

Cross-class 3D Object Synthesis Guided by Reference Examples

Xiaoyu Su^a, Xiaowu Chen^{a,*}, Qiang Fu^a, Hongbo Fu^b

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

^b*City University of Hong Kong*

Abstract

Re-combining parts of existing 3D object models is an interesting and efficient technique to create novel shape collections. However, due to the lack of direct parts' correspondence across different shape families, such data-driven modeling approaches in literature are mostly limited to the synthesis of in-class shapes only. To address the problem, this paper proposes a novel approach to create 3D shapes via re-combination of cross-category object parts from an existing database of different model families. In our approach, a reference shape containing multi-functional constituent parts is pre-specified by users, and its design style is then reused to guide the creation process. To this end, the functional substructures are first extracted for the reference shape. After that, we explore a series of category pairs which are potential replacements for the functional substructures of the reference shape to make interesting variations. We demonstrate our ideas using various examples, and present a user study to evaluate the usability and effectiveness of our technique.

Keywords:

cross-class synthesis, assembly-based modeling, structure analysis

1. Introduction

Creating large-scale man-made 3D shape collections is essential for modeling the virtual world. However, manually assembling such shapes would be tedious and extremely labor-intensive, especially when the target model to be designed is complicated in its structure and function.

Recently, several approaches [1, 2, 3] have been proposed to effectively synthesize 3D shapes of a single family through reusing existing object parts. In these approaches, a single-class shape collection is fed into the algorithm, which interchanges the parts among different 3D models to generate a large collection of novel shapes. However, although such approaches can achieve promising results in certain scenarios, the diversity of the synthesized shapes might be limited without attention to inter-class information. The challenge in cross-class 3D object synthesis is the lack of direct parts' correspondence: naively interchanging shape parts can easily destroy the shape plausibility. Moreover, for the probabilistic approaches [2, 3], it is hard to collect enough cross-class models for training.

In this paper, we present an approach to synthesize shapes using parts from a variety of model families under the guidance of a reference shape. The reference shape is required to have composite man-made designs with multi-functional components and complicated structures. Given the reference shape and a database of pre-segmented shapes from multiple categories, we first summarize their part structures using *relation graphs*. For each part of the shapes, its structural context is

then identified by considering the related parts which have a support relation. We denote the sub-graph constituted by a part as well as its structural context as a *substructure*. We notice that certain substructures are more critically related to the actual functionality of the models (e.g., a chair's seat and its support, a sunshade's awning and its support, etc.). Such a substructure is defined as the *functional substructure* of the shape. Then we use an *Harmonic Shape Descriptor (HSD)* based descriptor to match the substructures between the database shapes and the reference shape aimed to analyze the constituents of the reference shape (Section 4). The obtained correspondences could be leveraged for exploring potential component replacements based on a category suggestion algorithm for synthesizing novel shape collections (Section 5).

To validate the effectiveness of the proposed approach, we collect a database consisting of 15 model families, and conduct experiments on 9 complex reference shapes. The obtained results and an additional user study show that cross-class synthesis of novel 3D shapes could be effectively performed by reusing the composite design of a complex reference model.

2. Related Work

Assembly-based modeling. Recently, as model collections grew, researchers have focused on data-driven content creation. Modeling by example [4] provided an approach to create new objects by cutting and compositing parts in a 3D database. Chaudhuri and Koltun [5] provided suggestions for 3D modeling benefited from customized examples that stimulate creativity. Shen et al. [6] presented an approach which converts scanning data to 3D models with labeled semantic parts. Tang

*Corresponding author.

Email address: chen@buaa.edu.cn (Xiaowu Chen)

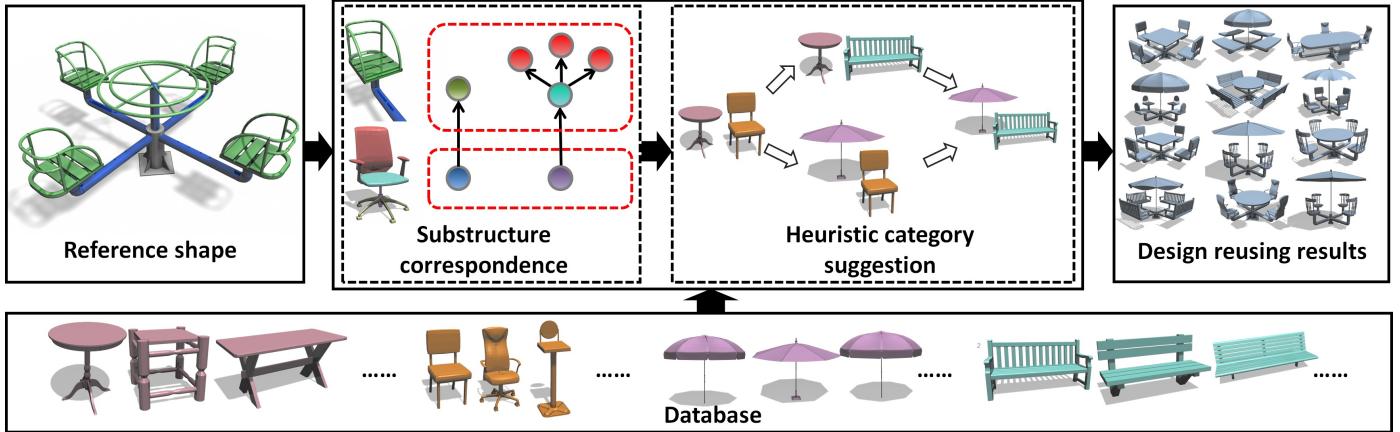


Figure 1: Starting from a reference shape, a series of shapes from different categories are suggested and recombined to synthesize various composite models.

et al. [7] presented a surface deformation method with local and nonlocal guidance, which supports mesh merging. In another approaches, probabilistic models were learned for shape synthesis [2, 3]. Jain et al. [8] proposed a system to create new shapes by blending between shapes from a database. Smart variations [9] proposed a geometric approach based on substructure to create functionally plausible model variations. The above methods achieve impressive results in shape synthesis with in-class shapes, or shapes across the categories which have similar structures. While it's still a hard work for synthesizing the shapes from the categories with different structures and functions. We leverage the reference shape to inspire shapes from such categories synthesizing novel multi-function composite models.

Shape analysis. Various approaches have been proposed to extract high-level hierarchies of shapes. Wang et al. [10] introduced symmetry hierarchy of man-made objects to represent a 3D model by a symmetry-induced, hierarchical organization of the models components. Semi-supervised learning method [11] used users' assists in the co-analysis by providing inputs iteratively to constrain the system. The survey [12] explored numbers of methods of extracting geometric symmetries and exploiting high-level hierarchies for a wide variety of geometry applications. In our work, we analyze the shape taking advantage of each component's structure context. Then we employ an analysis algorithm to recognize the functional substructures of the reference shape.

Exploring shape collections. With the fast growing of 3D databases that are available on the Internet, efficiently exploration of these shapes has becoming a new task for researchers. Attene et al. [13] proposed to perform segmentations and annotations of 3D surface meshes through ontology. Recently, more approaches have been extracted for organizing and exploring a collection of 3D shapes, such as deforming a base template [14], using fuzzy correspondences [15], and utilizing a qualitative analysis [16]. Besides, some works analyzed the relevancy between image and shape collection. For example, Averbuch-Elor et al. [17] proposed a distillation algorithm for

image collections which supports 3D applications like the construction of a 3D abstract model. Zhou et al. [18] used a single image to model a 3D garment. Su et al. [19] added depth to an image of an object by exploiting a collection of aligned 3D models of related objects. Huang et al. [20] proposed to jointly analyze a collection of images of different objects along with a smaller collection of existing 3D models. In our work, we employ a category group suggestion algorithm to explore the matched shape categories, which can be used to replace the certain substructures of the reference shape to synthesize novel composite models.

3. Overview

As shown in Figure 1, our method consists of an offline stage and an online stage. In the offline stage, we pre-analyze the 3D shape collections to facilitate computation. When online, an external reference shape is fed into our system with its design reused to synthesize novel composite models. We will briefly describe these two stages in the rest of this section.

3.1. Offline Database Pre-processing

The database of shape collections we use in this paper contains 15 categories (i.e., bathtubs, beds, benches, bikes, boats, chairs, dressers, cribs, lamps, pavilions, pianos, sofas, sunshades, tables and trolleys) collected from [21, 22, 23]. We assume that all the models in the database have been pre-segmented into meaningful parts. The state-of-the-art segmentation algorithms [13, 11] work well for this purpose. Note that we do not require the semantic labels of parts or their correspondences be available.

Our approach requires 3D models have approximately correct sizes as those presented in daily life. It is critical for our algorithm since some geometric features are deduced from the relative scales between parts. Besides, the method [24] is used to make sure that models have upright orientations. We also align the shapes globally to a common orientation to facilitate subsequent part synthesizing [9]. Finally, each shape is represented by a spatial relation graph, whose nodes and edges

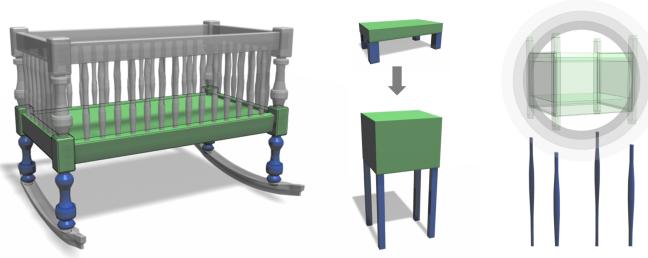


Figure 2: For each part, we first resize the object to make the part’s OBB to be a cube, and then calculate the support structure feature with the concentric spherical shells.

are formed by parts and their support relations, respectively. Note that the graph is directed since the support relations are not commutative.

3.2. Online Shape Synthesis

Given the input reference shape, the online shape synthesis consists of two main steps, namely, functional substructure matching and design reusing.

Functional substructure matching. This step aims at finding functional correspondences between the reference shape and a database shape. To this end, the input reference shape is first manually segmented into meaningful parts and represented by a relation graph as similar as in preprocessing database shapes. Afterwards, we seek for the substructures (i.e., components and their structural contexts) which are matched between the database shapes and the reference shape by a descriptor encoding shape geometry and support type. The matched substructures what we called *functional substructures* can be leveraged to describe the functional constituents of the reference shape, and establish the correspondences between the database shapes and the reference shape in part-level. The details of this step are summarized in Section 4.

Design reusing. After the correspondences between shapes are obtained, we reuse the design of the reference shape to inspire synthesis of novel cross-class models. Specifically, a suggestion algorithm is employed to encourage existing shapes from different categories to participate in synthesizing. Moreover, we adopt structure evolution to further diversify the created shapes. We refer the reader to Section 5 for more technical details.

4. Functional Substructure Matching

In this section, we introduce how to extract the functional substructures of the reference shape and database shapes with structural context descriptor, as well as establish their correspondences. We first add the support relation to the shape’s related graph to get the each components structural context (i.e., the structurally related parts) and every substructure which consists of a component and its structure context. Then we use the structural context descriptor to analyze the shape and support type of each component, and match the similar substructure in shapes from the database to explore the functional substructure of the reference shape.

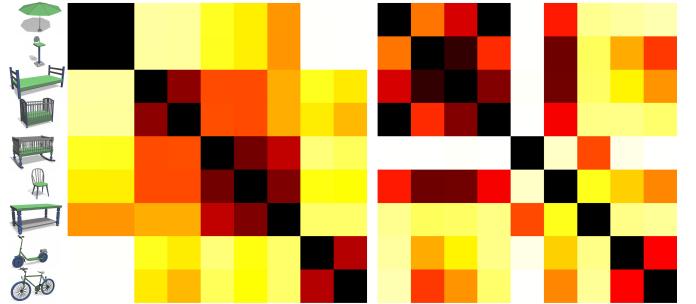


Figure 3: The classification of support type with our support structure feature (left), and regular HSD method (right).

4.1. Functional Substructure

For man-made shapes, some components are more critically related to the actual functionality of the models. Zheng et al. [9] leverage mutual (geometric) relations among different arrangements of shape parts to identify component-level compatible functional substructures. Since functionality is rarely explicitly encoded in the raw geometric descriptions, their method intends to simplify this problem by seeking for certain substructures which are often related to actual functionality of the models. Our method follows a similar idea. In our approach, the core component and its structural context (e.g., a chair’s seat, and its context), which are more critically related to the shape’s actual functionality, constitute the substructure that we term functional substructure. Single-functional database shapes always have single functional substructures, while multi-functional reference shapes have more than one functional substructure. We observe that the functional substructures which have similar functionalities are more likely to be matched.

As mentioned in Section 3, the reference shape and all shapes from the database are represented by directed spatial relation graphs. Given a model with its relation graph, we can extract the shape’s substructure as follows. For the i -th part denoted as P_i in the model, we use the directed graph to collect the set of parts O_i supported by P_i and the set of parts \mathcal{U}_i that support P_i . In this way, the substructure with P_i as the center part is the subgraph with the node set $\mathcal{R}_i = P_i \cup O_i \cup \mathcal{U}_i$. Afterwards, we aim to explore certain substructure which is able to represent the using function of the man-made shape. Specially, we call such substructure as the functional substructure. We can use the database shapes to analyze the functional constituents of the reference shape via matching the functional substructures.

4.2. Structural Context Descriptor

In order to measure the similarity of the substructures between the reference shape and the shapes from the database, we employ the structural context descriptor based on *Harmonic Shape Descriptor (HSD)* [25] to encode the shape and support of a substructure. The main idea of HSD is to decompose a spherical region into concentric spherical shells with different radii and compute the spherical harmonic decomposition for each of those shells, and then store the amplitudes of the harmonic coefficients within every frequency to form a feature vector for indexing and matching.

For the i -th substructure of a given model, its shape descriptor c_i is obtained through the following steps. First, the voxel-based representation of its center part P_i is obtained. Then, we intersect the model with 32 concentric spheres. The final feature is formed by concatenating the norms of frequency component at each radius based on 16 harmonic frequency decomposition of each spherical function. Note that since the reference shape and the shapes in the database may have different granularity of segmentation, for each component of the shapes in the database, we simultaneously calculate the shape descriptors c_i for the center part P_i and c'_i for the merged shape with parts $P_i \cup O_i$. We will discuss how the different granularity of segmentation influences the composite result in Section 6.

In addition to the shape descriptor c_i , we also define a support descriptor m_i based on the HSD to represent the type of support for the substructure. First, we resize the oriented bounding box (OBB) of center part P_i to a cube, and transfer the scale deformation to the part set \mathcal{U}_i which support P_i . Then, we use the cube's center as the center of sphere, and the cube's diagonal d_i as diameter to create a spherical shell. We add another 6 (3 inside and 3 outside) concentric spherical shells with the diameter range from $0.85 * d_i$ to $1.15 * d_i$ to enhance the descriptor. As shown in Figure 2, this spherical region is near the surface of center component (green), thus the descriptor is able to encode the support types and the connecting positions for all support parts.

The support descriptor can well measure the similarity of support types of two models. As shown in Figure 3, we choose 9 shapes with various types of support relations. Our support descriptors work well even in distinguishing different support types, while the original HSD features extracted on the parts engaged in the support relations do not (e.g. see the 1st model *umbrella* and the 3rd model *bed* in Figure 3).

4.3. Substructure Matching

We match the substructures between the reference shape and the shapes from the database by measuring their descriptors. For the i -th substructure of the reference shape and the j -th substructure of a database shape, the distances of their shape and support descriptors are given by

$$d_{sh}(\mathcal{R}_i, \mathcal{R}_j) = \min (\|c_i - c_j\|_2, \|c_i - c'_j\|_2), \quad (1)$$

$$d_{str}(\mathcal{R}_i, \mathcal{R}_j) = \|m_i - m_j\|_2, \quad (2)$$

where c_i, m_i, c_j, m_j are the descriptors of the substructures as previously defined. To normalize the distances to a unified scale, we further define the scale factor f_{sh} and f_{str} as:

$$f_{sh}(\mathcal{R}_i, \mathcal{R}_j) = \begin{cases} \frac{\|c_i + c_j\|_2}{n(c_i)}, & \text{if } \|c_i - c_j\|_2 < \|c_i - c'_j\|_2 \\ \frac{\|c_i + c'_j\|_2}{n(c_i)}, & \text{otherwise,} \end{cases} \quad (3)$$

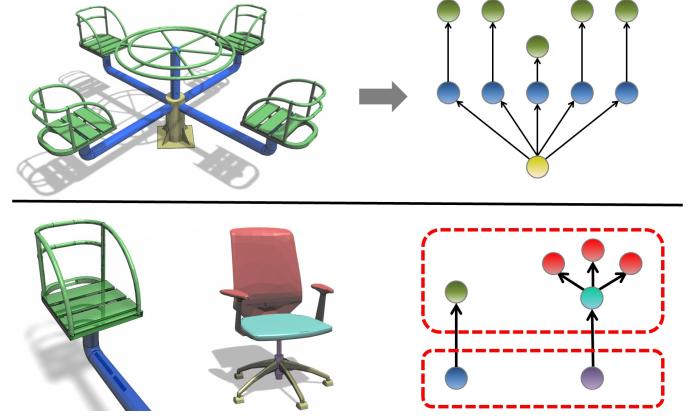


Figure 4: Correspondence of the substructures between reference shape and candidate shape.

$$f_{str}(\mathcal{R}_i, \mathcal{R}_j) = \frac{\|m_i + m_j\|_2}{n(m_i)}. \quad (4)$$

where $n(c_i)$ and $n(m_i)$ are the number of the shape descriptor matrix elements and support descriptor matrix elements respectively.

Given the distance criteria, for the k th substructure \mathcal{R}_k of the reference shape, we first suggest a category that contains the most similar substructure to \mathcal{R}_k . Denote the models in the c th category in database as \mathbb{G}_c , the suitability by replacing the substructure \mathcal{R}_k with a substructure of the c th category is given by

$$e(\mathcal{R}_k, \mathbb{G}_c) = \frac{w_1 \cdot d_{sh}^*(\mathcal{R}_k, \mathbb{G}_c)}{f_{sh}^*(\mathcal{R}_k, \mathbb{G}_c)} + \frac{w_2 \cdot d_{str}^*(\mathcal{R}_k, \mathbb{G}_c)}{f_{str}^*(\mathcal{R}_k, \mathbb{G}_c)} + w_3 \cdot d_s(\mathcal{R}_k, \mathbb{G}_c) + w_4 \cdot d_h(\mathcal{R}_k, \mathbb{G}_c), \quad (5)$$

where $d_{sh}^*(\mathcal{R}_k, \mathbb{G}_c) = \min_{\mathcal{R}' \in \mathbb{G}_c} d_{sh}(\mathcal{R}_k, \mathcal{R}')$ is the minimum disparity of the shape descriptor between \mathcal{R}_k and all substructures in \mathbb{G}_c , and $f_{sh}^*(\mathcal{R}_k, \mathbb{G}_c)$ is the corresponding scale factor. The minimum disparity of support descriptor d_{str}^* and the corresponding scale factor f_{str}^* are similarly defined. To take geometric similarity into consideration, we further introduce the distance measure d_s and d_h as

$$d_s(\mathcal{R}_k, \mathbb{G}_c) = \min_{\mathcal{R}' \in \mathbb{G}_c} r_x(\mathcal{R}_k, \mathcal{R}') + r_y(\mathcal{R}_k, \mathcal{R}') + r_z(\mathcal{R}_k, \mathcal{R}'), \quad (6)$$

$$d_h(\mathcal{R}_k, \mathbb{G}_c) = \min_{\mathcal{R}' \in \mathbb{G}_c} t_y(\mathcal{R}_k, \mathcal{R}'), \quad (6)$$

where r_x, r_y , and r_z are the scale ratio between the center parts of two given substructures along x, y and z dimension, respectively. We also measure the position disparity t_y between the mass of the given two center parts along y dimension. The weights $w_1 = 0.3, w_2 = 0.15, w_3 = 0.4, w_4 = 0.15$ are ascertained by experiment. The best category that matches the substructure \mathcal{R}_k is finally obtained by computing $\arg \min_{c \in \mathbb{C}} e(\mathcal{R}_k, \mathbb{G}_c)$, where \mathbb{C} is the index set of all categories.

Finally, for each substructure of the reference shape, we explore the matched substructure of the database shape by minimizing the energy of equation 5. To control the divergence between the synthesized model and the reference model, we allow the users to specify the core component of each category of



Figure 5: **Left:** The reference shape. **Right:** Four category suggestion results, in each of which shapes from the suggested category pair (left) are used to synthesize the composite model (right) guided by the reference shape.

shapes, and the amount of functional substructures N in reference shape. The top N substructures with the minimum energy are selected as the functional substructures. The functional substructures and their matched shapes in the suggested categories are fuzzily corresponded, as shown in Figure 4.

5. Design Reusing

With the obtained correspondences, we introduce in this section the heuristic category suggestion mechanism to reuse the reference shape to synthesize composite models, and some post-processing steps to further optimize and diversify the synthesized models.

5.1. Category Suggestion

In section 4, an energy function 5 is defined to measure the similarity between substructures. However, if we directly use this energy to suggest categories for replacement, we cannot generate functionally plausible and geometrically coherent models by setting the relative weights w_1 and w_2 between the similarity of shape geometry and support type determinately. Thus, in practice we adopt a heuristic and iterated suggestion mechanism. In each iteration, the algorithm selects a pair of functional substructures and generates four variations for them. Note that our algorithm operates on appropriate category pairs to account for reusing cross-category designs, and a reference shape with more than two different functional substructures can be easily handled by decomposing them into substructure pairs.

In an iteration, the suggestion algorithm works as follows. Given two substructures of the reference shape \mathcal{R}_1 and \mathcal{R}_2 , we first find the best matched two categories with the corresponding database substructures \mathcal{R}'_1 and \mathcal{R}'_2 by computing the minimum sum of the energies obtained by equation 5. To explore more other rational replacements, we then fix \mathcal{R}'_1 or \mathcal{R}'_2 and change another substructure by finding its best match. At this time, the relative weights w_1 and w_2 in the energy function 5 are set to 0.2 and 0.8 respectively to ensure a high structural similarity. After this step, we have two more pairs of database substructures, namely $(\mathcal{R}'_1, \mathcal{R}''_2)$ and $(\mathcal{R}'_2, \mathcal{R}''_1)$ for replacement. Finally, we produce one more pair $(\mathcal{R}''_1, \mathcal{R}''_2)$ by simply keeping the newest obtained database substructures. After each iteration, we obtain four substructure pairs for replacement. Then, we eliminate the used substructures in database out of the loop and proceed for the next iteration.

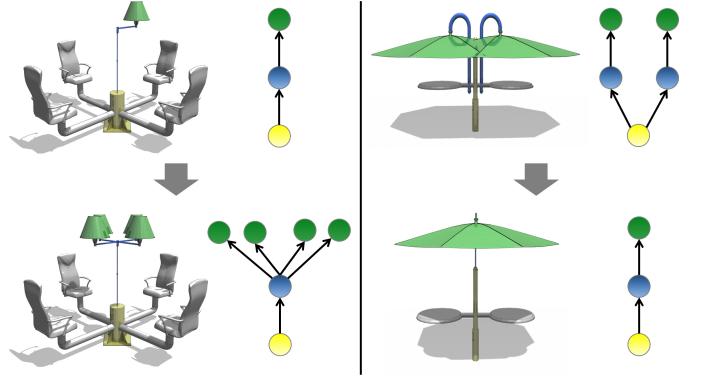


Figure 6: Structure variation with separate operation (left) and combine operation (right).

For instance, for the input reference shape in Figure 5(left) with two functional substructures, our suggestion mechanism first recommends the pair (chair, table), and then adds the pairs (chair, sunshade) and (bench, table) by fixing one and changing another, and finally gives the pair of bench and sunshade(Figure 5(right)). These operations are repeated to obtain replacements using more other categories.

5.2. Shape Synthesizing and Optimizing

After the previous step, we have obtained a series of database substructure pairs that could be used for plausible replacement. We replace the original substructure pairs by enumeration, obtaining a collection of initial synthesized models. Note that in the offline preprocessing stage (see Section 3.1), the database shapes are scaled and their orientations are globally aligned, the synthesis can be done directly. However, the replaced parts are still loosely placed together. Before further optimization, we first take two simple preprocessing steps to eliminate the models not visually coherent with the reference model. First, for each initial synthesized model, if the shape geometries of the support parts between the reference substructure and the replaced substructure are not similar (i.e. the distance of their shape descriptors exceeds a threshold), we retain reference shape's support parts and discard the replaced ones. Moreover, we duplicate the base part of the replaced substructure to make it have the same number of base parts as that of the reference substructure to ensure visual coherence.

To synthesize the initial models into holistic ones, we adopt an optimization procedure similar with that of [2]. In our implementation, we consider both contact and support relations in terms of contacting slots between two parts (see the survey [9] for more details). If two parts have different number of contact positions, we will treat them separately. For the part which has less number of connect slots , we consider the whole slots and define the number of slots as N . For the other one ,we only consider N slots of it which minimized the sum of the distance between the slots of two parts. In some extreme cases, there exists no suitable contact slots (i.e. the parts of the replaced shape mismatch those of the original shape). Since such case is rare, we manually choose the appropriate connect positions.

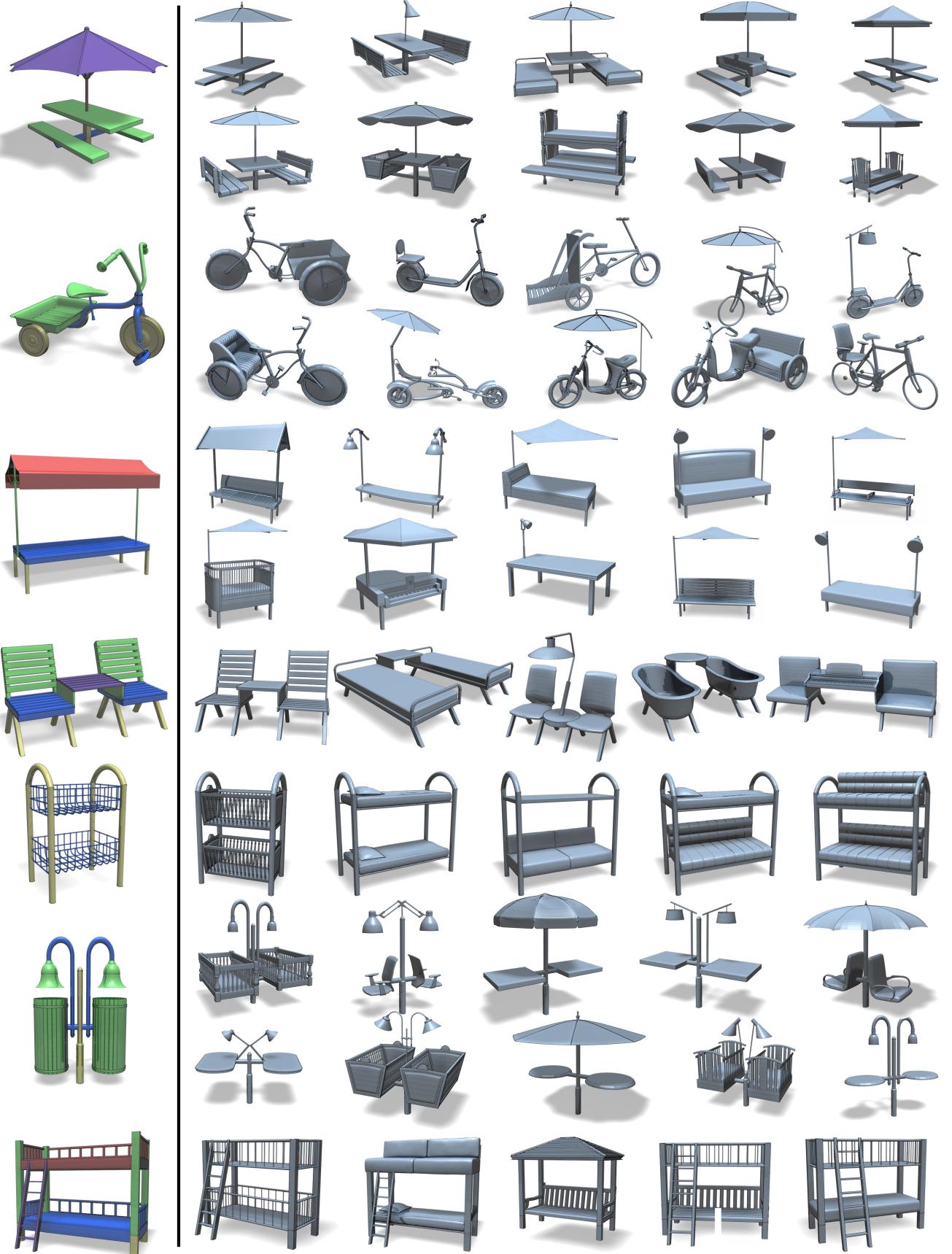


Figure 7: Guided by the reference shapes (left), our approach enable the non-trivial shape variations across classes (right) by synthesizing the suggested shapes from the database.

Finally, we adjust the orientation of the replace substructure. We use the global symmetry plan of the replaced functional substructure to rectify the upright orientation of the replace substructure. We use the method of [9] to address the ambiguities brought about by symmetry flipping.

5.3. Structure Variation

Some reference shapes may contain symmetric components which serve as a strong indication in its appearance and function. This property can be explored to further diversify the structures of the created composite models, as shown in Figure 6. We implement this idea through combining and separating operations to change the shape’s structure. Combining operation merges the symmetric components together and separating operation duplicates the single element, respectively. Specifically, for the symmetric parts in the reference shape, if the size of the replaced parts is larger than the original one in the reference shape, the synthesized model may have several components overlapped with each other. In this case, the combining operation is employed on the overlapped parts. On the other hand, for the independent part in the reference shape, when the size of the replaced part is smaller than the reference part and the remaining parts in the reference shape are symmetric, then the separating operation is employed to duplicate the replaced part to make them also symmetrical.

6. Results

Qualitative results. We tested our method with nine complex multi-functional reference shapes on a database with 15 different model families coming from [21, 22, 23]. Each model family has 15 to 30 shapes. Specially, for certain categories (e.g., chairs, tables and beds), we also test the design reusing process with a large database in which each family has around 100 shapes. This often leads to more various interesting shape variations. We illustrate some representative results in Figure 7.

Time analysis. Generally, the time cost of functional structure analysis and category suggestion is less than 10 seconds on an Intel i7-4790 3.60 GHz desktop with 16GB memory. The remaining synthesizing process is nearly automatical, with only several composite models whose base parts are manually adjusted. The whole running time for shape synthesis is typically in 30 seconds.

User study. We also conducted a user study to assess the results of our method. The user study was conducted with 15 participants, all graduated students of computer science. To prepare for the user study, we collected the top 4 category pairs suggested with and without our heuristic suggestion algorithm for several reference shapes, including the ones shown in Figures 1 & 5, and the top 4 ones in Figure 7, resulting in 48 synthesized results in total. We also compared our results with those generated by human designers. Specifically, we asked each user to select 4 category pairs with respect to each reference shape. Then we got the composite models synthesized based on those

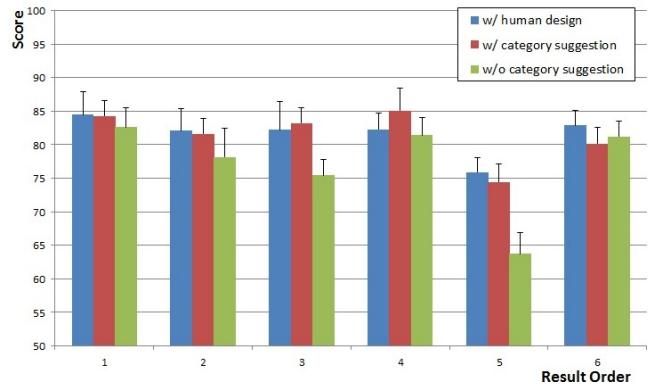


Figure 8: The average scores of the human design results (blue), the results by our heuristic suggestion mechanism (red) and by directly using the shape and support descriptor (green) in six experiments. From left to right, the respective results are shown in Figures 1 & 5, and the top 4 ones in Figure 7. The *error bars* here are the standard derivation of scores.

category pairs as the human designs. Note that for fair comparison the same shape synthesis method was applied to all suggestion category pairs (see Section 5.2). Afterwards, we asked each participant to blindly evaluate the rationality both in appearance and structure of our results and human-designed results (excluding the composite models designed by this participant), and to give a score in the range from 0 (poorest) to 100 (best). The evaluation results are summarized in Figure 8. It indicates that our heuristic suggestion mechanism performed better than directly using the shape and support descriptor in category suggestion. The composite results produced by our approach achieve a similar level of the manually designed results.

Limitations. Our work has two main limitations. First, our approach relies on good-quality pre-segmentation of the reference shape to reveal its functional substructures. Figure 9(left) shows two failed examples due to either under-segmentation (a) or over-segmentation (b). In such cases, manual intervention is needed to ensure an acceptable segmentation. Some parts which are irrelevant to the structure of the reference shape (e.g., the ladder of the last example in Figure 7) are also manually removed from the relation graph to avoid misleading the matching of substructures. Second, even though our HSD based descriptor often enables the extraction of certain constituents which play an important role in the shape’s functionality, such functional substructures are after all based on geometric properties only and thus do not always have semantic meanings. Therefore, some category suggestion results may fail to produce a semantically meaningful composite model, since their semantic functionalities are not suitable to be combined. For an example in Figure 9 (c), the crib is suggested to be placed around the table, making the table useless in practice. Another example is shown in Figure 9 (d), the synthesized tricycle is the combination of a bicycle and a chair. The wheel of tricycle is duplicated to satisfy the symmetric structure of the reference shape. However, there lacks a suitable axle to link the rear-wheels and support the seat.

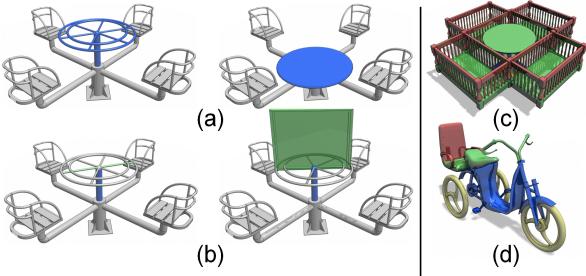


Figure 9: **Left:** Two less successful examples with coarsely segmented (a) and over segmented (b) reference shapes. In each case, we show one substructure which is highlighted in the reference shape (left) and the part replacement result (right). **Right:** Unreasonable suggestion category pairs (c) and the result lacking in strut members (d).

7. Conclusion

We present in this paper a novel approach for cross-class shape synthesis via reusing the design of a reference shape. In our approach, we first extract functional substructures for the reference shape. After that, we establish the correspondences between the substructures of the database shapes and the functional substructures of the reference shape in terms of their shape geometry and structural context. Given the correspondences, we reuse the design of the reference shape through a category suggestion algorithm to initialize a collection of cross-class synthesized models which are then optimized and expanded with more variations. In the experiments, we show various examples and present a user study illustrating the effectiveness of our method.

We believe that design reusing of a reference shape opens up new opportunities for shape synthesizing and can be helpful for generating complex multi-functional composite models. In the future, we will consider modeling approach jointly guided by structure and shape geometry of a reference shape, and improve the practicability of the synthesized models by introducing functional semantics of models.

Acknowledgments

We would like to thank the anonymous reviewers for their help in improving the paper. This work was partially supported by NSFC (61325011&61421003), 863 Program (2013AA013801), Research Grants Council of HKSAR, China (113513&11204014) and SRFDP (20131102130002).

References

- [1] Kai Xu, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen, Fit and Diverse: Set Evolution for Inspiring 3D Shape Galleries, ACM Transactions on Graphics, 31, 57:1–57:10 (2012)
- [2] Evangelos Kalogerakis, Siddhartha Chaudhuri, Daphne Koller, and Vladlen Koltun, A Probabilistic Model for Component-based Shape Synthesis, ACM Transactions on Graphics, 31, 55:1–55:11 (2012)
- [3] Siddhartha Chaudhuri, Evangelos Kalogerakis, Leonidas Guibas, and Vladlen Koltun, Probabilistic Reasoning for Assembly-based 3D Modeling, ACM Transactions on Graphics, 30, 35:1–35:10 (2011)
- [4] Thomas Funkhouser, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin, Modeling by Example, ACM Transactions on Graphics, 23, 652–663 (2004)
- [5] Siddhartha Chaudhuri, and Vladlen Koltun, Data-driven Suggestions for Creativity Support in 3D Modeling, ACM Transactions on Graphics, 29, 183:1–183:10 (2010)
- [6] Chao-Hui Shen, Hongbo Fu, Kang Chen, and Shi-Min Hu, Structure Recovery by Part Assembly, ACM Transactions on Graphics, 31, 180:1–180:11 (2012)
- [7] Yufeng Tang, Dongqing Zou, Jianwei Li and Xiaowu Chen, Local and Nonlocal Guidance Coupled Surface Deformation, SIGGRAPH Asia 2014 Technical Briefs, 10:1–10:4 (2014)
- [8] Arjun Jain, Thorsten Thormählen, Tobias Ritschel, and Hans-Peter Seidel, Exploring Shape Variations by 3D-Model Decomposition and Part-based Recombination, Computer Graphics Forum, 31, 631–640 (2012)
- [9] Zheng Youyi, Daniel Cohen-Or, and Niloy J. Mitra, Smart variations: Functional substructures for part compatibility, Computer Graphics Forum, 32, 2 (2013)
- [10] Yanzhen Wang, Kai Xu, Jun Li, Hao Zhang, Ariel Shamir, Ligang Liu, Zhiqian Cheng, and Yueshan Xiong, Symmetry Hierarchy of Man-Made Objects, Computer Graphics Forum, 30, 287–296 (2011)
- [11] Yunhai Wang, Shmulik Asafi, Oliver van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen, Active Co-analysis of a Set of Shapes, ACM Transactions on Graphics, 31, 165:1–165:10 (2012)
- [12] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan, Symmetry in 3D Geometry: Extraction and Applications, Computer Graphics Forum, 32(6), 1–23 (2013)
- [13] Marco Attene, Francesco Robbiano, Michela Spagnuolo, and Bianca Falcidieno, Characterization of 3D Shape Parts for Semantic Annotation, Computer-Aided Design, 41, 756–763 (2009)
- [14] Maks Ovsjanikov, Wilmot Li, Leonidas Guibas, and Niloy J. Mitra, Exploration of Continuous Variability in Collections of 3D Shapes, ACM Transactions on Graphics, 30, 33:1–33:10, (2011)
- [15] Vladimir G. Kim, Wilmot Li, Niloy J. Mitra, Stephen DiVerdi, and Thomas Funkhouser, Exploring Collections of 3D Models Using Fuzzy Correspondences, ACM Transactions on Graphics, 31, 54:1–54:11 (2012)
- [16] Shi-Sheng Huang, Ariel Shamir, Chao-Hui Shen, Hao Zhang, Alla Sheffer, Shi-Min Hu, and Cohen-Or Daniel, Qualitative Organization of Collections of Shapes via Quartet Analysis, ACM Transactions on Graphics, 32, 71:1–71:10 (2013)
- [17] Hadar Averbuch-Elor, Yunhai Wang, Yiming Qian, Minglun Gong, Johannes Kopf, Hao Zhang, and Daniel Cohen-Or, Distilled Collections from Textual Image Queries, Computer Graphics Forum, 34, 2 (2015)
- [18] Bin Zhou, Xiaowu Chen, Qiang Fu, Kan Guo and Ping Tan, Garment Modeling from a Single Image, Computer Graphics Forum, 32, 85–91 (2013)
- [19] Hao Su, Qixing Huang, Niloy J. Mitra, Yangyan Li, and Leonidas Guibas, Estimating Image Depth Using Shape Collections, ACM Transactions on Graphics, 33, 4 (2014)
- [20] Qixing Huang, Hai Wang, and Vladlen Koltun, Single-View Reconstruction via Joint Analysis of Image and Shape Collections. ACM Transactions on Graphics, 34, 4 (2015)
- [21] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser, The princeton shape benchmark, Shape Modeling Applications, 167–178 (2004)
- [22] Ibraheem Alhashim, Honghua Li, Kai Xu, Junjie Cao, Rui Ma, and Hao Zhang, Topology-Varying 3D Shape Creation via Structural Blending, ACM Transactions on Graphics, 33, 158 (2014)
- [23] Trimble, 2013. Trimble 3D warehouse, <http://sketchup.google.com/3dwarehouse/>.
- [24] Hongbo Fu, Daniel Cohen-Or, Gideon Drorand, and Alla Sheffer, Upright orientation of man-made objects, ACM Transactions on Graphics, 27, 42 (2008)
- [25] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz, Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors, Symposium on Geometry Processing, 6, 156–164 (2003)