

# Topology-Varying 3D Shape Creation via Structural Blending

Ibraheem Alhashim<sup>1</sup> Honghua Li<sup>1,2</sup> Kai Xu<sup>2</sup> Junjie Cao<sup>3</sup> Rui Ma<sup>1</sup> Hao Zhang<sup>1</sup>

<sup>1</sup>Simon Fraser University

<sup>2</sup>National University of Defence Technology

<sup>3</sup>Dalian University of Technology

## Abstract

We introduce an algorithm for generating novel 3D models via *topology-varying* shape blending. Given a source and a target shape, our method blends them topologically and geometrically, producing continuous series of in-betweens as new shape creations. The blending operations are defined on a spatio-structural graph composed of medial curves and sheets. Such a shape abstraction is structure-oriented, part-aware, and facilitates topology manipulations. Fundamental topological operations including split and merge are realized by allowing one-to-many correspondences between the source and the target. Multiple blending paths are sampled and presented in an interactive, exploratory tool for creative 3D modeling. We show a variety of topology-varying 3D shapes generated via continuous structural blending between man-made shapes exhibiting complex topological differences, in real time.

**Keywords:** 3D shape creation, topology variation, shape blending

**Links:** [DL](#) [PDF](#) [WEB](#) [DATA](#) [CODE](#)

## 1 Introduction

Shape blending is a common technique that has been most widely studied in the areas of interpolation-based animation and mesh morphing [Alexa 2002]. An essential step for most methods is to compute a mapping between the source and target meshes [Sheffer et al. 2006], assuming that the shapes have the same topology. Volumetric approaches based on implicit shape representations can be topology-altering, e.g., [Cohen-Or et al. 1998; Breen and Whitaker 2001]. However, like the case for mesh morphing, the resulting in-betweens only serve to visualize a temporal deformation process; they need to maintain certain physical properties of the source (target) or remain unchanged intrinsically, aside from articulation, throughout the process. In both cases, the in-betweens are typically computed via purely geometric means, governed by motion energy minimization and physical constraint satisfaction.

In this paper, we view and adopt shape blending from a different angle. Our goal is *shape creation*, i.e., to create novel and plausible 3D models using the in-betweens generated by shape blending. To obtain interesting and creative shape variations, we assume that the input shapes differ both in geometry and topology. We aim to obtain continuous and plausible blending with both geometric and topological variations. Our focus will be on the latter since topology plays the key role in altering *shape structures* and the novelty and diversity of the model creations are more fundamentally attributed to these structural variations; see Figure 1.

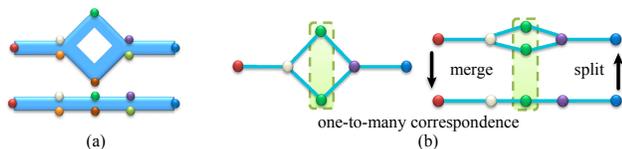


**Figure 1:** 3D shapes generated by topology-varying blending between a source (left) and a target (right), along three different blending paths. Part correspondence is shown by matching colors; some parts (neutral colors) are unmatched.

The technical challenges to our shape blending problem are two-fold. Not only must the computed in-betweens possess new topological structures that reflect a blending between the topologies of the input shapes, the blending results must also be properly controlled and filtered to attain a certain degree of plausibility. By “plausible”, we require the results to be functionally similar to the source and target. For example, as shown in Figure 1, the created in-betweens for two chairs should all be “chair-like”.

The key observation behind our approach is that the functional or semantic information about a shape is primarily built into its *part structure*, rather than geometric details. To obtain topology-varying shape blending with plausible in-betweens, two fundamental building blocks of any blending technique, namely, the shape representation and the correspondence scheme, ought to be structure-oriented, part-aware, and facilitate topological operations. In this paper, we focus on blending between mesh models representing man-made 3D objects which often possess rich structural and topological variations. We focus on structural blending at the part level which contributes to the overall preservation of shape functionality. Figure 3 provides an outline of our shape blending scheme.

**Spatio-structural graph.** For each 3D shape, we build a spatio-structural graph (or simply, a structural graph) composed of parametric 1D curves and 2D sheets which are medial abstractions of the shape parts; see Section 4.1. The graph nodes represent geometric entities that have spatial location and extent. The graph edges define connectivity between the shape parts. With such a graph representation, we refer to topological changes to a shape as any structural changes to the graph, e.g., adding or removing a node, altering edge connections, and merging or splitting of graph nodes, etc. The lightweight shape representation using only connected curves and sheets facilitates both correspondence computation and topological operations. For example, imposing topological changes on a curve is easier than on a surface since curve geometry is a lot simpler to handle, e.g., see Figure 2.



**Figure 2:** Topology blending between boundary representations (a) involves more geometry alteration than between 1D curves (b). Dots with matching colors specify point correspondences.

**Correspondence.** To enable merge and split of shape parts, one-to-many part correspondences between the source and target shapes are necessary. For example, a one-to-four mapping would trigger a split which turns a swivel chair into a four-legged one. When a part is missing from one of the two shapes, it is mapped to a *seed region* on the other shape. A seed region consists of a point or curve segment and it is where growing and shrinking of curves or sheets occur. Input shapes with significant topological dissimilarities make compelling cases for shape blending and creative modeling, but challenging cases for part correspondence. We compute part correspondence semi-automatically, allowing the user to specify a sparse initial correspondence (Section 4.2).

**Shape blending.** Given part correspondences, the blending problem becomes that of constructing morphing paths between corresponding parts/regions in the source and target shapes. Plausible blending results not only depend on meaningful correspondence, but also proper ordering of part transitions. In most cases, there is not a unique reasonable ordering, and at the same time, not all orderings lead to reasonable blended shapes. We take an approach which enables the user to explore a large portion of the space of blending paths, thus minimally constraining the creativity of the generative process, while exerting plausibility control.

Specifically, we generate sequences of blending operations stochastically and apply a series of *implausibility filters* to remove shapes or whole blending paths that are deemed to be sufficiently implausible in terms of functionality preservation. In the current work, our consideration of functionality focuses primarily on symmetry and connectivity relations between shape parts. Blending is first applied to input structural graphs and produces in-between structural graphs; see Sections 4.4 and 4.5. In the final step, a mesh can be reconstructed from an in-between graph by an inverse, skeleton-to-mesh, mapping and surface blending.

**Exploratory modeling.** Novel shape creation should naturally be an exploratory process, as free exploration of alternatives is key to a creativity support tool [Shneiderman 2006]. To this end, we develop an interactive and exploratory modeling tool to showcase our shape blending algorithm. Given a source and target shape, multiple blending paths are sampled and presented in the tool after implausibility filtering. Topologically and geometrically distinctive models are displayed upfront. The user can explore all the shape creations by refining the display. Any generated in-between is immediately reusable as a source or target for subsequent shape blending. Hence the user can browse and select preferred shapes to continuously evolve a shape collection.

**Contribution.** The main contribution of our work is a novel approach for 3D shape creation via topology-varying shape blending. To the best of our knowledge, our work represents the first attempt at producing *continuous* variations among 3D shapes that allow topological changes. The granularity afforded by our approach leads to richer shape variations which are unattainable using *discrete* part shuffling or recombination [Kreavoy et al. 2007; Kalogerakis et al. 2012; Jain et al. 2012; Xu et al. 2012], e.g., see the variety

of blended chair backs in Figure 1. Our work also contributes a useful tool for exploratory and creative 3D modeling leading to novel shape structures. We compare our modeling capabilities to those provided by existing modeling tools both qualitatively and through a user study. The continuous process of shape blending and user interaction with the modeling tool can be visualized in the accompanying video. Numerous blending results are reported in the paper as well as in the supplementary material. Source code for the blending algorithm and the modeling tool is also provided.

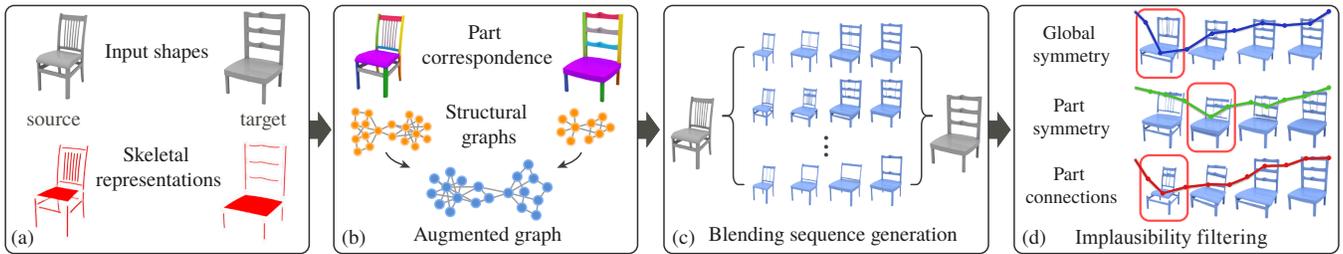
## 2 Related work

Extensive literature exists on shape blending [Alexa 2002], skeletal structures [Cornea et al. 2007], as well as topology analysis or processing for 3D shapes. For example, topological or skeletal characterizations have been employed in such applications as shape matching [Sebastian et al. 2004; Hilaga et al. 2001] and topology repair [Zhou et al. 2007]. Topological decomposition has been applied for mapping surfaces with complex topology [Li et al. 2009]. In what follows, we mainly discuss works that are most closely related to ours, i.e., those on structure or shape blending, as well as shape modeling in general, which allow topology changes.

Existing works on topology-altering mesh morphing, e.g., [Surazhsky et al. 2001; Takahashi et al. 2001; Kanonchayos et al. 2002], often produce unnatural shape transitions functionality-wise. Being purely geometry-driven, these methods are not designed to be structure-aware. Volume-based morphing, e.g., [Hughes 1992; Cohen-Or et al. 1998; Breen and Whitaker 2001], relies on distance fields and distance transforms to compute in-betweens. Often, some level of user control, e.g., by specifying anchor points [Cohen-Or et al. 1998], is necessary to achieve reasonable outcomes. Topological complexity is usually not an issue as volume processing through implicits does not need to track topology changes explicitly. However, like their mesh counterparts, these methods only blend geometry but not structure; they are not designed to preserve the functionality of the morphed shape during transition.

Editing tools capable of modifying shape topologies have been developed since the early days of computer graphics [Welch and Witkin 1994]. Cut-and-paste is a popular editing paradigm for changing topology. Fu et al. [2004] work with regions of non-zero genus to minimize distortions caused by incompatible geometry between the source and target. Sharf et al. [2006] further simplify the process by suggesting better placement of the pasted parts. Powerful real-time tools that support topology editing now exist, e.g., Meshmixer [Schmidt and Singh 2010] and GeoBrush [Takayama et al. 2011]. Most recently, Berstein and Wojtan [2013] developed a method to induce topological changes to arbitrary meshes for sculpting and deformation. Our work is not on shape editing, which would require interactive user input throughout while producing a single shape as outcome. Our goal is to create sequences of novel 3D models via shape blending. The blending process is largely automated, with user input only at the initialization stage.

There are various forms of graph representations which encode both shape structure and geometry. Widely used in CAD are constructive solid geometry (CSG) trees that represent constructions of complex objects using Boolean operators on 3D primitives. More recent works on generative shape analysis, e.g., [Wang et al. 2011], are in the same spirit. In procedural modeling, a shape grammar can be described by a graph defining space subdivision rules and the different geometric elements. Shock graphs and graph editing distances have been employed for shape recognition [Sebastian et al. 2004]. Topology of the shock graphs are altered via graph editing where the goal is to find a single, least-cost editing path which best reflects structural similarity. What is common about these graph



**Figure 3:** An overview of our shape blending algorithm. Given a source and target shape, we convert each into a skeletal curve-sheet representation (a). Part correspondences are identified semi-automatically (b). Each model is encoded by a structural graph and then an augmented graph to accommodate missing parts and one-to-many correspondences (b). Blending generates multiple paths of in-betweens (c) with clearly implausible results filtered out (d). Shapes circled in (d) receive low scores as they fail to preserve symmetry or connectivity.

representations and their use is the desire for shape understanding. Our spatio-structural graphs have a different purpose: to facilitate topology blending. Moreover, our graph transitions are meant to be diversified for the purpose of creative modeling.

Most relevant to our work are recent approaches for shape creation from existing examples via part replacement or recombination, e.g., [Jain et al. 2012; Kalogerakis et al. 2012; Xu et al. 2012; Zheng et al. 2013]. The “blending” in all of these approaches is discrete, in that entire shape parts are replaced by other parts which share similar geometric or structural properties. A part replacement can produce new topologies only when the new part has a different topology from the part to be replaced. In principle, any sequence of part replacements is realizable by a blending path in our approach; though we do not sample all possible blending paths. In addition, our blending generates potentially topology-varying transitions between each adjacent part replacement. Hence, the topology changes afforded by our approach are expected to be more fine-grained, resulting in more structural diversity. Moreover, owing to its continuous nature, topology blending induces more gradual changes to a shape which facilitate structure preservation and part connection.

### 3 Overview

Our blending algorithm takes as input two mesh models, a source and a target, which represent two man-made 3D objects. The output consists of multiple sequences of blended 3D shapes. We develop an interactive and exploratory modeling tool which allows a user to influence the blending process in various ways, to visualize and select blending results, and to export selected models so that they can be immediately reused for further blending.

**Structural graph.** The source and target models are assumed to have been segmented into meaningful parts with meaningful part connections. To facilitate topological operations, we abstract each input shape into a curve-sheet representation via skeleton extraction; see Figure 3(a). The fundamental data structure for our shape blending is the structural graph, whose nodes are the extracted curves and sheets from the shape parts and whose edges are defined by part connections; see Figure 3(b). Part correspondence between the source and target is computed semi-automatically.

**Graph augmentation.** Since not all nodes or edges in the source and target structural graphs have their counterparts, e.g., the armrests in the source chair may be missing in the target, we augment the two graphs so that they have corresponding sets of nodes and edges; see Figure 3(b). Augmented graphs allow us to keep track of evolving states of the shapes and maintain the topological operations needed for blending. Each pair of corresponding nodes in the augmented graphs induces an appropriate topological event, e.g., a

one-to-many correspondence implies split/merge and a *null* node (a missing part) implies growing/shrinking.

**Blending.** Executing the topological events in different orders produces different continuous blending paths; see Figure 3(c). To allow the user to explore the solution space well, we stochastically sample the possible blending paths and rely on an implausibility filter to eliminate clearly implausible in-betweens, keeping the number of models presented in the modeling tool manageable; see Figure 3(d). The filtering is based on structure preservation with respect to the source and target. Likewise, structure preservation also constrains shape blending. Finally, we reconstruct meshes from the generated in-between structural graphs by an inverse mapping from curves/sheets to surfaces and then surface blending.

**Manual vs. automatic processing.** Our blending process is not fully automatic. However, user assistance is only required in the preparation stage, namely, shape segmentation, part connection, and part correspondence, in challenging cases. Graph augmentation, shape blending, and implausibility filtering are all automatic. Compared to efforts required in processing large sets of topology-varying in-betweens, user involvement in our blending process is quite minimal. It is also necessary since none of the state-of-the-art shape analysis techniques is capable of automatically inferring meaningful part segmentation or correspondences amid significant shape variations in both geometry and topology.

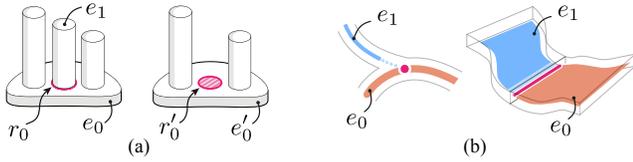
## 4 Topology-varying shape blending

In this section, we provide a detailed technical description of structural graphs, the blending algorithm, and implausibility filtering.

### 4.1 Structural graph

The input shapes are pre-segmented. For each shape part, we apply the skeletonization method of Tagliasacchi et al. [2012] which is able to extract a curve skeleton if the part is more tubular than disk-like or a 2D sheet otherwise. The resulting contracted geometries are *parameterically* fitted by B-Splines. For a curve skeleton, we perform shortest edge collapses to obtain a set of sample points along the skeleton. We then apply piecewise cubic B-spline fitting to obtain a parametric curve that best fits the sampled points. For a sheet, we first compute a quadrilateral patch that best fits the boundary of the contracted geometry. We then uniformly sample this patch by computing isolines along the two largest sides of the quadrilateral for a bi-cubic B-spline fitting. More advanced fitting techniques, e.g., [Pottmann et al. 2002], are also applicable.

For the source (or target) shape, we define a structural graph whose nodes correspond to the curves or sheets characterizing the shape



**Figure 4:** Seed region (marked red) determination, where  $e_1$  denotes the part that corresponds to a null node (a). If the seed part ( $e_0$ , by correspondence) is given, then the seed region is determined through natural geometric continuation (b).

parts. Node connections are implied from geometric connectivities between the corresponding parts in the input shape. If semantically, two parts ought to connect but they are disconnected in the shape, perhaps as a modeling artifact, the user can assist in imposing the connection. Since skeleton extraction involves geometric contraction, the curves or sheets associated with two connected parts are typically not physically connected; see Figure 3(a).

We encode functionality by part connection and *symmetry grouping*, which the shape generation process aims to preserve. Grouping relations among reflectional and rotationally symmetric parts are detected as in Wang et al. [2011] (without hierarchy construction) and stored in the graph nodes. Correspondences between symmetry groups are implied from part correspondences. Part connectivity is stored in graph edges. For each pair of connected parts, we store the contact points, i.e., the closest points on the two parts.

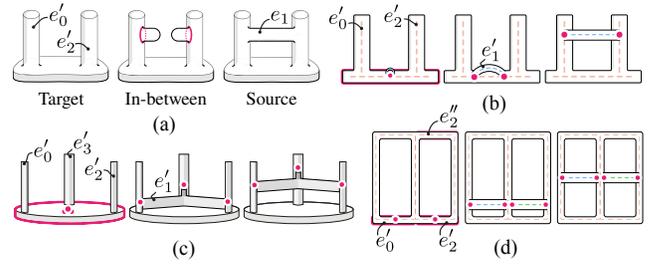
## 4.2 Correspondence

Continuous blending of two structural graphs requires a point-to-point correspondence between the curves and sheets. We first establish part/node correspondence then for corresponding nodes, compute point correspondence based on their parametric curve or sheet representations. With differing topologies in the source and target, one-to-many and *null* correspondences are possible, where the latter is between one part (a graph node) on the source/target and a corresponding region (a null node) over the target/source.

**Node/part correspondence.** The general problem of aligning or matching dissimilar man-made shapes is quite difficult. The challenge is even greater when the shapes are allowed to differ significantly in topology, which is typical and even desirable for modeling based on topology blending. Computing automatic shape correspondences is not a focus of our work, hence we follow a simple greedy scheme while allowing user assistance to prepare node correspondences for the blending task.

The source and target shapes are assumed to be upright oriented. We normalize them by global scale and align them by global reflectional symmetry axes with back and front distinguished by Hausdorff distance. Then for parts from the two shapes, we measure their pairwise Hausdorff distances on the control points of their parametric representations, which are efficient to compute. Part pairings with minimal Hausdorff distances are greedily selected before a similarity threshold (set as 0.5 throughout) is met. The remaining, unmatched parts are considered for null correspondence. After auto correspondence, the user can adjust the result as necessary.

**Curve and sheet correspondence.** For each pair of nodes in correspondence, we compute point-to-point correspondence between their associated curves or sheets to allow part-to-part linear interpolation. For curve-curve and sheet-sheet mappings, our simple strategy is boundary matching followed by parametric mapping in the interior; here let us recall that all our sheets are fitted by



**Figure 5:** Structure-aware seed part (red) selection. Seed region by natural continuation leads to disconnected growth at a null node if its corresponding node ( $e_1$ ) is connected to two parts (a). Instead, we define seed part as the central joint of a prominent symmetry group, e.g., a reflectional (b) or rotational (c) symmetry, so that the growth maintains part connectivity. Finally, seed parts are chosen symmetrically, whenever appropriate (d).

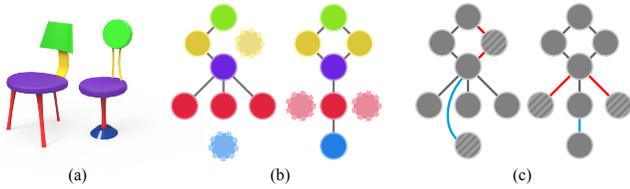
quadrilateral B-spline patches. Since the source and target shapes are already well aligned, we match the boundaries by pairing up the end (for curve) or corner (for sheet) points based on minimum Euclidean distance. We allow correspondence between a sheet  $s$  and a curve  $c$  by converting  $s$  to a curve counterpart. Specifically, we find a curve along either parametric directions on the sheet that best aligns with  $c$ . This allows for the flexibility of blending parts that significantly differ in both topology and geometry.

## 4.3 Seed region

For a null correspondence, we need to identify a seed region on the target (or source) for the missing part, from which the missing part can be grown to match its counterpart on the source (or target). Without loss of generality, let the missing part be on the target and denote by  $e_1$  the corresponding (unmatched) curve or sheet in the source. First, assume that the *seed part*  $e'_0$  for the null node, the part where growth originates, is known. We show how to locate the seed region on  $e'_0$  via *natural continuation*. Then we describe our scheme for identifying seed parts. Our approach takes both geometry and structure of the shapes into consideration.

**Natural geometric continuation.** Let  $e_0$  be the source node that matches  $e'_0$ . The seed region  $r'_0$  we seek should match a seed region  $r_0$  on  $e_0$  for a *perceived* growth of  $e_1$ ; see Figure 4(a). If the growth from  $e'_0$  is to be *natural*, then by an inversion,  $r_0$  should be the intersection of  $e_0$  with a *natural geometric continuation* of  $e_1$ . Since we use parametric representations for the curves and sheets, natural curve or sheet continuation can be obtained by extending the parametric domain. We simplify this by extending the curve or sheet by tangent lines at the boundary since in our case,  $e_0$  and  $e_1$  are always close in proximity. If  $e_1$  is a curve, then  $r_0$  consists of a single point; if  $e_1$  is a sheet, then  $r_0$  consists of a set of points; see Figure 4(b). If a tangent from  $e_1$  does not intersect  $e_0$ , then we find the point on  $e_0$  that is closest to the tangent.

**Structure-aware seed selection.** If, as shown in Figure 4(a),  $e_1$  is connected to only one part  $e_0$ , then the seed part is easy to select — it is  $e'_0$  and the seed region is determined by natural continuation. However, if  $e_1$  is connected to *two ends* to two parts  $e_0$  and  $e_2$ , then as shown in Figure 5(a), seeding by natural continuation would result in two disconnected seed regions on  $e'_0$  and  $e'_2$ . Then the growth of  $e'_1$  would start with two disconnected portions, breaking the connectivity of  $e_1$ . This situation is likely functionally invalid, e.g., consider the case of a back support of a seat. In general,  $e_1$  can be connected to multiple parts in the source. We develop a scheme for seed selection with *structure preservation* in mind.



**Figure 6:** Graph augmentation. Source and target shapes (a) with their corresponding structural graphs (b), shown with matching part-node colors. Split nodes (red and yellow) and one null node (blue) are shown with dashed borders; they augment the graphs to obtain 1-1 node correspondence (b). Augmented edges (c) may have corresponding nodes (red edges) or not (blue edge, involving a null node), leading to quasi edge correspondence.

Let  $S$  be the set of parts connected to  $e_1$  and  $S'$  the corresponding set in the target. We find a subset  $Q'$  of  $S'$ , if it exists, which is composed of the most prominent symmetry group, e.g., the vertical bars in Figures 5(b) and (c). If no such subset exists, then we simply choose any part in  $S'$  as the seed. For ranking symmetry groups, we follow the precedence rules defined in Wang et al. [2011] where rotational symmetries are more prominent than reflectional symmetries and among rotational symmetries, prominence is determined by the symmetry order. When there is a tie among different symmetries, we take the one whose centroid is closest to that of  $e_1$ .

We define the seed region for  $e'_1$  to be a point which lies at the center of the central joint  $q$  of the set of parts in  $Q'$ . If these parts intersect each other, by symmetry, there is a common region of intersection. Then  $q$  is defined as the point in that region which is closest (hence also equidistant) to the curves (or sheets) of the parts in  $Q'$ . Otherwise,  $q$  is the curve or sheet associated with the part, if it is unique, which is closest to and equidistant to all parts in  $Q'$  in the structural graph. If there are two such parts, which could occur when the parts in  $Q'$  form a reflectional symmetry, then  $q$  is the shortest line segment between the curves/sheets associated with the two parts in  $Q'$ . Figure 5(b) shows an example where  $Q'$  consists of a reflectional symmetry; it resolves the issues shown in (a). Intuitively, the point seed shown in (b) induces a growth of  $e'_1$  from a joint between  $e'_0$  and  $e'_2$  and the growth maintains connectivity of  $e'_1$ .

The above approach for seed part selection aims to preserve both part connectivity and symmetry during a growth. An additional measure for symmetry preservation simply requires that seed parts should be selected symmetrically whenever possible. For example, see Figure 5(d), if  $e'_0$  had been chosen as a seed part, then between  $e'_2$  and  $e''_2$ ,  $e'_2$  must be chosen due to symmetry.

#### 4.4 Augmented graph

Due to one-to-many and null correspondences, the source and target structural graphs  $G$  and  $G'$  may not have the same number of nodes or edges. We augment  $G$  and  $G'$  to obtain augmented graphs  $\hat{G}$  and  $\hat{G}'$ , which have exact node correspondence but only quasi edge correspondence. For brevity, we only describe the construction of  $\hat{G}$  as the same process applies to  $\hat{G}'$ . We start with  $\hat{G} = G$ .

**Node augmentation.** For a node in  $G'$  that has no correspondence in  $G$ , we insert a new null node in  $\hat{G}$ . For a one-to-many correspondence  $u \leftrightarrow \{u'_1, u'_2, \dots, u'_m\}$  with  $u \in G$ , we clone one node  $u_i \in \hat{G}$ , called a split node, with correspondence  $u_i \leftrightarrow u'_i$ , for  $i = 1, \dots, m$ . The split nodes  $u_1, \dots, u_m$  replace  $u$  in  $\hat{G}$ .

**Edge augmentation.** With node augmentation,  $\hat{G}$  and  $\hat{G}'$  now have one-to-one node correspondence. However, the newly added split and null nodes are not connected to the original nodes (those coming from  $G$ ) in  $\hat{G}$  yet. Edge augmentation only adds edges in  $\hat{G}$  involving new nodes and only when there is a corresponding original edge (edge coming from  $G'$ ) in  $\hat{G}'$ .

Let  $\{u, v\}$  be a pair of nodes in  $\hat{G}$  with a type combination  $\{\text{null}, \text{null}\}$ ,  $\{\text{split}, \text{split}\}$ , or  $\{\text{split}, \text{original}\}$ , then we add an edge  $(u, v)$ . Such augmentation ensures proper connections between corresponding nodes in the two augmented graphs. Edges between a null node and a split or original node are needed to perform a growth operation. Hence, we connect each null node in  $\hat{G}$  to its seed part, as defined in Section 4.3. See Figure 6 for an example of node and edge augmentation.

**Edge correspondence.** If two edges in  $\hat{G}$  and  $\hat{G}'$  have corresponding end nodes, then they obviously correspond. We also build correspondence for each original edge  $(u, v) \in \hat{G}$  with all original edges  $(u', w') \in \hat{G}'$ , where  $u$  and  $u'$  correspond while  $v$  and  $w'$  may not. Such edge pairings exist due to the topological discrepancies between the source and target and resolving such discrepancies during blending results in topology-varying in-betweens. At last, an edge between a null node  $u$  and its seed part is corresponded with edges between  $u'$  and nodes in the chosen symmetry group. Altogether, edges only have quasi correspondences, as the end nodes of two corresponding edges may not correspond.

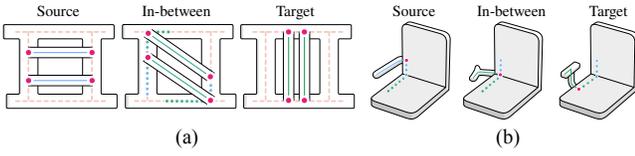
Figure 6 shows various kinds of edge correspondences. In (c), the four augmented edges in red are of the  $\{\text{split}, \text{original}\}$  type and they have edge+node correspondence. The blue edge is  $\{\text{null}, \text{original}\}$ ; it has a corresponding edge (in matching color) but the end nodes of the two edges do not correspond.

#### 4.5 Structural graph and shape blending

Our blending process can be thought of as an animation sequence where the source shape topologically and geometrically transforms towards the target. Every pair of corresponding nodes induces a blending task, which is one of three types: growing, shrinking, or morphing which includes splitting and merging. Each blending task transforms one or more nodes in  $\hat{G}$  to one or more nodes in  $\hat{G}'$ , resulting in an in-between structural graph. Blending tasks are executed in a structure-preserving way. Different ordering and concurrency of blending tasks leads to different blending paths.

**Stochastic sampling.** Each blending path is a stochastically sampled sequence of blending tasks. At each sampling step, we randomly select one node or a set of nodes that belong to a symmetry group in an in-between structural graph and execute the associated blending tasks. The task is always executed all the way to the target along the path and not interleaved with other tasks. Along each path, we automatically sample  $b^*$  in-betweens, where  $b^*$  can be greater than the number of blending tasks so that the in-betweens can sample shapes within the execution of a blending task. Each generated shape goes through implausibility filtering (Section 4.6) with shapes or entire paths below user-set thresholds removed.

**Geometric morph.** This occurs between two corresponding parts and involves interpolation of contact or attachment point positions simultaneously with morphing of curve or sheet shapes. The latter is fairly straightforward with our parametric curve or sheet representation. The morph linearly interpolates the control points. Each curve or sheet node is attached to one or more parts. The positions of the contact points are also linearly interpolated between the source and target, along the curves or sheets that connect the start



**Figure 7:** Two examples of interesting geometry morphs, where interpolation of part contact positions goes through multiple parts.

and end contact positions. Figure 7 shows two examples where transitioning of the contact points goes through multiple parts.

**Split and merge.** Split and merge are topological morphing tasks that are inverses of each other. Each split task “activates” one or more split nodes in an in-between structural graph into curve or sheet nodes, initially positioned according to the source shape. The number of activated split nodes is determined randomly and if symmetries exist among the split nodes, then only symmetric nodes can be activated at the same time by the split task. Once a split node is activated, it can be transformed position-wise and shape-wise to the target by a separate geometric morph, at a later time.

**Growing and shrinking.** Growing occurs at a null node, while shrinking is the inverse operation. For growing a null node  $u$  that originated from a single seed part  $s$ , as shown in Figures 4(a), we linearly interpolate along the path of natural continuation, starting from the seed point on  $s$  to the contact point on  $u$  between  $u$  and  $s$ . Once the growth is executed, the null node becomes a curve or sheet node in the in-between structural graph.

If the edge  $(u, s)$  corresponds to a set of edges in the target involving a symmetry group, then there is first a split at the seed point to start, then we interpolate along each part in the symmetry group to reach the target, as illustrated in Figure 5(d). When such a growth occurs, the null node first becomes a curve or sheet node and then it is *relinked* from the seed part to parts in the in-between structural graph belonging to the symmetry group.

**Structure preservation.** In our current work, we aim to preserve symmetry relations and part connections and have taken such measures throughout. Symmetry groups play a role during seed part selection and node selection when splitting as well as when sampling tasks for a blending path. Seed region selection for multiply connected null nodes is designed to not break a part; see Figure 5(b).

Part disconnection can also arise during a geometric morph since interpolated shape parts are positioned according to global shape alignment. The final interpolated part may be dangling in space. To enforce part connection, we carry out a relinking step when executing a morphing task. If the interpolated part  $p$  is connected to only one part, then it is translated to the contact point on that part. If  $p$  is connected to two parts, then we rigidly transform the control points of  $p$  to attach it at the contact points on the two parts. In the remaining cases, we consider parts with more than two contact points as rigid and simply translate the part’s center towards the centroid of the contact points. However, depending on the order of these operations, there may still be disconnections that are unresolved. This and other cases of structural incoherences are left to the implausibility filter (Section 4.6).

**Geometry reconstruction.** So far, the blending produces sampled sequences of in-between structural graphs. To go from an in-between curve-sheet representation  $G$  to a mesh model  $M(G)$ , for each point  $p$  on  $G$ , we locate its corresponding points  $p_s$  and  $p_t$  on the source and target graphs, respectively. The construction of the medial curves and sheets for each shape implies a mapping  $f$  from

the shape’s surface to its structural graph. For each surface point  $p'$ , we set up a local orthonormal frame with the tangent, normal, and binormal at the point  $f(p')$ . This frame provides a local encoding for  $p'$ . We linearly interpolate the local encodings for  $f^{-1}(p_s)$  and  $f^{-1}(p_t)$ , and through the local frame at  $p$ , compute the surface point corresponding to  $p \in G$ . Finally, we apply Poisson reconstruction [Kazhdan et al. 2006] over all the surface points for  $G$  to obtain its corresponding mesh model  $M(G)$ . Note that only parts that are blended go through mesh reconstruction.

## 4.6 Implausibility filter

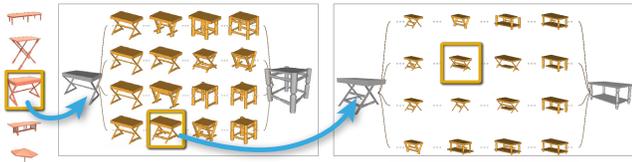
Defining appropriate criteria for plausibility of man-made objects is a non-trivial task which involves functional analysis; this is beyond the scope of this paper. We have taken measures in our blending algorithm to preserve part symmetry and connection. Despite these efforts, there are still complex situations, e.g., unmatched symmetry types in corresponding parts and conflicting constraints caused by multiple connected parts, which cause violation of these structural properties. We develop a series of implausibility filters, applied in post-processing, to remove clearly implausible in-betweens.

**Global reflectional symmetry.** If both the source and target possess a global reflection symmetry, we require the in-between shapes to preserve that symmetry. We detect such symmetries using the method of Mitra et al. [2006] while restricting the search to reflection planes parallel to the upright orientation. We measure the deviation of a shape from perfect symmetry as the Hausdorff distance between the shape and its reflection about the global reflection plane. The source or target shape is regarded as symmetric if the deviation is less than  $0.05\ell$ , where  $\ell$  is the length of shape’s bounding box diagonal averaged over the source and target. An in-between whose deviation is larger than  $0.1\ell$  is filtered out.

**Part group symmetry.** In addition, we require the in-betweens to preserve group symmetries among shape parts. We first detect all groups of mutually symmetric parts from both the source and target shape as in [Wang et al. 2011]. Given an in-between shape, we check existence of each group symmetry based on the presence of every member part of that group, utilizing the available part correspondence. If all members of a group are present in the in-between, we require that these parts preserve the group symmetry.

Currently, we focus on two types of group symmetries, reflection and rotation. To examine symmetry preservation in an in-between, we re-estimate its reflection plane or rotation axis and then compute the deviation again using Hausdorff distance. If the deviation of any present group symmetry is larger than  $0.05\ell$ , the shape is regarded as implausible and filtered out. For reflection symmetry, the reflection plane is computed as the perpendicular bisector plane of the barycenters of the two member parts. For rotation symmetry, its axis passes through the barycenters of all member parts and is perpendicular to the fitted plane of the barycenters.

**Part connection.** If two parts are connected in both the source and target, we require them to remain connected. Given an in-between, similar to the examination of part group symmetry, we first check the existence of all connected part pairs. If a connected pair is present in the in-between, we examine the positions of the contact points recorded on the two corresponding parts. If the distance between the contact points is larger than  $0.05\ell$ , the connection is considered broken and the in-between is filtered out. By examining the attachment of contact points, we can detect both part separation and protrusion.



**Figure 8:** Screenshots of our exploratory modeling tool. Source and target shapes are selected from a gallery. After initial part correspondence with possible user assistance, multiple blending paths are generated automatically. Topologically and geometrically distinct shapes are displayed upfront. The user can scroll between paths and refine the views by clicking between adjacent in-betweens. Any generated in-between can be immediately exported into the gallery to serve as source/target for subsequent blending.

#### 4.7 Exploratory modeling tool

We develop an interactive and exploratory tool that is built on our topology-varying shape blending scheme; see Figure 8 for a screenshot and the figure caption for how to use the tool. A demo of the tool can be found in the accompanying video.

Fitting for an exploratory process, we sample, compute, and filter blending results in fixed-size batches. In practice, we compute 20 paths at a time, which takes less than 10 seconds to produce, averaging over all shape pairs experimented with. The next batch is prepared while the user explores results from the current batch. Scores obtained from implausibility filtering are used to rank the blending paths, with top-ranked paths presented to the user. Note that while implausibility filtering serves to help discard obviously implausible creations, finding interesting novel shapes often entails exploring several batches of generated in-betweens.

To present to the user the most “interesting” in-betweens along each displayed blending path, we select  $k$  topologically and geometrically most distinct shapes, among the  $b^*$  sampled in-betweens. To this end, we perform  $k$ -medoids clustering, where the distance measure combines geometric similarity based on light field descriptor and topological graph dissimilarity based on a spectral descriptor [Wicker et al. 2013]. The  $k$  representative in-betweens, i.e., the cluster centers, are displayed upfront in the exploratory tool.

## 5 Results

In this section, we show results for topology-varying blending, recursive blending (reuse of generated in-betweens), and report timing and other statistics. We compare our modeling results to those obtained by part recomposition [Xu et al. 2012] and volumetric blending using level-sets [Breen and Whitaker 2001]. We also conduct a preliminary user study to evaluate our modeling tool. Finally, we show some unnatural blending results and discuss possible remedies. Extensive sets of results, code, the modeling tool, as well as a video describing our approach and demonstrating the blending tool are all available in the supplementary material.

**Input models.** We collected a set of 110 man-made models with rich topological and structural variations from Google warehouse, TurboSquid, and other online sources. These objects belong to diverse functional categories (furniture, airplanes, robots, etc.) and vary in complexity. All models were made watertight as it is a requirement for skeleton extraction. We used default parameters for the adopted skeleton extraction scheme [Tagliasacchi et al. 2012]. We have found that in practice, minor differences between the extracted skeletons did not significantly affect the blending results. We



**Figure 9:** Three different blending paths automatically generated, exhibiting rich topology variations. Multiple blending paths allow more control and variations in shape creations.

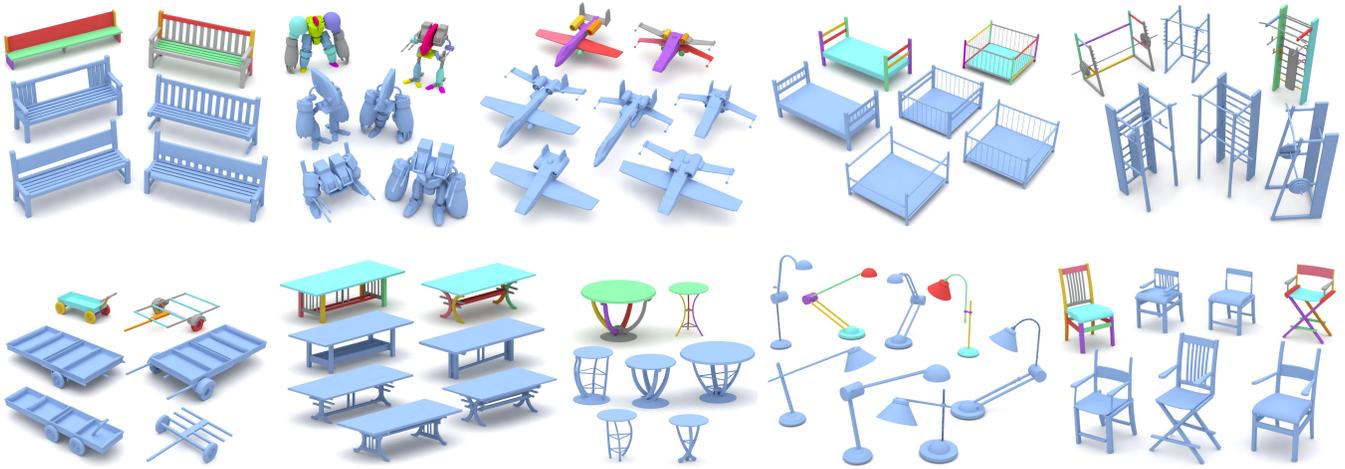
manually segmented some models that did not come with meaningful segmentation. After segmentation, the number of parts in the models range from 4 to 35. While encoding of the edges between parts are automatically computed, whether two parts are connected was specified by the user. This step is necessary since some models obtained online are imperfect and do not contain reliable connectivity information. Inferring such information automatically is not our focus. Once the skeletons of the parts and their connections are defined, the model is reusable for subsequent blending.

**Topology-varying blending.** Figure 10 presents a small gallery of 3D shapes generated by our blending method with various forms of topological changes including part growth, remission, split, and merge. Note also structure preservation in the results. These interesting creations were selected by the user while exploring the set of in-betweens created. Figure 9 shows multiple blending paths leading to diverse sets of results; these results were *automatically* selected (Section 4.7) after implausibility filtering, to appear as the first set of “interesting” generations.

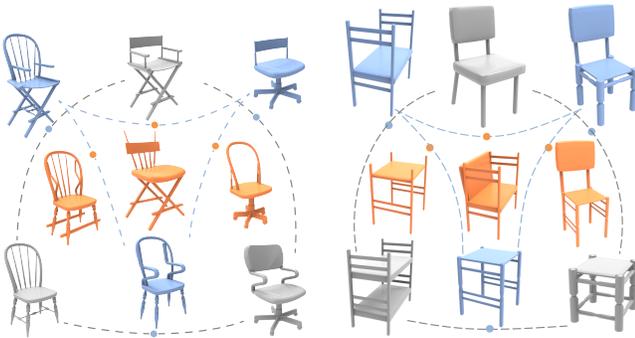
**Recursive blending.** Figure 11 shows newly generated shapes acting as source or targets for recursive blending. As a shape representation, the structural graphs of the newly generated in-betweens are readily usable to initiate new blending processes. However, part correspondences between each blended pair of shapes need to be obtained. With correspondences identified between the initial shape pairs, correspondences among these shapes and the generated in-betweens can be inferred via transitivity and simple inheritance rules, e.g.,  $\{a \leftrightarrow b_1, a \leftrightarrow b_2, a \leftrightarrow b_3, a \leftrightarrow b_4, c \leftrightarrow b_1, c \leftrightarrow b_2, d \leftrightarrow b_3, d \leftrightarrow b_4\}$  implies that  $\{a \leftrightarrow c, a \leftrightarrow d\}$ . The correspondence for the example shown in Figure 11 requires no user assistance beyond those involving the initial shapes. In general, however, automatic inference of part correspondences during recursive shape blending requires further effort in future work.

**Timing and statistics.** The preprocessing time mainly depends on model complexity. A typical model with around 12 parts would require several minutes for clean-up, segmentation, skeleton extraction, and possibly modifying the connected parts. This preprocessing was only required once for each model in the dataset. User-specified part correspondences are typically quite sparse (no more than five for each blending pair in all of our experiments), only involving quick mouse clicks on the parts. The time needed for surface re-sampling depends on the quality desired of the final mesh. We found that 20K samples per part is sufficient for a reasonable level of detail. A typical re-sampling process of the entire input shapes takes less than 10 seconds for each pair of input models.

Most critically, shape blending is highly parallelizable, typically requiring less than 5 seconds per thread. This allows the user to explore many different blending paths at interactive rates. The final mesh reconstruction is done on-demand, as a user selects a shape from the plausible choices presented. For most results shown, we



**Figure 10:** A gallery of topology-varying 3D shapes generated by our blending method (blue). Matching colors indicate part correspondences on the input pairs. Note the variety of topological and geometric variations exhibited by the user-selected in-betweens.



**Figure 11:** Two examples of shape creation via recursive blending. The process starts with three initial shapes colored grey. One in-between shape is generated along each of the three blending paths (grey edges). The three shapes (colored orange) in the middle are generated using the newly created shapes as sources or targets.

choose reconstruction at octree depth 7 which requires a couple of seconds to produce a final mesh. The timing is reported on a machine with a 3.0 GHz quad core processor and with 4GB of memory.

**Comparison to part recombination.** Figure 12 compares our method with set evolution of Xu et al. [2012] which relies on part mutation and crossover (a recombination) to generate shape variations. Discrete part recombination generates new topologies only through replacing or exchanging parts with different topologies. In contrast, our method enables varied transitions between the topologically different parts. In general, our blending produces a richer variety and more granular set of topological variations; see first row. As well, the structure blending is continuous rather than discrete via part exchange only. Another advantage of our method lies in structure preservation. When the input shapes contain incompatible structures, recombining parts with incompatible connections usually leads to incoherent results; see the bottom row. In contrast, structure variations in our method are carried out more gradually over time via continuous blending.

**Comparison to level sets.** Figure 13 shows a few blending results obtained by an implementation [Nielsen and Museth 2006] of the levelset based method of Breen and Whitaker [2001]. Such vol-

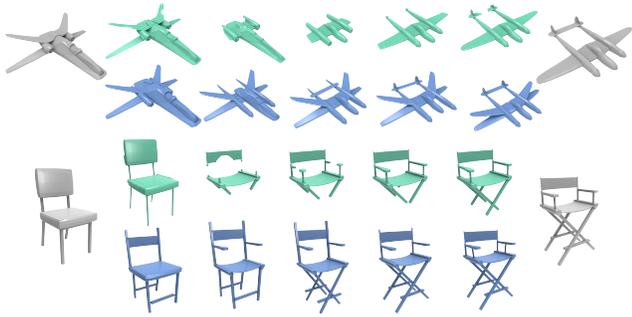


**Figure 12:** Comparison between our blending results (blue) and those from part recombination (green) [Xu et al. 2012]. Top row shows in-betweens generated by the two methods and chosen as the best in the user study. Our results exhibit more granularity in topology variations, e.g., various configurations in the back. Bottom row compares structural coherence of the generated shapes.

umetric blending approaches based on implicit representations do not operate at the part level, but based on Euclidean distance transforms after pre-aligning the source and target shapes. It is quite evident that the in-betweens are not structure-aware or plausible from a functional point of view. On the other hand, volumetric implicits are more powerful representations, compared to our spatio-structural graphs, in terms of modeling highly complex topological transformations; see results in Figure 15.

**User study.** We conducted a preliminary user study aimed at comparing our method with set evolution [Xu et al. 2012] in their capacity as creative modeling tools via shape blending. There are several fundamental differences between the two methods which make the comparison non-trivial. Our method relies on user-assisted correspondence, while set evolution uses fuzzy part correspondence which performs poorly on highly dissimilar input shapes. Also, our method generates in-betweens by sampling a large continuous space of possible in-betweens whereas set evolution relies on random part evolution from a rather limited pool of choices, when the initial population contains only two shapes. Thus, our study is designed to evaluate overall user preference with regards to the few best results generated by the two methods.

The user study was conducted on 40 participants, all graduate students from computing science. The participants were told that we are “researching tools aimed at creating interesting, novel, and plau-



**Figure 13:** Comparison between our results (blue) with those of levelset-based volumetric blending (green). The latter does not operate at the part level and generates implausible results due to a lack of part correspondence and structure preservation.

sible 3D shapes that are a blend of a source and a target shape”. To prepare for the study, we first collected ten pairs of 3D shapes that resulted in five or more reasonable (least geometric distortion) and diverse creations using set evolution. We then applied our method on the same set of pairs and sampled from in-betweens belonging to different blending paths. We collected two sets of five new shapes resulting from both methods and in both cases, we only choose the best results possible so as to fulfill the stated objectives (interesting+novel+plausible) for shape blending.

For each of the ten shape pairs, we asked the participants to select, based on the above stated modeling or shape creation objectives, what they regard as the “two shapes that they like the most and the two they like the least.” Note that we randomly shuffled the generated results from both methods when presenting them to the participants. No other criteria were asked of the users aside from what they regarded as preferred creations.

In 72.6% of the time, users chose shapes from our method as the most liked and only in 24.3% of the time as the least liked. In 9 out of the 10 pairs, our method was ranked as first. While these results from the user study are promising, we should emphasize that it is only meant as a preliminary assessment, rather than a scientific validation, of the effectiveness of our blending method for creative 3D modeling, in comparison to part shuffling or recombination. For example, despite our best efforts, there is still a potential risk of introducing biases in the selection of model pairs and blending results.

## 6 Conclusion, limitations, and future work

We present a method for novel 3D model creation via topology-varying shape blending. To the best of our knowledge, it is the first structure-oriented technique which produces continuous and plausible in-betweens undergoing topology variation. Our work operates at the part level of the input shapes and focuses on the most fundamental topological operations including split and merge, while also allowing growing and gradual removal of extra parts. Plausible and versatile blending results (through multiple blending paths) on man-made objects with moderately complex geometry and topology are presented. All the results shown were obtained in real time, further demonstrating the practical utility of our approach.

The most general form of the problem we are addressing is highly challenging. The current work is only a preliminary attempt. The ultimate goal of our work is to apply topology-varying shape blending for general and *large-scale* novel shape creation.



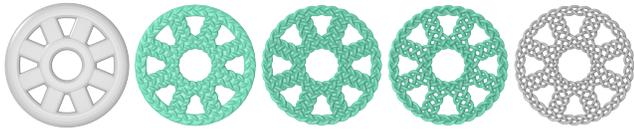
**Figure 14:** Implausible in-betweens generated by our method. Top row: lack of proper segmentation and thus part correspondences result in unnatural in-betweens. Other implausible results are due to violation of various physical or functional properties.

**Limitations.** Plausible blending results clearly depend on meaningful shape segmentation and part correspondence. Figure 14, top row, shows a sequence of implausible blended results from two chairs which are caused by an unnatural segmentation of the source shape; user assistance in part correspondence could not even correct the problem. More implausible in-betweens are shown in the figure and they occur since our implausibility filters are still rather primitive. Functional criteria such as balance, proportion, or physical stability have not been taken into account.

On the technical front, our current algorithm still leaves room for improvement. The structural graph representation is only suited to shapes composed of tubular and sheet-like parts. Though interestingly, this restriction is not as limiting for man-made shapes as one might have expected due to practical considerations during manufacturing. Recursive blending is not yet sufficiently automatic to allow easy generation of many variations from few input shapes. That would require an elaborate correspondence inference scheme for the generated in-betweens. Another limitation is that all topological variations are executed at the part level. For example, a part cannot be split in the middle to generate a handle and a disk-like sheet cannot be morphed into one with a hole in the center. Currently, the morphing scheme is only via linear interpolation; it does not take into account semantic information nor is it guaranteed to be collision-free. Last and not the least, our blending technique focuses on structure and not geometric details. As such, our mesh reconstruction scheme is quite rudimentary; it is not designed to reproduce high-quality blended surface geometry.

**Future work.** Aside from addressing issues raised in the above limitations, it would also be interesting to look for principled ways to improve the granularity of our spatio-structural shape representation to allow more fine-grained topological variation. Figure 15 shows what is possible for implicit representations; our method cannot generate such results. Hybrid representations, e.g., by combining skeletal and volumetric descriptions, seem to offer a promising direction for future investigation. As a means for creative modeling, it may be desirable for the blending process to more conveniently and more interactively incorporate user control, in contrast to the (auto)generate-then-select paradigm we currently adopt. From a theoretical point of view, one is tempted to consider a proper definition of “topology space” which would allow rigorous topological analysis and modeling on 3D shapes.

Finally, in this paper, we do not answer the question of what makes a 3D shape functionally plausible. Instead, we employ conservative criteria to filter out shapes that are considered as implausible, i.e., shapes which fail to preserve certain structural properties of the source and/or target. We leave as future work in-depth studies of functional plausibility of synthesized 3D objects.



**Figure 15:** The level-set approach excels at highly complex topology transformations, when the source and target are well aligned.

**Acknowledgements** We thank all the reviewers for their valuable comments and feedback. We also thank Oliver van Kaick for insightful discussions; Marco Attene and Alec Jacobson for providing code for mesh repair; Michael Kazhdan for the surface reconstruction code; Andrea Tagliasacchi for the skeletonization code; and to all the participants of the user study. This work is supported in part by grants from NSERC Canada (611370), NSF China (61202333, 61363048), CPSF (2012M520392), and the China Scholarship Council.

## References

- ALEXA, M. 2002. Recent advances in mesh morphing. *Computer Graphics Forum* 21, 2, 173–198.
- BERNSTEIN, G., AND WOJTAN, C. 2013. Putting holes in holey geometry: Topology change for arbitrary surfaces. *ACM Trans. on Graph.* 32, 4, 34:1–34:12.
- BREEN, D., AND WHITAKER, R. 2001. A level-set approach for the metamorphosis of solid models. *Visualization and Computer Graphics, IEEE Transactions on* 7, 2, 173–192.
- COHEN-OR, D., SOLOMOVIC, A., AND LEVIN, D. 1998. Three-dimensional distance field metamorphosis. *ACM Trans. Graph.* 17, 2, 116–141.
- CORNEA, N. D., MIN, P., AND SILVER, D. 2007. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. & Comp. Graphics* 13, 3, 530–548.
- FU, H., TAI, C.-L., AND ZHANG, H. 2004. Topology-free cut-and-paste editing over meshes. In *Proc. of Geometric Modeling and Processing*, 173–182.
- HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proc. of SIGGRAPH*, 203–212.
- HUGHES, J. F. 1992. Scheduled Fourier volume morphing. In *Proc. of SIGGRAPH*, 43–46.
- JAIN, A., THORMAHLEN, T., RITSCHER, T., AND SEIDEL, H.-P. 2012. Exploring shape variations by 3D-model decomposition and part-based recombination. *Computer Graphics Forum* 31, 2, 631–640.
- KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. on Graph.* 31, 4, 55:1–55:11.
- KANONCHAYOS, P., NISHITA, T., YOSHIHISA, S., AND KUNII, T. L. 2002. Topological morphing using Reeb graphs. In *Proc. of Cyber Worlds*, 0465–.
- KAZHDAN, M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Proc. of Symp. on Geom. Proc.*, 61–70.
- KREAVOY, V., JULIUS, D., AND SHEFFER, A. 2007. Model composition from interchangeable components. In *Proc. of Pacific Graphics*, 129–138.
- LI, X., GU, X., AND QIN, H. 2009. Surface mapping using consistent pants decomposition. *IEEE Trans. Vis. & Comp. Graphics* 15, 4, 558–571.
- MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3D geometry. *ACM Trans. on Graph.* 25, 3, 560–568.
- NIELSEN, M. B., AND MUSETH, K. 2006. Dynamic tubular grid: An efficient data structure and algorithms for high resolution level sets. *J. Sci. Comp.* 26, 3, 261–299.
- POTTMANN, H., LEOPOLDSEDER, S., AND HOFER, M. 2002. Approximation with active b-spline curves and surfaces. In *Proc. of Pacific Graphics*, 8–17.
- SCHMIDT, R., AND SINGH, K. 2010. meshmixer: an interface for rapid mesh composition. In *SIGGRAPH Talks*.
- SEBASTIAN, T., KLEIN, P., AND KIMIA, B. 2004. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pat. Ana. & Mach. Int.* 26, 5, 550–571.
- SHARF, A., BLUMENKRANTS, M., SHAMIR, A., AND COHEN-OR, D. 2006. SnapPaste: an interactive technique for easy mesh composition. *Vis. Comput.* 22, 9, 835–844.
- SHEFFER, A., PRAUN, E., AND ROSE, K. 2006. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.* 2, 2, 105–171.
- SHNEIDERMAN, B. 2006. Creativity support tools: Report from a US national science foundation sponsored workshop. *Int. J. of Human-Computer Interaction* 20, 6, 61–77.
- SURAZHSKY, T., SURAZHSKY, V., BAREQUET, G., AND TAL, A. 2001. Blending polygonal shapes with different topologies. *Computers & Graphics* 25, 2001.
- TAGLIASACCHI, A., ALHASHIM, I., OLSON, M., AND ZHANG, H. 2012. Mean curvature skeletons. *Computer Graphics Forum (Proc. of SGP)* 31, 5, 1735–1744.
- TAKAHASHI, S., KOKOJIMA, Y., AND OHBUCHI, R. 2001. Explicit control of topological transitions in morphing shapes of 3D meshes. In *Proc. of Pacific Graphics*, 70–79.
- TAKAYAMA, K., SCHMIDT, R., SINGH, K., IGARASHI, T., BOUBEKEUR, T., AND SORKINE, O. 2011. Geobrush: Interactive mesh geometry cloning. *Computer Graphics Forum* 30, 2, 613–622.
- WANG, Y., XU, K., LI, J., ZHANG, H., SHAMIR, A., LIU, L., CHENG, Z., AND XIONG, Y. 2011. Symmetry hierarchy of man-made objects. *Computer Graphics Forum* 30, 2, 287–296.
- WELCH, W., AND WITKIN, A. 1994. Free-form shape design using triangulated surfaces. In *Proc. of SIGGRAPH*, 247–256.
- WICKER, N., NGUYEN, C. H., AND MAMITSUKA, H. 2013. A new dissimilarity measure for comparing labeled graphs. *Linear Algebra and its Applications* 438, 5, 2331–2338.
- XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: set evolution for inspiring 3D shape galleries. *ACM Trans. on Graph.* 31, 4, 57:1–57:10.
- ZHENG, Y., COHEN-OR, D., AND MITRA, N. J. 2013. Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum (Proc. of Eurographics)* 32, 2, 195–204.
- ZHOU, Q.-Y., JU, T., AND HU, S.-M. 2007. Topology repair of solid models using skeletons. *IEEE Trans. Vis. & Comp. Graphics* 13, 4, 675–685.