K.-C., AND TAI, C.-L. 2011. Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum (Eurographics) 30*, 2, 563–572.

ZHENG, Y., COHEN-OR, D., AND MITRA, N. J. 2013. Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum (Eurographics) 32*, 2, 195–204.

ZHENG, Y., COHEN-OR, D., AVERKIOU, M., AND MITRA, N. J. 2014. Recurring part arrangements in shape collections. *Computer Graphics Forum (Eurographics)*.