2-D Shape Blending based on Visual Feature Decomposition

Wenwu Yang†  Jieqing Feng†  Xiaogang Jin†  Qunsheng Peng†  A.Robin Forrest‡

†State Key Lab of CAD&CG, Zhejiang University
Hangzhou, 310027, P.R.China
Tel. (+86)571 87951045  Fax. (+86)571 87951780
{wwyang | jqfeng | jin | peng} @cad.zju.edu.cn

‡School of Computing Sciences, University of East Anglia
Tel. (+44) 1603 592605  Fax. (+44) 1603 593345
forrest@cmp.uea.ac.uk

Abstract

2-D polygon shape blending has been widely used in 2-D character animation, pattern matching, and geometric modeling. Previous algorithms mostly tend to use the polygon’s geometric elements such as edge, angle, area and skeleton, to associate the regions on the two shapes which look alike. However they ignore the visual features of two polygonal shapes. This paper presents a new 2-D polygonal shape blending method based on the correspondence of visual features. By decomposing the two shapes into sub-shape pairs which have similar visual features compatibly, the sub-shape in the source polygon can be smoothly transformed into the corresponding sub-shape in the destination polygon. Since the feature-decomposition vertices are introduced and our blending algorithm is so simple, the user can edit the visual correspondence of features flexibly and intuitively in real time. Experimental results show that the source shape can be transformed into the destination shape as prescribed by the user whilst keeping feature correspondence.

Keywords: 2-D Shape blending, visual feature decomposition, local coordinate system

1. Introduction

The visually smooth transformation from a 2-D source polygon to a 2-D destination polygon is called 2-D shape blending. It has been widely used in 2-D character animation, pattern matching, and geometric modeling. There are two problems which need to be solved: vertex correspondence and vertex interpolation paths. The purpose of vertex correspondence is to establish the correspondences between the two polygons. The purpose of vertex paths is to determine how the source polygon is transformed into the destination polygon. The general criteria to assess 2-D shape blending algorithms include: (1) the transformation should be visually smooth; (2) unnatural effects such as self-intersection, shrinkage, and distortion of the intermediate polygons should be avoided; (3) the intermediate polygons should have visual features of both the source and the destination polygons.

Sederberg et al. introduced a physically based method to solve the vertex correspondence problem [1]. They assumed that each polygon is composed of pieces of wires. The vertex correspondence established by their approach is equivalent to the minimum work required to bend and stretch the source
wire to the destination wire. Zhang establishes the vertex correspondence based on the similarity between the pairs of triangles [2]. The two methods only consider the geometric elements of the polygons, e.g. edge, angle, area etc, but ignore the visual features. Thus it is difficult to guarantee the correspondence of visual features between the source polygon and the destination polygon. Carmel et al. proposed a visual feature correspondence approach between the two polygons [3]. In their method, vertex correspondence is established through warping one polygon to another polygon so that the corresponding feature vertices are coincident. The feature vertices are specified by the users. However, the vertex correspondence is not intuitive because it is controlled by the warping function. Furthermore, the computational cost is relatively high.

The simple vertex interpolation technique is linear interpolation of the two corresponding vertices globally. However, the approach suffers from shrinkage problems when the 2D shape blending is dominated by rigid body motion. Sederberg et al. proposed the intrinsic 2-D shape blending method. It interpolates the edge length and orientation of the polygons [4]. The method can achieve pleasant result for simple polygons. When the polygons are complex, the intermediate polygons generated tend to be distorted. In the methods proposed by Carmel et al. and Zhang [2,3], the vertex path transformation are decomposed into two parts: rigid body transformation and flexible transformation. The rigid body transformation is described by global rotation and translation. Carmel et al. adopt a warping function to achieve the flexible transformation, whilst Zhang transforms the polygons into local affine frames and uses linear interpolation in the local coordinate system to achieve the flexible transformation. However, as we know, it is difficult to represent the flexible transformation by single affine transformation.

These methods only take into account boundary information of the polygons. Shapira and Rappoport propose an algorithm based on the “star-skeletons” representation of the polygons [5]. The algorithm is suitable for star-shaped polygons. It can avoid self-intersection in the intermediate polygons. But how to generate the topological isomorphic skeletons for two significantly different polygons is still a question. Alexa et al. propose a good method to interpolate between two triangulated polygons [6]. An optimal affine transformation is defined for each pair of corresponding triangles. The global transformation is deduced such that it minimizes the local distortion. The method can blend both boundaries and interiors of polygons smoothly. But the SVD (Single Value Decomposition) for evaluating the optimal affine transformation may produce a large error for irregular triangulation, and the computational cost is high. The method proposed by Gotsman and Surazhsky interpolates the triangulated polygons too [7]. Their method can guarantee the intermediate polygons are free from self-intersection. Unfortunately, the input polygons need be enclosed by two convex polygons before interpolation. If the source polygon and the destination polygon differ significantly, area distortion may arise in the intermediate polygons.

This paper proposes a new 2-D shape blending method based on visual feature decomposition. In the proposed method, the two polygons are decomposed interactively into sub-polygon pairs of isomorphic topology according to their visual features. Then a local coordinate system is constructed for each sub-polygon, and each sub-polygon is transformed into its own local coordinate system. The shape blending is completed by interpolating the frame polygons (which will be described in the following section) and the local coordinates of vertices. Our method has the following features: (1) It is simple and easy to implement; (2) It provides the user a way to control and edit the blending effect intuitively and flexibly in real time; (3) It produces smooth intermediate polygons with feature preservation. The rest of the paper is organized as follows. The section 2 gives the definitions of feature decomposition of a polygon and isomorphic feature decomposition of two polygons. The section 3 describes the algorithms for vertex correspondence and vertex interpolation path in detail. The section 4 shows the implementation results and discusses the proposed method. The section 5 concludes our algorithm.
2. Compatible Feature Decomposition

The goal of feature decomposition is to decompose a polygon into several sub-polygons, where each sub-polygon represents a visual feature which is specified by the user. Figure 1 is an example to illustrate the concept.

Figure 1: The dashed line polygon is called the frame polygon

In Figure 1, the cock-shaped polygon is decomposed into ten sub-polygons by selecting 10 feature-decomposition vertices labeled 1, 2, ..., 10 in clockwise. It should be noted that the feature-decomposition vertices are polygon vertices which are selected by user to decompose the polygon into several sub-polygons in clockwise or anti-clockwise. Each sub-polygon represents a visual feature on the cock: mouth, face, cockscomb, back, tail, trunk, back foot, belly, front foot, and breast respectively. Here the feature decomposition vertices are not in general equivalent to the geometric feature vertices of the polygon. An example is shown in Figure 2. Vertices 1 and 2 are feature decomposition vertices. In general they are the geometric feature vertices, such as cusp, inflection etc. The vertex f in Figure 2 is a geometric feature vertex, rather than feature decomposition vertex. From geometric considerations, it is not difficult to extract all geometric feature vertices from a 2-D polygon since they can always be described by some formulae. However, it is difficult to extract the feature-decomposition vertices amongst all of the geometric feature vertices without semantic assistant. Thus automatic polygon feature decomposition is not a trivial task. In this method the work is performed interactively.

Figure 2: The vertex f on the cock mouth is a feature vertex, but not a feature decomposition vertex. The vertices 1 and 2 are the feature decomposition vertices

The closed polygon formed by connecting the feature-decomposition vertices sequentially is called the frame polygon, shown as a dashed line polygon in Figure 1. The frame polygon is a low-resolution version of the original polygon. Each edge of the frame polygon is associated with a sub-polygon. In the rest of the paper, we show that the frame polygon is also used as the local coordinate frame during shape blending.

Compatible feature decomposition is to decompose source and destination polygons into the same number of corresponding sub-polygons according to their similar visual features. The sub-polygon correspondences on source and destination polygons are established according to the labels of the feature-decomposition vertices. An example is given in Figure 3.

Figure 3: The source and destination polygons are decomposed into ten pairs of corresponding sub-polygons by choosing ten corresponding feature-decomposition vertices interactively
The purpose of the compatible feature decomposition is to set up the visual features correspondences between the source and destination polygons. The features on the source polygon will transform to that of the destination polygon smoothly during the blending process.

3. Vertex Correspondences and Vertex Interpolation Paths

After compatible feature decomposition, visual shape correspondences are obtained between the source and destination polygons. In this section, we discuss in detail how to construct the vertex correspondences between corresponding sub-polygon pairs. Then the vertex interpolation path is described, containing two steps: frame polygons interpolation and sub-polygon interpolation.

3.1 Vertex Correspondence between two sub-polygons

In general, the numbers of vertices in the corresponding sub-polygon pairs are different. To perform the vertex interpolation, they must have same number of corresponding vertices. This is achieved through a reparameterization step which merges the two set of vertices in the corresponding sub-polygon pair by normalized chord-length parameterization.

Given two corresponding sub-polygons $\tilde{P}^s_f$ and $\tilde{P}^d_f$:

\[
\begin{align*}
\tilde{P}^s_f &= \{p^s_{f_1}, p^s_{f_2}, \ldots, p^s_{f_m}\} \\
\tilde{P}^d_f &= \{p^d_{f_1}, p^d_{f_2}, \ldots, p^d_{f_n}\}
\end{align*}
\]

where $p^s_{f_1}, p^s_{f_2}, \ldots, p^s_{f_m}$ are the vertices in the source sub-polygon, the first and last vertices, $p^d_{f_1}$ and $p^d_{f_m}$, are the adjacent feature decomposition vertices in the source polygon. $p^d_{f_1}, p^d_{f_2}, \ldots, p^d_{f_n}$ are the vertices in the destination sub-polygon. $p^d_{f_1}$ and $p^d_{f_n}$ are two adjacent feature decomposition vertices in the destination polygon. For each vertex in the source sub-polygon $\tilde{P}^s_f$, it is reparameterized by the normalized chord-length parameterization whose parametric interval is $[0, 1]$. The parameter for each vertex in the $\tilde{P}^s_f$ can be evaluated through the following formula:

\[
t^s_{f_1} = 0; t^s_{f_i} = \frac{\sum_{j=2}^{i} l^j}{\sum_{j=2}^{m} l^j} (i = 2, 3, \ldots, m)
\]

where $l^j$ is the length of $p^s_{f_{j-1}}p^s_{f_j}$. Thus the parametric representation of the source sub-polygon is $\{0, t^s_{f_2}, \ldots, t^s_{f_{m-1}}, 1\}$.

Similarly, the vertices in the destination sub-polygon can also be reparameterized on the interval $[0, 1]$. Their parametric representation is $\{0, t^d_{f_2}, \ldots, t^d_{f_{n-1}}, 1\}$. After merging their chord-length parametric representation, we can obtain the combined parametric representation for the source and destination sub-polygons, i.e., $\{0, t^s_{f_2}, \ldots, t^s_{f_{m-1}}, 1\}$ and $\{0, t^d_{f_2}, \ldots, t^d_{f_{n-1}}, 1\}$. In fact, they are the same except for their notations. Then the two parametric representations are transformed back to the source and destination sub-polygons, $\tilde{P}^s_f$ and $\tilde{P}^d_f$. Thus the vertex correspondences are established for each sub-polygon pair. Let $P^s_f$ and $P^d_f$ denote the corresponding sub-polygons after vertex merger, then

\[
\begin{align*}
P^s_f &= \{p^s_{f_1}, p^s_{f_2}, \ldots, p^s_{f_m}\} \\
P^d_f &= \{p^d_{f_1}, p^d_{f_2}, \ldots, p^d_{f_n}\}
\end{align*}
\]

3.2 Transforming the vertices in the sub-polygons into local coordinate system

For each sub-polygon, the first and the last vertices are the feature decomposition vertices. They will be used to create a local coordinate frame. Then the vertices in the sub-polygon will be transformed into the local coordinate system.

Given a source sub-polygon $P^s_f = \{p^s_{f_1}, p^s_{f_2}, \ldots, p^s_{f_M}\}$, the first vertex $p^s_{f_1}$ (also a feature decomposition vertex) is chosen as the origin of local coordinate system, the first axis is chosen as the normalized vector $u^s$ from feature-decomposition vertex $p^s_{f_1}$ to $p^s_{f_M}$, the second axis $v^s$ is obtained by rotating the $u^s$ 90° counter clockwise. Thus the origin $p^s_{f_1}$ and two orthogonal vectors $u^s$, $v^s$ form the local coordinate system $(p^s_{f_1}, u^s, v^s)$.
vertices $p_i^s$ in the sub-polygon $P^s_f$ can be represented by the local coordinates $(u_i^s, v_i^s)$ in the local coordinate system. They can be evaluated by the following equation:

$$p_i^s = p_i^s + u_i^s u^s + v_i^s v^s \quad (i = 1, 2, 3, \ldots, M)$$

Obviously, $(u_1^s, v_1^s) = (0,0)$. The procedure can be applied to the destination sub-polygon $P^d_f$.

### 3.3 Vertex Interpolation Path

In our method, the vertex interpolation path is decomposed into two parts: (1) rotation and translation part; (2) stretch part. They are completed by two steps. The first step is to transform the source frame polygon into the destination frame polygon. According to the description in the subsection 3.2, frame polygon interpolation is equivalent to axis and origin interpolation of the local coordinate systems for the corresponding sub-polygon pairs. The second step is linear interpolation between the source sub-polygon and the corresponding destination sub-polygon in the local coordinate system.

As we know from the section 2, the frame polygon describes the rough shape of the polygon. Each edge of the frame polygon determines a local coordinate system for the corresponding sub-polygon. In our method, the intrinsic interpolation is adopted for frame polygon blending [4]. Since the frame polygon is simple and contains few vertices in general, intrinsic interpolation can produce smooth blending between them. An example is shown in Figure 4.

![Image](image_url)

**Figure 4: Interpolating results of the frame polygons of the two source polygons in Figure 3**

The second step is to interpolate the corresponding sub-polygon pairs in the local coordinate system. The interpolated local coordinate frame was obtained from the first interpolation step, i.e., interpolation of the frame polygons.

Given two corresponding sub-polygons $P^s_f$ and $P^d_f$, they have the local coordinate systems $(p_i^s, u^s, v^s)$ and $(p_i^d, u^d, v^d)$ respectively. The corresponding vertices $p_i^s$ and $p_i^d$ in $P^s_f$ and $P^d_f$ have the local coordinates $(u_i^s, v_i^s)$ and $(u_i^d, v_i^d)$ respectively. At the intermediate time $t$, an intermediate local coordinate system $(p_i^t, u^t, v^t)$ can be determined through interpolation between the source and destination frame polygons. Let $p_i^t$ be the vertex in the intermediate sub-polygon at time $t$, it can be evaluated by using the following equation:

$$p_i^t = p_i^s + tu_i^s u^s + tv_i^s v^s$$

### 4. Experimental results and discussion

We have implemented the proposed method and compared the results with those generated by the intrinsic shape-blending method [4] and Zhang’s method [2]. Figure 5 shows the result of blending between the two cock-shaped polygons using the proposed method. The feature-decomposition vertices which are selected interactively are shown in Figure 3.

Figure 6 shows a comparison of the results between our algorithm and the intrinsic shape-blending algorithm [4]. The vertex correspondence is established by the proposed algorithm. Figure 6(a) is the result generated by the intrinsic shape-blending algorithm, 6(b) and 6(c) are the results generated by our method using different feature-decomposition vertices. Figure 6(a) and 6(b) have the same corresponding vertex pairs. We can see that the intermediate polygons generated by intrinsic shape-blending algorithm have distortion, whilst results generated by the proposed algorithm are smooth with feature preservation. Another smooth blending result in Figure 6(c) is obtained by editing the feature-decomposition vertices in Figure 6(b).

The comparison between the three results generated by the intrinsic shape-blending algorithm, Zhang’s algorithm and our proposed
algorithm is shown in Figure 7. The vertex correspondences are established by our algorithm. The result in Figure 7(a) is generated by using the intrinsic shape-blending algorithm, the results in the (b) and (c) are generated by the Zhang’s algorithm and our algorithm respectively. We can see that the intermediate polygons are distorted, self-intersected and unnatural in Figures 7(a) and 7(b), whilst the intermediate polygons in Figures 7(c) generated by our algorithm are smooth with feature preservation.

Other blending examples generated by the proposed algorithm are shown in Figures 8, and 9, 10.

Figure 5: Blending of two cock-shape polygons, the compatible feature decomposition is shown in the Figure 3

Figure 6: Blending of a circle to a star-shaped object
(a) intrinsic shape-blending algorithm
(b) blending based on the feature decomposition algorithm
(c) blending based on the feature decomposition algorithm

Figure 7: Blending of trees, ten pairs of feature-decomposition vertices are used
(a) Intrinsic shape-blending algorithm
(b) Zhang’s algorithm
(c) Blending based on the feature decomposition algorithm

Figure 8: Blending of the runners, seven pairs of feature-decomposition vertices are used

Figure 9: Blending of dinosaurs, six pairs of feature-decomposition vertices are used
Experimental results show that the proposed method can avoid the distortion and generate smooth blending results. Compared with Zhang’s algorithm which uses a single rotation and translation to represent the variation of orientations and positions between two polygons, our algorithm defines a local coordinate system for each sub-polygon pair, and a rotation and translation are adopted to represent the variation of orientation and position between two corresponding sub-polygons. The blending results generated by our method are smoother and more natural than the results produced by Zhang’s method.

When setting up the vertex correspondences between sub-polygons, we merge two set of vertices in the two sub-polygons based on chord-length parameterization. The solution is simple and efficient because the source and destination polygons are decomposed based on visual similarity. If the two corresponding sub-polygons are greatly different, we suggest decomposing the sub-polygon pairs in more detail or taking into account additional geometric factors such as vertex, edge, angle, area, skeleton et al. to establish the vertex correspondence, where the physically based 2D blending method is a good choice [1].

5. Conclusions and Future Work

This paper presents a new 2-D shape blending method based on visual feature decomposition. In the proposed method, the source and destination polygons are compatibly decomposed interactively into sub-polygon pairs according to their visual feature correspondences. The feature-decomposition vertices form the frame polygons which represents the rough shape of the original polygon. Then a local coordinate system is constructed for each sub-polygon, and vertices in the sub-polygon are transformed into the local coordinate system. By interpolating the corresponding frame polygons and local coordinates, the visual features on the source polygon are smoothly transformed into the corresponding visual features on the destination polygon. Users can control the visual feature correspondence and blending effect flexibly and intuitively by modifying the feature decomposition vertices. This is very useful in 2-D shape blending editing. Compared with previous methods, our method provides the user a way to control the correspondence of visual features between two polygons precisely and intuitively in real time. Another advantage is that our method is simple and computationally efficient.

Future researches should be focused on how to select the feature decomposition vertices automatically. It is also worthwhile to generalize our method to 2-D multi-polygon blending with different topology.

Acknowledgements

This work is supported jointly by National Natural Science Foundation of China (No. 60373036, 60333010) and National Basic Research Project of China (No. 2002CB312101). We thank Li Yu for her help on writing the paper.

References
