

Structure-adaptive Shape Editing for Man-made Objects

Qiang Fu¹, Xiaowu Chen^{†1}, Xiaoyu Su¹, Jia Li^{1,2}, Hongbo Fu³

¹State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University

²International Research Institute for Multidisciplinary Science, Beihang University

³ School of Creative Media, City University of Hong Kong

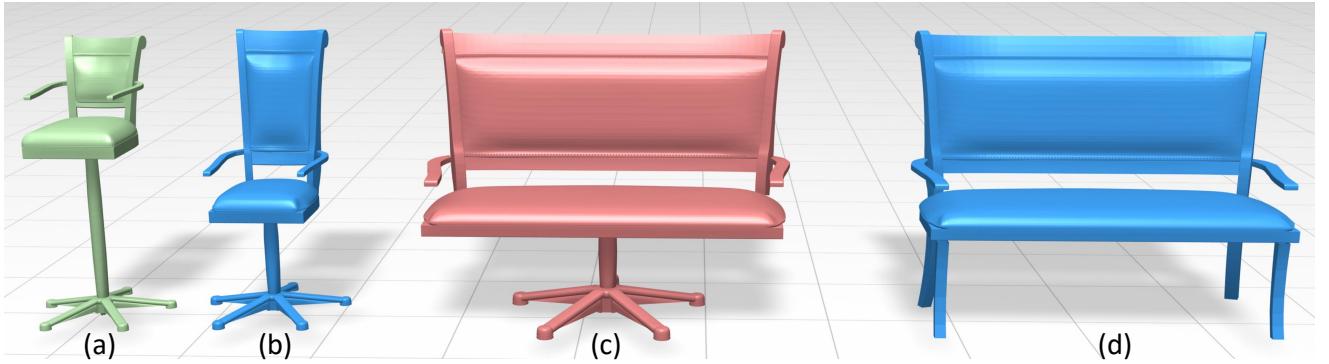


Figure 1: Given a man-made object (a), structure-preserving shape editing produces shape variation (b). In some scenarios, when the original structure does not match the shape variation (c), structure-varying shape editing might be more preferred (d). Our structure-adaptive editing tool automatically decides whether a structure should be preserved (b) or changed (d).

Abstract

One of the challenging problems for shape editing is to adapt shapes with diversified structures for various editing needs. In this paper we introduce a shape editing approach that automatically adapts the structure of a shape being edited with respect to user inputs. Given a category of shapes, our approach first classifies them into groups based on the constituent parts. The group-sensitive priors, including both inter-group and intra-group priors, are then learned through statistical structure analysis and multivariate regression. By using these priors, the inherent characteristics and typical variations of shape structures can be well captured. Based on such group-sensitive priors, we propose a framework for real-time shape editing, which adapts the structure of shape to continuous user editing operations. Experimental results show that the proposed approach is capable of both structure-preserving and structure-varying shape editing.

1. Introduction

Shape editing, which aims to generate new variations of existing shapes via a moderate amount of user interactions, has been extensively studied in the past decades. The recent efforts have been mainly put to the design of editing frameworks for man-made objects, which often exhibit rich structural information.

Among the existing shape editing approaches for man-made objects, some of them [GSMCO09, ZFCO*11] leverage the geomet-

rical characteristics of an input shape as constraints to optimize the edited shapes, while others [FAvK*14, YK14] are proposed to extract editing constraints from a shape collection via statistic analysis. However, the edited shapes by most of these approaches are structure-preserved. That is, editing manipulation on a given model leads to continuous shape variations, while the shape structure, namely the constituent parts of the model, keeps unchanged during editing. On the other hand, many shape synthesis approaches [SFCH12, KCKK12] focus on the creation of structure variations via a parts replacement. This inspires us that the shape structure, in many real-world scenarios, needs to be varied in the editing process

[†] Corresponding author (email:chen@buaa.edu.cn)



Figure 2: Chairs in three structure groups with different colors.

(*i.e.*, structure-varying editing, see Figure 1). Therefore, it is necessary to develop a unified framework that supports both structure-preserving and structure-varying shape editing.

To address this problem, this paper introduces a novel approach for structure-adaptive shape editing. Given a dataset of pre-segmented shapes, with editable parts and sufficient structure variations, we cluster the editable parts by their spatial positions and enumerate all part types in each shape category. As a consequence, the structure of each shape can be represented by a binary vector with each component indicating whether the shape contains a specific type of part. We define a distance metric to measure the similarity of structures, with which each shape category is divided into multiple structure groups by using spectral clustering (Figure 2). Based on these groups, we learn the inter-group and intra-group priors through statistical structural analysis and multivariate regression, respectively. Such group-sensitive priors enable an interactive tool for structure-adaptive shape editing. This is achieved through a two-step approach: after the user edits a certain part, the inter-group prior is first employed to determine a proper structure for the edited shape, and then all parts of the shape are refined according to the intra-group prior. Experimental results show that the proposed editing framework is simple and easy to use, and can be used to create interesting shape- and structure-variations.

Our work has made two main contributions: 1) we propose a unified framework that adaptively combines both structure-varying and structure-preserving shape editing; 2) we learn group-sensitive priors from each category of 3D models and use such priors for structure-adaptive shape editing.

2. Related Work

We focus our review on existing editing techniques for man-made objects.

Shape manipulation. Several methods have been proposed to facilitate the manipulation of man-made objects, especially for structure-aware shape processing [MWZ^{*}13]. The analyze-and-edit approaches like *iWIRES* [GSMCO09] and *Component-wise Controllers* [ZFCO^{*}11] used the relations between shape parts or features to constrain the shape during editing. Yang et al.

[YXG^{*}13] proposed a framework for multidomain subspace deformation. Lin et al. [LCOZ^{*}11] performed structure-preserving re-targeting of irregular 3D architecture models by decomposing the input model into a set of 1D structures. Shao et al. [SLZ^{*}13] used shape editing to interpret concept sketches. However, the structure in these approaches is preserved by the characteristics from the single input shape, but not inferred from a set of shapes as our goal.

Recently a few approaches like *Meta-representation* [FAvK^{*}14] and *Co-Constrained Handles* [YK14] have been proposed to leverage a category of database shapes to learn the deformation constraints for shape manipulation. These data-driven shape editing approaches are most related to our work. Their pairwise parameter constraints perform well in structure-preserving editing. However, they mainly focus the parameter pairs in the same kind (*e.g.*, the width of backrest and the width of seat), possibly missing some latent relations between different kinds of parameters (*e.g.*, the rotation of backrest and the length of seat). In contrast, our work employs multivariate regression to reveal the richer relations within different kinds of parameters associated with a shape category. Besides, compared to all the above works, our work supports structure-adaptive shape editing, enabling the creation of more variations that are structurally valid.

Analysis of shape collection. With the growing abundance of 3D shape collections, many approaches have been proposed to shape co-analysis to get semantic parts and even their structure hierarchy. Strategies, such as spectral clustering [SvKK^{*}11], subspace clustering [HFL12], active analysis [WAvK^{*}12], sparse reconstruction [XSX^{*}14], have been employed in shape segmentation, which is of great benefit to our work. Some other literatures such as [WXL^{*}11, VKXZ^{*}13] focused on the hierarchical structure of man-made shapes, and inspired us to handle the structure types with the constituent parts. Besides, Xu et al. [XLZ^{*}10] proposed style-content separation by analyzing the objects at the part level. Ovsjanikov et al. [OLGM11] presented a method for discovering and exploring continuous variability in a shape collection. Huang et al. [HSS^{*}13] presented a method for organizing a heterogeneous collection of 3D shapes for overview and exploration. Kim et al. [KLM^{*}13] produced a set of probabilistic part-based templates that capture the styles and variations in a category of 3D shapes. Averkiou et al. [AKZM14] directly parameterized the template space for coupled shape exploration and synthesis. In our work, we use the shape’s constituent parts to encode the structure, and cluster the shapes into several groups to learn the priors for structure-adaptive shape editing.

Assembly-based modeling. There have also been many works on assembly-based modeling, which automatically synthesizes novel shapes by recombining a set of part-based shapes. Funkhouser et al. [FKS^{*}04] proposed a *Modeling by Example* system to form new objects with retrieved source models. Kreavoy et al. [KJS07] used common component structures to create new models by exchanging parts. Shen et al. [SFCH12] presented a bottom-up structure recovery approach based on part assembly with respect to raw depth data. A *Sketch-to-Design* system [XXM^{*}13] was proposed to design models guided by user sketches. *Smart Variations* [ZCOM13] was a purely geometric approach based on functional substructures for part reshuffling. Alhashim et al. [ALX^{*}14] introduced an algorithm

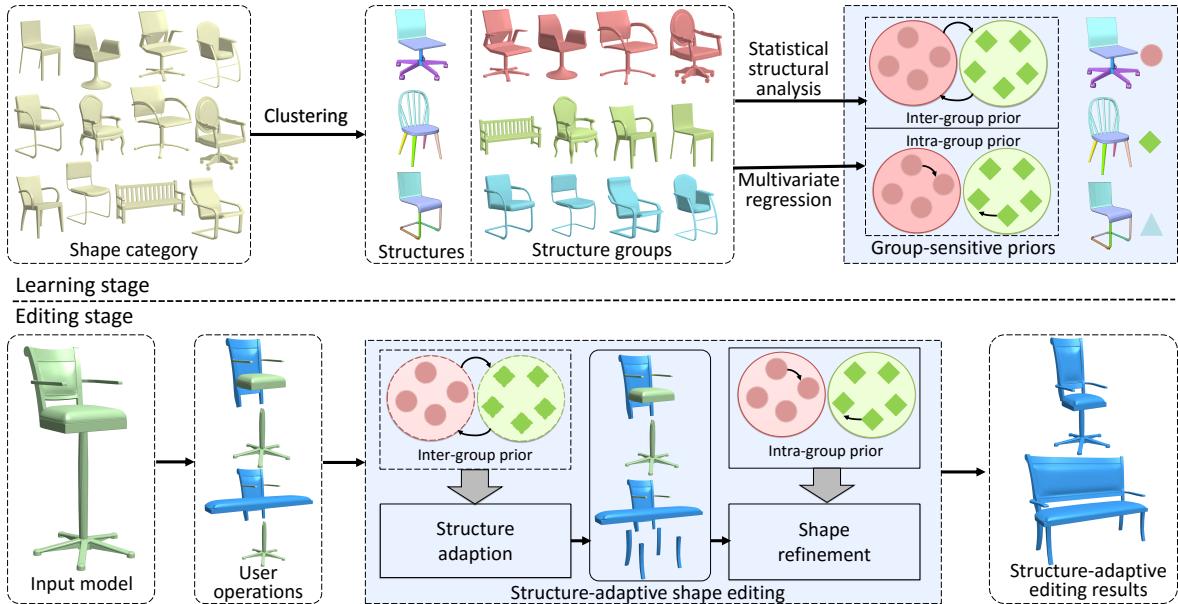


Figure 3: Overview of our framework. Two major stages are involved in the proposed editing framework, including a learning stage (offline) and an editing stage (online). The learning stage learns group-sensitive priors that describe the shape and structure variations among a category of shapes, while the editing stage employs such priors to adaptively refine the shape and/or structure driven by user editing operations. Note that in the given cases of shape editing, the original parts are highlighted in green, and the edited parts are highlighted in blue.

for generating novel 3D models via topology-varying shape blending. Chaudhuri et al. [CKG^{*}13] presented an approach for users to create visual content using relative semantic attributes expressed in linguistic terms. Schulz et al. [SSL^{*}14] proposed a data-driven method for designing actual physical objects that can be fabricated. Another way to shape synthesis is to learn the probabilistic models from a shape repository such as [CKGK11, KCKK12]. In our paper, when the structure of a shape needs to be changed, a part replacement step is employed with the proper new parts suggested based on the relevant existing parts, and a deformation optimization step is then employed to ensure all parts well connected.

3. Overview

In this work we propose an interactive shape editing approach that automatically adapts the structure of a shape being edited with respect to user edits. The same as [SFCH12], we focus on four categories of man-made objects, including chairs, airplanes, tables and bicycles. Most models are from the datasets in [SFCH12]. We also insert some new models like benches and tricycles into the datasets to enrich each category. As a preprocessing step, the centers, scale, and front and upright orientations of all shapes in the same category are normalized. Every model in each category is also decomposed into several meaningful parts, which can be edited by users. This can be achieved by using the state-of-the-art segmentation techniques [ARSF09, WAvK^{*}12]. Note that, we expect each shape to be adequately segmented to provide meaningful editable parts, and

it is not necessary to consistently or semantically label each part in the preprocessing step.

As illustrated in Figure 3, our framework consists of two major stages: an offline learning stage and an online editing stage. In the learning stage, shapes in each category are clustered into multiple groups according to their structures (i.e., the constituent parts). In the clustering process, a binary vector structure descriptor is first extracted to characterize the constituent parts of each shape. Based on such descriptors, a distance metric is then defined to measure the similarity of two structures so that each category of shapes can be divided into groups via spectral clustering [LZ04]. As a consequence, the shapes in the same group have similar structures, while the shapes in different groups have remarkably different structures (see Figure 2). Given these groups, two kinds of prior knowledge are learned through statistic analysis of structure and multivariate regression, respectively. These priors capture the inherent characteristics of inter- and intra-group variations of shapes, which will be used as the guidance for shape editing.

In the editing stage, each part of the input 3D model is first represented by an OBB (Oriented Bounding Box) by using the approach in [FavK^{*}14]. Each OBB is characterized by 9 parameters, including the 3D scale, position and orientation (see Figure 4 for OBBs and their parameters). Based on these OBBs, various user editing operations, including adding, removing, translating, rotating and scaling, can be applied to the OBBs so as to edit the input shape. The structure of the edited shape might be inappropriate after various editing operations (see Figure 1 (c)). Then a structure-adaptive

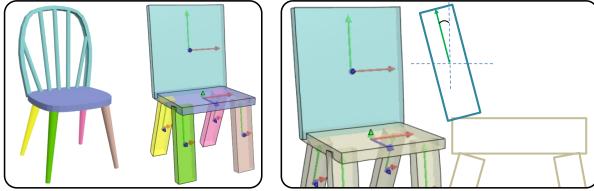


Figure 4: Each part of the shape is represented by an oriented bounding box (OBB). The part’s scale and position parameters are respectively described by the OBB’s axial lengths and the central position (relative to the model center), and the rotation parameters are described by the projection angle between the OBB’s axis and global coordinate axis as shown in right.

shape editing process is employed to automatically refine both the structure and the geometry of the edited shape. In this process, structure adaption is first applied to determine a proper structure for the edited shape. After that, a shape refinement step is employed to optimize the geometry of all parts based on the determined structure to obtain visually pleasing models.

To achieve structure-adaptive shape editing, we need to address two key problems:

- How to learn the group-sensitive priors that capture the geometric and structural variations of shapes in the same category?
- How to adaptively edit shapes with the assistance of the learned group-sensitive priors?

In the next two sections, we will show our detailed solutions to these two problems.

4. Learning Group-sensitive Priors

In this section we first introduce how to cluster shapes into groups according to their structures. We then describe how to learn the group-sensitive priors, namely the inter-group prior which describes how to change the structure of the edited shape from one group to another group, and the intra-group prior which describes how to refine the edited results inside a shape group.

4.1. Clustering Structure Groups

The component-based structure of shapes has been studied via hierarchical analysis [VKXZ^{*}13] or probabilistic model [KCKK12]. In our implementation, all input models have been adequately segmented to provide meaningful editable parts. We attempt to cluster structure groups with the components not semantically labeled.

Our key observation is that the similar parts are always position-dependent in man-made shapes, and some components cannot coexist in man-made objects due to the same function (e.g., the legs of a swivel-chair and a four-legged chair, the rear-wheel of a bicycle and a tricycle), which leads to the structural variations in a shape category. Therefore, we first cluster the parts into some types, and then use a distance metric to separate the shapes with incompatible types of parts into different structure groups.

	$x_i(1)$	$x_i(2)$	$x_i(3)$	$x_i(4)$	$x_i(5)$	$x_i(6)$	$x_i(7)$	$x_i(8)$	$x_i(9)$	$x_i(10)$	$x_i(11)$
	1	1	1	1	1	1	0	0	0	0	0
	1	1	0	0	0	0	1	1	0	0	0
	1	1	1	1	0	0	0	0	1	1	1

Figure 5: Structure descriptors (partial) of three representative shapes from different structure groups.

Since the shape categories in our experiments largely lie in a low-dimensional shape space defined with respect to the relative sizes and positions of shape parts [OLGM11], we first scale all shapes to a same size (i.e., making all shapes’ OBBs be the same size of cubes) and then perform a simple position-based clustering approach to establish part correspondence. Note that we heuristically choose the number of clusters by observing whether the different groups have different component parts. Alternatively, some advanced approaches like [KLM^{*}12, ZCAM14] can also be used to cluster the parts in a shape category. Let $\{\mathcal{S}_1, \dots, \mathcal{S}_K\}$ be the K models from a specific shape category (e.g., chairs). These shapes contain N types of parts which have been clustered, denoted as $\mathcal{P}_1, \dots, \mathcal{P}_N$. As a consequence, we can represent the structure of a shape \mathcal{S}_i with a binary vector \mathbf{x}_i with N components, where the n -th component $\mathbf{x}_i(n)$ equals to 1 if the shape \mathcal{S}_i contains the part \mathcal{P}_n , and 0 otherwise (Figure 5).

Then we need to define a distance metric $\phi(\cdot)$ between two structure descriptors \mathbf{x}_i and \mathbf{x}_j . To ensure that shapes with similar structures can be clustered into the same group, we define $\phi(\mathbf{x}_i, \mathbf{x}_j)$ as

$$\begin{aligned} \phi(\mathbf{x}_i, \mathbf{x}_j) = & \sum_{(n_1, n_2)} \mathbf{x}_i(n_1)(1 - \mathbf{x}_j(n_1))(1 - \mathbf{x}_i(n_2))\mathbf{x}_j(n_2) \\ & \cdot (1 - \max(\phi(n_1, n_2), \phi(n_2, n_1))), \end{aligned} \quad (1)$$

where the first term $\mathbf{x}_i(n_1)(1 - \mathbf{x}_j(n_1))(1 - \mathbf{x}_i(n_2))\mathbf{x}_j(n_2)$ is used to detect if both two structures have a specific component which does not belong to the other one. The conditional probability $\phi(n, n') = P(\mathbf{x}(n) = 1 | \mathbf{x}(n') = 1)$ is statistically calculated with all shapes in a category and indicates whether two types of parts can coexist or not. We introduce the second term to separate the shapes with the incompatible types of parts.

Based on such a distance metric, we employ the spectral clustering algorithm [LZ04] to divide the shapes in the same shape category into groups, denoted as $\{\mathbb{S}_m\}_{m=1}^M$. M is the number of groups, which is manually fine-tuned for each shape category to provide the best editing performance. See Figure 5 for an illustration of shapes in different groups and their structure descriptors. In Figure 6, we visualize all existing structures and shape examples for the three structure groups of chairs in our datasets. We can also broaden the scope of structures by adding new shapes into the existing datasets.

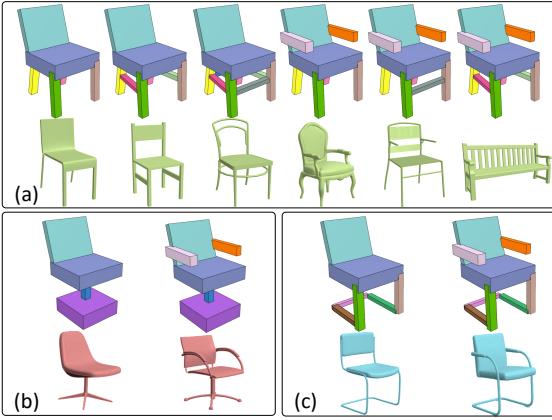


Figure 6: Structures and corresponding shape examples of the three structure groups ((a), (b) and (c)) in the chair dataset.

4.2. Learning Inter-group Prior

Shapes from two different groups may still share some common parts, e.g., the seat of chair, the body and wings of airplane, etc. If there exist large gaps between some parameters of such parts, these parameters might be used to control “switching” of the structures between two groups. This observation motivates us to model the inter-group prior to capture the structural variations between shapes in different groups.

Here we represent such prior knowledge with a set of $M \times M$ matrices, denoted as $\{\mathbf{T}_n^t | t = 1, \dots, 9\}_{n=1}^N$. The matrix \mathbf{T}_n^t describes whether the structure of a shape can be changed between any two of the M groups with respect to the t -th parameter for the OBB of part \mathcal{P}_n . That is, $\mathbf{T}_n^t(i, j)$ is a threshold obtained by the statistics of the database models, determining whether a shape in the group \mathbb{S}_i should be re-clustered into the group \mathbb{S}_j if the t -th parameter for the OBB of its part \mathcal{P}_n is changed during user editing operations.

Assume that \mathcal{P}_n is a part shared by shapes in both groups \mathbb{S}_i and \mathbb{S}_j . The sets of possible values for the t -th parameter of \mathcal{P}_n in \mathbb{S}_i and \mathbb{S}_j are defined as $\{\mathbb{B}_i | b_1^i, \dots, b_{N_i}^i\}$ and $\{\mathbb{B}_j | b_1^j, \dots, b_{N_j}^j\}$. We define $d(\mathbb{B})$ as the maximum distance within a group and $\Delta(b, \mathbb{B})$ as the point-to-set distance between parameter value b and set \mathbb{B} as follows:

$$d(\mathbb{B}) = \max_{i,j}(|b_i - b_j|), \text{ and } \Delta(b, \mathbb{B}) = \min_{b_i \in \mathbb{B}}(|b - b_i|). \quad (2)$$

With this distance definition, we use the averaged maximum distance on two groups $\lambda = \frac{d(\mathbb{B}_i) + d(\mathbb{B}_j)}{2}$ to estimate whether the t -th parameter of \mathcal{P}_n can be used as the switching parameter, and calculate the threshold $\mathbf{T}_n^t(i, j)$ for structure switching between groups \mathbb{S}_i and \mathbb{S}_j as

$$\mathbf{T}_n^t(i, j) = \begin{cases} \lambda & \text{if } \max_{b_n^j \in \mathbb{B}_j} \Delta(b_n^j, \mathbb{B}_i) > \lambda, \forall i \neq j \\ \infty & \text{otherwise} \end{cases}, \quad (3)$$

where $\max_{b_n^j \in \mathbb{B}_j} \Delta(b_n^j, \mathbb{B}_i) > \lambda$ indicates some shapes in group \mathbb{B}_j have a large gap for the t -th parameter of part \mathcal{P}_n . Such a prior will

be used as a threshold for structure-adaptive editing (Section 5.1). Note that \mathbf{T} depends on the numbers of clustered parts and structure groups. We will provide some statistics on \mathbf{T} matrices in Section 6.

4.3. Learning Intra-group Prior

Beyond the inter-group prior, we also learn the intra-group prior that describes how shapes can change inside each group. Different from the inter-group prior that focuses on the variation of structures, the intra-group prior emphasizes the geometric variations of shapes while the structure is preserved.

The geometric relationships between the parts of man-made objects can be leveraged as constraints for shape editing. Fish et al. [FAvK*14] proposed to encode the relations between part pairs, and used such relations to refine the edited models. Yumer et al. [YCHK15] used a set of semantic attributes learned by user study to create geometric deformations. We observe that some part relations such as symmetry and parallelism are linear. Therefore, we aim to use multivariate linear regression to learn the relations between different parts’ parameters in each structure group as the intra-group prior. Since the relations between some part parameters might not be linear, we use a weight based on the regression to alleviate the influence of such parameters (see more discussions in Section 5.2). Note that such relations also involve different types of parameters such as the relation between rotation and scale. For example, the rotation of a deck-chair’s backrest has a positive correlation with the length of the seat as a result of ergonomics design. In this way, some design knowledge of man-made shapes can be captured without being semantically predefined. However, there exist two problems to learn such a prior: 1) the constituent parts of each shape in one structure group are not always the same, e.g., some swivel-chairs have armrests while others may not, making it impossible to directly learn the regression coefficients with all member shapes as samples; 2) some parts’ parameters like the orientations of the chairs’ seats are nearly invariable, which might lead to inaccurate parts’ relationships captured by regression analysis.

To address these problems, we first select the parts which are not shared by all group members, and establish unitary regression between each parameter of such parts and each parameter of the parts shared by all group members. The parameter pair with the largest coefficient of determination of regression is chosen to estimate the parameter of the part which does not belong to a shape. For example, if the position of the armrest is strongly correlated to the width of the seat, we can use the seat which is shared by all shapes in a group, to estimate the position of the armrest for the shape which has no armrest (see Figure 7 top). Hence, all member shapes can be used as samples to learn the regression coefficients. Second, we ignore the invariable parameters by calculating the mean square errors. Such parameters would not be edited or influence other parts during the editing process (see Figure 7 bottom).

With the parts’ relations encoded by multivariate regression, we represent the intra-group prior with a set of regression coefficient matrices for each structure group, denoted as $\{\mathbf{A}^{(m)} | m = 1, \dots, M\}$. The matrix $\mathbf{A}^{(m)}$ consists of all the regression coefficients of all parts’ parameters $\mathbf{b} = [b_1, \dots, b_n]$ in the structure group \mathbb{S}_m . Let $\mathbf{a}_i = [\alpha_0, \dots, \alpha_n]$ denote the i -th row of $\mathbf{A}^{(m)}$, which is the regres-

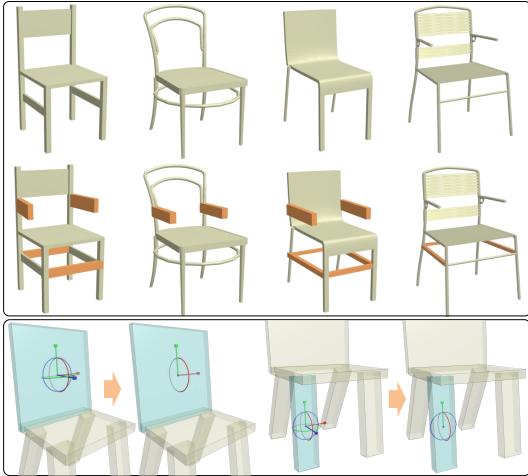


Figure 7: **Top:** for a set of shapes in the same structure group (top), we estimate the parameters of the parts (brown) which are not shared by all shape members, and show the OBBs with the estimated parameters (bottom). **Bottom:** two examples, in which we ignore the invariable parameters of scale and orientation.

sion coefficients of b_i . We use the following function to learn \mathbf{a}_i with all shape members of the structure group \mathbb{S}_m as samples:

$$\mathbf{a}_i = \arg \min_{\mathbf{a}_i} \|b_i - \alpha_0 - \sum_{j \neq i} \alpha_j \cdot b_j\|_2^2. \quad (4)$$

Not that, for all part parameters, some regression results which have low goodness of fits, should be filtered out when the intra-group regression prior is employed (Section 5.2). If the sample size (i.e., the number of shapes in a group) is less than the number of the part parameters, we take an approximation approach to learn the regression coefficients: we set $\alpha_j = 0$ for the parameter b_j that is irrelevant to b_i by calculating the correlation coefficient between b_j and b_i , until the number of the rest of part parameters is equal to the sample size.

5. Structure-adaptive Shape Editing

In this section we introduce how to apply the learned group-sensitive priors to structure-adaptive shape editing. We first use the inter-group prior for structure adaption to determine the valid structure of the edited shape, and then use the intra-group prior for shape editing and refinement.

5.1. Structure Adaption

Given an input model from a category of shapes already processed in the learning stage, the editing operations are performed by changing individual parameters associated with model parts. After a parameter b is edited, the edited shape might be re-clustered into another group through structure switching based on the inter-group prior \mathbf{T}_n^t (Equation 3). Let \mathbb{S}_i be the original structure group which the edited shape belongs to, and \mathbb{S}_j be a new group. To determine whether a shape should be re-clustered from \mathbb{S}_i into \mathbb{S}_j , we

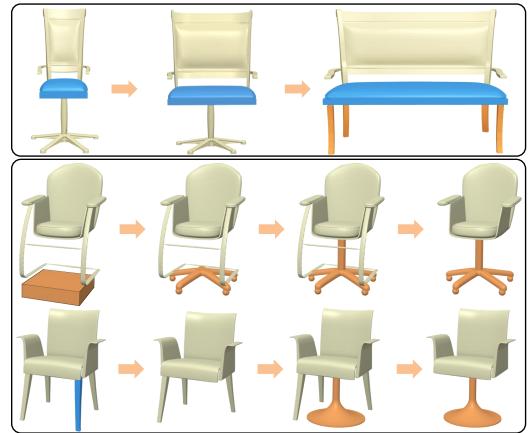


Figure 8: **Top:** structure preserved or changed with respect to the part parameter edited. **Bottom:** two cases of structure varying with new part explicitly added or original part removed. The edited parts are highlighted in blue, and the added box or introduced parts are highlighted in brown.

compute

$$K(b) = \begin{cases} 1 & \text{if } \Delta(b, \mathbb{B}_j) \leq \mathbf{T}_n^t(i, j) \text{ and } \Delta(b, \mathbb{B}_i) > \mathbf{T}_n^t(i, j) \\ 0 & \text{otherwise} \end{cases}, \quad (5)$$

where $K(b) = 1$ makes the structure changed from group \mathbb{S}_i to \mathbb{S}_j , while $K(b) = 0$ makes the original structure preserved. In this way, our approach automatically adapts the structure of the edited shape with the help of the inter-group prior learned from the same category of shapes. Take the case of structure switching shown in Figure 8 (Top) as an example, where the seat's width is edited. When it is widened too much (right), the original structure is switched to a bench's structure. Otherwise, the original structure is preserved (middle).

Besides, explicitly adding new parts by placing OBBs or removing existing parts might also trigger structure switching. That is, if the user adds a new part which does not belong to the current structure group, or removes a part which is indispensable to the current structure group, the edited shape would also be re-clustered into another group. For instance, as shown in Figure 8 (bottom), if the user adds a new box under the seat, our tool predicates that this box represents a swivel-chair's base, which does not belong to the current structure group, then the structure is switched by removing the four legs which does not belong to the new structure group, and introducing a gas lift to complete the new structure. Likewise, when the user removes one foreleg of a chair which is indispensable to that structure group, the original structure group turns to the one with only one base part.

A part suggestion step is necessary in this process. The new OBB added by the user is firstly labeled based on its position, and then we search for proper parts from the dataset, by calculating the difference of the scale between the added box and all parts' OBBs with the same label. For the missing parts which are indispensable

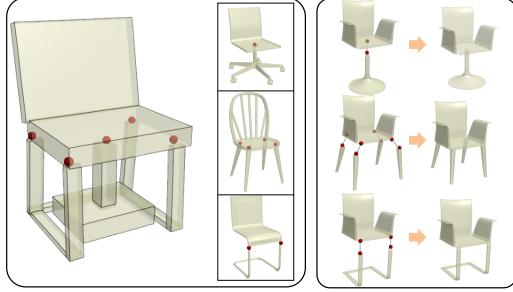


Figure 9: The seats of chairs from different structure groups share their contacts (left), hence the seat can be conjoined with different types of legs (right).

to the new structure, they are suggested by the similarity of scale to the remaining parts. For example, when a structure with one leg turns to another structure with four legs, we search for a four-leg chair with a similar seat and backrest to the edited shape to provide four legs for the edited shape.

5.2. Shape Editing and Refinement

When the structure of the edited shape is determined, a structure-preserving shape editing and refinement step is then performed to make the edited shape well connected and visually pleasing. The purpose of this step is to leverage the regression relation to refine the model with respect to user inputs.

For the structure group \mathbb{S}_m , we have obtained the intra-group prior or $\mathbf{A}^{(m)}$ as the constraints. In this manner, for all parts' parameters $\mathbf{b} = [b_1, \dots, b_n]$, assuming that the user adjusts a certain parameter b_c to B_0 , we obtain the parameters \mathbf{b} by solving the following equation using a least-square method:

$$\arg \min_{\mathbf{b}} \sum_{i=1}^n \|b_i - (\omega_i b'_i + (1 - \omega_i) \tilde{b}_i)\|_2^2 + \omega_c \|b_c - B_0\|_2^2, \quad (6)$$

where \tilde{b}_i is the original value of parameter b_i , and b'_i is the estimated value by the regression equation $b'_i = \sum_{j \neq i} \alpha_j \cdot b_j + \alpha_0$, whose coefficients $\mathbf{a}_i = [\alpha_0, \dots, \alpha_n]$ is obtained from the intra-group prior $\mathbf{A}^{(m)}$. ω_i is the coefficient of determination of this regression equation, as a weight to alleviate the influence of the parameters without linear relations having low goodness of fits. The term $\omega_c \|b_c - B_0\|_2^2$ is used to constrain the edited parameter to user inputs ($\omega_c = 10$ in our experiment). We show several editing results in Section 6 to demonstrate the effectiveness of our shape refinement with the intra-group regression prior.

Though our intra-group prior performs well in optimizing the scale, orientation and position of each part, some parts may not be well connected especially for the ones introduced by part replacement. To address this problem, we employ the contact-based deformation [ZLDM15] to conjoin those parts together. This method uses the contacts (in terms of 3D points) of each part as the constraints to enforce the contact relations to get a well connected shape. The parts with the same labels in different structure groups share their

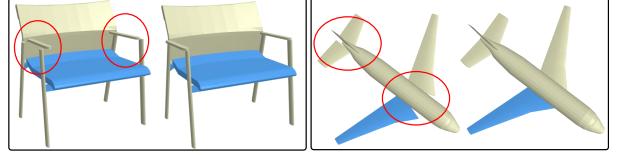


Figure 10: Two comparison examples with (right) and without (left) contact-based refinement. The contact-based deformation refines the position of adjacent parts (red circle).

contacts and the related parts, so that the introduced parts can easily find the contacts from the other parts in the original structure as illustrated in Figure 9. Note that we use such contact-based deformation to refine only the position of adjacent parts, and keep the scale and orientation of these parts unchanged. Figure 10 shows two comparison examples with and without contact-based refinement.

6. Experiments

In this section we first evaluate the efficiency of our framework, and then compare our approach with the state-of-the-art technique [FAvK*14]. Our dataset consists of four categories: chairs (64), airplanes (58), tables (39) and bicycles (20). Such man-made objects have interesting part-level characteristics such as symmetry and coaxially, and also have some structure characteristics like stability, functionality, etc., which should be considered during shape editing. Our experiments were conducted on an Intel Core i7-4790 3.60GHz PC with 16GB RAM. On average it took 7 seconds for part suggestion and replacement for structure-varying editing, and structure-preserving refinement ran in real-time.

Structure groups. Shapes in each of the four categories are gathered into 3 structure groups for chairs, airplanes and tables, and 2 groups for bicycles. We calculate the \mathbf{T} matrices only using the parts shared by any two groups as the thresholds for structure adaptation (Section 4.2). Note that, we also ignore the invariable parameters of the shared parts in this process. To better understand the \mathbf{T} matrices, we compute the ratios of the number of lambda values to the number of infinity in \mathbf{T} : 10/62 (chairs), 6/12 (tables), 2/38 (bicycles) and 1/62 (airplanes). Note that the \mathbf{T} matrices for the chair, table and airplane categories are 3×3 matrices and for bicycle 2×2 matrices. By ignoring the diagonal elements which are identical to infinity, we got the ratios of structure switching: 20.83% (chairs), 50% (tables), 10% (bicycles) and 2.38% (airplanes). It indicates that, for the categories like chairs and tables, the shared parts have obvious differences of part configurations, giving users more opportunities to change the original structure by editing the part parameters. Since the configurations of the shared parts are similar in categories like bicycles and airplanes, only a small number of part parameters can be used to change the original structure. Thus it might be easier for users to change the structure by adding new part or removing the existing part in these cases.

Editing results. We first give several results produced by our structure-adaptive shape editing tool. Figure 11 shows four results whose structures have no need to switch based on the inter-group prior, and Figure 12 shows six results whose structures are changed

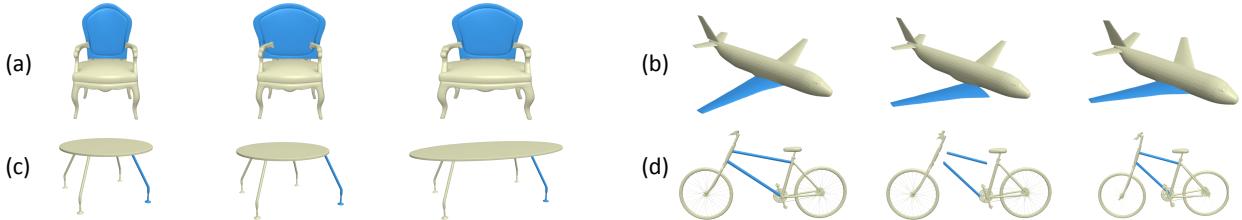


Figure 11: Gallery of shape editing examples with structure preserved. In each case, we show the input model (left), the model with user operation (middle) and the editing result (right). Note how the rest of the parts are changed with respect to the edited parts highlighted in blue.

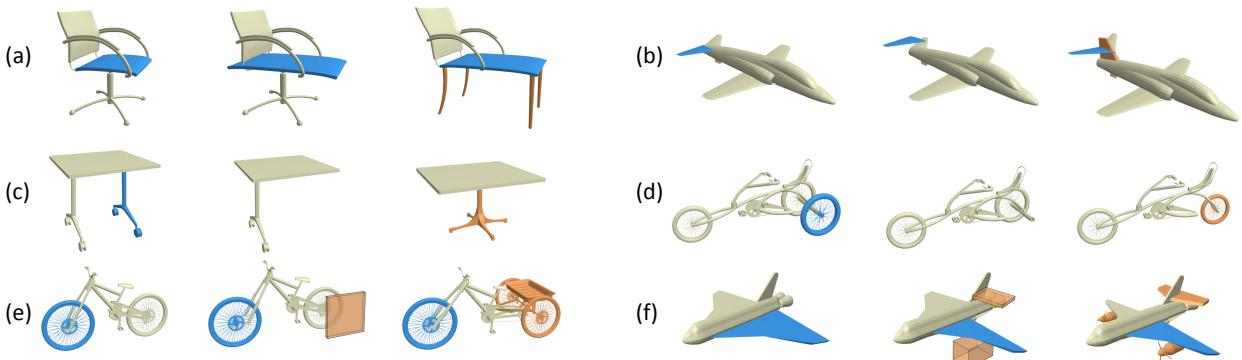


Figure 12: Gallery of shape editing examples with structures changed. The parts with user operations (scaling, translating or removing) are highlighted in blue, and the added boxes and the introduced parts are highlighted in brown. Note that how the structures of the input models are changed due to implicit or explicit structure editing operations.

Table 1: The number of operations and the time of both parameter editing and part adding/removing (in seconds) for each example.

Figure	11				12					
Example	(a)	(b)	(c)	(d)	(a)	(b)	(c)	(d)	(e)	(f)
Operations	1	1	1	1	1	1	1	1	2	3
Time (s)	4+0	5+0	3+0	2+0	5+0	4+0	1+0	1+0	14+5	25+10

due to adjusting one part’s parameter ((a) and (b)), removing an integrant part ((c) and (d)), or explicitly requesting new parts ((e) and (f)). Figure 12 (e) and (f) also show that our approach enables the creation of nontrivial variations with multiple simple manipulations (translating the front wheel of the tricycle, scaling the wing of the plane, and adding new boxes). Table 1 shows the number of operations (rotating, translating, scaling, adding and removing) and the timings of the above results with our approach. Most results were generated with only one user operation.

Figure 13 shows the individual effects of group-sensitive priors in structure-adaptive shape editing. Figure 13 (b) and (c) are the editing results using only the intra-group prior and only the inter-group prior, respectively. It suggests that such two priors should be used together in order to produce structurally valid, visually pleasing results (d).

Comparisons. We compared our method with [FAvK^{*}14] in a guided shape editing application. For fairness, the same level of

part segmentation was used. Both of the methods performed well in structure-preserving shape editing, while our approach was able to capture more complex relations. Four comparison examples are presented in Figure 14. The top two examples clearly show that our approach supports structure-adaptive shape editing, which extends the ability of structure-preserving shape editing. In the third example when rotating the backrest of a deck-chair, our intra-group prior refines the shape as well as stretching the length of the seat and shortening the legs making the chair more comfortable for the lying posture. The last example shows that when translating a bicycle’s front wheel forward, our approach rotates the suspension without changing the distance between the handle and the seat, to suit for the riding posture. Such interesting effects are impossible to achieve without using the learnt intra-group prior.

Limitations. Our approach relies on correct part clustering and requires good-quality pre-segmentation of repository models. All shapes in a category should be decomposed into meaningful parts

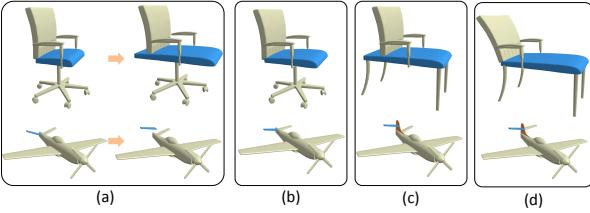


Figure 13: Editing a part of input models (a) using only the intra-group prior (b), only the inter-group prior (c), and both priors (d).

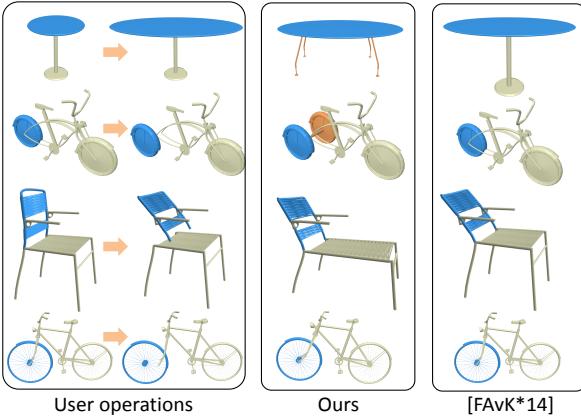


Figure 14: Comparisons with [FAvK*14].

in the same level, and a proper number of clusters should be chosen to ensure that different groups have different parts. Figure 15 shows a case where the too small number of clusters ignores the rich structure variations, making structure-adaptive editing impossible. Besides, since our approach employs the same method as [FAvK*14] to obtain part-level OBBs, we also meet the same problem of possibly incorrect alignment of OBBs in some cases like the base of the swivel chairs. In these cases, slight user assistance is helpful to improve the alignment of box’s orientation.

Our approach is largely based on the assumption that the structure variations within a shape category can be explained in terms of the constituent parts. Therefore, our approach might fail for certain man-made objects with permanent constituent like some appliances and tools (e.g., a television or a fork). Besides, since the structure groups in our approach depend on a conditional probability based distance metric, a shape category without the incompatible types of parts, needs user assistance to increase the distance between some parts. For example, no parts are incompatible in the category of airplane. We thus need to increase the distance between the horizontal and vertical tails to separate them into two groups (see Figure 16 left), and the shapes with both the horizontal and vertical tails are clustered into another group. Moreover, since the structure adaption needs obvious differences of part configurations between different structure groups, shapes with unusual configurations might invalidate the structure adaption. As illustrated in the right of Figure 16, if there exist several deck chairs with only one leg in the dataset,

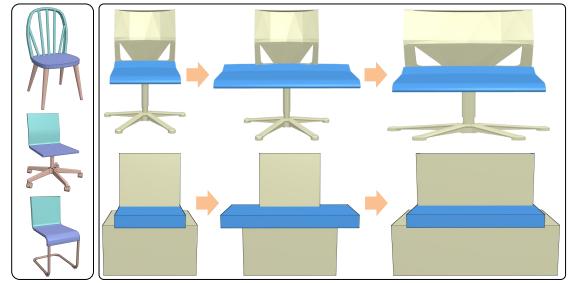


Figure 15: *Left:* clustering the parts of chair into only three groups will make all shapes have the same structure. *Right:* in this case, the structure cannot be changed adaptively and the regression constraint might not be properly derived.

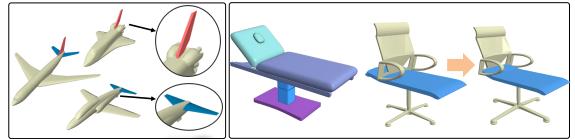


Figure 16: *Left:* user assistance is needed to increase the distance between the vertical tail (red) and the horizontal tail (blue) to separate the structures of the airplanes. *Right:* adding shapes with unusual configurations (e.g., the left deck chair) into the dataset will invalidate the structure adaption by lengthening the seat of chairs with one leg.

lengthening the seat will not change the original structure of the edited shape.

Furthermore, although our approach can reasonably well capture the statistical priors of shapes from a moderate number of models, too small datasets might make the multivariate regression sensitive to noisy data (e.g., by a strangely designed chair). It will also influence the group clustering performance when lacking of various structures. Our tool allows users to manually disable some regression constraints to alleviate the above problems for improving the editing performance.

7. Conclusions

We introduced a structure-adaptive shape editing for man-made objects. With the group-sensitive priors learned from the classified structure groups, we combine the structure-varying and structure-preserving editing into a unified framework. This is accomplished with an interactive shape editing tool, which is easy to use for non-professional users. We have evaluated our approach with several shape editing results, and compared it with the advanced data-driven shape editing technique. We demonstrate that our approach enables the creation of more variations that are structurally valid.

As a future work, we are interested in extending our approach to the inputs across different categories, which can not only increase the diversity of the editing results but also explore the universal editing regularity of man-made shapes especially for the parts with

similar function. With the growing accessibility of man-made objects, we expect this data-driven shape editing approach could enable more non-professional users to experience the creation and design benefited from the rich knowledge contained in the database.

Acknowledgements

We thank the anonymous reviewers for the constructive comments. This work was partially supported by grants from the the NSFC (61325011 & 61532003), Research and Development Program (2013BAH35F01), SRFDP (20131102130002) and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU113513, CityU11204014, and CityU11300615).

References

- [AKZM14] AVERKIOU M., KIM V., ZHENG Y., MITRA N. J.: Shapesynth: Parameterizing model collections for coupled shape exploration and synthesis. *Computer Graphics Forum*, 10 (2014). [2](#)
- [ALX*14] ALHASHIM I., LI H., XU K., CAO J., MA R., ZHANG H.: Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics* 33, 4 (2014), 158:1–158:10. [2](#)
- [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. [3](#)
- [CKG*13] CHAUDHURI S., KALOGERAKIS E., GIGUERE S., FUNKHOUSER T.: AttribIt: Content creation with semantic attributes. *ACM Symposium on User Interface Software and Technology (UIST)* (Oct. 2013). [3](#)
- [CKGK11] CHAUDHURI S., KALOGERAKIS E., GUIBAS L., KOLTUN V.: Probabilistic reasoning for assembly-based 3d modeling. *ACM Transactions on Graphics* 30, 4 (2011), 35:1–35:10. [3](#)
- [FAvK*14] FISH N., AVERKIOU M., VAN KAICK O., SORKINE-HORNUNG O., COHEN-OR D., MITRA N. J.: Meta-representation of shape families. *Transactions on Graphics* (2014). [1, 2, 3, 5, 7, 8, 9](#)
- [FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. *ACM Transactions on Graphics* 23, 3 (2004), 652–663. [2](#)
- [GSMC009] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: i-wires: an analyze-and-edit approach to shape manipulation. *ACM Transactions on Graphics* 28, 3 (2009), 33:1–33:10. [1, 2](#)
- [HFL12] HU R., FAN L., LIU L.: Co-segmentation of 3d shapes via subspace clustering. *Computer Graphics Forum* 31, 5 (2012), 1703–1713. [2](#)
- [HSS*13] HUANG S.-S., SHAMIR A., SHEN C.-H., ZHANG H., SHEFFER A., HU S.-M., COHEN-OR D.: Qualitative organization of collections of shapes via quartet analysis. *ACM Transactions on Graphics* 32, 4 (2013), 71:1–71:10. [2](#)
- [KCKK12] KALOGERAKIS E., CHAUDHURI S., KOLLER D., KOLTUN V.: A probabilistic model for component-based shape synthesis. *ACM Transactions on Graphics* 31, 4 (2012), 55:1–55:11. [1, 3, 4](#)
- [KJS07] KREAVOY V., JULIUS D., SHEFFER A.: Model composition from interchangeable components. In *in Proceedings of the Pacific Conference on Computer Graphics and Applications* (2007), pp. 129–138. [2](#)
- [KLM*12] KIM V. G., LI W., MITRA N. J., DiVERDI S., FUNKHOUSER T.: Exploring collections of 3d models using fuzzy correspondences. *ACM Transactions on Graphics* 31, 4 (2012), 54:1–54:11. [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. *Transactions on Graphics (Proc. of SIGGRAPH)* 32, 4 (2013). [2](#)
- [LCOZ*11] LIN J., COHEN-OR D., ZHANG H., LIANG C., SHARF A., DEUSSEN O., CHEN B.: Structure-preserving retargeting of irregular 3d architecture. *ACM Transactions on Graphics* 30, 6 (2011), 183:1–183:10. [2](#)
- [LZ04] LIU R., ZHANG H.: Segmentation of 3D meshes through spectral clustering. In *Proc. of Pacific Graphics* (2004), pp. 298–305. [3, 4](#)
- [MWZ*13] MITRA N., WAND M., ZHANG H. R., COHEN-OR D., KIM V., HUANG Q.-X.: Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses* (2013), pp. 1:1–1:20. [2](#)
- [OLGM11] OVSJANIKOV M., LI W., GUIBAS L., MITRA N. J.: Exploration of continuous variability in collections of 3d shapes. *ACM Transactions on Graphics* 30, 4 (2011), 33:1–33:10. [2, 4](#)
- [SFCH12] SHEN C.-H., FU H., CHEN K., HU S.-M.: Structure recovery by part assembly. *ACM Transactions on Graphics* 31, 6 (2012), 180:1–180:11. [1, 2, 3](#)
- [SLZ*13] SHAO T., LI W., ZHOU K., XU W., GUO B., MITRA N. J.: Interpreting concept sketches. *ACM Transactions on Graphics* 32, 4 (2013). [2](#)
- [SSL*14] SCHULZ A., SHAMIR A., LEVIN D. I. W., SITTHI-AMORN P., MATSIK W.: Design and fabrication by example. *ACM Transactions on Graphics* 33, 4 (2014), 62:1–62:11. [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. *ACM Transactions on Graphics* 30, 6 (2011), 126:1–126:9. [2](#)
- [VKXZ*13] VAN KAICK O., XU K., ZHANG H., WANG Y., SUN S., SHAMIR A., COHEN-OR D.: Co-hierarchical analysis of shape structures. *ACM Transactions on Graphics* 32, 4 (2013), 69:1–69:10. [2, 4](#)
- [WAvK*12] WANG Y., ASAIFI S., VAN KAICK O., ZHANG H., COHEN-OR D., CHEN B.: Active co-analysis of a set of shapes. *ACM Transactions on Graphics* 31, 6 (2012), 165:1–165:10. [2, 3](#)
- [WXL*11] WANG Y., XU K., LI J., ZHANG H., SHAMIR A., LIU L., CHENG Z.-Q., XIONG Y.: Symmetry hierarchy of man-made objects. *Computer Graphics Forum* 30, 2 (2011), 287–296. [2](#)
- [XLZ*10] XU K., LI H., ZHANG H., COHEN-OR D., XIONG Y., CHENG Z.: Style-content separation by anisotropic part scales. *ACM Transactions on Graphics* 29, 5 (2010), 184:1–184:10. [2](#)
- [XSX*14] XU W., SHI Z., XU M., ZHOU K., WANG J., ZHOU B., WANG J., YUAN Z.: Transductive 3d shape segmentation using sparse reconstruction. *Computer Graphics Forum* 33, 5 (2014), 107–115. [2](#)
- [XXM*13] XIE X., XU K., MITRA N. J., COHEN-OR D., SU Q., GONG W., CHEN B.: Sketch-to-design: Context-based part assembly. *Computer Graphics Forum* 32, 8 (2013), 233–245. [2](#)
- [YCHK15] YUMER M. E., CHAUDHURI S., HODGINS J. K., KARA L. B.: Semantic shape editing using deformation handles. *ACM Transactions on Graphics* 34, 4 (2015), 86:1–86:12. [5](#)
- [YK14] YUMER M. E., KARA L. B.: Co-constrained handles for deformation in shape collections. *ACM Transactions on Graphics* 33 (2014), 187:1–187:11. [1, 2](#)
- [YXG*13] YANG Y., XU W., GUO X., ZHOU K., GUO B.: Boundary-aware multidomain subspace deformation. *IEEE Transactions on Visualization and Computer Graphics* 19, 10 (2013), 1633–1645. [2](#)
- [ZCAM14] ZHENG Y., COHEN-OR D., AVERKIOU M., MITRA N. J.: Recurring part arrangements in shape collections. *Computer Graphics Forum* 33, 2 (2014), 115–124. [4](#)
- [ZCOM13] ZHENG Y., COHEN-OR D., MITRA N. J.: Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum* 32, 2pt2 (2013), 195–204. [2](#)
- [ZFCO*11] ZHENG Y., FU H., COHEN-OR D., AU O. K.-C., TAI C.-L.: Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum* 30, 2 (2011), 563–572. [1, 2](#)
- [ZLDM15] ZHENG Y., LIU H., DORSEY J., MITRA N. J.: Ergonomics-inspired reshaping and exploration of collections of models. *IEEE Transactions on Visualization and Computer Graphics* (2015). [7](#)