Geometric and Textural Blending for 3D Model Stylization

Yi-Jheng Huang, Wen-Chieh Lin, *Member, IEEE*, I-Cheng Yeh, and Tong-Yee Lee, *Senior Member, IEEE*

Abstract—Stylizing a 3D model with characteristic shapes or appearances is common in product design, particularly in the design of 3D model merchandise, such as souvenirs, toys, furniture, and stylized items. A model stylization approach is proposed in this study. The approach combines base and style models while preserving user-specified shape features of the base model and the attractive features of the style model with limited assistance from a user. The two models are first combined at the topological level. A tree-growing technique is utilized to search for all possible combinations of the two models. Second, the models are combined at textural and geometric levels by employing a morphing technique. Results show that the proposed approach generates various appealing models and allows users to control the diversity of the output models and adjust the blending degree between the base and style models. The results of this work are also experimentally compared with those of a recent work through a user study. The comparison indicates that our results are more appealing, feature-preserving, and reasonable than those of the compared previous study. The proposed system allows product designers to easily explore design possibilities and assists novice users in creating their own stylized models.

Index Terms—Computer graphics, modeling

1 Introduction

With the evolution of industrial aesthetics, style has become a critical component of product identity and economic value. The need for interesting, meaningful, effective, and enjoyable sensory experiences is prevalent. Therefore, product designers need to fulfill the basic requirement of functionality and meet the increasing demand for product style. A common design technique is to integrate artistic or design elements into a normal product to improve product style [1]. The resulting products, which are referred to as *style products*, are commonly used in daily life. Fig. 1 shows several examples. However, creating a style product by using conventional modeling tools is time consuming. Manually modifying geometrical details and setting texture coordinates are tedious. An approach that can automatically incorporate the geometric and textural features of a style model into a 3D model with several hints from users is presented in this paper. The style of a model is transferred to another through a process called *model stylization*. "Style" is defined in this study as the characteristics of representative objects that possess an interesting appearance or certain aesthetic values. These characteristics include color appearances (texture) and geometric features.

Inspired by the concepts of style product design, such as the effect of imitating distinctive perceptual features [2] and mimicking the appearances of natural subjects [1], as well as observations on existing style products, three commonly used strategies in style product design are identified in this study. One strategy is to change the form of a product. The form of a product may be changed to fit a certain style, such as the products shown in Figs. 1a and 1b. In these two examples, products are formed similar to a bear or capybara. Another strategy is to attach or merge several components of the style model to the product without deforming the geometry of the product. For example, in Figs. 1c and 1d, legs and ears are attached to the products. The last strategy is to change the texture. Many style products include the colors or textures of a given style, as presented in Figs. 1e and 1f. Only the texture of the functional model is changed, and the original form is unmodified in the products.

The proposed approach was designed based on these strategies. With *base* and *style models* as input, the proposed system produces several blended models that preserve the user specified shape features of the base model and the decorative attributes of the style model. The goal of shape feature preservation is to preserve the functionality of the base model. Several parts of the base model, such as the cup handle in Fig. 1a may have a specific functionality. Users specify these parts as shape features and request the proposed system to maintain the geometry of these parts and their connecting topology in the blending process. Preserving shape features is not necessary for preserving functionality but a straightforward approach. Understanding functionality beyond shape is out of the scope of this paper. The word "shape" implies geometry and topology in this paper.

Fig. 2 shows an overview of the proposed approach. The approach involves preprocessing, topology blending, and surface blending. In preprocessing, input models are

- *Y.-J. Huang and W.-C. Lin are with the Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan. E-mail: jennyhuang914@gmail.com, wclin@cs.nctu.edu.tw.*
- *I.-C. Yeh is with the Department of Computer Science and Engineering, Yuan Ze University, Taoyuan 32003, Taiwan. E-mail: ichenyeh@gmail.com.*
- *T.-Y. Lee is with the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan 70101, Taiwan. E-mail: tonylee@mail.ncku.edu.tw.*

Manuscript received 1 June 2016; revised 27 Dec. 2016; accepted 11 Jan. 2017. Date of publication 25 Jan. 2017; date of current version 29 Dec. 2017. Recommended for acceptance by T. Igarashi.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2017.2657751

Fig. 1. Examples of stylized products.

analyzed, and the features of the models that are utilized in the succeeding blending steps, such as curvature, segmentation, and topology, are extracted. In topology blending, two models are blended at the topological level. The blended topology describes the style components that are attached to or blended into the base model. Users can specify shape features in this stage. In surface blending, the geometry and texture of the two models are blended by applying a mesh morphing technique.

The experimental results show that the output models reflect the characteristics of the style models while preserving the user-specified shape features of the base models. The proposed system produces multiple blended models for each pair of style and base model. Each model is generated based on the three design strategies. Considering that the morphing technique is utilized to blend the two models, users can easily create a blended model that resembles its base or style model more by simply changing the blending degree. The system can be utilized as a simple design tool to enable amateurs to blend two models or as an auxiliary tool that product designers can use to create ideas and obtain a draft for editing.

This study provides the following contributions. First, a model stylization framework is established by geometrically and texturally blending a functional model with a style model. Second, a method for blending 3D models at the topological level is proposed. The method can suggest various plausible combinations of model components (see examples in Fig. 2d). A user can select different combinations depending on his or her interest and preference. Third, a morphing technique is developed. This technique can establish feature correspondence according to textural and geometric information while preserving the important characteristics of the two input models during the morphing process (e.g., preserving the texture patterns of the style model and the geometric features of the base model).

2 Related Work and Background

Model stylization is a popular subject among researchers. Li et al. [3] identified style as a set of 2D curves and transferred style by putting the curves on target contours. Han et al. [4] extended the work of Li et al. [3] to 3D models. Ma et al. [5] measured the analogous relationship between 3D models and applied this relationship to synthesize a new 3D model that possesses a specified style. Unlike these studies, in which "style" is defined as concise curves, "style" is referred to as the characteristics of representative objects in the current study. These characteristics include color appearances (texture) and geometric features. Recently, Duncan et al. proposed a computational approach to zoomorphic design [6] which blends a man-made mesh and an animal mesh. Unlike this work [6], the proposed method supports texture and morphing technique.

The proposed approach is closely related to 3D model blending. The concept of blending 3D models is not new in computer graphics. Various significant approaches were proposed previously. Alhashim [7] presented a comprehensive literature review on this field. This section only describes related studies on mesh morphing [8] and assembly-based model synthesis [9].

2.1 Mesh Morphing

Mesh morphing is commonly utilized in computer animations and films to deform one mesh to another through a seamless transition. Mesh morphing solves the correspondence problem and requires a blending operator to interpolate two meshes continuously. To smoothly transform a mesh, several studies [8], [10], [11], [12], [13] investigated the automatic alignment of features when the correspondence is set. A common strategy to establish correspondence between

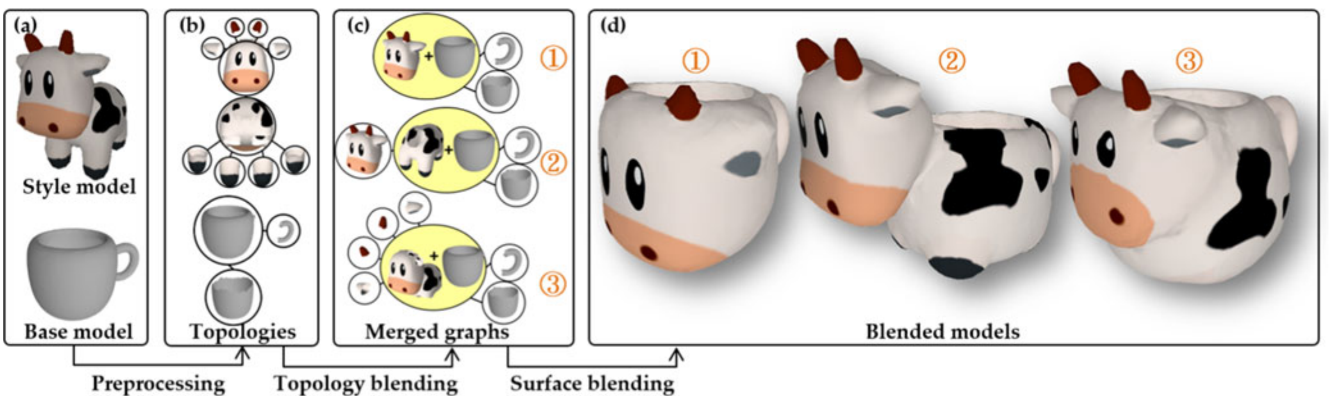


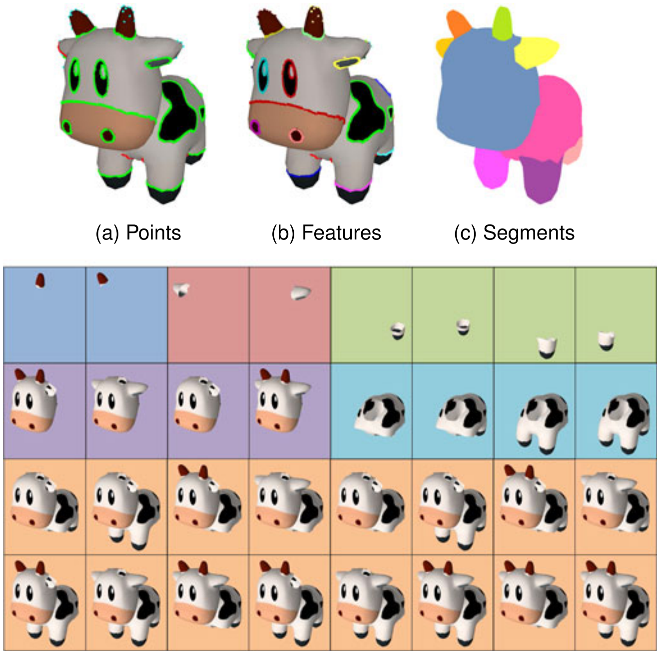
Fig. 2. Approach overview. (a) Input models. (b) Topologies of input models. Each model is presented as a graph whose nodes represent the segments of the model, and the edges show the connection between these segments. In this example, the base model has three segments: the exterior surface, inner surface, and handle of the cup. The exterior surface connects the inner surface and the handle. The segments of the inner and exterior surfaces allow the form of the inner surface to be maintained as a container while the geometry of the exterior surface is changed for different styles. (c-d) Merged graphs and output models. The labels ①②③ indicate three different cases.

meshes is to parameterize two meshes and map them to the same space, such as a disk [14] or a sphere [15], [16]. A variant of this strategy involves dissecting meshes into pieces and separately mapping each piece to a topological disk [17]. Several feature alignment methods, including matching prominent regions [8], embedding meshes in a spectral domain [10], deforming meshes [11], extracting curve skeletons for voting [12], and separating meshes into correspondence regions through region growing [13], have been proposed.

These methods map meshes intelligently but do not consider texture features when establishing correspondence. An automatic feature alignment technique that considers both geometry and texture when creating correspondence is presented in this paper. Texture features, such as characteristics or representative patterns of style models (e.g., facial features and special body geometries), are maintained in the morphing process. The proposed method does not require obvious feature correspondences between two models.

2.2 Assembly-Based 3D Model Synthesis

In Funkhouser et al.'s pioneering work on assembly-based modeling, "Modeling by Example" [9], they manually cut models into components and produced a new 3D model by combining components retrieved from a database. Kreavoy et al. [18] extended this work by automatically cutting a group of models and calculating interchangeable components between the models. Several component suggestion systems have also been proposed. In these systems, users provide a rough-shaped component, and the systems suggest possible components from a database [19], [20]. Chaudhur et al. improved the suggestion mechanism by applying a probabilistic model [21] and further used the probabilistic model to generate a group of meshes automatically [22]. Several mesh synthesis approaches have also been proposed aside from assembly-based modeling. Xu et al. [23] employed the genetic algorithm to synthesize many descendants of input models. Jain et al. [24] blended in-between geometries of given models by interchanging proximal components. Zheng et al. [25] defined symmetric functional arrangements and used them to synthesize functionally plausible models. The objective of the current study is similar to that of Zheng et al. [25]. However, the current study focuses on the variations of topologies. Alhashim et al. [26] blended two models topologically and geometrically and generated a series of in-between models. A limitation of these previous studies is that their methods can only blend models from the same family. The zoomorphic design approach [6] was recently used to create a series of zoomorphic meshes by merging an arbitrary functional mesh and an animal mesh. However, this approach considers geometrical information only. Our approach can blend models from different types; besides, we further consider color appearance to create more appealing results and we explore the possibility of various topological combinations. These features help users easily create varied and interesting results. In addition, the morphing technique is adopted to create in-between models to enable users to deform the models through various and flexible means. These differences provide several advantages. First, the proposed approach considers texture, which can efficiently convey the attribute of the style model. Second, our approach has less manual work in preprocessing. Third, the morphing technique is used to deform models instead of adopting linear blending skinning (LBS) and free-form deformation (FFD) methods used in [6]; the blended models are smoothened in many cases. The experimental comparison and user study show that participants prefer the results of our approach over the results of zoomorphic approach [6].



(a) Points (b) Features (c) Segments

(d) Parts and groups

Fig. 3. Information extracted from an input model in the preprocessing stage. (a) Extracted points. The green ones are texture edge points, the red ones are ridge points, and the light blue ones are curvature points. The definition of each type of points is presented in Section 3. (b) Features. Extracted points are grouped into several clusters with different colors. Each cluster represents a feature. (c) Segments. Each color denotes a segment. (d) Parts and groups. Each box represents a part. Boxes with the same color represent the parts that form a group. This case has 32 parts and six groups.

3 Preprocessing

In the preprocessing stage, the input models were automatically analyzed according to previous studies [27], [28], [29], [30], [31] with a few user-specified parameters. Figs. 2b and 3 illustrate the extracted information, including texture edge points, curvature points, ridge points, features, segments, parts, and topology. Each model needs to be preprocessed only once, and the extracted information can be used for blending with different models.

Texture edge points, curvature points, ridge points, boundary points, and features were the information used in the surface blending stage. In other words, the geometry and texture characteristics are represented in 3D points, and the positions of these points were recalculated in the surface blending stage. These points are located on the surface of a model. The 3D position, corresponding barycentric coordinate, and texture coordinate of each point were stored. Texture edge points located at texture discontinuities were extracted with the Canny edge detection algorithm [27]. Curvature points are vertices with large mean curvatures [28]. Ridge points are curves along which one of the principal curvatures has an extremum. These points were extracted with

TABLE 1
Variables Used in Sections 4 and 5

Variables	Definitions
B^s	Base segment
$\mathbf{B}^p$	Base part
S^s	Style segment
$\mathbf{S}^p$	Style part
$\mathbf{p}$	Point in a style feature
$\mathbf{d}$	Target position of a point $\mathbf{p}$
$\mathbf{F}$	Feature containing a group of points
$\mathbf{e}$	Edge vector connecting two features
$\mathbf{l}$	Link vector connecting two points in a feature
$\mathbf{M}$	Transformation matrix
$\mathbf{f}$	Triangle face in a mesh
$\mathbf{v}$	Vertex of a triangle face
$\mathbf{r}$	Barycentric coordinate

the method proposed in [29]. With a density-based clustering algorithm [30], these points were divided into several clusters according to their 3D positions. Each cluster was defined as feature $\mathbf{F}$, which may contain texture edge points, curvature points, ridge points, or all of these points. Another type of points called boundary points was dynamically generated according to the topology of the blended model. The boundary points are introduced in Section 5.5.

Segments, topology, and parts were the information used in the topology blending stage. Segments are disjointed submeshes of an input model. These segments were obtained with the shape diameter function segmentation algorithm [31]. Each input model was assumed to possess top and frontal sides (directions), which are provided by users. A model coordinate was then defined in an input model, with the coordinate origin located at the center of the model and the z-axis and y-axis toward the top and frontal directions, respectively. With this model coordinate, the proposed system identifies symmetrical segments along the YZ plane of the model. Users can specify the segments to be preserved in the output blended model. A topology of an input model is a structural graph in which each node represents a segment and an edge denotes the connection between two segments. Parts describe all possible combinations of the segments of an input model. They were generated by traversing the structural graph (topology) of an input model and gradually combining connected segments into parts. Symmetrical segments were operated together. Different parts may contain the same segment. The concepts of parts enhance the variations of output models (see Fig. 3d).

4 Topology Blending

To increase the variety of the results, the topologies of two input models were blended at the part level, as shown in Fig. 2c. By decomposing style and base models into multiple parts, different combinations of mixing and attaching operations could be applied on these parts to generate various possible outcomes. Some design heuristics can be considered when topology blending is executed. For instance, not all base parts were blended with style parts (e.g., the handle of the bear cup in Fig. 1a), and not all style parts were attached to base parts (e.g., the hands and legs of Spongebob are disregarded in Fig. 1e). Based on these considerations, *operational vector* and *merged graph* were designed to represent different operational and topological combinations of the style and base parts of the topologies of two input models. A *plausibility tree growing* process was also developed to search for plausible operations efficiently. Under this design, implausible results could be effectively excluded. To further explain the notations, the variables used in Sections 4 and 5 are tabulated in Table 1.

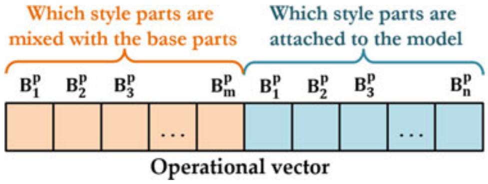


Fig. 4. Definition of an operational vector of $2\ \mathrm{m}$ base parts.

4.1 Operational Vector and Merged Graph

An operational vector consists of two portions that represent mixing and attaching operations, as defined in Fig. 4. The design of the operational vector allows various combinations of mixing and attaching base and style parts to be compactly represented. The mixing portion (orange) indicates the parts to be mixed, and the attaching portion (blue) denotes the style parts to be attached to the base model. If the value of the ith element of the mixing portion is j, base part i and style part j would be blended geometrically and texturally. If the value of the ith element is ϕ, base part i would not be blended with any style part. For the attaching portion, if the value of the ith element of the mixing portion is j, style parts $\overline{\mathbf{S}^p_j}$ will be attached to base part $\mathbf{B}^p_i$, where $\overline{\mathbf{S}^p_j}$ is the style model excluding the part $\mathbf{S}^p_j$. We define B^s_i and $\mathbf{B}^p_i$ as the ith base segment and part, respectively. S^s_j and $\mathbf{S}^p_j$ as the jth style segment and part, respectively.

Given an operational vector and two input topologies, a *merged graph* that represents a blended model could be constructed. Each node in a merged graph could represent a base part, a style part, or a blended part of both. The blended part is denoted as a yellow node in a merged graph (Fig. 2c) and is called a *mixture node*. This designation indicates that a base part and a style part are mixed geometrically and texturally in the surface blending stage. Each edge in a merged graph represents the connection between two parts. Fig. 5 provides an example of the relations among topologies, parts, an operational vector, and a merged graph.

4.2 Plausibility Tree Growing

Although the combination of m base parts and n style parts could generate $(n+1)^{2m}$ possible operational vectors, many of these combinations are implausible and could be discarded in the early stage. Instead of listing all combinations, we discover plausible operational vectors by growing a decision tree, as illustrated in Fig. 6. To fill the ith element of the vectors, the tree grows nodes, which contain all possible values in level i, namely, $\{\phi, 1, 2, \ldots, n\}$. For each newly added node, some design heuristics (to be discussed in the next paragraph) were employed to test if the outcome is plausible. Thus, the time devoted to identifying plausible operational vectors is reduced by considering and expanding only plausible nodes to the next level. Once a node reaches the $(2m)$th level, a plausible combination is generated. Back

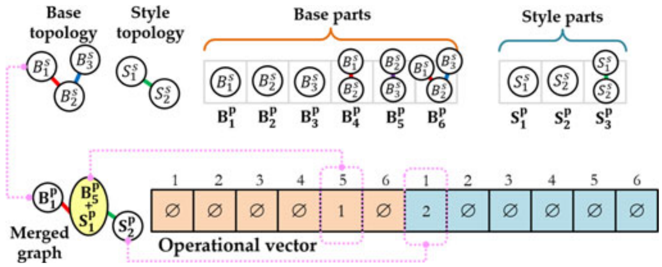


Fig. 5. Example of the construction of a merged graph. Each part of the base model, which is provided by a user, is expected to have its purpose. Thus, each part of the base model is assumed to be preserved. The proposed method can be easily extended to not to preserve all parts of the base model.

tracing could be conducted from the end of the searching path, and a plausible operational vector could be obtained.

In addition, because the operations on similar parts generate similar results, a grouping technique was applied to reduce similar results and further accelerate the tree-growing process. Similar parts were aggregated into a group, which was then regarded as a single node during the tree-growing process. Two parts were grouped together if their visual difference, which is computed by using the definition in [32], is less than threshold σ_1. Fig. 3d illustrates an example of parts and groups.

Plausibility Testing. Rules from six different aspects are recommended to filter out implausible results. These rules can be easily modified and expanded in the proposed framework. In the current implementation, the first three rules are mandatory. The other rules are optional, and users can control their strictness by setting thresholds. Rule 1 is enforced to maintain the correctness of topology blending. Rule 2 is enforced to satisfy user design. Rule 3 is enforced because it is a common design principle, as suggested by [33]. Rule 4 reduces the possibility of self-intersection of the blended model. Rules 5 and 6 maintain the original attributes of input models. Fig. 7 shows several examples with inappropriate thresholds. The plausibility rules are described as follows:

1) A merged graph cannot contain repeating segments.
2) *Importance preserving*: An important base segment (specified by users) cannot be blended with any style part, and all important style segments (specified by users) must exist in a merged graph.
3) *Symmetry preserving*: A symmetrical style part cannot attach to a base part individually, i.e., a pair of symmetrical style parts must attach together. Similarly, a symmetrical style part cannot blend to a symmetrical base part individually.

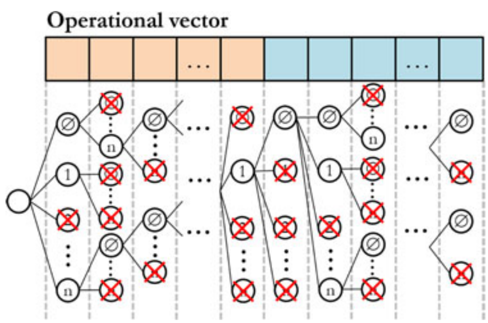


Fig. 6. Searching for plausible operational vectors by tree growing.

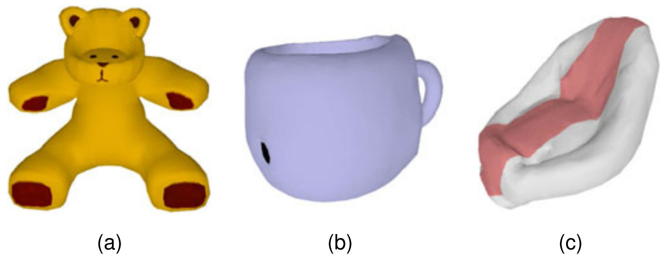


Fig. 7. Implausible cases of blended models. (a) Bear beanbag. The bear's face is deformed to the geometry of a beanbag. The functional attribute of the beanbag is not preserved because the shape of the blended model differs significantly from that of the base model. Rule 6, which preserves the volume of the base model, can filter out this type of implausible model. (b) Elephant cup. Without the important segments of the elephant (ears and nose), the cup cannot present the characteristics of the elephant. This case would be filtered out if the importance rule is enabled (Rule 2). (c) Rabbit beanbag. One of the rabbit ears is deformed to the form of a beanbag. The case would be ruled out if the symmetry rule is enforced (Rule 3).

4) *Collision avoidance*: The penetration rate of the axis-aligned bounding box (AABB) of two parts should be less than σ_2.
5) *Similarity*: The visual difference [32] of two parts in a mixture node should be less than σ_3.
6) *Size preserving*: The AABB volume of the output model should not exceed σ_4 times the volume of the base model.

Threshold Adjustment. Thresholds σ_1 to σ_4 influence the amount and variety of the output models. However, adjusting individual thresholds may be tedious. A single parameter, namely, *diversity*, is provided in the proposed system to allow users to control the amount and variety of the output models easily. The system automatically determines the threshold values according to a diversity value set by users. Fig. 8 shows the relationship between diversity and these thresholds.

5 Surface Blending

Morphing technique [34] was applied to blend the parts in a mixture node. First, the correspondence between the features of two parts was determined based on the textural, geometric and topological information extracted in the preprocessing stage (Section 3). As surface blending aims to preserve the characteristics of style parts, the proposed system establishes feature correspondence while maintaining the form of texture patterns and the relative positions among the texture patterns on style parts. For the texture pattern, the system enforces the deformation of patterns as

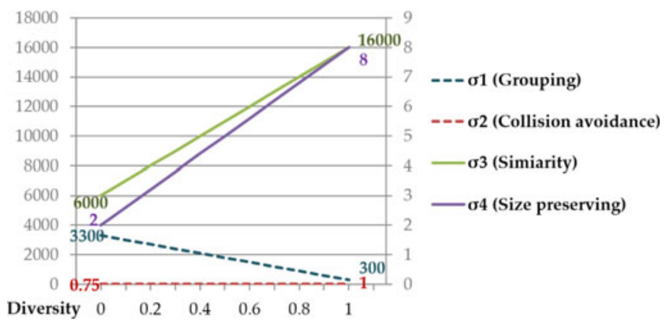


Fig. 8. Threshold values of σ_1 to σ_4 are automatically determined by the diversity value. This figure shows the relationships between diversity and thresholds.

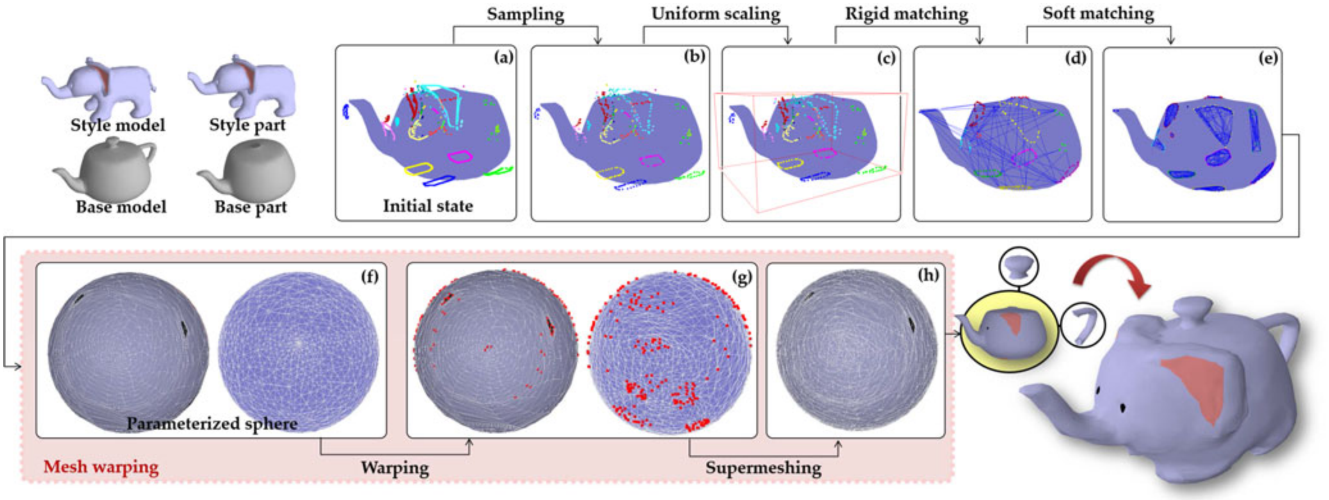


Fig. 9. Flowchart of surface blending.

rigid as possible to reduce the distortion of texture patterns. For the geometric features, the system maps the convex or concave regions of a base part to those of a style part whenever possible.

Fig. 9 shows the flowchart of surface blending. The system computes the correspondence between the features of two parts. The corresponding position on the base surface of each point in a style feature is calculated in a global-to-local manner, from uniform scaling and rigid matching to soft matching. Then, the surface of a base part is morphed to the corresponding style part by employing parameterization and warping on the spherical intermediate domain.

5.1 Initial Aligning and Scaling

The AABB center of a style part was aligned with that of a base part, as shown in Fig. 9a. Voxel-based downsampling was utilized. The system creates a 3D voxel grid over the points of a feature. All points in each voxel are approximated by their centroid point to reduce the number of points in a feature. The style features are uniformly scaled, such that the distributions of style features are fitted with the AABB of a base part. Figs. 9a, 9b, and 9c show each step of the preprocessing stage of surface blending. The border lines in Fig. 9c indicate the AABB of the base part.

5.2 Rigid Matching

Rigid matching maps each style feature onto the surface of a base part and preserves the geometric and textural characteristics of all style features. The relative positions between features should be maintained when they are mapped onto the surface of a base part. For example, if feature $\mathbf{F}_i$ is on the opposite side of feature $\mathbf{F}_j$, feature $\mathbf{F}_i$ should still be on the opposite side of feature $\mathbf{F}_j$ after rigid matching. Iterative closest point (ICP) method was applied to map the feature positions to the surface of a base part for blending and maintain the rigidity of the entire set of features by preserving the relative positions among them. For a pair of features $(\mathbf{F}_i, \mathbf{F}_j)$, edge vector $\mathbf{e}_{ij}$ was used to describe their geometric relationship. Our method tries to maintain direction and length of the edge vector. Fig. 9d illustrates the edges among features. The rigid matching problem was formulated as a nonlinear optimization problem and solved by using the Levenberg-Marquardt algorithm. The objective function is as follows:

$$\min_{\mathbb{T}} \sum_{\mathbf{F}_i \in \mathbb{F}} \left(\sum_{\mathbf{p}_i^k \in \mathbf{F}_i} \|\mathbf{M}_i \mathbf{p}_i^k - \mathbf{d}_i^k\|^2 + w_r \sum_{\mathbf{e}_{ij} \in \mathbb{E}} \|\mathbf{e}'_{ij} - \mathbf{e}_{ij}\|^2 \right), \quad (1)$$

where $\mathbb{T} = \{\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_n\}$ is the set of homogeneous transformation matrices that we solve for transforming each feature of a style part onto the surface of a base part. $\mathbb{F} = \{\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_n\}$ is the set of all features of the style part. $\mathbf{p}_i^k$ is the kth point of feature $\mathbf{F}_i$ and $\mathbf{d}_i^k$ its target position. The process of determining the target position of $\mathbf{p}_i^k$ is determined in the next paragraph. An edge vector is defined as $\mathbf{e}_{ij} = \mathbf{p}_i^* - \mathbf{p}_j^*$, where $\mathbf{p}_i^*$, $\mathbf{p}_j^*$ are the two closest points between features $\mathbf{F}_i$ and $\mathbf{F}_j$ before transformation. $\mathbf{e}'_{ij} = \mathbf{M}_i\mathbf{p}_i^* - \mathbf{M}_j\mathbf{p}_j^*$ is the edge vector $\mathbf{e}_{ij}$ after transformation. $\mathbb{E}$ is the set of all edges that link two associated features in $\mathbb{F}$. In addition, the spatial relation of nearby features should be preserved more strictly to ensure the integrity of the texture pattern. Therefore, instead of preserving all relative positions among features, only the relative positions of nearby features are maintained and a large rigid transformation for far features is allowed. Accordingly, all edge vectors $\mathbf{e}'_{ij}$ of a pair of features $(\mathbf{F}_i, \mathbf{F}_j)$ are sorted in ascending order based on lengths, and only the top 60 percent of the edges of a feature are retained when $\mathbb{E}$ is constructed.

Before optimization, for each point $\mathbf{p}_i^k$ in a feature, its target point $\mathbf{d}_i^k$ on the surface of the base part to be blended is determined as follows. If $\mathbf{p}_i^k$ is a texture edge point or a boundary point (for attaching a style part to a blended part), $\mathbf{d}_i^k$ is a point on the surface of the base part that is closest to $\mathbf{p}_i^k$. If $\mathbf{p}_i^k$ is a curvature or ridge point, we look for the closest curvature or ridge point on the surface of base part. However, if the distance of a matched pair is farther than a threshold, the point would be ignored to prevent outliers from biasing the optimization results and to ensure a reasonable correspondence between $\mathbf{p}_i^k$ and $\mathbf{d}_i^k$.

The first term in Equation (1) is related to local matching, and the second term is for global matching. Weight w_r is used to control the balance between the terms. If w_r is large, the features will preserve the relative position among features although the features may be far from the surface of the base part. If w_r is small, the features are close to the surface although the relative position among features may not be held. Therefore, instead of setting w_r as a constant, we iteratively optimized the feature positions as follows: we first optimized the relative positions of features and then locally adjusted the position of each feature point. In the current implementation, the optimization was executed three times. In each iteration, the position of $\mathbf{p}_i^k$ was updated by using previous optimized results, and the corresponding targets $\mathbf{d}_i^k$ were searched again. w_r was set to 1.0, 0.3, and 0.09 in the first, second, and third iteration in the current implementation.

5.3 Soft Matching

All points of a feature in rigid matching are regarded as a rigid point cloud, and the same transformation is applied to these points. Hence, these points may be close to the surface of the base part and must undergo soft matching to map all points of a feature onto the surface. The concept of as-rigid-as-possible was adopted to preserve the feature patterns. Link vectors $\mathbf{l}_{jk} = \mathbf{p}_j - \mathbf{p}_k$ were constructed between any two points in a feature, and the distance of these vectors were kept as fixed as possible. Fig. 9e shows the links of features (blue lines). To ensure that a point is on the surfaces, each point's barycentric coordinate $\mathbf{r} = (r_1, r_2, r_3)$ on a triangle $\mathbf{f} = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ on the surface was computed. $\mathbf{v}_i$'s are the vertices of the triangle. For each feature $\mathbf{F}$, the new position of its points after soft matching was calculated by optimizing a set of barycentric coordinates $\mathbb{R} = \{\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^n\}$ with the objective function

$$\min_{\mathbb{R}} \sum_{\mathbf{p}^k \in \mathbf{F}_i} \|B_p(\mathbf{f}^k, \mathbf{r}^k) - \mathbf{p}^k\|^2 + w_s \sum_{\mathbf{l}_{jk} \in \mathbb{L}_i} \|\mathbf{l}'_{jk} - \mathbf{l}_{jk}\|^2. \quad (2)$$

The first term of the objective function minimizes the movement of each point, and the second term maintains the relative position among the points (i.e., maintaining a texture pattern). In the first term, $\mathbf{f}^k$ is the triangle on which $\mathbf{p}^k$'s target position $\mathbf{d}^k$ is located. $B_p(\mathbf{f}^k, \mathbf{r}^k) = r_1^k\mathbf{v}_1^k + r_2^k\mathbf{v}_2^k + r_3^k\mathbf{v}_3^k$ is a point on triangle $\mathbf{f}^k$ described by a barycentric coordinate $\mathbf{r}^k$. In the second term, $\mathbf{l}'_{jk} = B_p(\mathbf{f}_j, \mathbf{r}_j) - B_p(\mathbf{f}_k, \mathbf{r}_k)$ is a link vector that represents the spatial relation of two points in a feature after soft matching. Equation (2) can be converted to a system of linear equations which can be solved by using any standard matrix-solving algorithm. w_s was set to 1 to solve the optimization problem.

$\mathbb{L}_i$ is the set of all links defined for feature $\mathbf{F}_i$. Two types of links are used in soft matching. The first type of links connects the points close to each other. These links can maintain the continuity and smoothness of a feature pattern locally. The second type of links randomly joins two points together and can maintain the global form of a feature. In several cases, two features may overlap spatially after the matching process. According to our observations and experiments, these points inside the overlapped space may generate improper mapping and cause the self-intersection of a style part. Therefore, the points of two features that are too close after soft matching were removed to solve this problem. In the implementation, each point in a feature had eight nearby links (type 1) and two random links (type 2).

The second term in Equations (1) and (2) is the difference between two vectors rather than their lengths. This design avoids rigid and soft matching to rotate a feature because each model is presumed to have a top and frontal direction, and textural or geometric features must maintain their original orientation in their model coordinates.

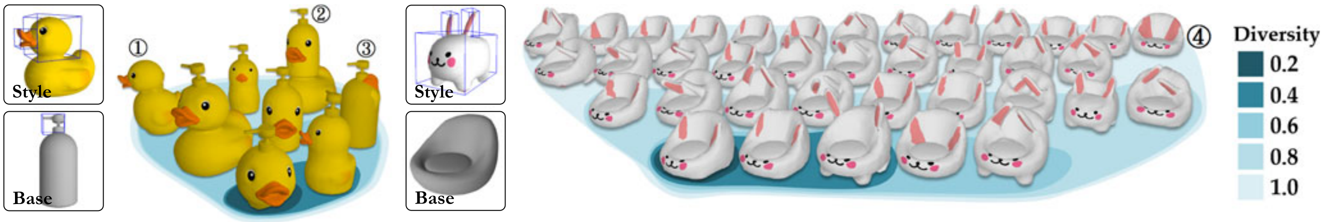
5.4 Mesh Warping

After rigid and soft matching, a set of correspondences between the features of style and base meshes was obtained. The points of features were aligned by applying a foldover-free warping method [35] on spherical intermediate meshes. Figs. 9f, 9g, and 9h illustrate the mesh warping procedure, which includes three steps: First, both style and base meshes were mapped to a unit sphere by utilizing relaxation-based spherical parameterization [15] or an area-preserving spherical parameterization method [16]. The relaxation-based approache [15] could produce good blended results that are smooth on their texture borders and blended geometries. However, not all meshes can be parameterized by [15]; in this case, the area-preserving spherical parameterization [16] is adopted. Second, the two spherical parameterizations were aligned with the features by using foldover-free warping [35]. Third, for smooth transformation between the style and base meshes, a supermesh that contains both style and base meshes information was built by adopting a previous approach [34].

In the second step of the mesh-warping procedure, meshes cannot be directly warped based on the correspondences of the feature points because feature points may not be not mesh vertices. Therefore, the vertices of a triangle are warped if any feature point exists inside the triangle. The red vertices in Fig. 9g indicate an example of these vertices. Because only the correspondence between the feature points was determined, the technique of weighted averages on surfaces [36] was utilized to deduce the correspondence of mesh vertices from these feature points. When the correspondence of mesh vertices was identified, two meshes were warped iteratively until the vertices of one mesh reached the target positions. Warping a mesh to align the target position may cause triangle foldover. When a foldover triangle was detected, the mesh was recovered to the previous iteration, and the mesh around the folded triangles was refined. When the number of folded triangles exceeded 5,000, warping was stopped and supermeshing was carried out directly.

5.5 Connection and Boundary Points

An edge in a merged graph represents the connection between two parts. A connection can be either from the edges of the topology of style model or base model. We refer the node in a merged graph containing only a base part as a base node, and that containing only a style part as a style node. The connection between a mixture node and a base node was obtained from the topology of base model because we assume the entire base model will be preserved and a mixture node will contain a base part. The connection between two mixture nodes was also obtained from the topology of base model. The connection for the style node, which was added by the attaching operation, was obtained from the topology of the style model.



(a) Duck shampoo (b) Rabbit beanbag

Fig. 10. Two sets of blended models.

In the surface blending stage, a connection represents as the boundary points from a style model or a base model. To maintain the form of boundaries between two parts in a merged graph, these boundary points are included as feature points. Because style features are always mapped to the surface of a base part, and boundary points are not always obtained from a style model, the computing process differs depending on the source of the connection. If the connection originates from the style topology, the boundary points will be regarded as a feature and processed with other types of features. If the connection comes from the base topology, the boundary points will not have a corresponding destination on the surface of the style part. Therefore, virtual boundary points must be identified on the surface by mapping the original boundary points to the surface of the style part. The virtual boundary points were determined by using the method described in Sections 5.2 and 5.3, and these points were processed with other types of features. The virtual boundary points were only mapped to their boundary points when correspondence was identified.

6 Results and Discussion

Fig. 10 shows two sets of blended models that are automatically generated by the proposed system with diversity varying from 0 to 1. All output models preserve their important segments and symmetry because of rules 2 and 3. The generated outputs increase when diversity is high. However, high diversity may also cause unconventional blended models and many similar blended results. Examples include the self-penetration of the model in Fig. 10① and the unusual design product in Fig. 10②. They can be filtered out by rules 4 and 6, respectively, if the diversity is sufficiently small. Figs. 10③ and 10④ illustrate two other failure cases.

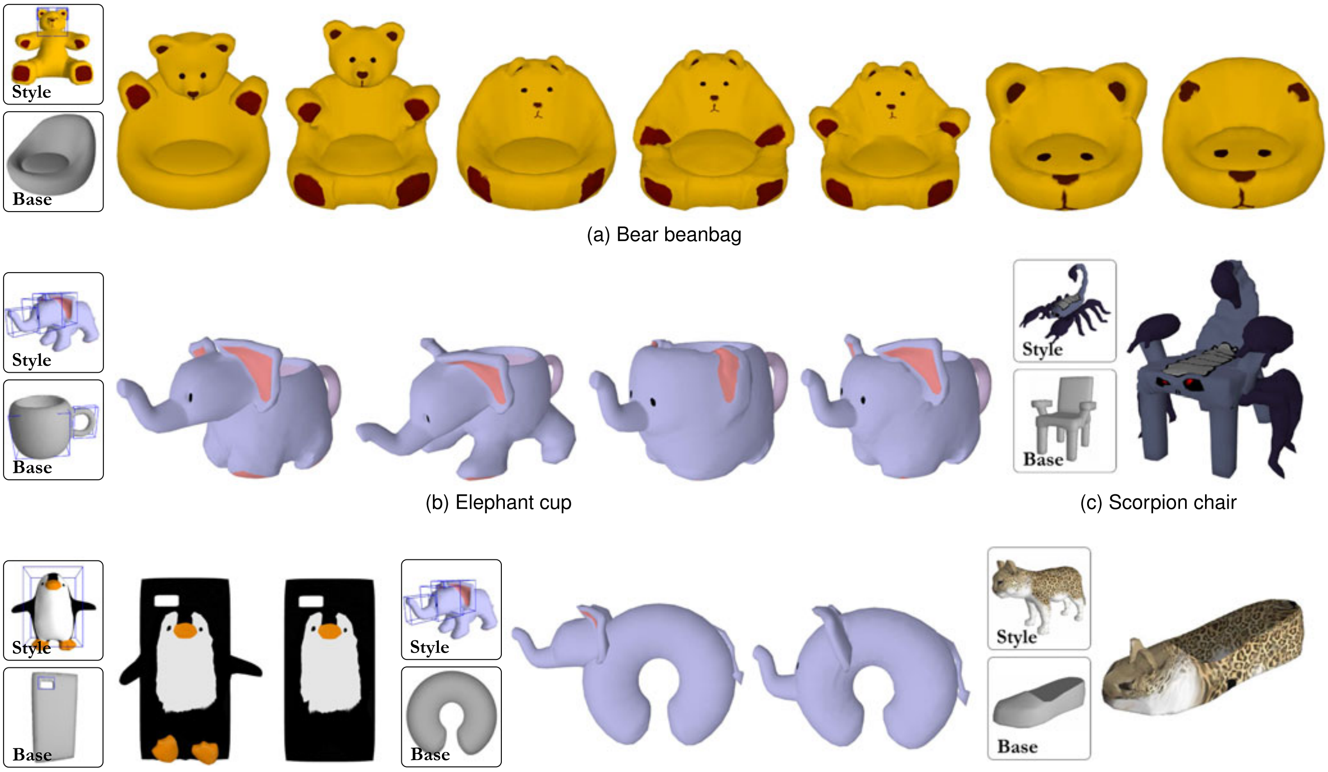
All the blended models preserve the functional attributes and style features in their base and style models. These blended models demonstrate different blending operations. For example, in Fig. 11a, base and style parts are blended by morphing the former to fit the characteristics of the latter. In the rightmost model in Fig. 11c, the texture of the style model is mapped to the base model (i.e., blended texturally) because texture (skin appearance) is a salient characteristic of the style model (penguin). In Fig. 11e, style parts (ears and legs of a rabbit) are attached to the base model (beanbag). An advantage of the method is that the segments of a style model can be separately mixed or attached to the base model. In Fig. 11c, the chelas of the scorpion are separated from its body. Non-cartoony texture could also be used, as shown in Fig. 11f. Although the texture of the blended model exhibits distortion in the front and back of the shoe, the model still can present the image of a leopard.

The number of output models generated by the system is influenced by the diversity parameter set by users and the topologies of input models. The topology determines the numbers of segments, parts, and groups of a model. Fig. 12 plots the number of output models generated with different diversity values and shows the numbers of segments and parts of each input model. The generated output models increase when diversity is high or when input models contain many segments. Browsing each model may be tedious when many output models exist. Therefore, we suggest users set the diversity value between 0.4 and 0.8 and adjust σ_1-σ_4 according to the input models and their needs when necessary.

For each blended model, the system supports a smooth deformation transition, which allows users to freely decide the degree of morphing between a base model and a style model. Fig. 13 shows an example of a shampoo bottle that gradually morphs into a bear model. Moreover, because the system extracts ridge and curvature points from input models, users can also locally adjust the geometry near the ridge and curvature points, as shown in Fig. 14. The geometry near the ears and horns can be locally controlled. Fig. 15 shows a comparison of surface blending with and without rigid and soft matching. With rigid and soft matching, the texture pattern of the style model is well preserved in the morphing process, whereas the texture demonstrates large distortions when the model is blended without feature alignment. Fig. 16 presents another comparison of the results with and without matching ridge and curvature points. The bottom parts of the bear and beanbag are aligned when the proposed approach is applied.

Although most procedures are automatically completed in the proposed system, several procedures require user assistance. During preprocessing, a user must define the top and frontal directions of an input model and set thresholds for curvature points, ridge points, segments and features. In the segmentation procedure of the preprocessing, a user is allowed to merge two segments. This step can avoid an exponential increase in part number caused by meaningless segments. User assistance is critical in topology blending. Users manually define important parts because the system cannot automatically assess the semantics of a part. Users are also allowed to adjust σ_1-σ_4 in topology blending. They can adjust σ_1-σ_4 individually or control σ_1-σ_4 together by using the parameter "diversity." The experiment described in Section 6.2 shows that "diversity" can satisfy the needs of users.

The proposed system was tested on Windows OS with Intel Xeon e3 CPU and 16 GB RAM. Most of the code was implemented in Visual C++, whereas the implementation of spherical parameterization [16] and the calculation of the



(a) Bear beanbag

(b) Elephant cup

(c) Scorpion chair

(d) Penguin case

(e) Elephant pillow

(f) Leopard shoe

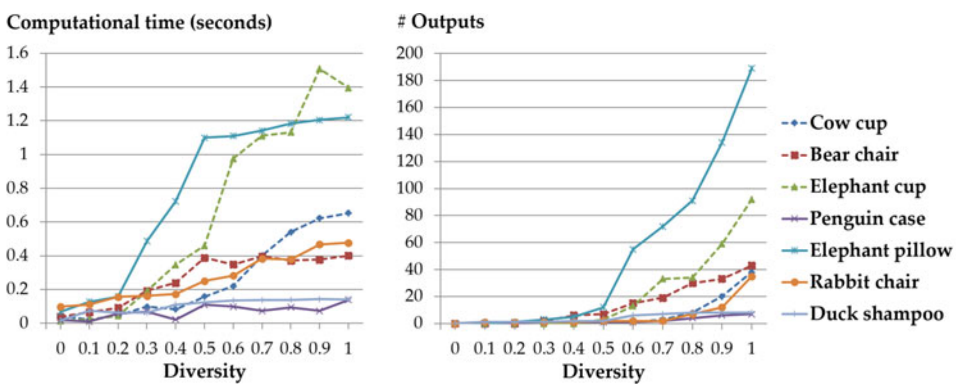
Fig. 11. Selected results of blended models and their inputs. The important segments defined by users are enclosed in blue boxes. Heads and ears are usually assigned as important segments in the experiment.

target position for warping [36] were run in MATLAB. With the non-optimized code, the computational performance of the proposed system was as follows: the topology blending process time, which ranges between 0.015 and 1.507 s, roughly increases with diversity values (Fig. 12). The running time of the topology blending also increases with the increase in the number of parts in the input models. However, the filtering process of the six rules effectively prevents an exponential blowup. Fig. 17 lists the computational time of surface blending in the five tested examples. The computational time ranges from 1 to 4 min. The computational time of rigid matching depends on the number of features. The computational time of soft matching increases as the number of points of each feature increases. The warping time increases with the number of features and mesh polygons. The process time in building a supermesh is influenced by the number of polygons.

6.1 Comparison with Zoomorphic Results

A user study was conducted to compare the results generated by the proposed approach and zoomorphic approach [6]. A total of 15 males and 15 females aged between 19 and 30 years old were invited to join the user study. Eight of these individuals have a background in design. The experiment was first introduced to each participant for two minutes. A participant watched eight pairs of blended models played in a random order on a computer screen. For each pair, a participant was asked to judge the models from four aspects. The first is personal preference in which the participants were asked to subjectively choose their preferred models. The second aspect is the reasonableness of the way style and function attributes are combined. The third and fourth aspects are style and functionality preservation, respectively.

The zoomorphic results were provided by the authors [6]. For fairness, only the results of geometric models were



	# Segments	# Parts
Chair	1	1
Pillow	1	1
Cup	3	6
Penguin	5	8
Duck	5	11
Bear	8	20
Rabbit	8	23
Cow	10	32
Elephant	10	52

Fig. 12. Computational time and output amount increase with the diversity value and the number of input parts.

Fig. 13. Proposed system allowing users to freely decide the degree of morphing between a base model and a style model. The body of a shampoo bottle morphs into a bear continuously.

Fig. 14. Proposed system allowing users to locally control the geometry near the ridge or curvature points (ears and horns).

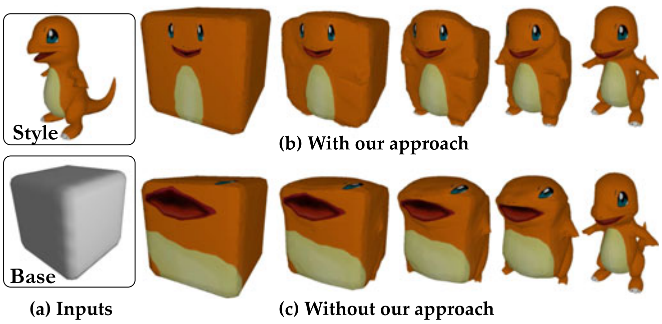


Fig. 15. Comparison of surface blending with and without rigid and soft matching. With the proposed approach, the texture pattern of the eyes, mouse, and belly are preserved during the morphing process.

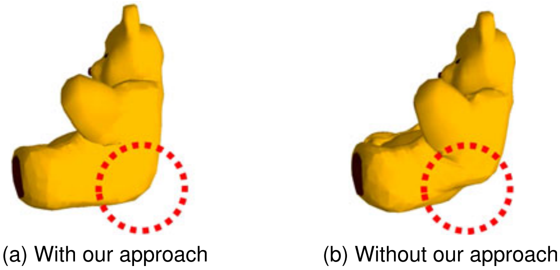


Fig. 16. Comparison of results with and without matching the ridge and curvature points.

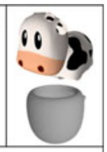
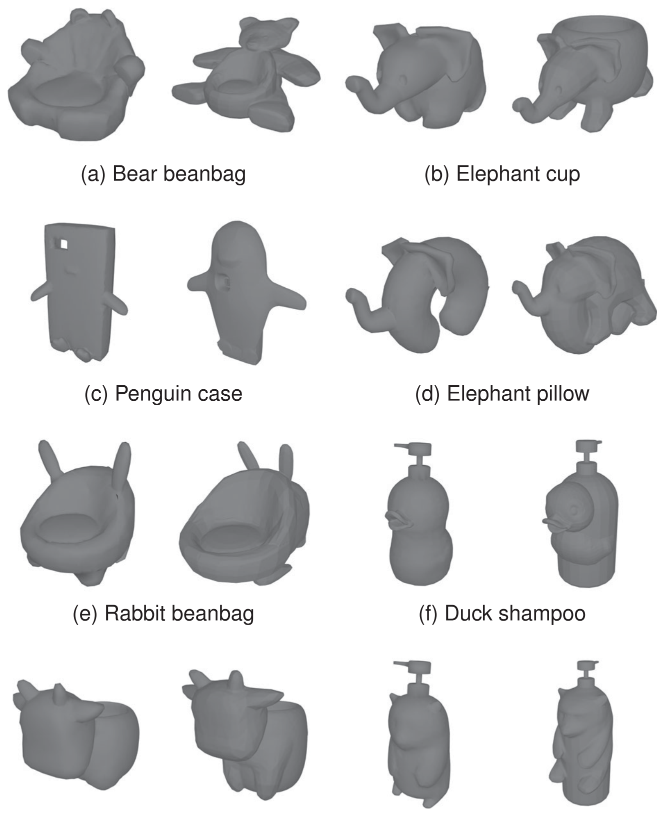
Style parts and base parts					
Data					
# Style faces	738	424	962	798	1782
# Base faces	760	760	760	2634	2634
# Features	8	1	6	14	21
# Feature points	357	228	182	491	702
Computing time (seconds)					
Rigid matching	0.705	0.036	0.288	0.812	1.72
Soft matching	0.035	0.146	0.026	0.109	0.132
Warping	38.624	26.639	50.257	111.667	219.018
Supermeshing	1.948	0.66	8.446	7.927	21.077
Total	41.742	27.661	59.017	120.525	241.947

Fig. 17. Computational time of the five tested examples in the surface blending process.

compared because the zoomorphic approach does not handle textures. A model was presented by showing participants a nine-second video in which a model rotates on its vertical axis. After a participant completed the pairwise-comparison questionnaire, the textured results were shown and a short interview was conducted to understand the influence of texture.

Fig. 18 shows the blended models for comparison. The corresponding textured models are shown in Fig. 11. Table 2 lists the parameters used to generate the results in Fig. 18. Diversity was controlled, and σ_1-σ_4 were adjusted with diversity. Table 3 shows the results of the user study, which compared the results generated by our approach and the zoomorphic design. For each pair in each aspect, the percentage of participants who prefer our blended results over the zoomorphic results is listed. The participants generally prefer our results and agree that these results are more reasonable than those produced by the zoomorphic design [6]. In the interview, twelve participants mentioned that the proposed approach retains a more appropriate aspect ratio in blending the parts from the input models. One participant said, "The head (in the zoomorphic design's penguin case) is too large. It's very inconvenient. I don't like it." Ten of the participants thought our blended models are smoother than the zoomorphic results. "It (zoomorphic's duck shampoo) looks like two models are forced to stick together, so it looks discorded." Meanwhile, as presented in Table 3, averagely only 44.2 percent of the participants thought our approach can better preserve the style attribute of the style model. In particular, in the bear beanbag and penguin case, our approach performed poorly in style preservation. For the bear beanbag, most participants described the shape of the zoomorphic models as "looks more like the style model," whereas "the head (in our bear beanbag) disappears." This may be because only geometry was provided in the comparison, but the head of our bear beanbag was mainly represented by texture in order to preserve the user-specified shape features of the beanbag. For the penguin case, the blended model does not maintain the rounded form of the penguin; instead, the form of the phone case was maintained. This is a trade off because the rounded form may lose the functionality of the phone case. Although our penguin case has a low score in style preservation, 90 percent of the participants agreed that it better preserves functionality.

In the interview, all the participants agreed that the textured models generated by the proposed system can better present the attributes of style models than the non-textured models. The participants preferred textured models and said, "The texture compensates for the lack of inconspicuous contours", "Textured models are better because of their presenting capability. Take the rabbit beanbag for example; before texturing, I thought it was the rabbit's butt (the butt faces front). After texturing, I realized it was the rabbit's face."



(a) Bear beanbag (b) Elephant cup

(c) Penguin case (d) Elephant pillow

(e) Rabbit beanbag (f) Duck shampoo

(g) Cow cup (h) Bear shampoo

Fig. 18. Blended models generated by the proposed method and zoomorphic [6]. For each pair, the left model is generated by our method and the right model is generated by zoomorphic design.

TABLE 3
Comparison Result: The Percentages That Participants Favored Our Blended Models in Each Aspect

	Personal preference	Reasonableness	Style preservation	Functionality preservation
(a) Bear beanbag	63.3%	60.0%	6.7%	83.3%
(b) Elephant cup	80.0%	73.3%	83.3%	53.3%
(c) Penguin case	70.0%	66.7%	3.3%	90.0%
(d) Elephant pillow	93.3%	90.0%	33.3%	90.0%
(e) Rabbit beanbag	76.7%	73.3%	36.7%	76.7%
(f) Duck shampoo	96.7%	96.7%	70.0%	76.7%
(g) Cow cup	70.0%	63.3%	50.0%	56.7%
(h) Bear shampoo	93.3%	90.0%	70.0%	76.7%
Average	80.4%	76.7%	44.2%	75.4%

For instance, 70 percent of personal preference means that 7 out of 10 participants subjectively prefer the results of the proposed approach over those of the zoomorphic design.

6.2 System Evaluation

Another user study was conducted to test the usability of the proposed system for users with different backgrounds (e.g., novices, 3D modelers, and designers). Eighteen users were invited to test the system and provide feedback. Ten of these users have 3D modeling software using experiences (modelers) and the other eight users have neither modeling or design experiences (novices). Five of the ten modelers also work in the design industry or study in a design school (designers). The user interface of the proposed system was explained to each user, and the user freely used the system. After using the system, the users were asked to fill out a questionnaire on the usability of the system. A short interview was also conducted. The experiment lasted for about 30 min.

The content and results of the questionnaire are shown in Table 4. The novices agreed that the proposed system allows them to easily create 3D blending models. They also agreed that the proposed system can provide customized, novel, and reasonable blending models. A user said, "One advantage of your system is its ability to generate reasonable results." However, not all users agreed that the system allows them to modify blended models. A user explained, "I want to move the position of a part or adjust the angle of components. Your system doesn't provide these functions." This comment suggests that although users feel that modifying the blended model by adjusting the morphing degree between the style and base models is intuitive and helpful, several users may also require lower-level editing functions in conventional model editing tools. Users who have a design background (designers) agreed that the system can assist them in designing products, save time, provide drafts, and inspire them. The designers commented, "Sometimes the system generates unexpected results. They can inspire me. The system can save my time to generate the drafts. It is helpful in communicating with customers." Lastly, 3D modelers believed that the system can save time in generating 3D blended models.

TABLE 2
Parameters for Producing the Results in Fig. 18

Parameters	Diversity	σ_1	σ_2	σ_3	σ_4
(a) Bear beanbag	0.7	1,800	0.925	13,000	6.19
(b) Elephant cup	0.7	1,800	0.925	13,000	6.19
(c) Penguin case	0.8	1,300	0.95	14,000	6.8
(d) Elephant pillow	0.6	2,300	0.9	12,000	5.6
(e) Rabbit beanbag	0.4	3,300	0.85	10,000	4.4
(f) Duck shampoo	0.4	3,300	0.85	10,000	4.4
(g) Cow cup	0.8	1,300	0.95	14,000	6.8
(h) Bear shampoo	0.3	3,800	0.82	9,000	3.8

TABLE 4
Questionnaire Evaluating the Proposed System from Three Types of Users

Questions	Avg. score
1. Do you agree that our system has the following functions? (novices only)	
A. Easily create blended models	4.4
B. Provide customized blended models	4.4
C. Provide unexpected results	4.3
D. Provide reasonable results	4.0
E. Modify blending models	3.7
2. Do you agree that our system can provide the following assistance? (designers only)	
A. Assist in designing products	4.0
B. Save creation time	4.0
C. Inspire designers	4.0
D. Provide various draft	4.2
3. Do you agree that our system can save time when generating blended models? (3D modelers only)	4.4

The right column is the statistics of user feedback to the questionnaire in a five-point Likert scale. High scores represent a strong agreement to the questions.

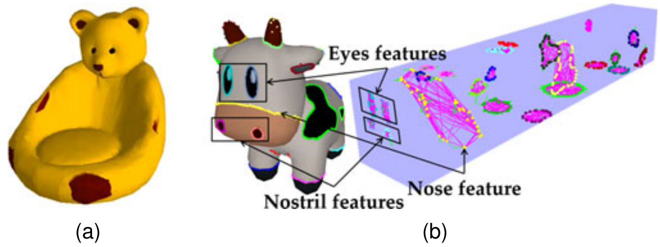


Fig. 19. Failure cases. (a) The texture pattern of bear hands is not preserved well because the inappropriate feature matching caused the mesh to collapse. (b) The relative positions among features are not maintained. In this example, cow features are mapped to a long box. Nostril features are originally inside the nose feature, and the eye features are outside the nose feature. However, after matching, both the nostril and eye features are inside the nose feature.

The users were also interviewed about their interactive experience with the proposed system. The users generally agreed that the interface is easy to use. One user said, "The interface is friendly. I only need to select important segments, drag the parameter bar, and then click 'Blend' to generate results." Although several users mentioned that they experienced difficulty controlling parameters σ_1-σ_4 individually because the parameters are difficult to understand, 13 of 18 users agreed that parameter diversity is helpful. They could see the change in an output set with the change in diversity, which implies that the diversity parameter is intuitive for users.

6.3 Failure Cases and Discussion

Aside from the failure cases shown in Fig. 10, Fig. 19 also provides several failure cases in the surface blending procedure. Three possible reasons caused these failure cases. First, the shape of input models influences blending quality. If two input meshes differ substantially in shape, such as the inputs in Fig. 11f, a distortion in texture will occur. Second, the method of identifying target point **d** can be improved. In the current method, the target point is identified by considering only the point's type and distance. This condition caused the failure cases shown in Figs. 10③, 19a, and 19b. The failure cases in Fig. 10③ and Fig. 19a are caused by the inappropriate feature matching, and that in Fig. 19b is caused by the incorrect relative position among features. These factors should be considered when identifying target points. Third, the method of mesh warping (Section 5.4) can also be improved. To avoid foldover faces, not all vertices are warped to their desired positions. This condition caused the failure case shown in Fig. 10④, where the rabbit's ears cannot maintain their texture pattern.

7 Conclusion, Limitations and Future Work

An approach that can automatically generate stylized models by considering the topology, texture, and geometry of input models was developed. At the topological level, operational vectors were designed and graphs were merged to encode all possible combinations of parts in the two input models and develop plausible tree growing to efficiently search for plausible combinations. At the geometric and textural levels, a global-to-local matching method was proposed to preserve the characteristics of the style models and the user-specified shape features of the base models. The results showed that the proposed system can blend two models in various topologies and geometries while preserving the texture patterns and geometric features of the style model.

The proposed system possesses several limitations. First, although the six rules defined in Section 4.2 can filter out many unreasonable combinations, the rules do not address the core of functionality and may be heuristic. Several of these rules are designed according to the daily experiences of individuals. In the future, we plan to study other methods to define the rules, e.g., learn the rules from examples. Second, the current approach does not focus on stylization on an abstract level. Some product designers may only bring an abstract-style image to the style products. In the future, we plan to explore the idea suggested by Goes et al. [37] with regard to abstract-style features. Third, the morphing technique was applied to deform geometries in the current study. In the future, we would like to apply other deforming methods, such as Laplacian surface editing [38], to add more editing tools or functionalities to the proposed system. Another possible future work direction is to extract the skeleton of input models and deform them by transforming the skeleton. Many of the style models in this study had an animal or human shape. Thus, rotating their limbs according to their skeletons would create a natural look. We believe the blended models could preserve the geometric and topological properties of the base model if deformation by the skeleton technique is applied.

Despite these limitations, the proposed system can adequately and effectively provide designers previews of various possible designs. This system can also be used as a tool for novices to design stylized models or products. With the rapid development of 3D printing, the proposed system can readily assist novice users to develop their stylized models, thereby possibly motivating other novice users to create stylized models as well. We hope that this work would encourage further research on model stylization and potentially affect the development of support techniques for stylized product designs in the future.

Acknowledgments

We thank the reviewers for their insightful comments, Noah Duncan for providing the zoomorphic design results, and Yi-Hsuan Huang and Pin-Xuan Huang for assisting the user studies. This work was supported in part by the Ministry of Science and Technology of Taiwan under grant no. MOST-102-2221-E-009-082-MY3, MOST-104-2221-E-006-044-MY3, MOST-104-2628-E-009-001-MY3, MOST-105-2221-E-009-095-MY3, MOST-105-2218-E-009-013-, and MOST-105-2218-E-155-005.

References

[1] T. Wu and H. Chen, "Products with biomimetic shapes convey emotions more effectively," in *Proc. 4th Int. Conf. Des. User Experience Usability: Des. Discourse Held Part HCI Int.*, 2015, pp. 559–566.

[2] F. V. Horen and R. Pieters, "Consumer evaluation of copycat brands: The effect of imitation type," *Int. J. Res. Marketing*, vol. 29, no. 3, pp. 246–255, 2012.

[3] H. Li, H. Zhang, Y. Wang, J. Cao, A. Shamir, and D. Cohen-Or , "Curve style analysis in a set of shapes," *Comput. Graph. Forum*, vol. 32, no. 6, pp. 77–88, 2013.

[4] Z. Han, Z. Liu, J. Han, and S. Bu, "3D shape creation by style transfer," *Visual Comput.*, vol. 31, no. 9, pp. 1147–1161, 2015.

[5] C. Ma, H. Huang, A. Sheffer, E. Kalogerakis, and R. Wang, "Analogy-driven 3D style transfer," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 175–184, 2014.

[6] N. Duncan, L. Yu, S. K. Yeung, and D. Terzopoulos, "Zoomorphic design," *ACM Trans. Graph.*, vol. 34, no. 4, 2015, Art. no. 95.

[7] I. Alhashim, "Modeling and correspondence of topologically complex 3D shapes," *arXiv preprint arXiv:1506.06855*, 2015.

[8] M. Alexa, "Recent advances in mesh morphing," *Comput. Graph. Forum*, vol. 21, no. 2, pp. 173–196, 2002.

[9] T. A. Funkhouser, et al., "Modeling by example," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 652–663, 2004.

[10] V. Jain and H. Zhang, "Robust 3D shape correspondence in the spectral domain," in *Proc. Int. Conf. Shape Model. Appl.*, 2006, Art. no. 19.

[11] H. Zhang, A. Sheffer, D. Cohen-Or , Q. Zhou, O. van Kaick, and A. Tagliasacchi, "Deformation-driven shape correspondence," *Comput. Graph. Forum*, vol. 27, no. 5, pp. 1431–1439, 2008.

[12] O. K. Au, C. Tai, D. Cohen-Or , Y. Zheng, and H. Fu, "Electors voting for fast automatic shape correspondence," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 645–654, 2010.

[13] T. Athanasiadis, I. Fudos, C. Nikou, and V. Stamati, "Feature-based 3D morphing based on geometrically constrained spherical parameterization," *Comput. Aided Geometric Des.*, vol. 29, no. 1, pp. 2–17, 2012.

[14] M. S. Floater, "Parametrization and smooth approximation of surface triangulations," *Comput. Aided Geometric Des.*, vol. 14, no. 3, pp. 231–250, 1997.

[15] M. Alexa, "Merging polyhedral shapes with scattered features," *Visual Comput.*, vol. 16, no. 1, pp. 26–37, 2000.

[16] L. Shen and F. Makedon, "Spherical mapping for processing of 3D closed surfaces," *Image Vis. Comput.*, vol. 24, no. 7, pp. 743–761, 2006.

[17] A. D. Gregory, A. State, M. C. Lin, D. Manocha, and M. A. Livingston, "Interactive surface decomposition for polyhedral morphing," *Visual Comput.*, vol. 15, no. 9, pp. 453–470, 1999.

[18] V. Kraevoy, D. Julius, and A. Sheffer, "Model composition from interchangeable components," in *Proc. Pacific Conf. Comput. Graph. Appl. Pacific Graph.*, 2007, pp. 129–138.

[19] J. Lee and T. A. Funkhouser, "Sketch-based search and composition of 3D models," in *Proc. 5th Eurographics Conf. Sketch Based Interfaces Model. Annecy*, 2008, pp. 97–104.

[20] S. Chaudhuri and V. Koltun, "Data-driven suggestions for creativity support in 3D modeling," *ACM Trans. Graph.*, vol. 29, no. 6, 2010, Art. no. 183.

[21] S. Chaudhuri, E. Kalogerakis, L. J. Guibas, and V. Koltun, "Probabilistic reasoning for assembly-based 3D modeling," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 35.

[22] E. Kalogerakis, S. Chaudhuri, D. Koller, and V. Koltun, "A probabilistic model for component-based shape synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 55:1–55:11, 2012.

[23] K. Xu, H. Zhang, D. Cohen-Or, and B. Chen, "Fit and diverse: Set evolution for inspiring 3D shape galleries," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 57:1–57:10, 2012.

[24] A. Jain, T. Thormählen, T. Ritschel, and H. Seidel, "Exploring shape variations by 3D-model decomposition and part-based recombination," *Comput. Graph. Forum*, vol. 31, no. 2, pp. 631–640, 2012.

[25] Y. Zheng, D. Cohen-Or , and N. J. Mitra, "Smart variations: Functional substructures for part compatibility," *Comput. Graph. Forum*, vol. 32, no. 2, pp. 195–204, 2013.

[26] I. Alhashim, H. Li, K. Xu, J. Cao, R. Ma, and H. Zhang, "Topology-varying 3D shape creation via structural blending," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 158:1–158:10, 2014.

[27] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[28] G. Taubin, "Estimating the tensor of curvature of a surface from a polyhedral approximation," in *Proc. 5th Int. Conf. Comput.*, 1995, pp. 902–907.

[29] F. Cazals and M. Pouget, "Smooth surfaces, umbilics, lines of curvatures, foliations, ridges and the medial axis: A concise overview," *Int. J. Comput. Geometry Appl.*, vol. 15, no. 5, pp. 511–536, 2005.

[30] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[31] L. Shapira, A. Shamir, and D. Cohen-Or , "Consistent mesh partitioning and skeletonisation using the shape diameter function," *Visual Comput.*, vol. 24, no. 4, pp. 249–259, 2008.

[32] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, "On visual similarity based 3D model retrieval," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 223–232, 2003.

[33] W. Lidwell, K. Holden, and J. Butler, *Universal Principles of Design, Revised and Updated: 125 Ways to Enhance Usability, Influence Perception, Increase Appeal, Make Better Design Decisions, and Teach Through Design*. Rockport, WA, USA: Rockport Pub, 2010.

[34] J. Parus, "Morphing of meshes," Univ. West Bohemia, Pilsen, Czech Republic, Tech. Rep. DCSE/TR-2005-02, 2005.

[35] Y. Ma, J. Zheng, and J. Xie, "Foldover-free mesh warping for constrained texture mapping," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 3, pp. 375–388, Mar. 2015.

[36] D. Panozzo, I. Baran, O. Diamanti, and O. Sorkine-Hornung , "Weighted averages on surfaces," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 60:1–60:12, 2013.

[37] F. de Goes, S. Goldenstein, M. Desbrun, and L. Velho, "Exoskeleton: Curve network abstraction for 3D shapes," *Comput. Graph.*, vol. 35, no. 1, pp. 112–121, 2011.

[38] O. Sorkine, D. Cohen-Or , Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geometry Process.*, 2004, pp. 175–184.

Yi-Jheng Huang received the BS degree from the Department of Computer Science and Information Engineering, National Dong Hwa University, Taiwan, in 2007 and the MS degree in computer science from National Chiao Tung University, Taiwan, in 2009. Now, she is working toward the PhD degree in the Department of Computer Science, National Chiao Tung University. Her research interests include computer graphics and data visualization.

Wen-Chieh Lin received the BS and MS degrees in control engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1994 and 1996, respectively, and the PhD degree in robotics from the School of Computer Science, Carnegie Mellon University, Pittsburgh, in 2005. He joined the Department of Computer Science, National Chiao Tung University, in 2006 and is a professor now. His research interests span several areas of computer graphics, computer animation, human-computer interaction, and visualization. In particular, he is interested in physics-based animation and applying perceptual studies to improve computer graphics and HCI techniques. He is a member of the IEEE.

I-Cheng Yeh received the BS degree from the Department of Computer and Information Science, National Chiao Tung University, Taiwan, in 2004 and the PhD degree in computer science and information engineering from National Cheng Kung University, Taiwan, in 2011. Now, he is an assistant professor in the Department of Computer Science and Information Engineering, Yuan Ze University. His research interests include computer graphics, surfaces parameterization and manipulation, media retargeting, and data visualization.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a chair professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He is a senior member of the IEEE Computer Society and a member of the ACM.