DOI: 10.1111/j.1467-8659.2012.03218.x
Pacific Graphics 2012
C. Bregler, P. Sander, and M. Wimmer
(Guest Editors)

Volume 31 (2012), Number 7

Structure Preserving Manipulation and Interpolation for Multi-element 2D Shapes

Wenwu Yang[1,2] Jieqing Feng[†1] Xun Wang[2]

[1]State Key Laboratory of CAD&CG, Zhejiang University [2]Zhejiang Gongshang University

Abstract

This paper presents a method that generates natural and intuitive deformations via direct manipulation and smooth interpolation for multi-element 2D shapes. Observing that the structural relationships between different parts of a multi-element 2D shape are important for capturing its feature semantics, we introduce a simple structure called a feature frame to represent such relationships. A constrained optimization is solved for shape manipulation to find optimal deformed shapes under user-specified handle constraints. Based on the feature frame, local feature preservation and structural relationship maintenance are directly encoded into the objective function. Beyond deforming a given multi-element 2D shape into a new one at each key frame, our method can automatically generate a sequence of natural intermediate deformations by interpolating the shapes between the key frames. The method is computationally efficient, allowing real-time manipulation and interpolation, as well as generating natural and visually plausible results.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Geometry—Shape Deformation

1. Introduction

2D shape deformation is useful in many applications, such as real-time live performance and enriching graphical user interfaces. This technique can also be commonly found in commercial software for video editing or 2D vector-based animation, such as Adobe After Effects, Adobe Flash, or Toon Boom Studio.

A general representation of a 2D shape is the usage of a simple polygon to represent the boundary of the shape, while the interior of the shape is described using the interior texture of the polygon. However, in real-world inputs arising from practical demands, different meaningful parts of a shape are not only disjoined but also often overlapped, i.e., occlusion occurs (see the left column of Fig. 1(a)). For such a shape, a simple polygon is insufficient to fully capture its semantic boundaries, such that the different parts of the shape may fail to move separately during deformation (see Fig. 1(c)).

Multi-element 2D shapes: In practical applications such as vector-based cartoon animation, artists create a complex cartoon character typically by drawing different parts of the character in different layers that are then combined together. Similarly, we propose a multi-element 2D shape representation which can correctly describe the semantic boundaries of the shapes like that in Fig. 1(a). To create a multi-element 2D shape, the user needs to draw a closed or open curve to represent the boundary of each meaningful part of the shape (see the right column of Fig. 1(a)), where each curve is called a sub-shape. With the user-specified layer levels [SSJ*10], all of the sub-shapes are then layered together to form the multi-element 2D shape.

Recently, a variety of 2D shape deformation algorithms that model the rigidity of the shapes [IMH05, WXW*06] have been introduced. These methods allow the user to directly manipulate a shape through a click-and-drag interface and are able to generate intuitive and physically plausible deformations for various 2D shapes [IMH05]. While these methods work well for the 2D shapes whose boundary can be represented by a simple closed polygon, they cannot be applied to the multi-element 2D shapes straightforwardly. For example, in the method of [IMH05], a simple closed polygon is used to describe the shape boundary so that a triangulated mesh can be generated inside the boundary: the deformation

† Correspondence: jqfeng@cad.zju.edu.cn

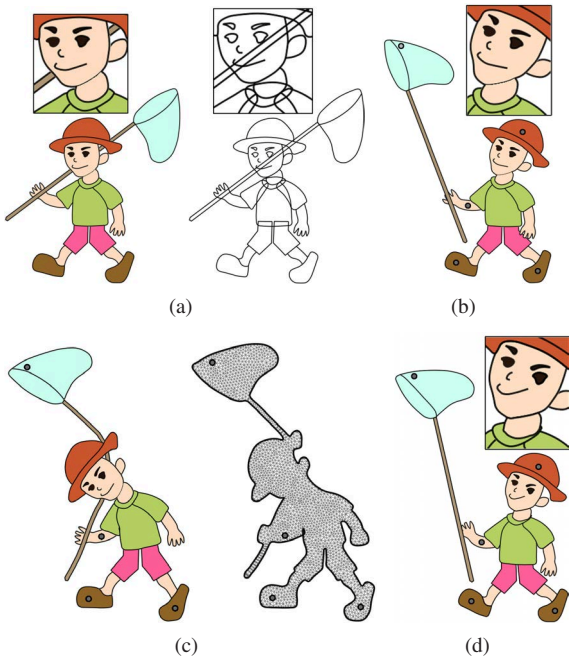


Figure 1: Manipulation of a cartoon boy. (a): input object (left) and the sub-shapes (right); (b): our manipulation approach; (c): the method of Igarashi et al. [IMH05], where the input object is enclosed by a simple polygon and the interior of the polygon is triangulated (right); (d): the approach of [YF09a], where the semantic features on the face of the boy and around his neck are distorted.

is first conducted on the triangle mesh, and then the original shape is mapped from the original mesh to the deformed mesh. As shown in Fig. 1(c), since the triangulation corresponds to a connected manifold mesh, the sub-shapes of a multi-element 2D shape that are spatially overlapped but semantically separated will be tied together, possibly leading to unnatural deformation results. Furthermore, the deformation flexibility of the multi-element 2D shape would be reduced, since a single triangulated mesh could not reveal the separately moving parts of the shape.

One possible approach to address the above problems would be to connect all of the sub-shapes as a whole using line segments, where vertices connected by a line segment are considered as neighbors, and generate a triangulated mesh for each sub-shape, and then conduct the deformation on the triangulation of each sub-shape separately. A similar work had been done in [YF09a]. However, this approach does not account for the structural relationships between different sub-shapes, which would inappropriately distort semantic features in underlying subjects (see Fig. 1(d)).

In this paper, we present a method that generates natural and intuitive deformations for multi-element 2D shapes, where the different shape parts are allowed to deform separately and the structural relationships between them are well preserved. This method is motivated by the observation that the structural relationships between different parts of a multi-element 2D shape are important for capturing its feature semantics. The key idea is to introduce a simple structure which we call the feature frame to represent these structural relationships. The feature frame allows us to preserve the feature semantics of the input multi-element 2D shape during subsequent manipulation and interpolation (see Fig. 1(b)). Additionally, the feature frame is closely related to the shape geometry and naturally reflects the local characteristics of the shape, thus making it convenient to retain local shape features. Furthermore, the feature frame is easy to construct and is applicable to multi-element 2D shapes of arbitrary subjects.

To manipulate a multi-element 2D shape, the user places handles on the shape and moves the handles around (see Fig. 1(b)). Beyond deforming the input multi-element 2D shape into a new one at each key frame, the method can interpolate the shapes between each two successive key frames to generate a sequence of intermediate deformations. In combination, the method of manipulation and interpolation for multi-element 2D shapes enables the user to create visually pleasing animations from single input multi-element 2D shapes of any subjects.

The main contributions of this paper are:

- At first, we introduce a simple structure called the feature frame for encoding the structural relationships between parts of a multi-element 2D shape.
- Based on the feature frame, we then develop a unified framework for the manipulation and interpolation of multi-element 2D shapes.

2. Related work

Our work is closely related to the techniques of 2D shape manipulation and 2D shape interpolation: we cover relevant references below.

Shape manipulation. Many deformation techniques have been developed for shape manipulation. Skeleton-driven technique uses a pre-defined skeleton to manipulate the input shape [LCF00, YN03]. It associates each point of the shape with one or more coordinate frames, each defined by a skeleton bone, and then changes the overall shape relative to the skeleton that is manipulated by the user. However, binding a shape to a skeleton, either manually or automatically, is still a challenging task, especially for the shapes that lack an obvious jointed structure. In FFD, a space deformation is defined via a simple control object for shape manipulation [GB08]. The user manipulates the control object and then the system interpolates the deformation of the control object to the entire space. However, the space warp is

generally defined in a pure mathematical manner (e.g., under the criteria of smoothness and continuity), oblivious to the geometric or structural characteristics of the input shape. Thus, the feature semantics of the shape that is deformed by the space warp could be easily destroyed.

In recent years, a variety of 2D shape deformation algorithms that model the rigidity of the shapes have been introduced [IMH05, WXW*06, SDC09]. These methods generate natural deformation results by minimizing local shape distortions. (These approaches can be regarded as variants of detail-preserving mesh deformation techniques [YZX*04, SCOL*04, BS08].) As mentioned in the introduction, these methods, however, are not suitable for the multi-element 2D shapes. Hornung et al. [HDK07] introduce an augmented version of the as-rigid-as-possible shape manipulation technique to the context of deforming multi-element 2D shapes. However, their method needs the adjacent sub-shapes to share common boundary vertices in order to convey the impression of a connected object, thus is unsuitable for the shapes with disjoined or overlapped parts. Yang et al. [YF09a] unleash this vertex-sharing limitation by using line segments to connect disjoined sub-shapes, and conduct the deformation of each sub-shape separately. Our work on multi-element 2D shape manipulation is also free of this limitation. Moreover, our method takes into account the structural relationships between parts of the shape, which is crucial in preserving the semantic features in the underlying subjects (see Fig. 1(b) and (d)).

Further related to our work, Gal et al. [GSMCO09] and Zheng et al. [ZFCO*11] show that when editing man-made 3D objects, high-level structures (e.g., symmetry, parallelism) can be effectively maintained using an analyze-and-edit approach. Xu et al. [XWY*09] preserve the nature between components of 3D mechanical objects using a joint-aware deformation framework. However, neither of these methods demonstrates how to extend to the case of multi-element 2D shapes for preserving the structural relationships between parts. Ho et al. [HKT10] and Zhou et al. [ZXTD10] represent spatial relationships between object parts using edges that connect vertices between close components. However, the vertex edges may connect the parts that are spatially close but semantically separated, and how to correct such inappropriate part connection with a light work has not been mentioned in their papers. In this work, we define a simple and effective structure called the feature frame to represent the structural relationships between parts of a multi-element 2D shape. The feature frame allows the user to modify alternative semantic relations with a very small amount of assistance. Our work is similar to the method of Sumner et al. [SSP07], where a deformation graph is used to build the local transformations of the input objects. The key difference is that our feature frame focuses on the structural relationships between parts, while such relationships have not been dealt with in the deformation graph.

Shape interpolation. Shape interpolation involves the creation of a smooth transition from a source shape to a target shape [WNS*10]. Because simple linear interpolation of corresponding vertices tends to generate intermediate shapes with shrinkages, Sederberg et al. [SGWM93] propose to interpolate the intrinsic properties (e.g., edge length, vertex angle) of the input shapes. To preserve local volumes, the interior of the shapes is further taken into account by applying the blend to the compatible triangulations of the input objects [ACOL00, BIBA09]. However, these methods are primarily designed for 2D shapes that are represented by a single polygon and are not suitable for the interpolation of multi-element 2D shapes, since they do not account for the structure relationships between shape parts. Additionally, in the interior-considering methods, the automatic creation of compatible triangulations is involved, and though possible for the shapes that are represented by a simple polygon, it is still a challenging work for multi-element 2D shapes where all sub-shapes should be taken into account.

Motivated by the interpolation techniques that are based on multi-resolution representations [HBCC07], Yang et al. [YF09b] introduce a hierarchical interpolation approach. The approach decomposes the input shapes into a pair of coarse polygons and several pairs of corresponding features, and then interpolates the coarse polygons and feature pairs using the interior-considering method [ACOL00] and intrinsic method [SGWM93], respectively. This approach achieves pleasing results: not only the overall shapes of the input objects are blended naturally, but also the details of the input objects are well preserved. Our work on shape interpolation is similar to this method, but extends to the context of interpolating multi-element 2D shapes, where preserving the feature semantics of the input objects is important in obtaining natural results.

3. Feature Frame

This section presents the details of feature frame construction and establishes the association between the input multi-element 2D shape and its feature frame.

3.1. Feature frame construction

Intuitively, the visual appearance or structure of a shape is mainly determined by its salient features. Therefore, it is reasonable to delegate the structural relationships between parts of a multi-element 2D shape to its salient features. Then, the feature frame that represents these structural relationships is constructed through the following steps:

Feature extraction. To extract the salient features, we identify feature points – those that attract more attention in human perception than others. For a multi-element 2D shape, it is reasonable to assign high visual salience to the cusp, inflection, and curvature extrema points of its sub-shapes, as well as the intersection points between the sub-

shapes and the end points of each of open sub-shapes. In our implementation, a simple and efficient algorithm of Chetverikov and Szabo [CS99] is used to detect the cusp, inflection, and curvature extrema points of the sub-shapes. These feature points then segment each sub-shape into several salient features, each of which corresponds to a curve segment that is delimited by two successive feature points. An example of the feature points and salient features is demonstrated in the right sub-figure of Fig. 2(a).

Sub-frame generation. Intuitively, the structural relationships between salient features are determined by their spatial positions and orientations. For each sub-shape, we sequentially connect its feature points to form a polyline or a polygon, as illustrated by the thin line segments in Figure 2(d). We call the polyline or polygon as a sub-frame since it seems like a scaffolding of the corresponding sub-shape. Obviously, each edge of the sub-frame corresponds to a salient feature and thus is called a feature line for clarity. It is clear that each feature line describes the position, size, and orientation of the associated feature in general. Hence, for each sub-shape a sub-frame can be used to encode the structural relationships between its salient features.

Sub-frame stitching. During deformation, sub-shapes of the input multi-element 2D shape cannot move independently, but have to be "stitched" together to convey the impression of a connected body. Due to the one-to-one correspondence between the salient features and the feature lines, the "stitching" can be established by connecting sub-frames together. Moreover, to maintain the deformation flexibility of a multi-element 2D shape, individual sub-shapes should be allowed to move separately. Hence, sub-frames are expected to be connected together, but in a loose way.

We can model the problem as graph connection. The graph contains one node N_i per sub-frame SF_i, with an edge (i, j) between each pair of nodes N_i and N_j. The edge length of (i, j) is defined as the distance between the sub-frames SF_i and SF_j:

$$\mathbf{d}(SF_i, SF_j) = min||v_{i,l} - v_{j,m}||$$

where $v_{i,l}$ and $v_{j,m}$ are the vertices of SF_i and SF_j respectively. The problem is then to decide when a pair of nodes are to be connected in the graph so that the graph is connected with minimum edges. The Euclidean Minimum Spanning Tree (EMST) solves the problem and tends to connect sub-frames that are near each other.

For each edge in the EMST, we put a line segment to connect the vertices of the corresponding sub-frames that define the edge length. For clarify, we call this line segment as a sewing line, to distinguish from the feature lines of sub-frames. An example is shown in the left sub-figure of Fig. 2(c), where the output includes only one sewing line, which is labeled as 'SL 1' (the sewing lines 'SL 2' and 'SL 3' are imposed by the user later, as mentioned below). Obviously,

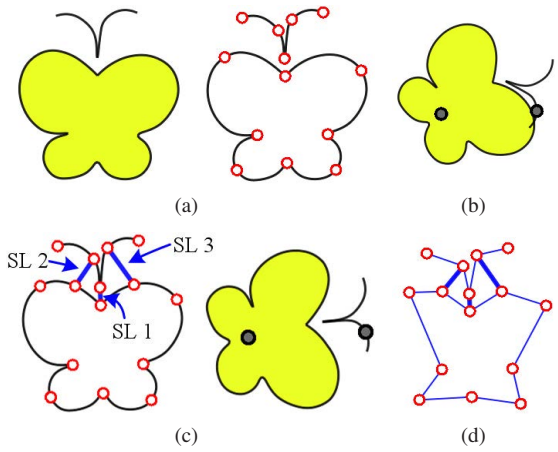


Figure 2: Feature frame construction. (a): input shape and its feature points (right: circular dots); (b): deformation result with only sewing line 'SL 1'; (c): the automatically generated sewing line 'SL 1' and the manually specified sewing lines 'SL 2' and 'SL 3' (left), as well as deformation result with the sewing line 'SL 1' and the additional sewing lines 'SL 2' and 'SL 3' (right); (d): The final feature frame.

through the sewing line 'SL 1', all parts of the butterfly (left and right antenna and body) are "stitched" together.

On the other hand, each sewing line is also used to indicate that the structural relationships between the parts, which are connected by the sewing line, should be preserved during deformation. However, the EMST over the sub-frames may not be sufficiently dense in edges to serve the purpose. Moreover, while the sewing lines derived from the EMST's edges tend to connect neighboring parts, they may join together the parts that are spatially close but semantically separated. We let the user manually modify the sewing lines for desired deformation effects.

The modification process is simple and intuitive, where the user only needs to add, delete or move the sewing lines between shape parts. If the local visual appearance or structure between two parts of the shape are desired to be preserved during deformation (e.g., the left antenna and the left upper body or the right antenna and the right upper body in Fig. 2(c)), a sewing line (e.g., 'SL 2' or 'SL 3' in Fig. 2(c)) is then put by the user between the nearly middle points of the two parts. The effect is demonstrated by comparing the deformation results without and with the additional sewing lines, as shown in Fig. 2(b) and the right sub-figure of Fig. 2(c), respectively. In our implementation, for each end vertex of the new sewing line, if its distance to an existing feature point is less than a threshold, the two points will be merged; otherwise it will be inserted as a new feature point on the involved sub-shape and the corresponding sub-frame is updated accordingly. We use a distance threshold of a quarter

of the average edge length of the input shape for the examples in the paper. Similarly, the user can untie the shape parts by deleting redundant sewing lines between them.

Discussion: In our experiments, the modification of sewing lines seems intuitive to the user, since only necessary connections between the parts, which typically correspond to the semantic features of the shape, are involved. Moreover, the automatically generated sewing lines from the EMST over the sub-frames will connect the neighboring parts. Hence, a small amount of user interaction is sufficient in general. As shown in Table 1, for all examples shown, the number of sewing lines is within 20, where only several of them (2-6) are specified by the user and the interaction time is less than 2 minutes. It bears noting that the sewing lines can be adjusted on the fly during the shape manipulation, which provides the user intuitive and flexible control of the deformation results.

Finally, the feature frame, which represents the structural relationships between parts of a multi-element 2D shape, is formed by the feature and sewing lines as well as their vertices, as illustrated in Fig. 2(d).

Notation. Throughout, we denote 2-vectors by boldface lower case letters, while we denote 2×2 matrices by boldface upper case letters. Furthermore, we denote a feature frame vertex using an index, e.g., i. Similarly, we also denote a feature or sewing line using an index, e.g., j. Specifically,

- $\bar{\mathbf{v}}_i$ and $\mathbf{v}_i$ denote the initial and deformed positions of each feature frame vertex i, $i \in [1 \dots q]$, respectively. An example of feature frame vertices is shown in Fig. 2(d), i.e., the circular dots.
- $\bar{\mathbf{v}}_j(t)$ and $\mathbf{v}_j(t)$, $t \in [0,1]$, denote the initial and deformed positions of each point of the feature or sewing line j, respectively. An example of feature and sewing lines is shown in Fig. 2(d), i.e., the thin and thick line segments, respectively.

3.2. Parametric representation

The feature frame can be considered a simplified version of a multi-element 2D shape. To facilitate the preservation of local features and structural relationships during subsequent manipulation and interpolation, we further take the feature frame as a proxy of the shape by recording the shape vertices in parametric coordinates.

Given a feature line j $(\bar{\mathbf{v}}_j^1, \bar{\mathbf{v}}_j^2)$, a local orthogonal coordinate system can be defined as: $(\bar{\mathbf{v}}_j^1, \bar{\mathbf{v}}_j^1\bar{\mathbf{v}}_j^2, \bar{\mathbf{v}}_j^1\bar{\mathbf{v}}_j^{2\perp})$, where $\bar{\mathbf{v}}_j^1\bar{\mathbf{v}}_j^2 = \bar{\mathbf{v}}_j^2 - \bar{\mathbf{v}}_j^1$ and $\perp$ is a 2D operator such that: $(x,y)^\perp = (-y,x)$. Then, for the salient feature associated with the feature line j, each of its vertices is parameterized using two local coordinates, with respect to the local orthogonal coordinate system. This approach is similar to the skeleton coordinate systems [LCF00, YN03]. However, our approach has two main advantages:

- While binding a shape and skeleton bones is a nontrivial work, the association between a shape and feature lines of the feature frame is explicit.
- Furthermore, a skeleton is more appropriate to the shapes that have an obvious articulated structure; however, the feature frame is general and applicable to arbitrary shapes, since it is mainly determined by the shape geometry, but not the shape's jointed structure.

With these configurations, we then develop a unified framework for the manipulation and interpolation of multi-element 2D shapes.

4. Shape manipulation

As the user manipulates the input multi-element 2D shape, a constrained optimization is solved to find optimal deformed shapes. The user-specified handles serve as positional constraints for the optimization problem, in which the deformed positions of the feature frame vertices comprise the unknown variables. Additionally, the objective function contains two other constraints, i.e., rigidity constraints for preserving local shape features, and coupling constraints for maintaining the structural relationships between parts of the shape. With the new configuration of the feature frame, the deformed shape is determined straightforwardly using the recorded parametric coordinates.

4.1. Deformation constraints

Rigidity constraints. It is a well-established goal for any interactive shape manipulation tool to preserve local features when a broad change in shape is made. Achieving that using the feature frame is very straightforward. Since the shape vertices are recorded in parametric coordinates, the feature lines will drive the local shape deformations. Thus, by confining the feature lines to undergo rigid transformations, the local features of the shape will be well preserved (Fig. 3). We implement such rigidity constraints by adding an energy term $\mathrm{E}_{\mathrm{rig}}$, which sums on each feature line the squared distances between the actual transformed position of each of its points and an optimal rigid transformation applied to the point:

$$\mathrm{E}_{\mathrm{rig}} = \sum_j \int_0^1 \left\| \mathbf{v}_j(t) - \mathbf{v}_j^c - \left(\bar{\mathbf{v}}_j(t) - \bar{\mathbf{v}}_j^c\right)\mathbf{R}_j \right\|^2 \mathrm{d}t \qquad (1)$$

where $\mathbf{R}_j$ is a normalized orthogonal matrix corresponding to the optimal rotation of the points of the feature line j, while $\bar{\mathbf{v}}_j^c$ and $\mathbf{v}_j^c$ are the initial and deformed rotation center, corresponding to the middle point of the feature line j.

Coupling constraints. As described in the section introduction, it is important to preserve the structural relationships between shape parts, so as to maintain the feature semantics of the underlying subject. To satisfy this requirement, we couple neighboring feature lines together and let

them deform as a rigid body, which will preserve the relative positions and orientations between them, resulting in the maintenance of the structural relationships of our interest (Fig. 3). In the coupling constraints, we let the points of the neighboring feature lines transform as a rigid unit:

$$\mathrm{E_{cou}} = \sum_i \sum_{j \in \mathcal{N}(i)} \int_0^1 \|\mathbf{v}_j(t) - \mathbf{v}_i^c - (\overline{\mathbf{v}}_j(t) - \overline{\mathbf{v}}_i^c)\hat{\mathbf{R}}_i\|^2 \, \mathrm{d}t \quad (2)$$

where $\mathcal{N}(i)$ is the indices set of the neighboring feature and sewing lines that are incident to the feature frame vertex i; $\hat{\mathbf{R}}_i$ is the optimal rotation of the set $\mathcal{N}(i)$, while $\overline{\mathbf{v}}_i^c$ and $\mathbf{v}_i^c$ are the initial and deformed rotation centers, corresponding to the feature frame vertex i.

Positional constraints. To satisfy the user-specified positional constraints, a positional energy term is defined as follows. Given a handle k, its source and target positions are $\overline{\mathbf{c}}_k$ and $\mathbf{c}_k$. We find the shape point closest to the handle and assume $(\overline{\mathbf{v}}_i, \overline{\mathbf{v}}_j)$ be the feature line associated with the shape feature where the closest shape point lies. With respect to the local orthogonal coordinate system $(\overline{\mathbf{v}}_i, \overline{\mathbf{v}}_i\overline{\mathbf{v}}_j, \overline{\mathbf{v}}_i\overline{\mathbf{v}}_j{}^{\perp})$, which is derived from the feature line $(\overline{\mathbf{v}}_i, \overline{\mathbf{v}}_j)$, the source position $\overline{\mathbf{c}}_k$ has a unique expression:

$$\overline{\mathbf{c}}_k = \overline{\mathbf{v}}_i + u\overline{\mathbf{v}}_i\overline{\mathbf{v}}_j + v\overline{\mathbf{v}}_i\overline{\mathbf{v}}_j{}^{\perp} \quad (3)$$

where (u, v) is the local coordinates of $\overline{\mathbf{c}}_k$ in the coordinate system. Then the penalty term, which preserves the position of the handle k relative to its nearest shape feature, is:

$$\mathrm{Pos}(\mathbf{c}_k) = \|\mathbf{v}_i + u\mathbf{v}_i\mathbf{v}_j + v\mathbf{v}_i\mathbf{v}_j{}^{\perp} - \mathbf{c}_k\|^2 \quad (4)$$

Finally, the energy term for all handles is: $\mathrm{E_{pos}} = \sum_k \mathrm{Pos}(\mathbf{c}_k)$.

4.2. Optimization

To obtain the deformed positions $\mathbf{v}_i$, $i \in [1 \ldots q]$, of the feature frame vertices, we solve the following optimization problem:

$$\arg\min_{\mathbf{v}_1, \ldots \mathbf{v}_q} \quad w_\mathrm{r}\mathrm{E_{rig}} + w_\mathrm{c}\mathrm{E_{cou}} + w_\mathrm{p}\mathrm{E_{pos}} \quad (5)$$

We use the weights $w_\mathrm{r} = 1$, $w_\mathrm{c} = 1$, and $w_\mathrm{p} = 100$ for the examples shown in the paper. It bears noting that while the individual rigidity and coupling constraints here demonstrate the different deformation effects intuitively (Fig. 3), they can be combined as one in the implementation, since the coupling constraints implicitly impose the local rigidity. Eq. (5) is a nonlinear optimization problem, which contains $2q$ unknown variables, i.e., $\{\mathbf{v}_i = (x_i, y_i)\}_{i=1}^q$. Similar to [SA07], this nonlinear optimization is solved using an alternating minimization approach.

In the optimization, the objective function of Eq. (5) is treated as a functional depending on both $\{\mathbf{v}_i\}$, and $\{\mathbf{R}_j\}$ of Eq. (1), as well as $\{\hat{\mathbf{R}}_i\}$ of Eq. (2). The optimization is then performed in an alternating manner: fixing $\{\mathbf{v}_i\}$ to compute the optimal $\{\mathbf{R}_j\}$ and $\{\hat{\mathbf{R}}_i\}$; using $\{\mathbf{R}_j\}$ and $\{\hat{\mathbf{R}}_i\}$ to find the best $\{\mathbf{v}_i\}$, and repeating until convergence. In our experiments, the optimization will converge after about 12 iterations.

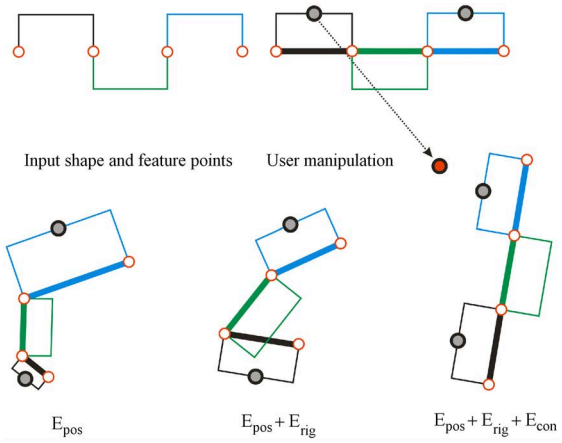


Figure 3: A simple deformation case shows the effect of the three constraint terms of the objective function. Top left: input shape and its feature points (circular dots). For clear demonstration the feature points are manually specified so that the shape is segmented into three intuitive parts, i.e., three rectangles. As illustrated in the top right, each feature line (i.e., a thick line segment) and its associated shape part (i.e., a rectangle) are drawn in the same color. Without the rigidity term, unnatural distortions of local features can occur (bottom left). The coupling term preserves the structural relationships between shape parts (bottom right).

In each iteration, each optimal $\mathbf{R}_j$ is computed independently, since $\mathbf{R}_j$ is only related to the energy $\mathrm{E_{rig}}$ of Eq. (1) and each term in the sum of Eq. (1) involves only one rotation $\mathbf{R}_j$. Thus, we seek an $\mathbf{R}_j$ that minimizes the corresponding term in Eq. (1). In 2D case, the solution to $\mathbf{R}_j$ has an analytic form [SMW06]. Similarly, the analytic solution for each optimal $\hat{\mathbf{R}}_i$ can also be obtained. In order to obtain $\{\mathbf{v}_i\}$ from the given $\{\mathbf{R}_j\}$ and $\{\hat{\mathbf{R}}_i\}$, we should compute the gradients of $\mathrm{E_{rig}}$, $\mathrm{E_{cou}}$ and $\mathrm{E_{pos}}$ with respect to each $\mathbf{v}_i$. While the gradient computation is simple for $\mathrm{E_{pos}}$, it seems complicated for $\mathrm{E_{rig}}$ and $\mathrm{E_{cou}}$, since they both involve the integral calculations. Thanks to the carefully designed energy terms $\mathrm{E_{rig}}$ and $\mathrm{E_{cou}}$, their partial derivatives with respect to $\mathbf{v}_i$ are both analytically integrable (see Appendix A). Finally, we obtain a sparse linear system for $\{\mathbf{v}_i\}$. The system matrix is constant at all iterations; thus, the full factorization can be reused [Dav04], resulting in a very efficient solver. As shown in Table 1, the optimization of Eq. (5) requires only about 5 ms for the presented examples.

5. Shape interpolation

One can make 2D animations by deforming the multi-element 2D shapes via direct manipulation frame by frame.

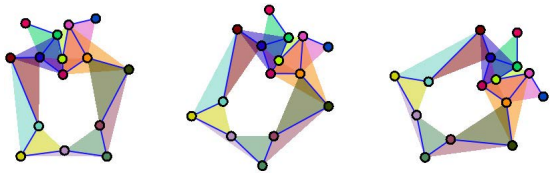


Figure 4: Interpolation of feature frames and triangle sets of the butterfly shape in Fig. 2(a). (left, right): a source and target feature frames and the corresponding triangle sets, where each triangle is rendered with alpha blending using the same color as the feature frame vertex from which it is derived. (middle): an intermediate triangle set and feature frame.

However, this is a tedious process and we reduce the manual effort by using the shape interpolation technique. To create a 2D animation, the user creates key deformations of the input shape via direct manipulation at only several keyframes, and then the system automatically generates smooth intermediate deformations by interpolating the shapes between each two consecutive keyframes. Our goal is to interpolate shapes in a shape-preserving way, so as to obtain visually pleasing animation.

Let S_1 and S_2 be two deformed multi-element shapes, and F_1, F_2 the associated feature frames. During interpolation, the key is to maintain the structural relationships encoded in the feature frames F_1 and F_2, so that the feature semantics of the input shapes are preserved. With this goal in mind, we propose the following algorithm to interpolate the shapes S_1 and S_2. The algorithm includes two steps.

Step 1: *Structural relationships transferring.* In this step, we construct two triangle sets T_1 and T_2 from the feature frames F_1 and F_2, respectively. The process of triangle set construction is on the basis of feature frame vertex. Given a feature frame, for each of its vertices, let r be the total number of the feature and sewing lines that are incident to the vertex: if r is equal to 2, one triangle is created; if r is greater than 2, r triangles are created. An example is shown in Fig. 4. It is evident that one-to-one correspondence exists between the triangle of sets T_1 and T_2, since F_1 and F_2 have identical connectivity. In this simple way, the structural relationships between parts of a shape that are encoded by the relative positions and orientations between the neighboring feature lines of the corresponding feature frame are explicitly transferred into a set of triangles.

Step 2: *Structure preserving interpolation.* This step performs an as-rigid-as-possible interpolation between the compatible triangle sets T_1 and T_2, as shown in Fig. 4 (The concept of as-rigid-as-possible transformations was itself first introduced by Alexa [ACOL00], thus we refer the readers to [ACOL00] for details). Due to the nature of as-rigid-as-possible shape interpolation in minimizing redundant distortions, the transferred structural relationships of S_1 and S_2 in the triangle sets T_1 and T_2 are preserved during interpolation.

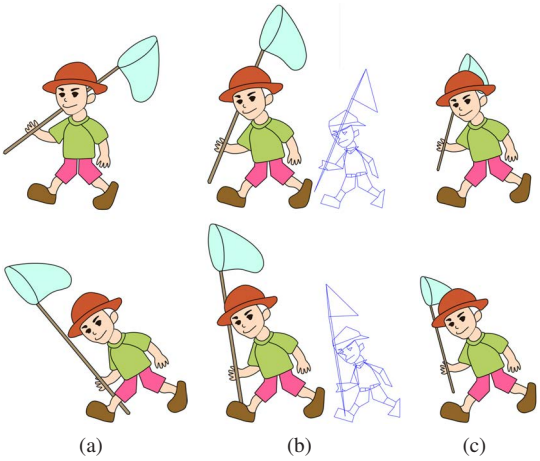


Figure 5: Interpolation of two deformed multi-element 2D shapes of the cartoon boy in Fig. 1(a). (a): input shapes; (b): our interpolation approach; (c): linear blending method.

Based on the vertex association between the triangle set and the feature frame, it is straightforward to obtain the intermediate feature frames from the intermediate triangle sets (Fig. 4). Furthermore, by applying the recorded parametric coordinates of shape vertices to the intermediate feature frames, the intermediate shapes are finally determined (Fig. 5(b)).

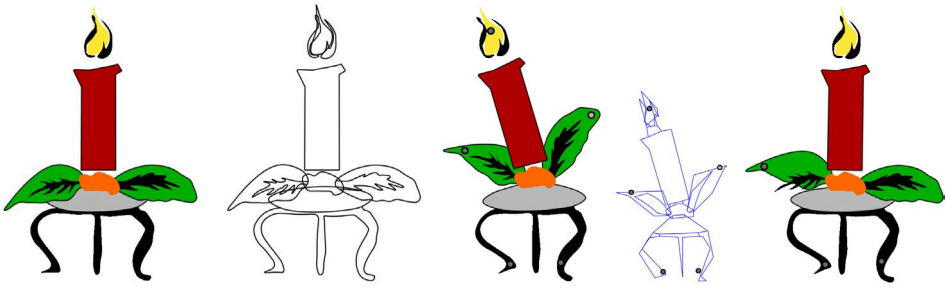
6. Results

We have implemented the proposed method of manipulation and interpolation for multi-element 2D shapes on a PC with 2.4GHz Intel Core 2 Duo CPU and 2GB RAM. Our implementation is single threaded and uses only one core. Live manipulation and interpolation examples are demonstrated in the accompanying video, and key results are highlighted in this section.

Fig.	S. VER	F. LINES	F. VER	PRE	SOL
1(b)	1918	138 (13, 5)	122	1.60	3.59
6(b)	1030	96 (13, 6)	80	1.51	2.67
8(b)	1396	103 (12, 5)	96	1.56	2.81
9(b)	458	55 (4, 2)	42	1.10	1.87

Table 1: Statistics and performance data for shape manipulation. Timing is measured in milliseconds.

Shape manipulation. Table 1 shows time statistics of manipulation for the multi-element 2D shapes in this paper. In the table, "S. VER" denotes the number of shape vertices, "F. LINES" the number of the feature and sewing lines of the feature frame (brackets: number of the total sewing lines and number of the user-specified sewing lines, respectively), "F. VER" the number of the feature frame vertices, "PRE"



(a) (b) (c)

Figure 6: Manipulation of a cartoon candle. (a): input shape; (b): our manipulation approach; (c): the method of [YF09a], where the semantic feature on the left "arm" of the candle is distorted.

runtime of pre-factorization, and "SOL" runtime of all iterations. Moreover, the time for determining deformed shapes from deformed feature frames is less than 1 ms for all examples, and thus is neglected. Obviously, the shape deformation can be performed completely interactively.

Fig. 1 and 6 demonstrate that our manipulation approach for multi-element 2D shapes is more appropriate to complex inputs than previous rigidity-modeling methods [IMH05, WXW*06] and the stick figure manipulation approach [YF09a]. Our method preserves both local shape features and the structural relationships between shape parts, thus obtaining visually plausible deformation results. In these figures, the pleasing deformation results are obtained by manipulating only several handles, which can be challenging with the previous shape manipulation techniques.

Fig. 8 demonstrates that our manipulation approach retains more deformation flexibility of the multi-element 2D shapes than the previous rigidity-modeling methods [IMH05, WXW*06, SDC09]. Fig. 8(c) shows the deformation result of a girl shape using the method of Igarashi et al. [IMH05]. Since the girl shape is entirely embedded in a triangulation, all of its parts is tightly coupled. Consequently, lifting the girl's right leg will pull her body accordingly, leading to a unnatural pose. Furthermore, the girl's hair and left arm are inseparable during manipulation, because the simple polygonal outline cannot fully capture the semantic boundaries of the overlapped parts. However, in our approach, different parts of the input shape are loosely connected and thus are allowed to move separately, finally leading to natural deformation results, as shown in Fig. 8(b). Fig. 9 further demonstrates the deformation flexibility of our manipulation approach.

It bears noting that the concept of feature frame is general. It is possible to construct a feature frame using arbitrary points of the shape instead of its feature points. This is demonstrated with a spring-like curve (Fig. 7), where we have replaced the feature points of the shape with its sample points at uniform intervals of arc-length for feature frame construction. In this case, our deformation framework is similar to the as-rigid-as-possible curve editing [IMH05]. By using different sample intervals, e.g., half, two times, and eight times the average edge length of the curve, respectively, feature frames with three different high-to-low resolutions are constructed, and then interesting soft-to-stiff stiffness behaviors are effectively mimicked (see the lower row of Fig. 7).

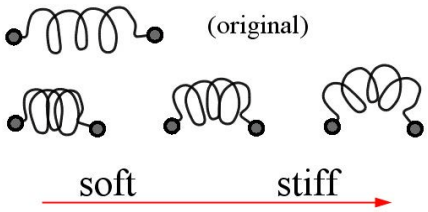


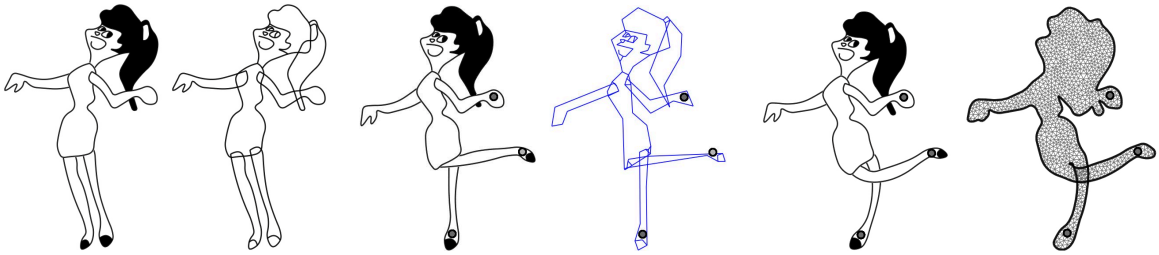
Figure 7: Squashing of a spring-like curve.

Shape interpolation. Since both the vertex number and the number of the feature and sewing lines of feature frame are small, as shown in Table 1, the triangle set for feature frame will have a small number of vertices and triangles. Finally, our interpolation approach is very efficient: it is able to generate over 100 intermediate shapes per second for the examples shown.

Fig. 5 demonstrates that our shape interpolation approach generates natural results. As shown in Fig. 5(b), the local shapes of the input objects and their feature semantics are effectively preserved. Fig. 5 also demonstrates that our interpolation approach yields intuitive rotations of the source shape towards the target shape, thus effectively avoid the shrinkage effects, which are common in naïve linear blending methods (Fig. 5(c)). More examples are shown in the accompany video.

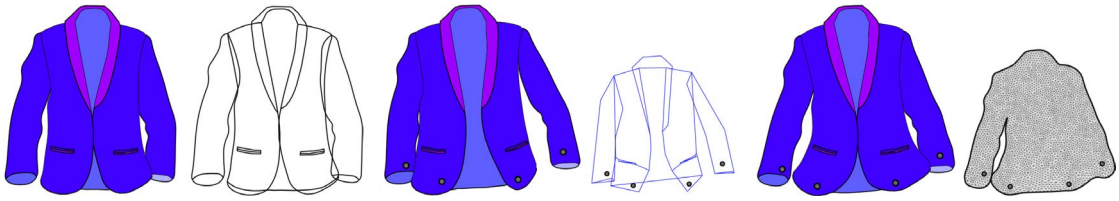
7. Conclusion

We have presented a method that generates natural and intuitive deformations via direct manipulation and smooth in-



(a) (b) (c)

Figure 8: Manipulation of a figure. (a): input shape; (b): our manipulation approach; (c): the method of Igarashi et al. [IMH05].



(a) (b) (c)

Figure 9: Manipulation of a suit. (a): input shape; (b): our manipulation approach; (c): the manipulation method of Igarashi et al. [IMH05], where the two front pieces of the suit are tied together and the body of the suit and its left sleeve are inseparable, leading to unnatural deformation result.

terpolation for multi-element 2D shapes. The multi-element 2D shape representation can correctly describe the semantic boundaries of arbitrary shapes even when different parts of the shape are overlapped. We introduce a simple structure called the feature frame to represent the structural relationships between parts of the shape. Based on the feature frame, a unified framework is developed for the manipulation and interpolation of multi-element 2D shapes. Our method is computationally efficient, allowing real-time manipulation and interpolation.

There are still improvements needed. While it is a simple task to draw sub-shapes for most objects, as shown in our examples, there are some objects which have so much detailed decoration that it will be tedious for the user to draw the large number of sub-shapes for them. One possible solution is the usage of interior texture to describe the subtle details. Furthermore, we intend to investigate the possibility of automatically extracting multi-element 2D shapes from images [ZCZ*09].

In the extreme cases where a handle is dragged *too far* from the shape, the structural relationships between neighboring parts might be destroyed to some extent, so as to satisfy the user-specified handle constraints. In these extreme cases, an unnatural jaggedness between neighboring local shape features might occur. We tried to smooth out the jaggedness by blending adjacent local transformations, but the result was not very satisfactory. Hsu and Lee [HL94] propose a skeleton-driven deformation model based on an ideal material with non-localized deformation, which can effectively eliminate the jaggedness-like effects, such as wrinkles or fold-backs. We would like to experiment with this by finding a way to integrate their deformation model into our deformation framework for shape manipulation in Section 4.

Furthermore, we hope to apply the method to multi-component 3D models. The main problem seems to be the construction of the "feature frame"-like structure for the models. The intelligent wires [GSMCO09] or component-wise controllers [ZFCO*11] might be a good starting point for this purpose.

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments. This research was partially funded by the NSFC (No. 61003189, 60933007), the Foundation of Zhejiang Educational Committee (No. Y201017028), the Program for New Century Excellent Talents in University (NCET-10-0728), and the 973 program of China (No. 2009CB320801).

References

[ACOL00] ALEXA M., COHEN-OR D., LEVIN D.: As-rigid-as-possible shape interpolation. In *SIGGRAPH '00* (2000), pp. 157–164.

[BIBA09] BAXTER III W. V., BARLA P., ANJYO K.-I.: Com-

patible embedding for 2d shape animation. *TVCG 15*, 5 (2009), 867–879.

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *TVCG 14*, 1 (2008), 213–230.

[CS99] CHETVERIKOV D., SZABO Z.: A simple and efficient algorithm for detection of high curvature points in planar curves. In *Proceedings of 23rd Workshop of the Austrian Pattern Recognition Groups* (1999), pp. 175–184.

[Dav04] DAVIS T. A.: Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software 30*, 2 (2004), 196–199.

[GB08] GAIN J., BECHMANN D.: A survey of spatial deformation from a user-centered perspective. *TOG 27*, 4 (2008), 1–21.

[GSMCO09] GAL R., SORKINE O., MITRA N. J., COHEN-OR D.: iwires: an analyze-and-edit approach to shape manipulation. *TOG 28*, 3 (2009), 1–10.

[HBCC07] HAHMANN S., BONNEAU G.-P., CARAMIAUX B., CORNILLAC M.: Multiresolution morphing for planar curves. *Computing 79*, 2 (2007), 197–209.

[HDK07] HORNUNG A., DEKKERS E., KOBBELT L.: Character animation from 2d pictures and 3d motion data. *TOG 26*, 1 (2007), 1.

[HKT10] HO E. S. L., KOMURA T., TAI C.-L.: Spatial relationship preserving character motion adaptation. *TOG 29*, 4 (2010), 1–8.

[HL94] HSU S. C., LEE I. H. H.: Drawing and animation using skeletal strokes. In *SIGGRAPH '94* (1994), pp. 109–118.

[IMH05] IGARASHI T., MOSCOVICH T., HUGHES J. F.: As-rigid-as-possible shape manipulation. *TOG 24*, 3 (2005), 1134–1141.

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *SIGGRAPH '00* (2000), pp. 165–172.

[SA07] SORKINE O., ALEXA M.: As-rigid-as-possible surface modeling. In *SGP '07* (2007), pp. 109–116.

[SCOL*04] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *SGP '04* (2004), pp. 175–184.

[SDC09] SÝKORA D., DINGLIANA J., COLLINS S.: As-rigid-as-possible image registration for hand-drawn cartoon animations. NPAR '09, pp. 25–33.

[SGWM93] SEDERBERG T. W., GAO P., WANG G., MU H.: 2-D shape blending: an intrinsic solution to the vertex path problem. In *SIGGRAPH '93* (1993), pp. 15–18.

[SMW06] SCHAEFER S., MCPHAIL T., WARREN J.: Image deformation using moving least squares. *TOG 25*, 3 (2006), 533–540.

[SSJ*10] SÝKORA D., SEDLACEK D., JINCHAO S., DINGLIANA J., COLLINS S.: Adding depth to cartoons using sparse depth (in)equalities. *Computer Graphics Forum 29*, 2 (2010), 615–623.

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. *TOG 26* (2007).

[WNS*10] WHITED B., NORIS G., SIMMONS M., SUMNER R. W., GROSS M., ROSSIGNAC J.: Betweenit: An interactive tool for tight inbetweening. *Computer Graphics Forum 29* (2010), 605–614.

[WXW*06] WENG Y., XU W., WU Y., ZHOU K., GUO B.: 2d shape deformation using nonlinear least squares optimization. *The Visual Computer 22*, 9 (2006), 653–660.

[XWY*09] XU W., WANG J., YIN K., ZHOU K., VAN DE PANNE M., CHEN F., GUO B.: Joint-aware manipulation of deformable models. *TOG 28* (2009), 35:1–35:9.

[YF09a] YANG W., FENG J.: 2d shape manipulation via topology-aware rigid grid. *Computer Animation and Virtual Worlds 20*, 2–3 (2009), 175–184.

[YF09b] YANG W., FENG J.: 2d shape morphing via automatic feature matching and hierarchical interpolation. *Computers and Graphics 33* (2009), 414–423.

[YN03] YAMANE K., NAKAMURA Y.: Natural motion animation through constraining and deconstraining at will. *TVCG 9*, 3 (2003), 352 – 360.

[YZX*04] YU Y., ZHOU K., XU D., SHI X., BAO H., GUO B., SHUM H.-Y.: Mesh editing with poisson-based gradient field manipulation. *TOG 23*, 3 (2004), 644–651.

[ZCZ*09] ZHANG S.-H., CHEN T., ZHANG Y.-F., HU S.-M., MARTIN R. R.: Vectorizing cartoon animations. *TVCG 15* (2009), 618–629.

[ZFCO*11] ZHENG Y., FU H., COHEN-OR D., AU O. K.-C., TAI C.-L.: Component-wise controllers for structure-preserving shape manipulation. In *Computer Graphics Forum* (2011), vol. 30, pp. 563–572.

[ZXTD10] ZHOU K., XU W., TONG Y., DESBRUN M.: Deformation transfer to multi-component objects. *Computer Graphics Forum 29* (2010), 319–325.

Appendix A

Here we provide the analytical expressions of the partial derivatives of energy terms E_{rig} of Eq. (1) and E_{cou} of Eq. (2) with respect to $\mathbf{v}_k$ (the deformed position of any feature frame vertex k). We take the computation of the derivative of $\partial E_{cou}/\partial \mathbf{v}_k$ as example.

Let $(\bar{\mathbf{v}}_j^1, \bar{\mathbf{v}}_j^2)$ and $(\mathbf{v}_j^1, \mathbf{v}_j^2)$ be the initial and deformed positions of the two end vertices of each feature or sewing line j in the set $\mathcal{N}(i)$ of Eq. (2), we have:

$$\begin{cases} \bar{\mathbf{v}}_j(t) = (1-t)\bar{\mathbf{v}}_j^1 + t\bar{\mathbf{v}}_j^2, & \bar{\mathbf{v}}_i^c = \alpha\bar{\mathbf{v}}_j^1 + \beta\bar{\mathbf{v}}_j^2 \\ \mathbf{v}_j(t) = (1-t)\mathbf{v}_j^1 + t\mathbf{v}_j^2, & \mathbf{v}_i^c = \alpha\mathbf{v}_j^1 + \beta\mathbf{v}_j^2 \end{cases} \tag{6}$$

Since every feature or sewing line j in the set $\mathcal{N}(i)$ is incident to the feature frame vertex i, which corresponds to the rotation center in Eq. (2), it is evident that $(\alpha = 0, \beta = 1)$ or $(\alpha = 1, \beta = 0)$. By substituting Eq. (6) into Eq. (2), we have:

$$\begin{aligned} \frac{\partial E_{cou}}{2\partial \mathbf{v}_k} &= \sum_i \sum_{j\in\mathcal{N}(i)} \int_0^1 \left(\mathbf{v}_j(t) - \mathbf{v}_i^c - \left[\bar{\mathbf{v}}_j(t) - \bar{\mathbf{v}}_i^c\right]\hat{\mathbf{R}}_i \right) \frac{\partial\left[\mathbf{v}_j(t) - \mathbf{v}_i^c\right]}{\partial \mathbf{v}_k}\, dt \\ &= \sum_i \sum_{j\in\mathcal{N}(i)} \left[\left(\frac{1}{3} - \alpha + \alpha^2\right)\mathbf{v}_j^1 + \left(\frac{1}{6} - \frac{\alpha+\beta}{2} + \alpha\beta\right)\mathbf{v}_j^2 \right] \frac{\partial \mathbf{v}_j^1}{\partial \mathbf{v}_k} + \\ &\quad \sum_i \sum_{j\in\mathcal{N}(i)} \left[\left(\frac{1}{6} - \frac{\alpha+\beta}{2} + \alpha\beta\right)\mathbf{v}_j^1 + \left(\frac{1}{3} - \beta + \beta^2\right)\mathbf{v}_j^2 \right] \frac{\partial \mathbf{v}_j^2}{\partial \mathbf{v}_k} - \\ &\quad \sum_i \sum_{j\in\mathcal{N}(i)} \left[\left(\frac{1}{3} - \alpha + \alpha^2\right)\bar{\mathbf{v}}_j^1\hat{\mathbf{R}}_i + \left(\frac{1}{6} - \frac{\alpha+\beta}{2} + \alpha\beta\right)\bar{\mathbf{v}}_j^2\hat{\mathbf{R}}_i \right] \frac{\partial \mathbf{v}_j^1}{\partial \mathbf{v}_k} - \\ &\quad \sum_i \sum_{j\in\mathcal{N}(i)} \left[\left(\frac{1}{6} - \frac{\alpha+\beta}{2} + \alpha\beta\right)\bar{\mathbf{v}}_j^1\hat{\mathbf{R}}_i + \left(\frac{1}{3} - \beta + \beta^2\right)\bar{\mathbf{v}}_j^2\hat{\mathbf{R}}_i \right] \frac{\partial \mathbf{v}_j^2}{\partial \mathbf{v}_k} \end{aligned}$$