1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

POLYGON MORPHING USING FOURIER PARAMETERIZATIONS

Ding-Horng Chen and Yung-Nien Sun

Department of Computer Science and Information Engineering
National Cheng-Kung University, Tainan, Taiwan, R.O.C.
Email:chendh@vision.iie.ncku.edu.tw

ABSTRACT

A novel algorithm for polygon morphing was proposed in this paper. We adopted the parametric curve representation based on Fourier descriptor estimation to transfer the traditional morphing process in spatial domain [1] into the process in parametric space instead [3,7]. The principles were to express the polygon with estimated Fourier description parameters, and then interpolated the parameters of both source and target polygons. Fourier descriptors of the sampled x and y profiles for both the source and target polygons were obtained efficiently by using the fast Fourier transform (FFT) algorithm. Intermediate contours in-between the source and target polygons were then reconstructed based on interpolation of the obtained Fourier descriptors of the two polygons. The experiment results were superior in appearance to the ones obtained by other addressed works.

Keywords: Polygon morphing, Parametric representation, Fourier descriptor

1. INTRODUCTION

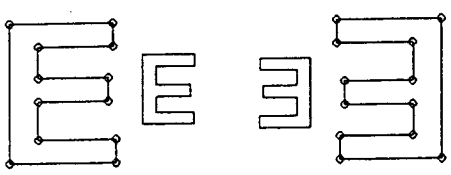
"Morphing", "metamorphosis" or "warping" is a term that describes a procedure to smoothly deform the desired content from the source image to the target one. The morphing technique is quite popular as a special effect in motion pictures or computer animation. Generally speaking, morphing techniques can be divided into two major categories, one is the gray image morphing, which transforms one gray level image to the other gray level image; another approach is the binary image morphing which deforms the binary object such as polygons into another desired polygons. In this paper, we focus ourselves in the polygon morphing.

Polygon morphing attracts a lot of attention in the past decade. Lots of literatures addressed about this topic [1,2,3,4,5]. The morphing procedure can be roughly divided into two sequential steps. The first is the establishment of the corresponding matching vertices between the source and the target objects. This step is normally achieved by human intervention, i.e. the corresponding matching vertices are assigned manually. Some researchers try to derive an automatic method to establish the vertex correspondence [5]. The second step is to find a reasonable and smooth transition function that warps the source image or object to the target one. The intermediate images now can be calculated by the transition function. The gray level image morphing can also be achieved by cross-dissolving the gray levels of the two images after the polygon morphing.

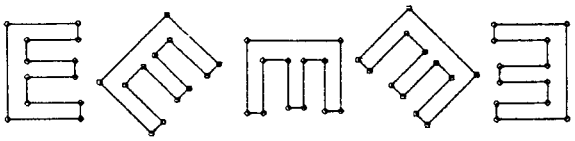
Given the source polygon, say P_0, and the target polygon, say P_1, the morphing procedure can be defined formally by:

$f: t \to P_t, t \in [0,1]$, where $f(0) = P_0$, $f(1) = P_1$, and $f(t) = P_t$ is the desired intermediate object. It is trivial that the morphing problem is a multidimensional interpolation problem, and, is an ill-posed problem. That means the derivation of the transition function f can be as many as possible. Some natural constraints would be given to constrict the solution space for finding the transition functions. For example, if the shapes of the source and the target objects are closed and simple, the intermediate object's shape should also be closed and simple. The area of the intermediate object should be gradually varied between the areas of the source and the target polygons. Moreover, The intermediate objects should be translation and rotation invariant if the source and the target objects are exactly the same one.

The traditional algorithm for solving the morphing problem is just linearly interpolating the corresponding matching vertices between the source and the target polygons, i.e., the so-called "vertex interpolation". As Fig. 1 shows, this method suffers the drawbacks of not satisfying the natural constraints mentioned above. While the alphabet "E" is morphed into the turnover "E", we expect the in-between objects could keep the size and rotated smoothly. But the traditional vertex interpolation method can not satisfy such requirement. The physically based or feature based approach reduces such disadvantages by imposing constraints on the angle of the source and the target polygons [6]. But this method is working on the physical domain, and it tends to distort the polygon area. Meanwhile, Shapira et. al [3] proposed a complex morphing algorithm and obtains a better result by human visual inspection. They introduce a special representation of the morphed polygons that called "star-skeleton". The morphing process is performed on the star-skeleton. The advantage of this method is that it transforms the physical domain problem into a parametric space problem. But the representation used in the algorithm is quite complicated, it is necessary to derive a simple but efficient representation to retain the natural constraints.



(a) Tranditional vertex intepolation



(b) Improved vertex intepolation

Fig. 1 The vertex interpolation. (a) The alphabet "E" is morphed into the turnover "E". The intermediate polygons tend to become smaller and intersect itself. (b) The improved vertex interpolation.

Goldstein et al. [7] represent the polygon boundary by a multi-resolution scheme. They first employ curve evolution step by step to transform the original polygon into a convex object and record the vertex paths. The morphing procedure is performed by interpolating and merging the corresponding vertex paths between the transformed source and target polygons. This method also transforms the morphing work from a physical domain problem into a parametric domain problem. However, the step of "curve evolution" in this algorithm needs a lot of computation and the maintenance of the vertex paths is very complicated.

In this paper, we propose a simple and novel method that satisfies the natural morphing constraints and obtains better results compared with other algorithms. The basic idea of our method is to transform the traditional vertex interpolation method into the parametric curve manipulation. Thus, the morphing procedure is performed in the parametric space with the polygons represented by parameters of the Fourier description. To retain the details near the corner position, the transverse axis scale of the shape profiles is sampled non-uniformly according to their curvature values. This step allows the morphing operations keep the details of the original shape around the corner areas. The true morphing work is implemented in the parametric space and then polygons are reconstructed from the interpolated parameters.

This paper is organized as follows. In section two, we introduce the preliminary theories. Section three illustrates the algorithm. Some experimental results are demonstrated in section four and conclusions are given in section five.

2. PARAMETRIC CURVE AND FOURIER PARAMETERIZATION

In this section, the basic concepts of the Fourier shape description and the relevant terminology that we used in our study are introduced. Algorithm and examples are given in the subsequent sections.

2.1 Parametric Curve

A parametric curve is used to represent a shape of a closed or open curve and map the curve from the physical domain into a real interval [0,1]. Suppose that C is a parametric curve, the mapping of C can be written as:

$$C:[0,1]\rightarrow R^2 \qquad (1)$$

The points of the curve would map from the [0,1] into the 2-D image; that is,

$$C \mapsto v(c)=(x(c),y(c)), \quad \text{where } c\in[0,1].$$

The regularized representation of parametric curve is then used in the image morphing algorithm. The advantage of employing the parametric curve is to avoid the many-to-few matching problem in the physical domain [2]. It is not necessary to maintain the vertex paths of the polygon as in the conventional method [7]; the correspondence between the source and the target polygons is now transformed into the correspondence between the real interval pair.

2.2 Fourier Parameterizations

The goal of the parameterization for a polygonal boundary is to find an efficient representation of the original boundary that can also be effectively manipulated to achieve the morphing operations. The boundary parameterization has the advantage that it dramatically reduces the number of the parameters compared with the free-formed or original representation. The additional restrictions, such as smoothness, can also be involved by the parameterization.

Curve parameterization expresses the shape based on the linear combination of some orthonormal bases. It is implemented by the weighted sum of a set of known basis functions. Given a continuous curve, $X(t)$; and a set of basis function $\phi_k(t)$, the parametric curve $X(t)$ can be defined as

$$X(t)=\sum_{k=1}^{\infty}\beta_k\cdot\phi_k(t), \text{ where } \beta_k=\int_0^t X(t)\phi_k(t)dt \qquad (2)$$

The coefficients β_k are the weights for distinct basis functions, on the other hand, they are also the projections of the function $X(t)$ onto the k basis functions. Many orthogonal polynomial functions fit the requirement of orthonormality as basis functions. Fourier parameterization that uses the sinusoids or trigonometric functions

$$\phi=\left\{\frac{1}{2\pi},\frac{\cos x}{\pi},\frac{\sin x}{\pi},\frac{\cos 2x}{\pi},\frac{\sin 2x}{\pi},\ldots\right\} \qquad (3)$$

as the basis is one of the popular examples.

The sinusoidal basis is used to transfer the original data to the transform domain that is usually called the frequency domain. The discrete version of Fourier transformation expressed in (4) is usually computed via the fast Fourier transform (FFT). The truncated version of (2)

$$X(t)=\sum_{k=1}^{n_0}\beta_k\cdot\phi_k(t), \text{ where } \beta_k=\int_0^t X(t)\phi_k(t)dt \qquad (4)$$

eliminates the higher frequency terms, and still preserves the accuracy of the shape.

The standard Fourier parameterization is used in both open and closed curves. But if we focus ourselves only on the closed curve, the elliptic Fourier representation could be

more appropriate [8,9]. The elliptic Fourier representation (or *Fourier descriptor*) assumes the closed curve as a combination of successive ellipses. The parametric form of the elliptic Fourier representation is expressed by

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} a_0 \\ c_0 \end{bmatrix} + \sum_{k=1}^{\infty} \begin{bmatrix} a_k & b_k \\ c_k & d_k \end{bmatrix} \begin{bmatrix} \cos kt \\ \sin kt \end{bmatrix} \tag{5}$$

where

$$a_0 = \frac{1}{2\pi}\int x(t)dt \qquad c_0 = \frac{1}{2\pi}\int y(t)dt$$
$$a_k = \frac{1}{2\pi}\int x(t)\cos ktdt \quad b_k = \frac{1}{2\pi}\int x(t)\sin ktdt$$
$$c_k = \frac{1}{2\pi}\int y(t)\cos ktdt \quad d_k = \frac{1}{2\pi}\int y(t)\sin ktdt$$

The truncated version of (5) where ∞ is replaced with a fixed basis number n_0 is allowed to simplified the data representation while still keep good approximation of the original shape.

3. ALGORITHM

The morphing algorithm is illustrated as Fig. 2. Given one source polygon, one target polygon and the matching vertices, we have to build the correspondence between the matching segments. We first transform the global coordinates for each matching segment into the local coordinate system. For preserving the detail in shape, we adjust the axis scale based on curvature for each pair of corresponding segments. The Fourier parameterization for local coordinate is computed and then used for the parameter interpolation. The intermediate polygons are then reconstructed by substituting the interpolated Fourier parameters into (5).

The details of the proposed morphing algorithm will be illustrated in the following subsections.

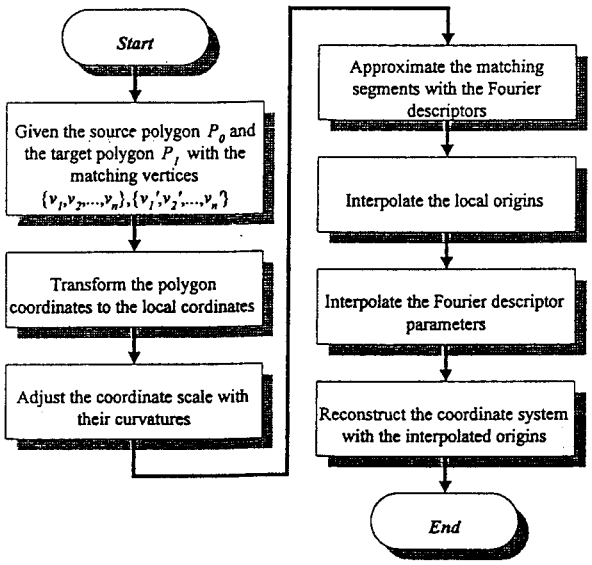


Fig. 2(a) The flowchart of the proposed algorithm

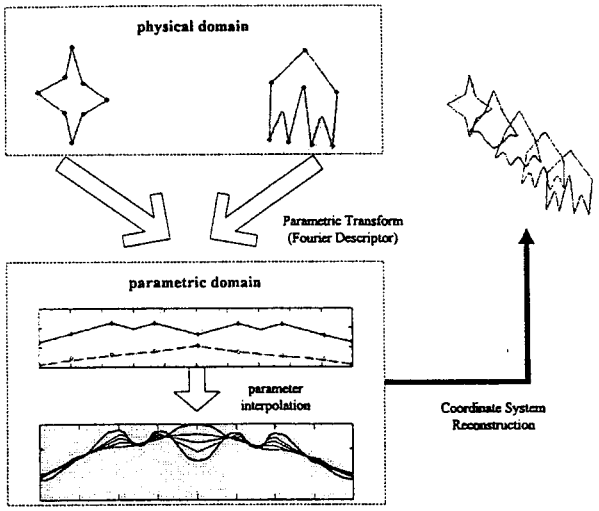


Fig. 2(b) The conceptual sketch of Fig. 2(a)

3.1 Coordinate Transformation

A polygon is expressed in the discrete form in a computer system. A polygon is usually represented with a set of chain codes or connected points. In the morphing system, the matching vertices are given manually. Two polygons and their corresponding matching vertices are expressed as

$$\begin{aligned} P_0 &: \{p_1, p_2, ..., p_M\} \approx [v_1, v_2, ..., v_n] \\ P_1 &: \{p_1', p_2', ..., p_N'\} \approx [v_1', v_2', ..., v_n'] \end{aligned}, \tag{6}$$

where M is the total number of points of the source polygon P_0, N is the total number of points of the target polygon P_1, and n is the number of the matching vertices. The vertex points of the source polygon, $v_1, v_2, ..., v_n$ are assigned to be morphed to the ones of the target polygon, $v_1', v_2', ..., v_n'$, respectively.

The global coordinate system is converted to the local coordinate for aligning the shape of the source and the target polygons. The transform is quite straightforward, the transformed coordinates of the vertices are defined as

$$\begin{aligned} &\tilde{P} : \{\tilde{p}_1, \tilde{p}_2, ..., \tilde{p}_N\} = \{p_1 - \bar{p}, p_2 - \bar{p}, ..., p_N - \bar{p}\}, \\ &\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i \end{aligned} \tag{7}$$

Here we can assume that the origins of the source and the target polygons $\tilde{P}_0, \tilde{P}_1$ are well aligned by the local coordinate transformation. The polygon areas denoted as Ω_0 and Ω_1 are defined by

$$\Omega_0 = \sum_{i=1}^{M} \|\tilde{p}_i\|, \Omega_1 = \sum_{i=1}^{N} \|\tilde{p}_i'\|. \tag{8}$$

An example of the matching segments is illustrated in Fig. 3 and will be explained in detail subsequently.

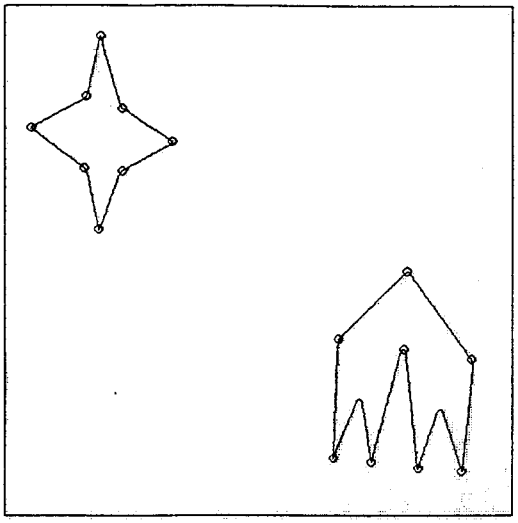


Fig. 3 An example for the morphing algorithm. The top-left polygon will be morphed to the bottom-right polygon. The circles on the polygons indicate the matching vertices.

3.2 Axis Scale Sampling Based on Curvature

Using the parametric curve representation, the x and y coordinates can be represented, or drawn, as a profile. The morphing of the complete boundary is decomposed into the combination of the morphing of boundary segments. To preserve the detail near the salient point, the curvature is calculated and used to adjust the scale on the object profile. To avoid the shape blurring effect during the morphing process, we stretch the object scale with the high curvature points, and squeeze the scale with the low curvature points. Therefore, the original shape is sampled with more points in the high-curvature areas and with less points in the low-curvature ones.

Define the first and the second derivatives for the i-th point p_i as follows:

$$\dot{x}_i = \frac{x_{i+1} - x_{i-1}}{s_{i+1} - s_{i-1}}, \dot{y}_i = \frac{y_{i+1} - y_{i-1}}{s_{i+1} - s_{i-1}},$$

$$\dot{x}_i^{for} = \frac{x_{i+1} - x_i}{s_{i+1} - s_i}, \dot{x}_i^{back} = \frac{x_i - x_{i-1}}{s_i - s_{i-1}}$$

$$\dot{y}_i^{for} = \frac{y_{i+1} - y_i}{s_{i+1} - s_i}, \dot{y}_i^{back} = \frac{y_i - y_{i-1}}{s_i - s_{i-1}},$$

$$\ddot{x}_i = \frac{\dot{x}_i^{for} - \dot{x}_i^{back}}{s_{i+1} - s_{i-1}}, \ddot{x}_i = \frac{\dot{y}_i^{for} - \dot{y}_i^{back}}{s_{i+1} - s_{i-1}} \quad (9)$$

where $s_i = \sum_{j=1}^{i-1} \| p_{j+1} - p_j \|$.

The curvature at point p_i can be computed by

$$k_i = \frac{\dot{x}_i \ddot{y}_i - \ddot{x}_i \dot{y}_i}{\left(\dot{x}_i^2 + \dot{y}_i^2\right)^{3/2}}. \quad (10)$$

Thus, we can sample the axis by the following equation:

$$\lambda' = \frac{\exp(\alpha \cdot \|k_\lambda\|)}{\sum_\lambda \exp(\alpha \cdot \|k_\lambda\|)}, \quad (11)$$

where λ and λ' are the sampled scales on the original and adjusted boundary data, respectively, k_λ is the corresponding curvature for λ, and α is a scale factor emphasizing the importance of the curvature. If we set α to zero, then equation (11) does not change the data sampling.

Fig.4(a) illustrates an example of boundary sampling according to the curvature. As Fig.4(b) shown, the original x profile is a straight line, but after the sampling process, the line come into a curve. In our algorithm, the sampling process is proceed on both x and y profile in the source and target polygons.

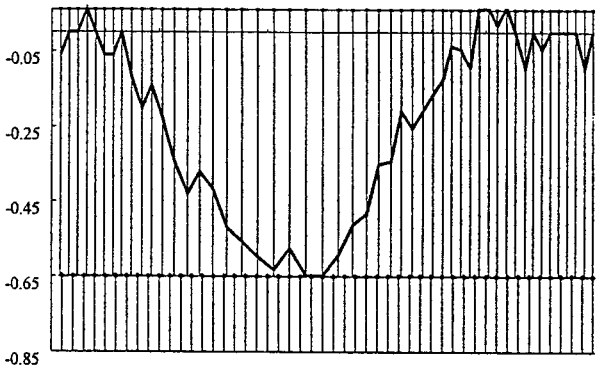


Fig.4(a) The axis sampling. The bottom grid is the original scale. The upper grid is the data scale adjusted according to the curvature. The adjusted data are then resampled to generate more points on the high-curvature areas.

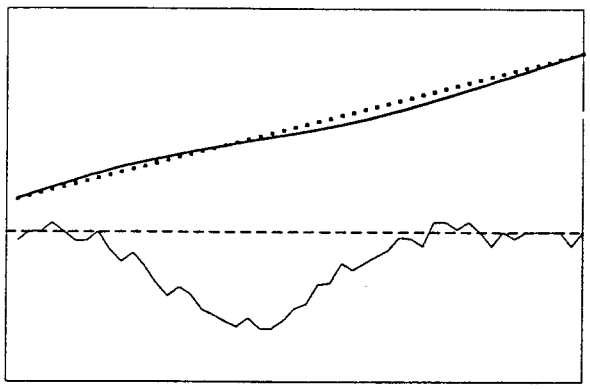


Fig.4(b) The bottom curve indicates the curvature for the corresponding x profile. The dashed line indicates the original x profile. After sampling process, the original straight line becomes a curve expressed by the solid line.

3.3 Fourier Descriptor Estimation

The source and the target polygons can be expressed as the combination of the matching segments, i.e.

$$P_0 = \sum_{i=1}^{n} \overline{v_i v_{i+1}}, v_{n+1} = v_1$$
$$P_1 = \sum_{i=1}^{n} \overline{v'_i v'_{i+1}}, v'_{n+1} = v'_1 \quad (12)$$

By the parametric curve representation, the corresponding radian of the matching segments for the source and the target polygons are defined as $\{t_1, t_2, ..., t_i, ..., t_n\}$, $\{t'_1, t'_2, ..., t'_i, ..., t'_n\}$, respectively. Now we want to approximate the matching segments with the Fourier descriptor. The Fourier descriptor approximation is proceeded on both the x and y profiles. In general, the number of basis or *harmonic* n_0, determines the details of the estimation. Since the global coordinate system is transformed to the local coordinate system, the origin vector $[a_0 \quad c_0]^T$ in equation (5) is simply a zero vector. The estimation is performed in each matching segment pair $(\overline{v_i v_{i+1}}, \overline{v'_i v'_{i+1}})$ to obtain the Fourier descriptor parameters $(a_{ik}, b_{ik}, c_{ik}, d_{ik})$, $(a_{ik}', b_{ik}', c_{ik}', d_{ik}')$, k=1...n_0. The Fast Fourier Transform (FFT) can also be employed to reduce the computation cost.

3.4 Morphing between the Source and the Target Polygons

The morphing between the source and the target polygons is consisted of two parts. The first part is the shape and size interpolation, and the second part is the shape reconstruction. The shape interpolation is carried out in the parametric space, i.e. the individual Fourier parameters $(a_{ik}, b_{ik}, c_{ik}, d_{ik})$, $(a'_{ik}, b'_{ik}, c'_{ik}, d'_{ik})$, i=1...n, are used for interpolation in this part. Suppose there are S intermediate morphing steps, the s-th parameters (s=1..S) can be calculated by

$$\begin{aligned} a_{ik}^{(s)} &= \frac{S-s+1}{S+1} \cdot a_{ik} + \frac{s}{S+1} \cdot a'_{ik} \\ b_{ik}^{(s)} &= \frac{S-s+1}{S+1} \cdot b_{ik} + \frac{s}{S+1} \cdot b'_{ik} \\ c_{ik}^{(s)} &= \frac{S-s+1}{S+1} \cdot c_{ik} + \frac{s}{S+1} \cdot c'_{ik} \\ d_{ik}^{(s)} &= \frac{S-s+1}{S+1} \cdot d_{ik} + \frac{s}{S+1} \cdot d'_{ik} \end{aligned} \quad ,i = 1...n \quad (13)$$

Let $\bar{p}_0 = (\bar{x}_0, \bar{y}_0)$, Ω_0, and $\bar{p}_1 = (\bar{x}_1, \bar{y}_1)$, Ω_1 be the origins and the area of the source and the target polygons, respectively. For the s-th intermediate morphing step, the origins of the local coordinate system and the size-scaling factor of the morphed object in the intermediate morphing steps are also defined as the linear interpolation of the source and target polygons as follows:

$$\bar{p}^{(s)} = \frac{S-s+1}{S+1} \cdot \bar{p}_0 + \frac{s}{S+1} \cdot \bar{p}_1 \quad (14)$$

$$\tau^{(s)} = \sqrt{\frac{S-s+1}{S+1} + \frac{s}{S+1} \cdot \frac{\Omega_1}{\Omega_0}}. \quad (15)$$

The origins record the trajectory of the morphing process. Because the shape size is proportional to the square of its length, we take the square root in (15). For a specific s step, we can reconstruct the coordinate system of the intermediate polygon by applying the revision of (5), (13), (14), and (15):

$$\begin{aligned} P^{(s)} &= \bar{p}^{(s)} + \tau^{(s)} \cdot \sum_i \sum_{k=1}^{n_0} \begin{bmatrix} a_{ik}^{(s)} & b_{ik}^{(s)} \\ c_{ik}^{(s)} & d_{ik}^{(s)} \end{bmatrix} \cdot \begin{bmatrix} \cos kt_i \\ \sin kt_i \end{bmatrix} \\ &= \begin{bmatrix} \bar{x}^{(s)} \\ \bar{y}^{(s)} \end{bmatrix} + \tau^{(s)} \cdot \sum_i \sum_{k=1}^{n_0} \begin{bmatrix} a_{ik}^{(s)} & b_{ik}^{(s)} \\ c_{ik}^{(s)} & d_{ik}^{(s)} \end{bmatrix} \cdot \begin{bmatrix} \cos kt_i \\ \sin kt_i \end{bmatrix} \end{aligned} \quad (16)$$

Equation (16) preserves the polygon size and satisfies the natural constraints of morphing. The interpolated origin $\bar{p}^{(s)} = [\bar{x}^{(s)} \quad \bar{y}^{(s)}]^T$ locates the translation between the source and the target polygons. The size-scaling factor $\tau^{(s)}$ makes the size of the intermediate object varied smoothly.

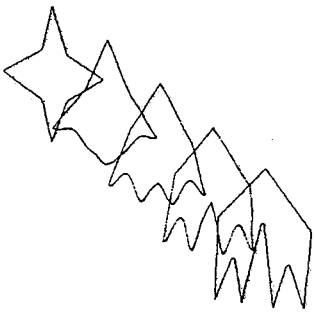
4. EXPERIMENTAL RESULTS

It is difficult to evaluate the performance of the morphing algorithms quantitatively. The most popular criterion adopted in recent reports is visual inspection by human eyes. The experiments were made to testify the correctness and performance of our algorithm. We have made some comparison with other researcher's results. As Fig. 5 shows, our method is apparently superior to the algorithms proposed by Sederberg et. al and Goldstein et. al; and have the similar appearance compared with the algorithm of Shapera et. al. The morphing on the corner in the intermediate polygons is reasonable and smooth. But the shape size of our method seems varied more smoothly than Shapera's method. Fig. 6 shows the intermediate polygons by our method.The x and y profiles of Fig. 3 are shown in Fig. 7. To reconstruct the morphed coordinate system, the interpolated parameters used for Fourier descriptor are employed with the sampled scale and are shown in Fig. 8. Fig. 9 demonstrates another comparative result for another polygon.

(a) Sederberg's method

(b) Shapira's method

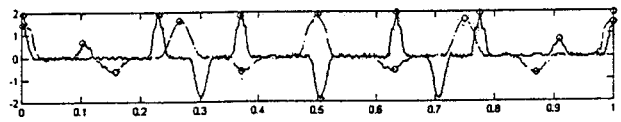
(c) Goldstein's method



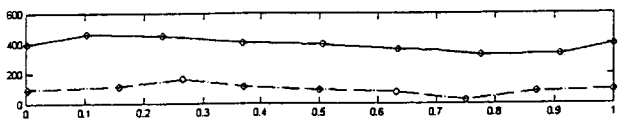
(d) Our method

Fig. 5 Comparative results of morphing algorithms. The assignment of the matching vertices is the same with other researchers. Our method transitionally morphs the source polygon from the top left to the bottom right target polygon.

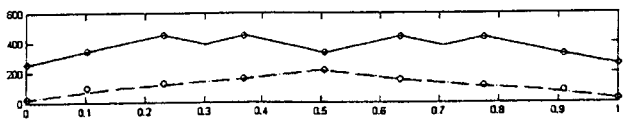
Fig. 6 The intermediate morphing polygons by our method.



(a)

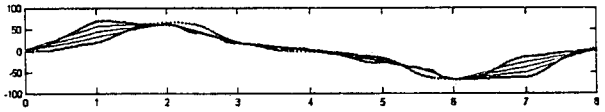


(b)

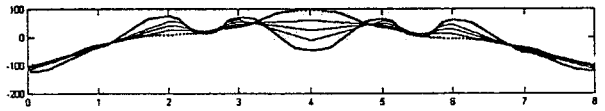


(c)

Fig. 7 Curvature and x, y profiles for Fig. 3 with parametric curve representation. The circles indicate the matching vertices; dash line: source polygon, solid line: target polygon. (a) The global x profile without local coordinate transform; (b) The global y profile.

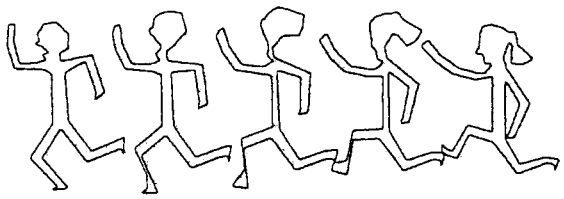


(a)

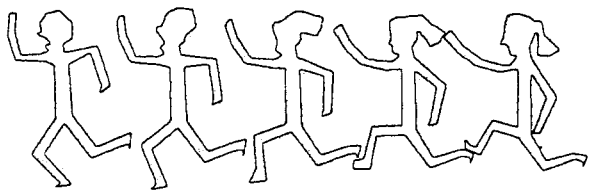


(b)

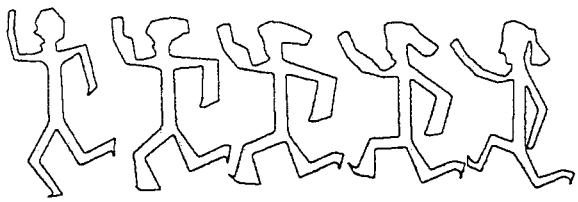
Fig. 8 Parameter interpolation. The bold line indicates the estimated Fourier descriptor representation with the sampled scale for source and target polygons, the in-between solid lines indicate the interpolated objects. (a) x profile; (b) y profile.



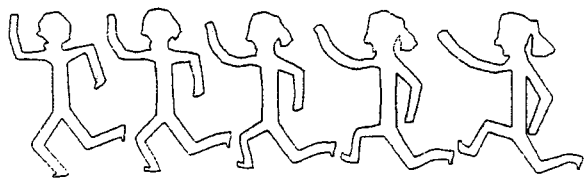
(a) Sederberg's method



(b) Shapira's method



(c) Goldstein's method



(d) Our method

Fig. 9 Comparative studies for another polygon.