Yang WW, Hua J, Yao KY. CR-Morph: Controllable rigid morphing for 2D animation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(5): 1109–1122 Sept. 2019. DOI 10.1007/s11390-019-1963-3

CR-Morph: Controllable Rigid Morphing for 2D Animation

Wen-Wu Yang[1], *Member*, *CCF*, Jing Hua[1,2], and Kun-Yang Yao[1]

[1] *School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310012, China*
[2] *Department of Computer Science, Wayne State University, Michigan 48202, U.S.A.*

E-mail: wwyang@zjgsu.edu.cn; jinghua@wayne.edu; 792282189@qq.com

Received December 3, 2018; revised July 24, 2019.

Abstract This paper presents a rigidity-preserving morphing technique that blends a pair of 2D shapes in a controllable manner. The morphing is controllable in two aspects: 1) motion dynamics in the interpolation sequences can be effectively enhanced through an intuitive skeleton control and 2) not only the boundaries but also the interior features of the source and target shapes are precisely aligned during the morphing. We introduce a new compatible triangulation algorithm to decompose the source and target shapes into isomorphic triangles. Moreover, a robust and motion-controllable rigidity-preserving transformation scheme is proposed to blend the compatible triangulations, ultimately leading to a morphing sequence which is appearance-preserving and with the desired motion dynamics. Our approach constitutes a powerful and easy-to-use morphing tool for two-dimensional animation. We demonstrate its versatility, effectiveness and visual accuracy through a variety of examples and comparisons to prior work.

Keywords compatible triangulation, disc-based interpolation, rigid morphing, skeleton-based dynamics

1 Introduction

Two-dimensional morphing (or blending), which transforms one 2D (two-dimensional) object (source) into another (target), has a wide range of applications such as producing visual effects and cell animation[1−2]. In the two-dimension case, planar objects can be described by vector graphics, i.e., curves or polygons, or can be represented as images. Accordingly, 2D morphing techniques can be generally classified into two categories: image-based and shape-based. Image morphing works with discretizations of planar space, while shape morphing deals with explicit representations of objects, i.e., curves or polygons. Image morphing techniques[3−8] have been widely deployed in commercial tools and through them it is convenient to handle abundant details of the objects in image space. However, in some scenarios such as 2D character animation where drastic local or global geometric changes tend to occur, it is challenging for pure image-space methods to generate in-between objects that logically retain their original appearance and properties. For example, it is difficult for the image morphing techniques to generate the drastic morphing of gourd vine shown in Fig.1. Beyond this, the entire morphing example in this figure is also difficult to achieve with image-based techniques since foreground and background behave differently.

Shape morphing techniques have the potential to deal with large geometric changes. As demonstrated in the shape morphing techniques[9−12] aesthetic and intuitive morphing should treat the meta-physical objects which are implicitly represented by the blended shapes as rigidly as possible, i.e., in a rigidity-preserving fashion, in order to avoid superfluous global or local deformations. While some methods can minimize the deformation of the boundaries[9,11−14], it is suggested in [10, 15-19] that a proper morph should take into account not only the boundary but also the interior of the objects. In these methods, compatible triangulations are used to represent the objects' interior and to easily handle the abundant details of the objects, i.e., as textures of the triangulations. However, existing compatible triangulations algorithms[19−24] mainly focus on generating high-quality triangulations with a

Regular Paper

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61003189, U1609215 and 61472363, and the US National Science Foundation under Grant Nos. 0915933, 0937586, and 1647200.

small number of Steiner vertices, but cannot guarantee the corresponding interior features of the shapes to be aligned in the resulting compatible triangulations. Consequently, undesirable results such as "ghosting" artifacts are easy to occur especially for the objects whose internal features are rather different in shape or position.

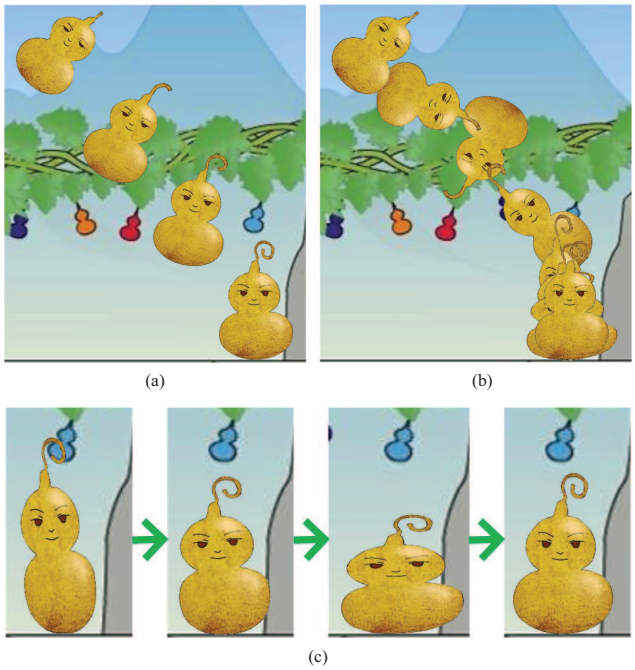


Fig.1. Given a gourd in the air (source shape) and the other on the ground (target shape), morphing animation is generated using our approach. (a) Original interpolation sequence. (b) Improved visual impact by enhancing motion dynamics over the interpolation sequence. (c) Last four frames which are overlapped to each other in the dynamics-enhanced morphing sequence.

Considering the morphing generation, a sequence of in-between shapes, which is locally least-distorting and visually pleasing, can be obtained by applying an as-rigid-as-possible (ARAP) interpolation scheme to blend the compatible triangulations[10,15,17–19]. In practice, mesh quality is essential for the stability of ARAP interpolation, and thus a time-consuming post-optimization is necessary for the compatible triangulations. Furthermore, real applications often require to enhance motion dynamics through the interpolation sequence in order to improve visual impact. Unfortunately, offering an intuitive and ease-to-use tool for controlling motion dynamics is always considered difficult in keyframe interpolation[4,25]. To our knowledge, there is no work yet for effectively incorporating such control into a rigidity-preserving shape interpolation process.

In this paper, we introduce CR-Morph a rigidity-preserving morphing technique that blends a pair of 2D shapes in a controllable manner. As shown in Fig.1, our method accommodates drastic local or global geometric changes between the source and the target shapes and allows the user to easily improve the visual impact by enhancing motion dynamics over the interpolation sequences. Finally, our approach constitutes a powerful and easy-to-use morphing tool for 2D animation. The main technical contributions of the paper are as follows:

- a rigidity-preserving interpolation algorithm which is insensitive to the mesh quality and is not only suitable for compatible triangulations but also suitable for arbitrary compatible point sets,
- an effective skeleton control scheme which allows users to enhance motion dynamics over the interpolation sequences in an easy and intuitive way, and
- a new compatible triangulation algorithm which can align both the boundary and interior features of the shapes in the compatible triangulations.

2 Related Work

The work in 2D morphing falls into two categories: morphing of images[5,8] and morphing of planar shapes[9,26]. Our approach takes advantage of these two, handling abundant details in image space (i.e., as textures of the triangulations), like image morphing, while taking geometry into account, as in shape morphing.

Image Morphing. Morphing of images is a long-studied problem[3,4,8,27], and is generally realized by coupling image warping with color interpolation. Image warping retains the geometric alignment between the corresponding features of the images, while color interpolation blends their colors. In image morphing, the corresponding features between the images are usually manually tagged by users using points, lines, or curves. In order to reduce user interaction, some (semi-)automatic approaches[7,8] were proposed, where the user draws only a small number of point correspondences and the system automatically determines a full correspondence between the features, and these method works well for the objects of similar appearance.

In general, the warping functions are defined in a pure mathematical manner (e.g., under the criteria of smoothness and continuity), but not motivated by or related to geometric principles since they are geometry-unaware. Therefore, while the image morphing techniques generate visually pleasing results in cases where the shapes of the source and the target objects are somewhat similar, it is typically difficult for them to

handle drastic global or local geometric changes. Similarly, our approach lets the user draw several curves to specify the corresponding features of the objects; however, different from pure image-space methods, our approach takes into account the objects' geometry and can naturally blend even the objects of significantly different shapes.

2D Shape Morphing. Shape morphing techniques consider geometric characteristics of the objects. In [9], Sederberg *et al.* introduced an intrinsic interpolation approach that minimizes the distortion of the shape boundaries by considering their edge lengths and vertex angles.

In [10, 15, 17-19], the source and the target shapes are dissected into a pair of compatible triangulations, so as to take into account not only the boundary but also the interior of the objects. To achieve aesthetic and intuitive morphing, local non-distorting morphs are determined for each pair of corresponding triangles. In practical implementation, the high quality of the triangles in the compatible triangulations is important; otherwise, numerical problems may occur[13]. Different from these methods, we introduce a disc-based rigidity-preserving interpolation technique for blending the compatible triangulations. The algorithm is insensitive to the triangle quality and is not only suitable for compatible triangulations but also suitable for arbitrary compatible point sets.

Shape Triangulation. Assuming that a boundary vertex correspondence between the source and the target shapes is given, many algorithms can be used to generate compatible triangulations within the shapes[18−21]. These algorithms mainly focus on generating high-quality triangulations with a small number of Steiner vertices, but cannot guarantee the corresponding interior features of the shapes to be aligned in the resulting compatible triangulations. In contrast to them, our compatible triangulation algorithm accounts for the correspondences between not only the boundaries but also the interior features of the shapes.

In [17], Tal and Elber also considered the interior features by isolating them via the so-called bridges. A compatible triangulation is constructed for each feature and then all of them are combined together. Obviously, the features should be described by a set of simple polygons, i.e., each feature as a hole. However, while the bridge technique can be straightforwardly extended to the interior features that are tagged by a point[21], it is not a trial work to apply such technique to the interior open curves. Since many features correspond to open curves in practice, as shown in Fig.2(b), the bridge technique is thus difficult to be used in these common cases.

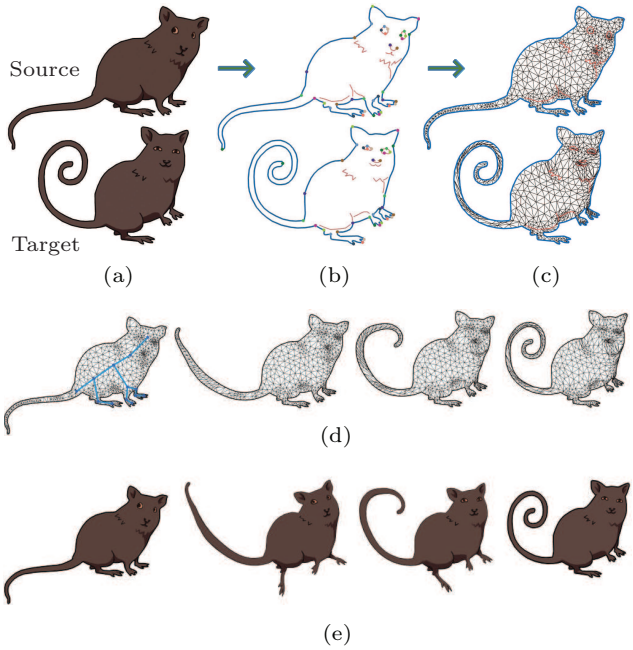


Fig.2. Algorithm overview. (a) Input shapes. (b) Boundaries, feature curves, and vertex correspondence, where for clarity only several corresponding salient points are shown. (c) Compatible triangulations. (d) Disc-based rigidity-preserving interpolation. (e) Interpolation sequence with enhanced motion dynamics.

Morphing Control. There is relatively little work on controlling the motion dynamics during the morph animation. A simple approach is to offer several interpolation paths for different portions[4,28]; however, it is difficult to visualize points of simultaneity on several paths such that the in-between shapes could be easily perceived at any instant. While secondary physical effects in shape interpolation are possible[25,29,30], it seems tedious and nonintuitive to tune abstract physical parameters such as vibration frequencies, amplitudes and damping, so as to achieve desired motion dynamics. Based on the observation that the animator often sketches stick figure representations to visualize a complex movements, Burtnyk and Wein[31] suggested the use of skeleton for enhancing motion dynamics in key frame animation. Inspired by this idea, we design a technique to effectively incorporate the use of skeletons into our disc-based rigidity-preserving interpolation, so that the user is able to enhance motion dynamics over the interpolation sequence through an easy and intuitive skeleton control. Such incorporation is not easy since the motion dynamics derived from the user-controllable skeleton is often inconsistent with the

original motion sequence determined by the disc-based rigidity-preserving interpolation.

3 Algorithm Overview

As shown in Fig.2, the source and the target shapes are imported by extracting the polygonal outline of the objects in the input images. The user specifies their corresponding features using open or closed curves (Fig.2(b)), which we call "feature curves". The system will automatically establish the vertex correspondence between boundaries and between each pair of corresponding feature curves using the feature-based vertex correspondence approaches[15], where several salient points are first detected and associated, and then the remaining vertices are aligned (Fig.2(b)). If needed, the user can alter the vertex correspondence by modifying the corresponding salient points.

A pair of compatible triangulations are generated within the source and the target shapes through our compatible triangulation algorithm (Section 5) (see Fig.2(c)). Then, our disc-based rigidity-preserving interpolation approach (Subsection 4.1) is used to transform the source triangulation into the target one (Fig.2(d)). Finally, a sequence of appearance-preserving in-between shapes is obtained by transferring the underlying source and target shapes to the intermediate triangulations using simple linear texture mapping, i.e., taking the input images as the textures of the source and the target triangulations and blending them for the intermediate triangulations. The user can further enhance motion dynamics via the skeleton control. To do so, the user first creates a skeleton on the source shape (the first column of Fig.2(d)). We call this skeleton as "reference skeleton" which can be defined regionally to apply the control selectively. Then, the user edits the skeleton at any intermediate frame and the system will automatically enhance motion dynamics over the interpolation sequence to meet the user's expectation (Fig.2(e)).

4 Motion-Controllable Morphing

Given a pair of compatible triangulations, we transform the source triangulation into the target one in a rigidity-preserving manner. Such an interpolation scheme includes two basic steps: 1) computing local non-distorting morphs and 2) assembling these local transformations into a global coherent least-distorting transformation. In the traditional rigidity-preserving approaches[13,20], the local transformations are determined based on triangle-to-triangle morph. Instead, we compute local non-distorting disc-to-disc transformations where a disc corresponds to a local point set around a point. This method has two main advantages: 1) the interpolation algorithm is point-based, or meshless, hence insensitive to the quality of mesh, and 2) the disc is general and can be easily constructed for a variety of representations such as triangulations and point clouds. In order to allow the user to enhance motion dynamics via a skeleton, we introduce a two-phase propagation scheme to combine the skeleton transformations with the local disc-to-disc transformations in a smooth and consistent fashion.

4.1 Disc-Based Rigidity-Preserving Interpolation

We consider a general case for the disc-based rigid interpolation. Let $\{\boldsymbol{p}_i\}$ and $\{\boldsymbol{q}_i\}$ be the sets of sampling points for the source and the target objects, respectively, where each source point $\boldsymbol{p}_i$ corresponds to a target point $\boldsymbol{q}_i$. For each point $\boldsymbol{p}_i$, a disc (denoted as $\mathcal{P}_i$) is constructed, which includes a local point set around $\boldsymbol{p}_i$. Similarly, we can construct for each target point $\boldsymbol{q}_i$ a disc ($\mathcal{Q}_i$) which contains the corresponding target points of those source points in $\mathcal{P}_i$. As shown in Fig.3, each disc describes the local geometric shape of the source or target object.

For each pair of corresponding discs $\mathcal{P}_i$ and $\mathcal{Q}_i$, a local transformation is determined to transform $\mathcal{P}_i$ into $\mathcal{Q}_i$. A straightforward solution is to use the best-fit affine mapping between the two point sets of $\mathcal{P}_i$ and $\mathcal{Q}_i$[32,33]. However, an affine mapping is typically insufficient for representing such local transformation when the number of points is greater than 3 (see Fig.3(b)). Our solution is to represent the local transformation using a collection of linear mappings. Taking $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ as rotation centers, each point $\boldsymbol{p}_j$ in $\mathcal{P}_i$ can be transformed to its corresponding point $\boldsymbol{q}_j$ in $\mathcal{Q}_i$ by a linear mapping $\boldsymbol{L}_{(i,j)}$ represented by a rotation angle α and a scale factor s:

$$\boldsymbol{L}_{(i,j)}(\boldsymbol{p}_j - \boldsymbol{p}_i) = \boldsymbol{R}_\alpha(\boldsymbol{p}_j - \boldsymbol{p}_i)s = \boldsymbol{q}_j - \boldsymbol{q}_i,$$

where $\boldsymbol{R}$ corresponds to a rotation matrix with respect to a specified angle, α is the angle from vector $\boldsymbol{p}_i\boldsymbol{p}_j$ to $\boldsymbol{q}_i\boldsymbol{q}_j$, and s is the ratio between the length of $\boldsymbol{q}_i\boldsymbol{q}_j$ and the length of $\boldsymbol{p}_i\boldsymbol{p}_j$. Note that the translation is not taken into account since it does not change any shape property of the disc.

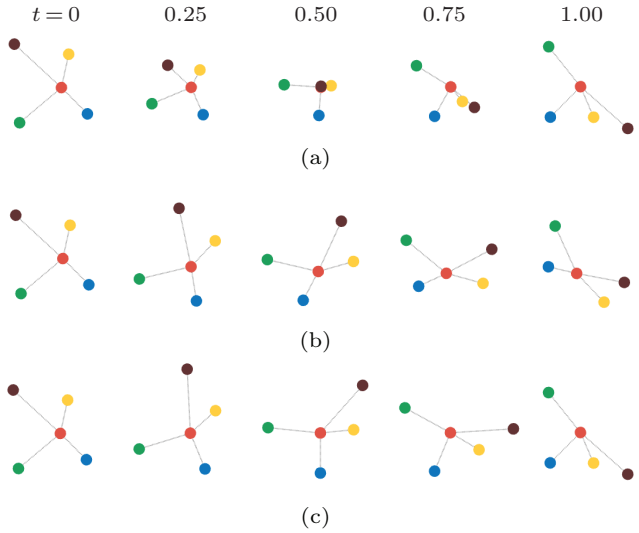


Fig.3. Transformations of a single disc. Given a pair of discs that are around a pair of corresponding points, i.e., the red dots, the transformations between them are determined by (a) linear vertex interpolation, (b) an affine mapping, and (c) a collection of linear mappings, respectively. The first and the last columns of (a) and (c) correspond to the source and the target discs, respectively. In (b) and (c), the transformations are conducted in a rigidity-preserving way, where it is shown in (b) that an affine mapping cannot fully represent the transformation since the source disc has not been exactly transformed into the target one at $t = 1.0$.

Therefore, the shape of intermediate disc for each pair of corresponding discs $\mathcal{P}_i$ and $\mathcal{Q}_i$ can be described by the collection of vectors $\{\boldsymbol{L}^t_{(i,j)}(\boldsymbol{p}_j - \boldsymbol{p}_i)\}_j$. To achieve local non-distorting disc-to-disc morphing, we determine each $\boldsymbol{L}^t_{(i,j)}$ by interpolating the rotation angle of $\boldsymbol{L}_{(i,j)}$ separately from its scale factor, i.e., in a rigidity-preserving way (Fig.3(c)):

$$\boldsymbol{L}^t_{(i,j)}(\boldsymbol{p}_j - \boldsymbol{p}_i) = \boldsymbol{R}_{t\alpha}(\boldsymbol{p}_j - \boldsymbol{p}_i)(1 - t + ts), t \in [0, 1].$$

To determine the vertices of intermediate objects $V(t) = \{\boldsymbol{v}^t_i\}$, we assemble all of the intermediate discs into a whole by minimizing the following quadratic error function, which is similar to the assembly process in [33, 34]:

$$E_{V(t)} = \sum_{\mathcal{P}_i} \sum_{\boldsymbol{p}_j \in \mathcal{P}_i} \|\boldsymbol{L}^t_{(i,j)}(\boldsymbol{p}_j - \boldsymbol{p}_i) - (\boldsymbol{v}^t_j - \boldsymbol{v}^t_i)\|^2. \quad (1)$$

The general disc-based rigid interpolation approach can be applied to arbitrary compatible point sets. For example, to generate morphing between a pair of compatible triangulations, we take their vertices as the source and the target sampling points respectively, and construct a disc for each triangulation vertex using its one-ring neighbors, i.e., the vertices that share an edge with it. Finally, a sequence of in-between triangulations which is locally least-distorting is achieved by solving (1), as shown in Fig.2(d).

4.1.1 Globally Consistent Rotation

For large rotations (no less than 180 degrees) between the source and the target objects, rotational inconsistencies may occur causing severe artifacts (Fig.4(c)). This problem stems from that the rotation angle of each local transformation (i.e., α of $\boldsymbol{L}_{(i,j)}$) is determined independently (the small magnitude rotation is chosen), but oblivious to its neighbors. Given two neighboring local transformations $\boldsymbol{L}_{(i_1,j_1)}$ and $\boldsymbol{L}_{(i_2,j_2)}$ with α_1 and α_2 as their rotation angles, in general, if there exists no significant rotation difference, i.e., $|\alpha_1 - \alpha_2| < \pi$, the rotations are consistent; otherwise, either α_1 or α_2 should be corrected. We can correct a rotation angle α by: $\alpha = \alpha \pm 2k\pi$ for $k \in \mathbb{N}^*$. In our implementation, the smallest k, which can eliminate the significant rotation difference, is used. Our idea is therefore to find a consistent global rotation where rotations are consistent between all pairs of neighboring local transformations.

We can solve the problem using a graph. The graph contains one node N_k per local transformation $\boldsymbol{L}_{(i,j)}$. Given two local transformations $\boldsymbol{L}_{(i_1,j_1)}$ and $\boldsymbol{L}_{(i_2,j_2)}$, we consider them to be neighbors when the point $\boldsymbol{p}_{j_1}$ is in the disc $\mathcal{P}_{j_2}$ or the point $\boldsymbol{p}_{j_2}$ is in the disc $\mathcal{P}_{j_1}$. For each pair of neighboring local transformations, we add an edge between their associated nodes in the graph. To make a globally consistent rotation, a rotation is determined for certain local transformation $\boldsymbol{L}_{(i,j)}$, and then the consistent rotation is "propagated" to neighboring local transformations via the graph. Intuitively, for smooth morphing, we would like to choose an order of propagation that favors the propagation from $\boldsymbol{L}_{(i_1,j_1)}$ to $\boldsymbol{L}_{(i_2,j_2)}$ if their rotation magnitudes are close. To this end, we assign each edge (k_1, k_2) in the graph the cost, $||\alpha_1| - |\alpha_2||$. Then, by traversing the minimal spanning tree of the resulting graph, the favorable propagation order can be achieved. In implementation, we use the small magnitude rotation at default and choose the local transformation whose rotation magnitude is minimal as the initial one. Then, rooting at this initial node, the tree is traversed in depth-first order, assigning each local transformation a rotation that is consistent with that of its parent. That is, if during traversal, the current local transformation $\boldsymbol{L}_{(i_1,j_1)}$ has been assigned the rotation α_1 and $\boldsymbol{L}_{(i_2,j_2)}$ is the next local transformation to be visited, then α_2 is corrected if $|\alpha_1 - \alpha_2| \geqslant \pi$. As shown in Fig.2 and Fig.4(d), this consistent rotation algorithm works well no matter when globally or locally large rotations occur.

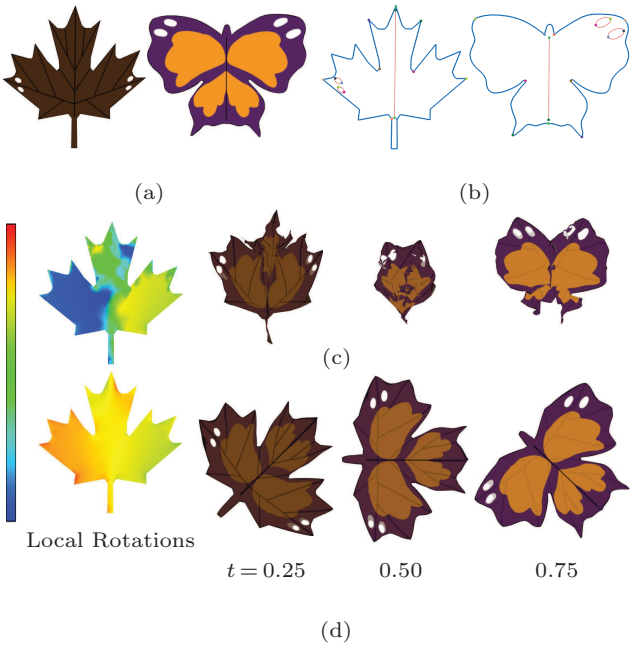


Fig.4. Morphing with large rotations. (a) Input shapes: a leaf and a butterfly. (b) Feature curves and vertex correspondence. Note that 1) the root of the leaf corresponds to the head of the butterfly, thus existing a large rotation (about 180 degrees) from the source to the target shape, and 2) just half of the four white ellipse-like features in the shapes are aligned when generating the compatible triangulations. For this case, the morphing results without and with making consistent global rotation are shown in (c) and (d), respectively. The first column of (c) and (d) shows the local rotations from the source to the target shape during morphing. Additionally, ghosting artifacts are demonstrated in (d) where two white ellipse-like features are not aligned in the compatible triangulations.

4.2 Enhancing Motion Dynamics via Skeleton

To increase visual impact of the morphing sequence, the user is allowed to enhance its motion dynamics via a skeleton. To do so, a sequence of skeletons is first derived from the intermediate and target shapes, with respect to the pre-defined reference skeleton and the triangle-to-triangle correspondence among the interpolation sequence (see Figs.5(a) and 5(b)). The user then selects any intermediate skeleton and edits it, where each of modified intermediate skeletons is stored as a "control skeleton". The system will automatically apply the transformations of the control skeletons to the morphing sequence, leading to enhancing motion dynamics (see Fig.5(c)). The whole process is conducted using a two-phase propagation scheme.

4.2.1 Phase-1: Frame-to-Frame Propagation

The first step is to propagate the transformations of the control skeletons which are made on one or multiple intermediate frames over the whole interpolation sequence. We do this by taking the reference skeleton, control skeletons and the skeleton of the target shape as keys, and interpolating among them to adjust the sequence of intermediate skeletons (Fig.6). As shown in Fig.6(b), directly interpolating the skeleton joints along a linear trajectory or a higher order polynomial curve tends to produce shrinkages, especially when large rotations occur. To avoid shrinkages, we instead interpolate the intrinsic parameters of the skeletons, i.e., joint angles and bone lengths. During interpolation, each adjusted intermediate skeleton can be reconstructed from their intrinsic parameters using the forward kinematic scheme[35] (see Fig.6(a)).

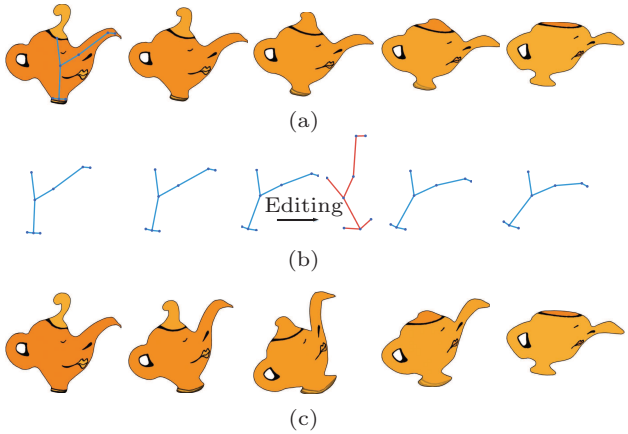


Fig.5. Motion-controllable morphing. (a) An interpolation sequence and the pre-defined reference skeleton on the source shape. (b) The derived skeleton sequence which can be modified at any intermediate frame. (c) The skeleton transformation is propagated over the interpolation sequence, leading to desired motion dynamics.

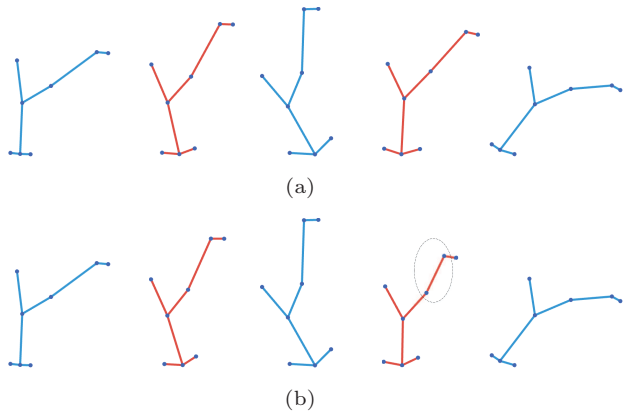


Fig.6. Skeleton interpolation. The left, middle and right are the keys while the others are intermediate skeletons. (a) Interpolation using intrinsic parameters. (b) Interpolation of skeleton joints along a cubic polynomial curve, where shrinkages occur on the skeleton bones, as shown in the dashed ellipse.

4.2.2 Phase-2: Skeleton-to-Triangulation Propagation

Now, for each intermediate frame, two skeletons exist: one describes the pose of the intermediate shape in the rigidity-preserving interpolation and the other corresponds to the pose that is desired in the improved motion dynamics (see Fig.7(a)). We denote the positions of the joints in the two skeletons as $\{\boldsymbol{J}_k\}$ and $\{\boldsymbol{J}_k{}'\}$, respectively. Given an intermediate frame, in order to make the intermediate triangulation at this frame consistent with the desired pose represented by $\{\boldsymbol{J}_k{}'\}$, a relatively simple-minded approach is to recompute the vertex positions of the intermediate triangulation by incorporating the positional constraints of $\{\boldsymbol{J}_k{}'\}$. Since each joint $\boldsymbol{J}_k$ lies in a triangle of the intermediate triangulation, we can compute its barycentric coordinates relative to the triangle vertices and assemble them into a row vector $\boldsymbol{C}_k$. The corresponding $\boldsymbol{J}_k{}'$ can thus be represented by: $\boldsymbol{J}_k{}' = \boldsymbol{C}_k \boldsymbol{V}_k(t)$, where $\boldsymbol{V}_k(t)$ is a column vector composed of three 2D vectors corresponding to the updated vertex positions of the triangle. Thus, the vertex positions of the intermediate triangulation can be updated by replacing (1) with:

$$\arg\min_{V(t)} \quad E_{V(t)} + \sum_k \| \boldsymbol{C}_k \boldsymbol{V}_k(t) - \boldsymbol{J}_k{}' \|^2. \qquad (2)$$

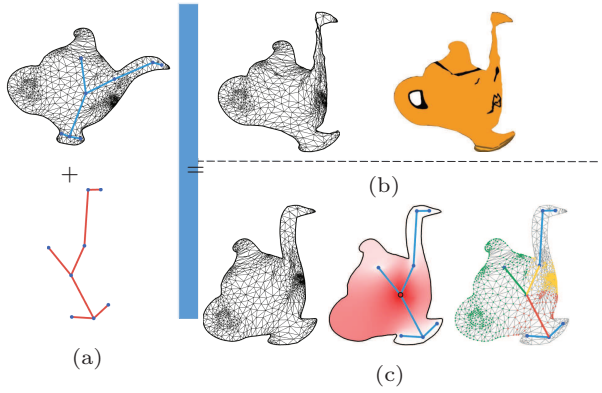


Fig.7. Skeleton-to-triangulation propagation. (a) Two poses described by skeletons exist at any intermediate frame, e.g., at $t = 0.5$. (b) Direct incorporation of the desired pose as positional constraints. (c) Pose incorporation by spreading the bones' transformations (the first column). In (c), the second column shows the influence of a skeleton joint (i.e., the red dot) over the triangulation (the lighter the color, the less the influence), while the last column highlights three selected bones and the triangulation vertices that are "bound" to them.

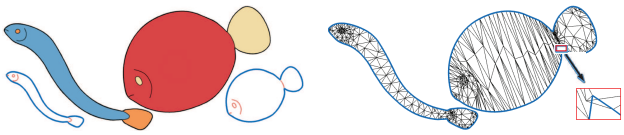
However, in practice we find that the motion sequence determined by the rigidity-preserving interpolation is usually quite different from the motion dynamics defined by the skeleton, i.e., large transformation may exist between the two skeletons represented by $\{\boldsymbol{J}_k\}$ and $\{\boldsymbol{J}_k{}'\}$, respectively. Consequently, this simple method tends to produce artifacts (see Fig.7(b)).

To tackle the above problem, we spread the transformations of the skeleton bones over the intermediate triangulation in a smooth and localized manner. For each skeleton bone (k_1, k_2), we determine its transformation as the rotation angle from vector $\boldsymbol{J}_{k_1}\boldsymbol{J}_{k_2}$ to $\boldsymbol{J}_{k_1}{}'\boldsymbol{J}_{k_2}{}'$ and the scale ratio between $\|\boldsymbol{J}_{k_1}{}' - \boldsymbol{J}_{k_2}{}'\|$ and $\|\boldsymbol{J}_{k_1} - \boldsymbol{J}_{k_2}\|$. We use the harmonic weights to propagate the bones' transformations. The harmonic weights are computed using the approach of [36] and need to be calculated only once with respect to the source triangulation and the reference skeleton, since each intermediate skeleton is derived from the reference skeleton. Then, the intermediate triangulation can be "bound" to the skeleton via a harmonic weight $\lambda_k(v)$ of each skeleton joint k at every triangulation vertex v. As shown in Fig.7(c), the harmonic weights are smooth and localized, and thus can be used as the influence of each skeleton joint to the triangulation vertices. Using the influence weights, each triangulation vertex v can be further "bound" to a skeleton bone (k_1, k_2) for which the value of $(\lambda_{k_1}(v)+\lambda_{k_2}(v))$ is maximum among all the skeleton bones. We then apply the transformation of each skeleton bone to the triangulation vertices that are bound to it. In order to establish a natural adjustment of the intermediate triangulation, the bones' transformations should be combined with those determined by the rigidity-preserving interpolation in a consistent way. It can be done as follows. For each triangulation vertex, let $\boldsymbol{p}_i$ and $\boldsymbol{q}_i$ be its corresponding source and target sampling points respectively in the disc-based rigid interpolation, and we add the rotation and scale components of the bone to which the vertex is bound, respectively, to the corresponding components of each intermediate local transformation in the set of $\{\boldsymbol{L}^t_{(i,j)}\}_j$ which is associated with the discs $\mathcal{P}_i$ and $\mathcal{Q}_i$. Finally, we solve (2) to obtain the new vertex positions of the intermediate triangulation (see Fig.7(c)). Please note that the inconsistency may occur between the updated intermediate local transformations, but the optimization of (2) will spread the inconsistency over the whole triangulation so that the side effect is reduced to the minimum.

5 Compatible Triangulation

Given the source and the target shapes with boundaries and interior feature curves, we construct the compatible triangulations in an intuitive way: triangulate one shape (the source shape) and map it to the

other (the target shape), and vice versa. We use Triangle[37], a constrained Delaunay triangulation algorithm, to generate the triangulation of the source shape first, where the boundary of the source shape and its interior feature curves are taken as the constraints. Then, we adopt the barycentric mapping[38] to map the source triangulation to the target one. While many forms of barycentric coordinates exist[38], we use the harmonic coordinates of [36], since they are well-defined for not only the boundary of a polygonal domain but also its interior vertices and lines. An example is shown in Fig.8. Note that the harmonic coordinates can also be substituted with other alternatives such as the (bounded) biharmonic weights[39].



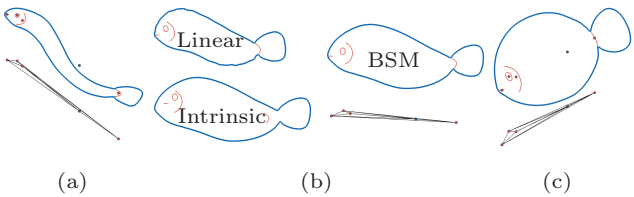
(a) (b)

Fig.8. Compatible triangulation. Given (a) input shapes with the boundaries and the interior feature curves, the source triangulation is first generated and then mapped to (b) the target one via barycentric mapping with harmonic coordinates. Note that triangle flips may occur during the direct mapping especially when large geometric changes exist from the source to the target configurations, as shown in the red box.

However, since there is no set of barycentric coordinates which can yield bijective mappings for any choice of source and target polygons[40], triangle flips may happen in the resulting target triangulation (see Fig.8). To address this problem, our solution is based on the observation that the concept of composite barycentric mapping can be straightforwardly applied to the harmonic coordinates. In the composite barycentric mapping[41], it was theoretically proven that any barycentric mapping is bijective if the source and the target polygons are sufficiently close. Therefore, the main idea is to create between the source and the target shapes a sequence of intermediate boundaries and feature curves which are sufficiently close, and then to successively map the triangulation vertices from one to the next. We refer the readers to [41] for the algorithm details.

The generation of intermediate boundaries and feature curves between the source and the target shapes corresponds to the shape interpolation problem. A straightforward approach is to conduct the interpolation by linearly interpolating the corresponding vertex positions. Such an approach is simple but it does not take into account the geometric shape of the boundaries and feature curves, possibly leading to shape distortion (see Fig.9(b)). The intrinsic parameters[9], i.e., the edge lengths and vertex angles, can be used to describe the shape of individual curves. However, we find that the spatial relationships between the boundaries and the interior feature curves are also important for their natural transition from the source to the target configurations. Therefore, we design a barycenter-based shape model (BSM) to represent the overall shape of the boundaries and feature curves. In the BSM model, we represent the geometric shape of individual curves using their intrinsic parameters, while describing the spatial relationships between the curves using a structure called a barycenter mesh. As proposed in [9], we use the edge lengths and vertex angles of the curves as their intrinsic parameters. To construct the barycenter mesh, we take the barycenter of each curve as its spatial proxy and then encode the spatial relationships between the curves by connecting their barycenters to each other, as shown in Figs.9(a) and 9(c). Let $\mathcal{M}^0$ be the BSM model of the boundary and the interior feature curves in the source shape. The model includes two components: $\mathcal{M}_s^0$ and $\mathcal{M}_p^0$, where $\mathcal{M}_s^0$ corresponds to the set of intrinsic parameters of all of the individual curves and $\mathcal{M}_p^0$ is the corresponding barycenter mesh. Similarly, we can define $\mathcal{M}^1$ for the boundary and the interior feature curves in the target shape. At arbitrary interpolating time t, the intermediate boundary and feature curves can thus be determined by computing an intermediate BSM model $\mathcal{M}^t$: for each intermediate curve, the intrinsic parameter set of $\mathcal{M}_s^t$ decides its geometric shape while $\mathcal{M}_p^t$ fixes its barycenter, i.e., the curve's position. The intrinsic parameters in $\mathcal{M}_s^t$ can be obtained by interpolating the corresponding intrinsic parameters in the sets $\mathcal{M}_s^0$ and $\mathcal{M}_s^1$:

$$\mathcal{M}_s^t = \mathcal{M}_s^0 + t(\mathcal{M}_s^1 - \mathcal{M}_s^0).$$



(a) (b) (c)

Fig.9. Barycenter-based shape model. In (a) and (c), the bottoms are the corresponding barycenter meshes of the source and the target shapes, where the circular dots correspond to the barycenter of each individual curve. In (b), the intermediate boundary and feature curves at $t = 0.5$ are obtained by interpolating vertex positions along linear trajectory, intrinsic parameters, and BSM models, respectively; additionally, the result of interpolating the source and the target barycenter meshes at $t = 0.5$ is also shown.

Similarly, $\mathcal{M}_p^t$ can be determined by interpolating the barycenter meshes $\mathcal{M}_p^0$ and $\mathcal{M}_p^1$. To preserve the encoded spatial relationships, the interpolation should be least-distorting, i.e., in a rigidity-preserving way. Such a requirement can be easily achieved by applying our disc-based rigidity-preserving interpolation technique to the source and the target barycenter meshes, i.e., $\mathcal{M}_p^0$ and $\mathcal{M}_p^1$, since they correspond to a pair of compatible point sets. A result of composite barycentric mapping is shown in Fig.10.

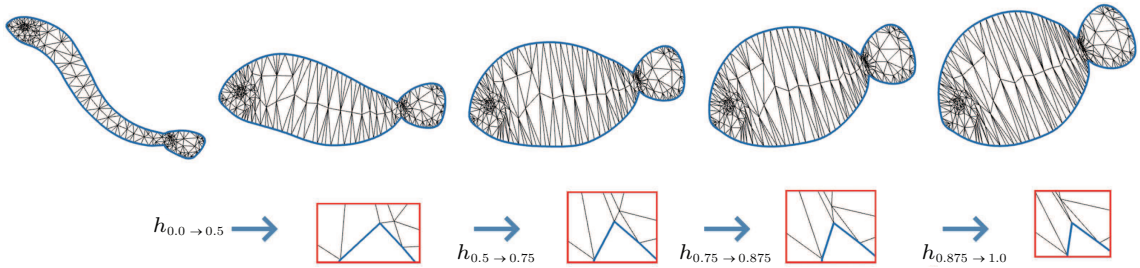


Fig.10. Composite barycentric mapping: the triangle flips are avoided by generating a sequence of intermediate boundaries and feature curves which are sufficiently close and then successively conducting the barycentric mapping from one to the next, where $h_{t_1\to t_2}$ represents the barycentric mapping from the interpolating time t_1 to t_2.

6 Results

We have implemented our method on a PC with 3.0 GHz Intel Core 2 Duo CPU and 8 GB RAM (single thread) and the performance of current implementation is summarized in Table 1. The bottleneck of the computational cost lies in the generation of compatible triangulations, which is mainly determined by the iteration number of barycentric mapping and the vertex number of the triangulation. In our experiments, using the BSM model, the mapping can be generally established within a reasonable number of iterations (4−13 iterations). One can reduce the cost by using a coarse triangulation, but it might cause non-smooth morphing results. In general, the delay of constructing compatible triangulations is acceptable since it needs to be conducted only once. In addition, the optimization problems of (1) and (2) can be solved by using a standard linear least squares minimization. In our implementation, we use UMFPACK[42] to solve them and the solving process always converges in our experiments.

Table 1. Performance Statistics

Fig.	Curves	#Tri.Vert	Time (s)	FPS
Fig.1	10	551	2.53	131
Fig.2	10	704	7.23	66
Fig.5	9	876	8.43	64
Fig.12(b)	3	672	1.04	117
Fig.12(c)	3	683	1.27	108
Fig.14	21	673	4.48	106
Fig.15	30	357	1.07	161

Note: "Curves" corresponds to the number of feature curves, "#Tri.Vert" is the vertex number of compatible triangulations, "Time" denotes in seconds the computational time for generating the compatible triangulations, and "FPS" shows the frame rate of the disc-based rigidity-preserving interpolation.

6.1 Results of Compatible Triangulation

We have applied the proposed approach to various examples such as cartoon characters or images. As shown in Fig.11, since our compatible triangulation algorithm takes into account both the boundaries of the shapes and their interior features, undesirable results such as "ghosting" artifacts are effectively eliminated. The existing bridge technique also allows to incorporate interior closed curves or points when constructing the compatible triangulations[17,21]; however, it is difficult to apply such technique to the interior open curves.

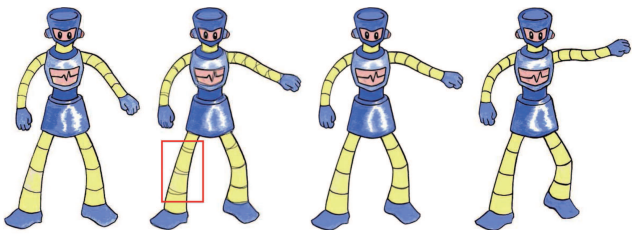


Fig.11. Comparison between [21] and our algorithm. Given (a) source and (d) target shapes, the compatible triangulations are generated using the approach of [21] and ours and the corresponding interpolation results at $t = 0.5$ are shown in (b) and (c), respectively. (b) is depicted from Fig.11 of [21]. Since the bridge technique is difficult to incorporate the interior open curves, it cannot be applied to align the line-like features, as illustrated in the red box of (b).

Moreover, our algorithm tends to produce "good" triangles in the resulting triangulations since the barycentric mapping is inherently smooth. In Fig.12, we show that when constructing the compatible triangulations the mapping not only can be performed from the source to the target shape but also can be performed from the target to the source shape. Though

skinny triangles may still occur when the boundaries or feature curves between the source and the target shapes are rather different, time-consuming triangulation optimization process is not necessary yet, since our disc-based rigidity-preserving interpolation approach is insensitive to the triangle quality, as shown in Fig.13(b).

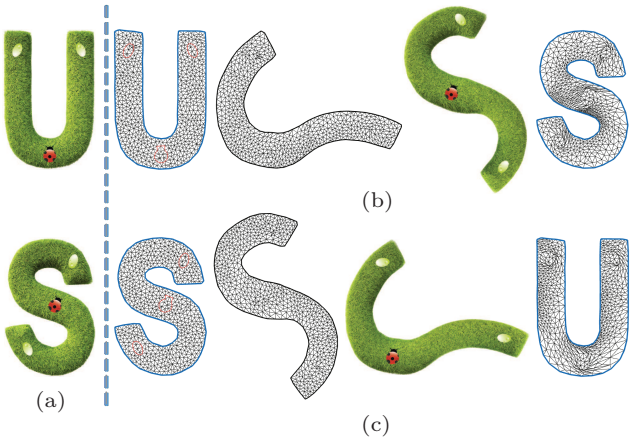


Fig.12. Morphing between the letters U and S. (a) Input shapes. (b) Morphing from U to S. (c) Morphing from S to U. Note that in (b) or (c), the first corresponds to the source triangulation where the feature curves are shown in pink and the last corresponds to the target triangulation.

6.2 Results of Motion-Controllable Rigid Interpolation

Our disc-based rigidity-preserving interpolation approach generates the examples that are comparable to the traditional triangle-based rigid interpolation approaches such as [13, 19, 20]. However, our approach has two advantages: 1) it is suitable for not only compatible triangulations but also arbitrary compatible point sets such as barycenter meshes (see Fig.9), and 2) it does not need to compute the affine transformation between each pair of corresponding triangles, and thus is robust to the skinny triangles, as shown in Fig.13. Furthermore, as demonstrated in Figs.1, 2, 4(d), and 12, our approach works well in the scenarios of globally or locally large rotations by employing the graph-based rotation-consistent scheme.

To increase visual impact, our approach allows the user to enhance motion dynamics over the morphing sequence via an easy and intuitive skeleton control, as shown in Figs.1, 2, 5, and 14. As shown in the accompanying video[①], for example, a vivid animation has been created via our approach using only six keyframes. To the best of our knowledge, our approach is the first one to incorporate the skeleton control into the rigidity-preserving interpolation process for enhancing motion dynamics over the morphing sequence. Consequently, our approach constitutes a powerful and easy-to-use morphing tool for 2D animation.

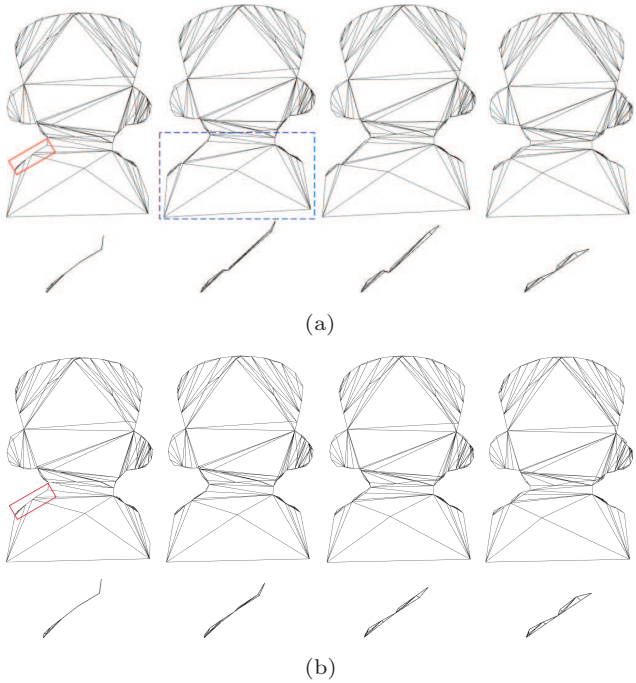


Fig.13. Robust interpolation of the compatible triangulations. (a) Result generated by the traditional triangle-based rigid interpolation of [13]. (b) Result generated by our disc-based rigid interpolation. The compatible triangulations are generated using [22] without the post optimization. Note that some triangles in the source triangulation are nearly degenerated to a line, thus skinny triangles occur, as shown in the red box. While skinny triangles are naturally transformed in (b), distortions occur in (a) since the triangle-based rigid interpolation is sensitive to the triangle quality, as shown in the dashed blue box.

6.3 Artistic Control and User Interaction

In our experiments, the workload of user interaction in drawing feature curves is not heavy. That is because 1) the user only needs to place a small number of feature curves on the major features, as shown in Table 1, and 2) the system does not set up any constraint when the user draws the feature curves and it will automatically establish the point-to-point correspondence between each pair of corresponding feature curves. Despite this, the automatic extraction of feature curves is preferable and we leave this to the future work.

Note that though our approach belongs to the category of shape morphing, it can be straightforwardly applied in the scenarios of image morphing, thus providing a useful alternative in the toolbox for the image mor-

①https://pan.baidu.com/s/1TrYvIiI2QRCIv2L88djTdQ (code: kzcx), July 2019.

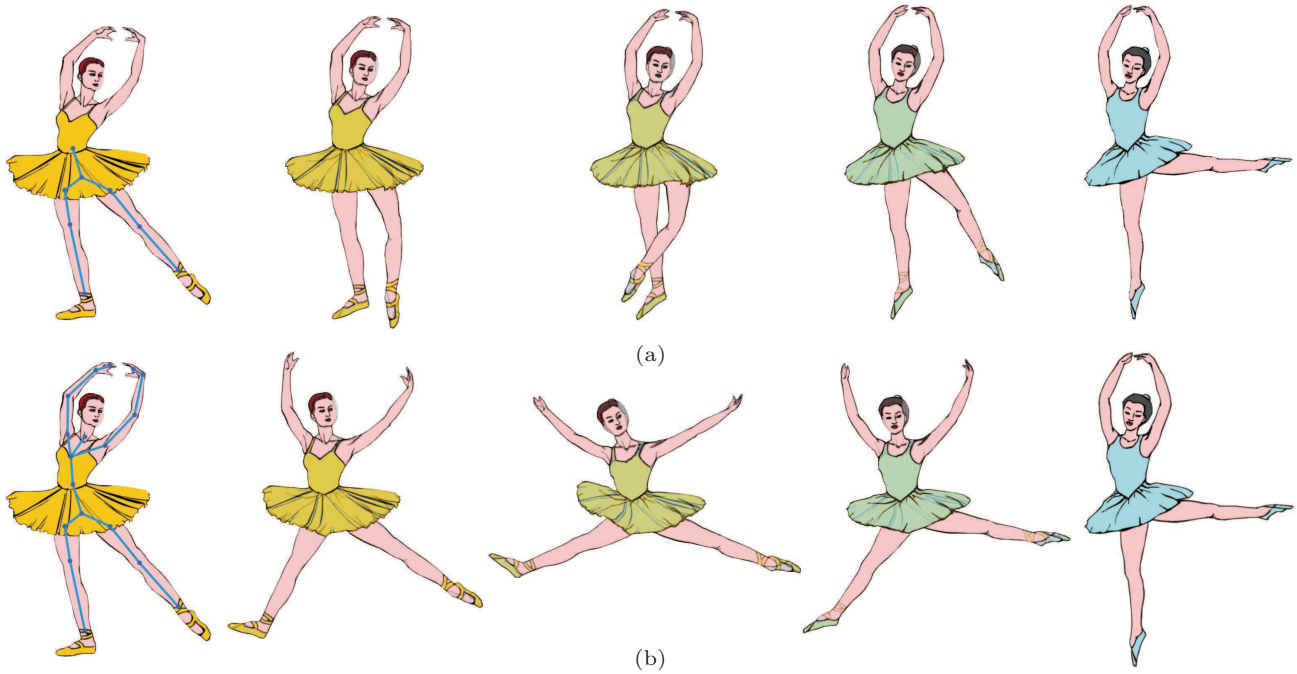


Fig.14. Morphs between two dancing poses using skeleton control. (a) Skeleton defined regionally for only the control of the lower body. (b) Full control of the body. At each row, the pre-defined reference skeleton is shown on the source shape and the third column corresponds to the frame where the skeleton is modified.

phing applications. Fig.15 demonstrates the versatility of our approach. Additionally, it is also shown that the feature curves can provide the user precise and intuitive artistic control over the local morphing behavior, which is very useful in certain scenarios.

6.4 Limitation

A limitation of our approach is that the new compatible triangulation algorithm cannot guarantee to produce the desired compatible triangulations for arbitrary source and target shapes, as illustrated in Fig.16. In this figure, the topology of some adjacent features changes between the source and the target shapes, e.g., two features are intersected in one shape but separated in the other. Obviously, if feature curves are specified on these features, the composite barycentric mapping will fail to generate the desired target triangulation without triangle flips, because it implies that the connectivity of the corresponding triangles covering these features should be different between the source and the target triangulations. Furthermore, though the BSM model can preserve well the spatial relationships among the boundary and feature curves during the interpolation, the intersection between the curves may still occur especially when the spatial layout among the adjacent curves differs greatly from the source to the target shape, as shown in Fig.16(b) where a circle-like feature curve lies to the right of a line-like one in the source shape but moves to its left in the target shape.

7 Conclusions

This paper presents a rigidity-preserving morphing technique that blends a pair of 2D shapes in a controllable manner. First, we adopted the progressive mapping to construct the compatible triangulations for the source and the target shapes. In the compatible triangulations, not only the boundaries but also the interior features of the source and the target shapes are precisely aligned. Second, a robust and motion-controllable rigidity-preserving transformation scheme was proposed to blend the compatible triangulations, ultimately leading to a morphing sequence which is appearance-preserving and with the desired motion dynamics. Our approach constitutes a powerful and easy-to-use morphing tool for 2D animation.

Considering the future work, we plan to extend our approach to 3D (three-dimensional) objects. Since 3D analog to compatible triangulations is still considered challenging, it is worthwhile developing a 3D composite mapping to dissect a pair of 3D meshes into compatible simplicial complexes. It is attractive to extend the disc-based rigidity-preserving interpolation to generate

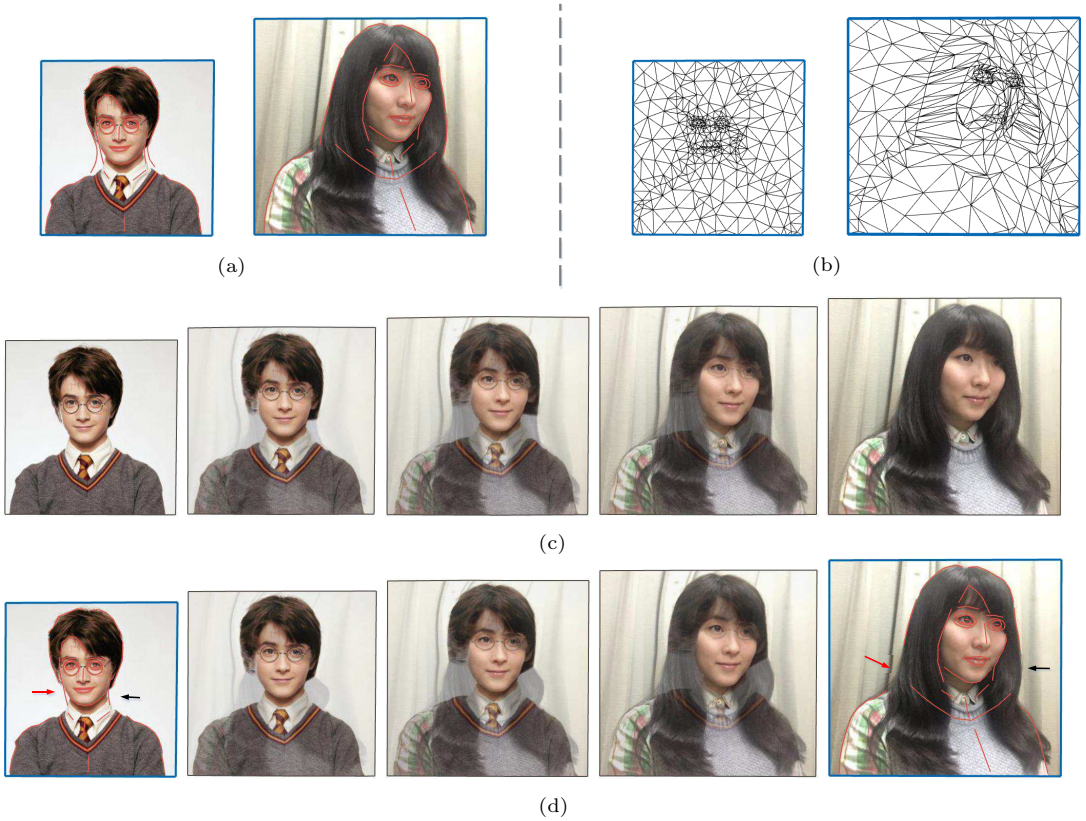


Fig.15. Image morphing using our method. (a) Source and target images as well as their feature curves where the bounding box of each image is taken as its boundary. (b) Compatible triangulations and (c) morphing sequence generated by our approaches. Note that the side hair of Harry Potter naturally grows out according to the shape of the corresponding feature curves instead of the shape of the corresponding hair of the girl. Without the control of the feature curves, (d) an unnatural result might occur since the shape of the Harry's side hair is not consistent with the girl's. Also, note that the feature curves can provide precise control to preserve the shape of the Harry's glasses, as shown in (c).

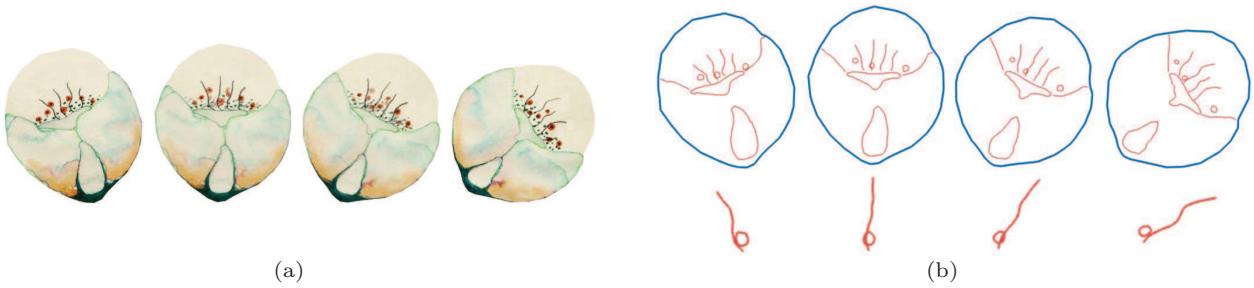


Fig.16. Limitation of our compatible triangulation algorithm. (a) Some features cannot be aligned in the morphing results due to that most of the red circular features change their spatial relations with the black curve-like features between the source and the target shapes. (b) Intersection between feature curves occurs during the interpolation.

the morphing sequence for 3D objects such as meshes or point clouds. Furthermore, we plan to investigate the possibility of automating the extraction of feature curves while effortlessly incorporating artistic control whenever needed. For some specific objects such as human faces or those with similar appearance, the existing automatic alignment approaches such as [8, 43] can already provide some insight to do this work.

References

[1] Fekete J D, Bizouarn É, Cournarie É, Galas T, Taillefer F. TicTacToon: A paperless system for professional 2D animation. In *Proc. the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, August 1995, pp.79-90.

[2] Yang W. Context-aware computer aided inbetweening. *IEEE Transactions on Visualization and Computer Graphics*, 2018, 24(2): 1049-1062.

[3] Beier T, Neely S. Feature-based image metamorphosis. In *Proc. the 19th Annual Conference on Computer Graphics and Interactive Techniques*, July 1992, pp.35-42.

[4] Lee S Y, Chwa K Y, Shin S Y. Image metamorphosis using snakes and free-form deformations. In *Proc. the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, August 1995, pp.439-448.

[5] Wolberg G. Image morphing: A survey. *The Visual Computer*, 1998, 14(8/9): 360-372.

[6] Fang H, Hart J C. Detail preserving shape deformation in image editing. *ACM Transactions on Graphics,* 2007, 26(3): Article No. 12.

[7] Glocker B, Komodakis N, Tziritas G, Navab N, Paragios N. Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis*, 2008, 12(6): 731-741.

[8] Liao J, Lima R S, Nehab D, Hoppe H, Sander P V, Yu J. Automating image morphing using structural similarity on a halfway domain. *ACM Trans. Graph.*, 2014, 33(5): Article No. 168.

[9] Sederberg T W, Gao P, Wang G, Mu H. 2-D shape blending: An intrinsic solution to the vertex path problem. In *Proc. the 20th Annual Conference on Computer Graphics and Interactive Techniques*, August 1993, pp.15-18.

[10] Fu H, Tai C l, Au K C. Morphing with Laplacian coordinates and spatial-temporal texture. In *Proc. the 13th Pacific Conference on Computer Graphics and Applications*, October 2005, pp.100-102.

[11] Sumner R W, Zwicker M, Gotsman C, Popović J. Mesh-based inverse kinematics. *ACM Trans. Graph.*, 2005, 24(3): 488-495.

[12] Xu D, Zhang H, Wang Q, Bao H. Poisson shape interpolation. In *Proc. the 9th ACM Symposium on Solid and Physical Modeling*, June 2005, pp.267-274.

[13] Alexa M, Cohen-Or D, Levin D. As-rigid-as-possible shape interpolation. In *Proc. the 27th Annual Conference on Computer Graphics and Interactive Techniques*, July 2000, pp.157-164.

[14] Hahmann S, Bonneau G P, Caramiaux B, Cornillac M. Multiresolution morphing for planar curves. *Computing*, 2007, 79(2/3/4): 197-209.

[15] Yang W, Feng J. 2D shape morphing via automatic feature matching and hierarchical interpolation. *Computers and Graphics*, 2009, 33(3): 414-423.

[16] Dym N, Shtengel A, Lipman Y. Homotopic morphing of planar curves. *Computer Graphics Forum*, 2015, 34(5): 239-251.

[17] Tal A, Elber G. Image morphing with feature preserving texture. *Computer Graphics Forum*, 1999, 18(3): 339-348.

[18] Floater M, Gotsman C. How to morph tilings injectively. *Journal of Computational and Applied Mathematics*, 1999, 101(1/2): 117-129.

[19] Surazhsky V, Gotsman C. Intrinsic morphing of compatible triangulations. *International Journal of Shape Modelling*, 2003, 9(2): 191-201.

[20] Baxter III W, Barla P, Anjyo K. Rigid shape interpolation using normal equations. In *Proc. the 6th International Symposium on Non-Photorealistic Animation and Rendering*, June 2008, pp.59-64.

[21] Baxter III W V, Barla P, Anjyo K. Compatible embedding for 2D shape animation. *IEEE Transactions on Visualization and Computer Graphics*, 2009, 15(5): 867-879.

[22] Aronov B, Seidel R, Souvaine D. On compatible triangulations of simple polygons. *Computational Geometry*, 1993, 3: 27-35.

[23] Surazhsky V, Gotsman C. High quality compatible triangulations. *Engineering with Computers*, 2004, 20(2): 147-156.

[24] Liu Z, Leung H, Zhou L, Shum H P H. High quality compatible triangulations for 2D shape morphing. In *Proc. the 21st ACM Symposium on Virtual Reality Software and Technology*, November 2015, pp.85-94.

[25] Huang J, Tong Y, Zhou K, Bao H, Desbrun M. Interactive shape interpolation through controllable dynamic deformation. *IEEE Transactions on Visualization and Computer Graphics*, 2011, 17(7): 983-992.

[26] Whited B, Noris G, Simmons M, Sumner R W, Gross M, Rossignac J. BetweenIT: An interactive tool for tight inbetweening. *Computer Graphics Forum*, 2010, 29(2): 605-614.

[27] Ruprecht D, Müller H. Image warping with scattered data interpolation. *IEEE Comput. Graph. Appl.*, 1995, 15(2): 37-43.

[28] Reeves W T. Inbetweening for computer animation utilizing moving point constraints. In *Proc. the 8th Annual Conference on Computer Graphics and Interactive Techniques*, August 1981, 263-269.

[29] Popović J, Seitz S M, Erdmann M, Popović Z, Witkin A. Interactive manipulation of rigid body simulations. In *Proc. the 27th Annual Conference on Computer Graphics and Interactive Techniques*, July 2000, pp.209-217.

[30] Zhu Y, Popović J, Bridson R, Kaufman D M. Planar interpolation with extreme deformation, topology change and dynamics. *ACM Trans. Graph.*, 2017, 36(6): Article No. 213.

[31] Burtnyk N, Wein M. Interactive skeleton techniques for enhancing motion dynamics in key frame animation. *Commun. ACM*, 1976, 19(10): 564-569.

[32] Baxter W. Point-based rigid shape interpolation. In *Proc. the 33rd International Conference on Computer Graphics and Interactive Techniques*, July 2006, Article No. 92.

[33] Sorkine O, Alexa M. As-rigid-as-possible surface modeling. In *Proc. the 5th Eurographics Symposium on Geometry Processing*, July 2007, pp.109-116.

[34] Igarashi T, Moscovich T, Hughes J F. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics*, 2005, 24(3): 1134-1141.

[35] Craig J. Introduction to Robotics: Mechanics and Control (3rd edition). Pearson, 2004.

[36] Joshi P, Meyer M, DeRose T, Green B, Sanocki T. Harmonic coordinates for character articulation. *ACM Trans. Graph.*, 2007, 26(3): Article No. 71.

[37] Shewchuk J R. Triangle: Engineering a 2D quality mesh generator and delaunay triangulator. In *Proc. the 1st ACM Workshop on Applied Computational Geometry*, May 1996, pp.203-222.

[38] Floater M S. Generalized barycentric coordinates and applications. *Acta Numerica*, 2015, 24: 161-214.

[39] Jacobson A, Baran I, Popovic J, Sorkine O. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 78.

[40] Jacobson A. Bijective mappings with generalized barycentric coordinates: A counterexample. *Journal of Graphics Tools*, 2013, 17(1/2): 1-4.

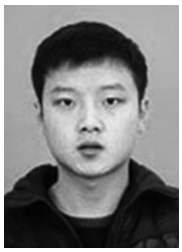
[41] Schneider T, Hormann K, Floater M S. Bijective composite mean value mappings. *Comput. Graph. Forum*, 2013, 32(5): 137-146.

[42] Davis T A. Algorithm 832: UMFPACK V4.3 — An unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 2004, 30(2): 196-199.

[43] Zhang Z, Luo P, Loy C C, Tang X. Learning deep representation for face alignment with auxiliary attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 38(5): 918-930.

Wen-Wu Yang is currently an associate professor at the School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou. Dr. Yang received his Ph.D. degree in computer science from the State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, in 2009. His research interests include cartoon animation, motion capture and motion synthesis.

Jing Hua is a professor and the founding director of Computer Graphics and Visualization Laboratory at Computer Science Department at Wayne State University, Michigan. Dr. Hua received his Ph.D. degree (2004) in computer science from the State University of New York at Stony Brook, New York. His research interests include computer graphics, visualization, image analysis and informatics.



Kun-Yang Yao is currently a graduate student at the School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou. His research interests include computer graphics and image analysis.