Perceptually Based Approach for Planar Shape Morphing

Ligang Liu* Guopu Wang[1†] Bo Zhang Baining Guo Heung-Yeung Shum
Microsoft Research Asia [1]Tsinghua University

Abstract

This paper presents a novel approach for establishing vertex correspondences between two planar shapes. Correspondences are established between the perceptual feature points extracted from both source and target shapes. A similarity metric between two feature points is defined using the intrinsic properties of their local neighborhoods. The optimal correspondence is found by an efficient dynamic programming technique. Our approach treats shape noise by allowing discarding small feature points, which introduces skips in the traversal of the dynamic programming graph. Our method is fast, feature preserving, and invariant to geometric transformations. We demonstrate the superiority of our approach over other approaches by experimental results.

1. Introduction

Shape morphing is the gradual transformation of one shape into another which has wide applications such as modeling, animation, medicine, and entertainment.

The morphing problem has been investigated in many contexts [12, 26]. In particular, object-space morphing has two subproblems. First is the vertex correspondence problem - determining the vertex correspondence pairs. Second is the vertex path problem - finding paths that the corresponding vertices traverse during the morphing process.

1.1. Related work

The establishment of suitable correspondence between the shapes is the primary concern for the morphing. There are many attempts to find methods for automatically finding vertex correspondences between shapes in the literature. Sederberg and Greenwood [22] proposed to use a physically based approach to minimize some work function to determine the correspondence. Zhang [27] introduced a fuzzy vertex correspondence based on maximizing a similarity function between vertices. This method is similar to the physically based approach but uses a similarity function instead of a work function. Ranjan etal. [19] represented an object by a union of circles. Correspondence is established by considering the sizes and relative locations of the circles and the in-between objects are generated by blending corresponding circles. Recently Mortara and Spagnuolo[17] used an approximate skeleton to describe the shape and established the vertex correspondence through finding a reasonable match among the approximate skeletons of the shapes.

In addition, in the morphing of curved shape, existing methods usually assume a uniformly distributed set of vertices approximating the shape. Sederberg etal. [23] extended the physically based approach [22] for establishing correspondences between knots of closed B-spline curves. Cohen etal. [5] used an approximated solution exploiting dynamic programming over the discrete sample sets of the two curves and their unit tangent vector fields. In industrial design context, Hui and Li [10] proposed a technique for locating significant vertices along curved shapes, which represent shape features. Correspondences between features of the objects are then established. Recently, Sebastian etal. [20] introduced an approach to find a correspondence between two curves based on a notion of an alignment curve which treats both curves symmetrically.

The vertex correspondence problem also arises in surface reconstruction [15], shape recognition and retrieval [2].

Other object-space methods were designed to solve the path problem assuming that the correspondence is given. Most of the research on solving the path problem concentrates on trying to eliminate shrinks and self-intersections and preserve the geometric properties of the intermediate shapes [3, 7, 21]. Two curves were morphed by blending their curvature signatures in [24]. Recent work of [1, 8, 11] blended two polygonal shapes by interpolating their compatible triangulations.

∗ 3F Beijing Sigma Center, No 49 Zhichun Road, Haidian District, Beijing 100080, P R China. Email: ligang_liu@hotmail.com (Ligang Liu), {bzhang, bainguo, hshum}@microsoft.com

† This research was done when Guopu Wang was working as part-time intern at Microsoft Research Asia. Email: wangguopu00@mails.tsinghua.edu.cn

1.2. Perceptually based approach

This paper mainly focuses on the vertex correspondence problem for planar morphing. A novel perceptually based approach is proposed to establish the correspondences between planar shapes. In this paper, a planar shape is represented as a series of contiguous curve segments or line segments. The shapes are first scaled to the same size so that the relative sizes of the shapes will not affect the correspondence process.

In previous approaches, correspondence is usually established by measuring the similarity between series of vertices approximating the shapes. It can be seen that correspondence between visual features other than vertices is essential to retain certain characteristics of the original shapes for morphing at a human level of competence. For instance, consider the morphing of two dancers. It is desirable that the visual parts such as head, arms, and legs in both shapes are in correspondence. The corresponding parts are to be morphed respectively to form the intermediate shape.

A shape can be generally interpreted as composed of some different perceptual or meaningful visual parts. Feature points segment a shape into meaningful subparts and play a dominant role in shape perception by humans. If the feature points are identified and matched properly for two shapes, the morphing result will conform to the perception of the human.

Given a source shape and a target shape, the perceptual feature point of both shapes are first extracted. We propose a novel approach to automatically establish correspondence between the feature points. Our approach takes into consideration the local information of the feature points according to some reasonable similarity criteria. The correspondence between these feature points is computed by optimizing a global cost function using dynamic programming (DP) technique. We also propose a mechanism for computing the cost for discarding relatively small and unimportant feature points. Discarding feature point has a similar effect to that of smoothing several short feature segments in a shape to produce a single longer feature segment, but without actually performing the costly smoothing operation. Discarding operation of feature points is incorporated in the dynamic programming scheme which introduces ”skips” in the traversal of the DP graph.

Our approach has several advantages over previous methods [5, 20, 22]. First, our approach establishes the correspondence between perceptually feature points at a human level of competence instead of the vertices of polygonal approximation. So the corresponding features can be preserved during the morphing sequence. Second, similarity measurement between a pair of feature points in our approach is determined by a wide range of local region of the feature points, while only a few nearby vertices are used in measuring similarity between vertices in their approaches. Third, the number of vertices approximating a shape is much more than the number of feature points detected on the shape. Thus our algorithm is much faster. Finally, our approach is resistant to moderate amounts of noise by discarding small and unimportant feature points in DP.

The contributions of our perceptually based approach for shape morphing are summarized in the following:

- **Feature preserving:** Correspondence between perceptually visual features other than vertices is essential to retain certain characteristics of the original shapes for morphing at a human level of competence. Overall shape as well as features can be preserved during the morphing process.
- **Fast:** Optimal correspondence is found by effective DP technique. It is faster than other methods in that there are only a few feature points used in the DP process.
- **Robust and flexible:** Skips are allowed in the traversal of DP graph by discarding unimportant feature points. Thus our approach is resistant to moderate amounts of noise.
- **Geometric invariant:** Both the correspondence and interpolation methods are invariant under geometric transformations such as translation, rescaling, and rotation.

1.3. Overview

The paper is organized as follows. Section 2 defines some geometric quantities for feature points. A general framework to find an optimal correspondence between feature points of two shapes is described in Section 3. Section 4 introduces the method to create the intermediate morphing sequences. Additional experimental results are illustrated in Section 5. We conclude the paper in Section 6 with the summary and future work.

2. Perceptual Feature Points for Planar Shapes

Feature detection is a well-studied research area in many scientific fields, including computer vision, medical imaging and computational fluid dynamics [4, 14, 28].

Given a shape $\mathcal{P}$, we first select a set of potential feature points, i.e., points that with high probability belong to a feature point. For curved shapes, potential feature points include the curvature extrema, cusp, inflection points, and the discontinuities of curvature. For polygonal shapes, we simply set all the vertices as potential feature points as they are actually discontinuous points. End-points should also be regarded as feature points for non-closed shape. We then use a simple and efficient method [4] to extract the feature points from the potential candidate points. Figure 1 shows the results of feature point detection for two human shapes.

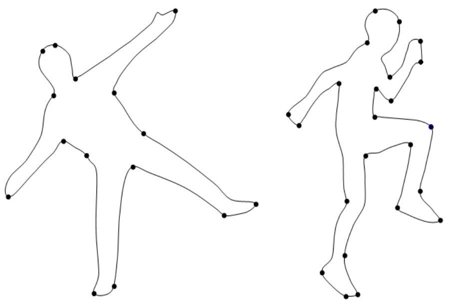


Figure 1. Some results of feature point detection.

As a pre-processing, the shape $\mathcal{P}$ is first densely and semi-uniformly sampled and represented by a sequence of points $P_i, i = 0, 1, \ldots, n$, while keeping all the feature points in the sampled points. The sampled points are used in computation of the covariance matrix introduced in the following.

2.1. Covariance matrix analysis of local neighborhood

We define a region of support (ROS) of a feature point P_i as a local neighborhood

$$ROS_h(P_i) = \{P_j | j = i - h, i - h + 1, \ldots, i + h\},$$

for some interger h.

Various researches have used principal component analysis of local point neighborhoods to estimate local shape properties [9, 18]. The measure for the shape property of the feature point P_i is derived from the statistical and geometrical properties associated with the eigenvalue-eigenvector structure of the covariance matrix of sample points over the region of support $ROS_h(P_i)$.

Let $\bar{P}_i = (\bar{x}_i, \bar{y}_i)$ be the center of $ROS_h(P_i)$. The 2×2 covariance matrix of $ROS_h(P_i)$ is defined as:

$$C(P_i) = \frac{1}{2h+1} \sum_{j=i-h}^{i+h} (P_j - \bar{P}_i)(P_j - \bar{P}_i)^T.$$

The eigenvectors $\{e_0, e_1\}$ of matrix $C(P_i)$ together with the corresponding eigenvalues $\{\lambda_0, \lambda_1\}$ define the correlation ellipse that adopts the general form of the neighbor points $ROS_h(P_i)$, see Figure 2. Thus eigenvalues $\{\lambda_0, \lambda_1\}$ can be utilized to measure the local form at the feature point over its neighbor $ROS_h(P_i)$. The two eigenvectors e_0, e_1 closely point in tangent direction and normal direction of P_i respectively. We call the eigenvector which points in tangent direction as tangent eigenvector, denoted by e_T , and similarly for normal eigenvector e_N. Eigenvalue λ_T and λ_N that correspond to e_T and e_N respectively are called tangent eigenvalue and normal eigenvalue respectively (Figure 2).

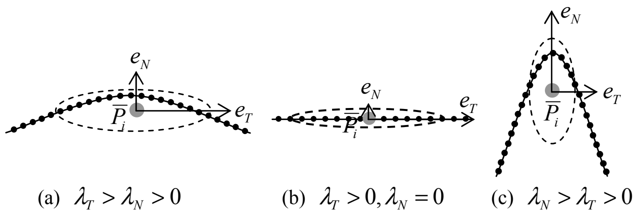


Figure 2. Eigenvalue-eigenvector structure of the covariance matrix of a point over a region of support.

It is noted that the size of ROS (ie. the number of data points used to construct the covariance matrix) will affect the eigenvalues. In fact, increasing the size of the local neighborhood is similar to applying a low-pass smoothing filter. This becomes intuitively clear as the covariance matrix is defined as sums of squared distances from the neighborhood's centroid. If we increase the neighborhood size, each individual point contributes less to the variation. Hence high frequency oscillations are attenuated, analogous to standard low-pass filter behavior. Perceptually ROS should be selected based on the level of detail represented by the shape. As increasing the size of the local neighborhood eventually violates the prerequisite that all points of the neighborhood belong to the same feature element of the underlying feature point, we suggest we use the same size of ROS for each feature point. Typically, one selects h in the range between 20 and 30 for sampling period 2.

We observe by some experiments that the eigenvalue-eigenvector structure of the covariance matrix is robust and reliable for shape under varying orientations and sampling ratio.

2.2. Geometric quantity properties of feature point

The extracted feature points represent different perceptually visual subparts of a shape. A feature element is a portion of the shape bounded by two successive feature points. A feature point P_i is the junction of its two nearby feature elements which are called left element and right element and denoted as $ROL(P_i)$ and $ROR(P_i)$ respectively.

We now define some geometric quantities for describing the local properties for a feature point.

(1) Feature variation

We define feature variation of feature point P_i as

$$\sigma(P_i) = \xi \frac{\lambda_N}{\lambda_N + \lambda_T},$$

where $\xi = 1$ if P_i is convex and $\xi = -1$ if P_i is concave. Feature variation measures deviation of neighbor of P_i from tangent direction at P_i. Value of $\sigma(P_i)$ is within the interval $[-1, 1]$ with the property that it is close to 0 as the neighborhood of P_i looks like a line segment (see Figure 2(b)) and it tends to 1 or -1 as the neighborhood shape of P_i is highly bended (see Figure 2(c)).

(2) Feature side variation

Side feature variation of feature point P_i is defined by

$$\tau(P_i) = \frac{\sigma(ROL(P_i)) + \sigma(ROR(P_i))}{2},$$

where $\sigma(ROL(P_i)) = \frac{\lambda_N^L}{\lambda_N^L + \lambda_T^L}$, $\sigma(ROR(P_i)) = \frac{\lambda_N^R}{\lambda_N^R + \lambda_T^R}$, λ_T^L and λ_N^L are eigenvalues of covariance matrix of $ROL(P_i)$, and λ_T^R and λ_N^R are eigenvalues of covariance matrix of $ROR(P_i)$. We use $\tau(P_i)$ to measure the flatness of its side neighbors.

(3) Feature size

Feature size of feature point P_i is defined by

$$\rho(P_i) = \frac{\rho^L(P_i) + \rho^R(P_i)}{2},$$

where $\rho^L(P_i)$ and $\rho^R(P_i)$ are respectively the proportions of lengths of $ROL(P_i)$ and $ROR(P_i)$ with respect to total length of the shape. Feature size $\rho(P_i)$ measures how dominant the feature on the shape boundary.

Note that all the geometric quantities are invariant under rescaling, rotation, and the sampling ratio.

3. Proposed Approach for Correspondence Problem

Let $\mathcal{S} = \{S_i | i = 0, 1, \ldots, m\}$ and $\mathcal{T} = \{T_j | j = 0, 1, \ldots, n\}$ be the source shape and target shape respectively, where S_i and T_j are the feature points of shape $\mathcal{S}$ and $\mathcal{T}$ respectively. If shape $\mathcal{S}$ (or $\mathcal{T}$) is closed, then $S_m = S_0$ (or $T_n = T_0$). Elements of $\mathcal{S}$ (or $\mathcal{T}$) are indexed by i (or j).

3.1. Similarity measurement of two feature points

The similarity between pair of feature points is measured by the geometric quantity properties. Similar features should have similar feature variation, similar feature side variation, and similar feature size. The similarity cost of a feature point S_i on $\mathcal{S}$ with a feature point T_j on $\mathcal{T}$ is computed as

$$SimCost\,(S_i, T_j) = \Psi(S_i, T_j) \sum_{q=\sigma,\tau,\rho} \omega_q \Delta_q(S_i, T_j), \quad (1)$$

where the term Δ_q is the cost associated with the difference in feature geometric quantities q (i.e., σ, τ, ρ) defined by

$$\Delta_\sigma(S_i, T_j) = |\sigma(S_i) - \sigma(T_j)|,$$

$$\begin{aligned}\Delta_\tau(S_i, T_j) &= \frac{1}{2}\left(|\sigma(ROL(S_i)) - \sigma(ROL(T_j))|\right.\\ &+ \left.|\sigma(ROR(S_i)) - \sigma(ROR(T_j))|\right),\end{aligned}$$

$$\Delta_\rho(S_i, T_j) = \frac{1}{2}\left(\left|\rho^L(S_i) - \rho^L(T_j)\right| + \left|\rho^R(S_i) - \rho^R(T_j)\right|\right),$$

and $\omega_q \geq 0$ are weights with sum to 1. The coefficient $\Psi(S_i, T_j)$ is a weight term associated with the importance of this feature correspondence defined by $\Psi(S_i, T_j) = \max[\rho(S_i), \rho(T_j)]$ which emphasizes the importance of matching large parts from both shapes similarly to the way humans pay more attention on large parts when judging the quality of correspondence.

The similarity cost takes value from 0 to 2 with the property that the value is close to 0 as the two feature points are very similar and it tends to 2 as the two feature points are rather dissimilar.

3.2. Penalty measurement of discarding a feature point

Intuitively, a feature point can be possibly discarded if its local neighborhood is small and flat enough. Therefore we define the cost of discarding a feature point S_i on shape $\mathcal{S}$ as follows:

$$DisCost\,(S_i) = \Phi(S_i) \sum_{q=\sigma,\tau,\rho} \omega_q \,|q(S_i)|,$$

where $q(S_i)$, $q = \sigma, \tau, \rho$, are the feature quantities of S_i, $\Phi(S_i) = \rho(S_i)$, and the weights ω_q are same as in Equation 1. The intuition behind the coefficient $\Phi(S_i)$ is to measure the importance of the discarded feature point relative to the whole shape. The cost of discarding a feature point T_j on shape $\mathcal{T}$ is similarly defined.

3.3. Minimization problem for correspondence

Establishing correspondence is to locate similar feature points between two shapes. Thus we can define the similarity function of $\mathcal{S}$ and $\mathcal{T}$ by using the similarity measurement between feature points. A correspondence between $\mathcal{S}$

and $\mathcal{T}$ is a mapping $J : \{S_i\} \rightarrow \{T_j\}$. We thus define the similarity function of $\mathcal{S}$ and $\mathcal{T}$ as follows:

$$SimCost(\mathcal{S}, \mathcal{T}, J) = \sum_{i=0}^{m-1} SimCost(S_i, T_{J(i)}),$$

An optimal correspondence (S_i, T_j) will be obtained if $SimCost(\mathcal{S}, \mathcal{T}, J)$ is a minimum. So the optimization problem that needs to be solved is

$$\min_J SimCost(\mathcal{S}, \mathcal{T}, J).$$

The above minimization problem can be effectively solved by dynamic programming technique. We will show the complete algorithm as follows.

3.4. Dynamic programming (DP) algorithm

All feature correspondences can be represented in an $m \times n$ rectangular DP graph defined with m rows corresponding to feature points $\{S_i\}$ of $\mathcal{S}$ and n columns corresponding to feature points $\{T_j\}$ of $\mathcal{T}$. The graph node at the intersection of row i and column j is referred to as node $node(i, j)$, which signifies a correspondence between S_i and T_j.

A complete correspondence can be represented on the graph as a string of dots starting at $(0, 0)$ and ending at (m, n). This is illustrated in Figure 3, where the dots are connected by bold lines. We will refer to such a sequence of dots (not necessarily adjacent) as a path, denoted by $\Gamma = ((i_0, j_0), (i_1, j_1), \dots (i_R, j_R))$ in the DP graph, where $(i_0, j_0) = (0, 0)$, $(i_R, j_R) = (m, n)$. The node $node(i_{r-1}, j_{r-1}), 1 \le r \le R$, is called parent of node $node(i_r, j_r)$. We denote a sequence of consecutive feature points of $\mathcal{S}$, e.g., $S_i, S_{i+1}, \dots, S_{i+k}$, as $S(i|i + k)$; similarly for the notation $T(j|j + l)$.

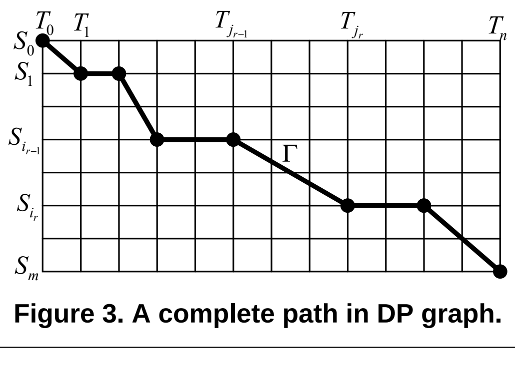


Figure 3. A complete path in DP graph.

The cost of correspondence between feature points of $\mathcal{S}$ and $\mathcal{T}$ is defined as

$$Cost(\mathcal{S}, \mathcal{T}) = \min_{\Gamma} \{Cost(\mathcal{S}, \mathcal{T}, \Gamma)\},$$

where $Cost(\mathcal{S}, \mathcal{T}, \Gamma)$ is the cost of the correspondence for the complete path Γ. In turn, $Cost(\mathcal{S}, \mathcal{T}, \Gamma)$ is defined as follows:

$$Cost(\mathcal{S}, \mathcal{T}, \Gamma) = \sum_{r=1}^{R} \delta\left(S(i_{r-1}|i_r), T(j_{r-1}|j_r)\right),$$

where $\delta\left(S(i_{r-1}|i_r), T(j_{r-1}|j_r)\right)$ represents the similarity cost between $S(i_{r-1}|i_r)$ and $T(j_{r-1}|j_r)$, defined by

$$\begin{aligned} \delta\left(S(i_{r-1}|i_r), T(j_{r-1}|j_r)\right) &= DisCost\left(S(i_{r-1}|i_r)\right) \\ &+ DisCost\left(T(j_{r-1}|j_r)\right) \\ &+ \lambda \cdot SimCost\left(S_{i_r}, T_{j_r}\right), \end{aligned}$$

where $DisCost\left(S(i_{r-1}|i_r)\right) = \sum_{i=i_{r-1}+1}^{i_r-1} DisCost(S_i)$ (note that there might be no item in the sum formula) is the total sum of the discarding cost of the feature points between $S_{i_{r-1}}$ and S_{i_r} , $DisCost\left(T(j_{r-1}|j_r)\right)$ can be similarly defined. Constant λ represents the relative importance of discarding feature points and similarity cost. High values of λ encourage discarding and, conversely, low values of λ inhibit discarding feature points. For example, matching shapes with much detail must employ high values of λ. We set $\lambda = 1$ in our paper. We have seen experimentally that the correspondence result is relatively insensitive to the choice of λ.

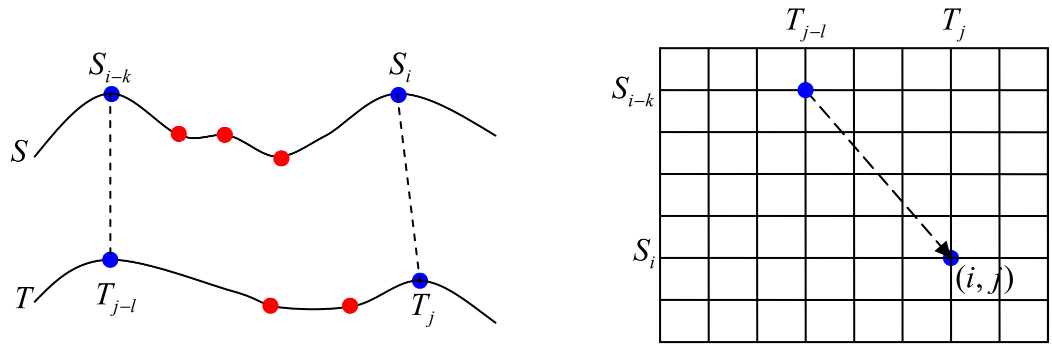
3.5. Implementations

In our implementation, we compute the optimum cost of the incomplete path ending at each node:

$$\begin{aligned} node(i, j) = \min_{k,l} &[node(i - k, j - l) \\ &+ \delta\left(S(i - k|i), T(j - l|j)\right)] \end{aligned} \quad (2)$$

where the minimum is over all possible values of $(k, l), k, l \ge 0$. Equation 2 determines the minimum cost transition from node $node(i - k, j - l)$ to node $node(i, j)$ for all possible values of $k, l \ge 0$. See Figure 4 for an instance of $k = 3$ and $l = 2$, where the skip from $node(i - 3, j - 2)$ to $node(i, j)$ shown in 4(b) means that the red feature points in Figure 4(a) would be discarded. Indices $i - k$ and $j - l$ are stored in node $node(i, j)$ and can be used to retrace the path from node $node(i, j)$ back to its starting point. It is seen that there will be skips in the traversal of the DP graph which allows more flexible optimization process.

The above algorithm is optimal, in that it always finds the path with the least cost. Equation 2 implies that the algorithm computes the minimum cost transition from each allowable node $node(i-k, j-l)$ to node $node(i, j)$. However, the algorithm may become very slow especially on large DP graphs. Notice that transitions on the DP graph correspond



(a) Correspondence between feature points (The red feature points are discarded)

(b) Update cost of node (*i*,*j*) from node (*i*-*k*,*j*-*l*)

Figure 4. Skip in DP graph.

to skips of feature points. It seems reasonable to restrict the maximum number of skips to a constant C. Thus skips in DP graph does not increase the complexity.

It can be shown [13] that the run-time complexity of the algorithm is $O(mn)$ if there is no skip in the DP graph, where m and n are number of feature points of the two shapes respectively. By restricting discarding to C feature points (usually, $C << m, n$), the time complexity becomes $O(C^2mn)$. Note that the number of feature points is much less than the number of sampling points in the previous method [5, 20, 22]. Therefore our algorithm is much faster than these methods.

When the shapes are non-closed, we use a constraint that the endpoints of the source and the target shapes correspond alternatively. For closed shapes we might not know in advance the initial correspondence. We handle the case with different starting points by repeating the algorithm for each possible case and by taking the least cost as the best correspondence.

It is worthwhile to indicate that previous methods [5, 20, 22] determined the globally optimal correspondence on existing vertices. The reason the optimization search is restricted to existing vertices is that otherwise it becomes a non-linear constrained optimization problem whose solution is very expensive and whole global optimality is difficult to verify. Note that the cost of discarding a feature point at one shape can be seen as the similarity cost between this feature point and an infinitesimal flat and short ”feature point”, with all feature quantities being zeros, at the other shape. Thus the optimization in our algorithm is essentially not restricted to existing feature points, which makes the algorithm more flexible and robust.

4. Generating Morphing Sequences

A feature point of source shape may correspond to multiple feature points of target shape or vice versa by the DP process. For this case, we keep the corresponding pair of feature points with minimal similarity cost and ignore the other corresponding pairs. The intuition behind this process is that the reasonable correspondence between two shapes will be exhibited by a few perceptually important feature points as the way humans pay primary attention on dominant feature parts when judging the quality of correspondence. Once we have a one-to-one correspondence between feature points of the shapes, other points can be generated on the shape based on the proportional length principle so that we can obtain a one-to-one point correspondence between the two shapes both for feature points and non-feature points.

To generate the in-between morph sequence, the path along which each of the points travels from the source to the target has to be defined. As the simple linear interpolation method generally yields distortion and shrink in the intermediate shapes, we use the intrinsic method[21], in which interpolated entities are edge lengths and angles between edges rather than the Cartesian coordinates of its vertices. This intrinsic method produces more satisfactory blending than the linear interpolation method generally. It handles many situations successfully, including cases where the shapes are affine transformations of each other or where parts of the shapes are transformed affinely.

5. Experimental Results

In this section we show several examples of morphing between planar shapes that illustrate the behavior of our approach. All examples in the paper are the results of an implementation of the proposed algorithm in a 2D user interface design system in C# developed at our lab. From our experimental results, the proposed algorithm is fast. The correspondence process of each example takes less than 0.6 second in all the experiments. The intermediate shapes can be computed in real time. We can generate more than 45 intermediate shapes per second for all the examples shown in this paper. The computation is performed on machine with Pentium IV 2.6G.

There are some parameters used in our approach. In practices, weights of $\omega_q = 1/3$, $q = \sigma, \tau, \rho$, in Equation 1 and the maximal skip steps $C = 2$ seem to perform well over a wide range of shapes.

Our approach is robust to the sampling number of the curves. This is due to the fact that the geometric quantity properties of feature point are not sensitive to the sampling number.

Some of the correspondence results are shown in Figure 5. Feature points are indicated by red points. Corresponding points are marked by the same numbers. Feature points without marked numbers are those skipped in the travesal of DP graph. The blue points are non-feature points which are generated by the proportional length interpolation process mentioned in Section 4.

Figure 6 shows examples of morphing sequences for shapes shown in Figure 5. Figure 6(a) shows the morphing between two dancer shapes. Note that the head, arms, and legs of the two dancer shapes are correctly in correspondence respectively and the corresponding features in the intermediate shapes are well preserved. Figure 6(b) shows the morphing between a desk shape and a turtle shape. The corresponding features are well preserved for the legs of the shapes. Figure 6(c) shows an example of morphing between two polygonal shapes. The head, legs, and tail are correctly in correspondence in the morphing sequence. A more complicated example is shown in Figure 6(d) where a horse shape is morphed into an elephant shape.

We compared our approach with those of [5] and [20]. Figure 7 shows some of these comparisons. Our algorithm (c) obtains more natural intermediate sequences in the examples. In the second example of morphing between a horse shape and a wolf shape, there is a geometrical similarity between the two shapes, including six large salient perceptual parts (which describe the four legs, the head, and the tail). There are many places where the two shapes may be considered locally similar. This local similarity is captured by the optimization process in our correspondence approach so that the corresponding feature parts are in correspondence respectively. But some perceptual parts of the two shapes are mismatched using the approaches [5] and [20].

Figure 8 plots the computational times of our approach and the approaches [5] and [20]. The statistical figures are obtained for the second example in Figure 7. Note that the running time of our approach is relevant to the number of feature points but not the number of sampling points. It shows that our approach is much faster than the other two approaches when the sampling number becomes large.

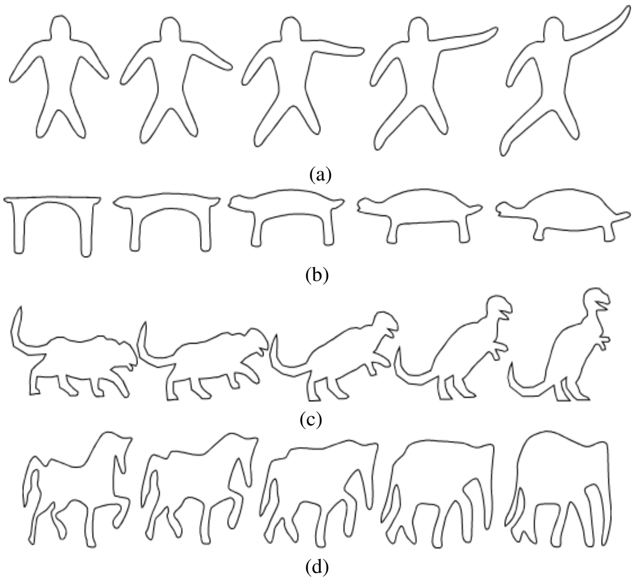


Figure 6. Morphing sequences.

Our approach is robust with respect to moderate amounts of noise due to allowing skips in the traversal of DP graph by discarding small and unimportant feature points. This can be seen in Figure 9 where some noises are added at the right arm of the target dancer shape shown in Figure 5. The correspondence result shown in Figure 9(a) is generated without any skip, i.e.,$C = 0$, while the result shown in Figure 9(b) is generated by allowing skips with $C = 2$. We can see the correspondences between the parts of right arm in Figure 9(b) is much better than in 9(a).

Observe that most of the intermediate shapes are meaningful in these examples, although in some examples self intersections can occur. More complicated interpolation method based on compatible triangulations could be used to avoid the self-intersections during the morphing sequence [8].

Our approach might fail in the case when the input shape is rather simple like circle or helical curve. This is caused by the fact no feature point can be detected for such shape. In our system, we can deal with such simple shape by inserting some ”virtual” feature points such that the polygon constructed by the feature points is a rough approximation for the shape within a specified large tolerance.

The submitted accompanying movies show the shape morphing animations of all the examples presented in this paper.

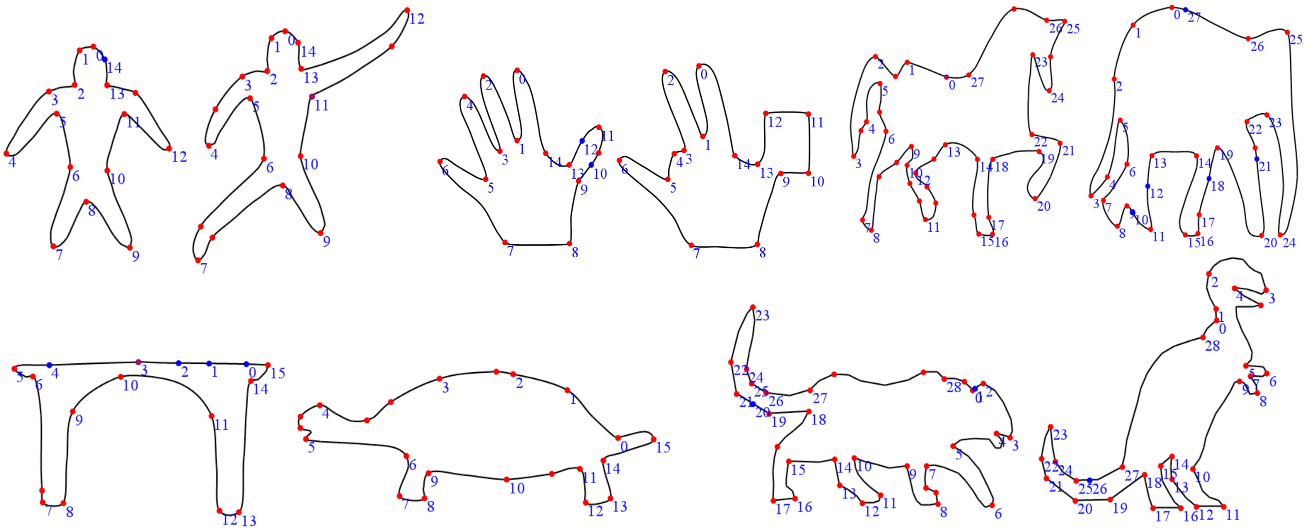


Figure 5. Correspondence results using our approach.

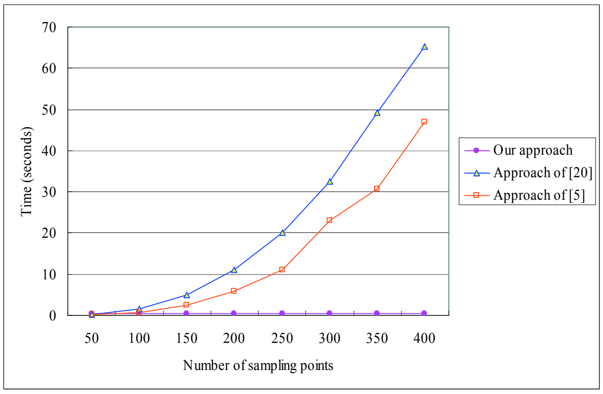


Figure 8. Comparison of running time with other approaches [5] and [20].

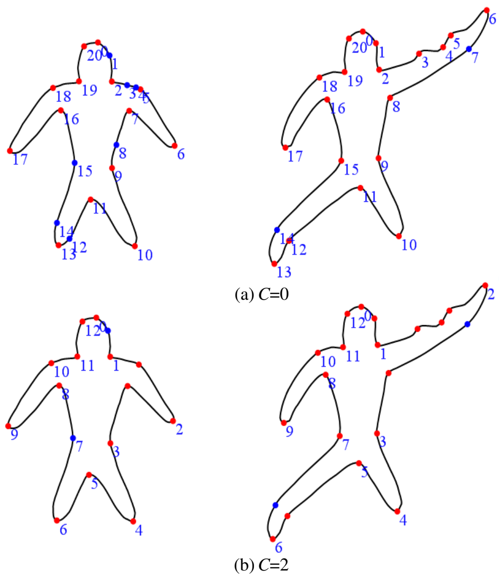


Figure 9. Robustness with respect to noise.

6. Conclusions and Future Work

In this paper, we have proposed a novel approach for morphing between two-dimensional shapes. Correspondences are built between the perceptual feature points extracted from both source and target shapes. The optimal correspondence is established using a dynamic programming process that allows skips during the process. Our algorithm is fast enough for interactive morphing applications because there are only a few feature points dealt in the process of dynamic programming. We demonstrate the superiority of our approach over traditional approaches by many experimental results.

Our approach still has much room for improvements and extensions. First, we are primarily concerned with features based on local cues in this paper. Potential extensions could include dealing with more global shape features, such as the transition from one type of curvature behavior to another, global symmetries and repeated structure, and intersections between analytic curves with different parameters. Second, correspondences between shapes would be affected by large noises on the shapes. Some work tried to address this issue using coarse-to-fine strategy based on the multi-resolution representations of shape [16, 6]. We found that mismatching in coarse level would cause distinct errors in finer levels because coarse matching is independently carried out without considering finer matching. For correct matching, it is necessary to consider multi-levels simultaneously in the similarity measurements like matching pixels of texture pyramid in texture synthesis [25]. Finally, as the proposed approach is generic for matching two shapes, we would like use the approach in other applications such as surface reconstruction, character recognition, and shape retrieval in the future.

References

[1] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *Proc. SIGGRAPH '00*, pages 157–164, 2000.

[2] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[3] E. Carmel and D. Cohen-Or. Warp-guided object-space morphing. *The Visual Computer*, 13(9-10):465–478, 1997.

[4] D. Chetverikov and Z. Szabo. A simple and efficient algorithm for detection of high curvature points in planar curves. In *Proc. 23rd Workshop of the Austrian Pattern Recognition Group*, pages 175–184, 1999.

[5] S. Cohen, G. Elber, and R. Bar-Yehuda. Matching of freeform curves. *Computer Aided Design*, 29(5):369–378, 1997.

[6] A. Finkelstein and D. Salesin. Multiresolution curves. In *Proc. SIGGRAPH '94*, pages 261–268, 1994.

[7] E. Goldstein and C. Gotsman. Polygon morphing using a multiresolution representation. In *Proc. Graphics Interface'95*, pages 247–254, 1995.

[8] C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. 25(1):67–75, 2001.

[9] S. Gumhold, X. Wang, and R. McLeod. Feature extraction from point clouds. In *Proc. 10th Int. Meshing Roundtable*, pages 293–305, 2001.

[10] K. C. Hui and Y. D. Li. A feature-based shape blending technique for industrial design. *Compute Aided Design*, 30(10):823–834, 1998.

[11] H. Johan and T. Nishita. Blending multiple polygonal shapes. In *Proc. 11th IEEE Pacific Graphics*, pages 434–440, 2003.

[12] F. Lazarus and A. Verroust. Three-dimensional metamorphosis: a survey. *The Visual Computer*, 14(8-9):373–389, 1998.

[13] M. Maes. On a cyclic string-to-string correction problem. *Information Processing Letters*, 35(2):73–78, 1990.

[14] R. Mehrotra, S. Nichani, and N. Ranganathan. Corner detection. *Pattern Recognition*, 23(11):1223–1233, 1990.

[15] D. Meyers, S. Skinner, and K. Sloan. Surface from contours: the correspondence and branching problems. *ACM Trans. Gr.*, 11(3):228–258, 1992.

[16] F. Mokhtarian and K. Mackworth. A theory of multiscale, curvature-based shape represention for planar curves. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 24(8):789–805, 1992.

[17] M. Mortara and M. Spagnuolo. Similarity measures for blending polygonal shapes. 25(1):13–27, 2001.

[18] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. 22(3), 2003. (Proc. Eurographics'03).

[19] V. Ranjan and A. Fournier. Matching and interpolation of shapes using unions of circles. 15(3):C129–C142, 1996. (Proc. Eurographics'96).

[20] T. B. Sebastian, P. N. Klein, and B. B. Kimia. On aligning curves. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 25(1):116–125, 2003.

[21] T. W. Sederberg, P. Gao, G. Wang, and H. Mu. 2d shape blending: an intrinsic solution to the vertex path problem. In *Proc. SIGGRAPH '93*, pages 15–18, 1993.

[22] T. W. Sederberg and E. Greenwood. A physically based approach to 2d shape blending. In *Proc. SIGGRAPH '92*, pages 25–34, 1992.

[23] T. W. Sederberg and E. Greenwood. *Mathematical Methods in CAGD III*, chapter Shape blending of 2-D piecewise curves, pages 1–3. 1995.

[24] T. Surazhsky and G. Elber. Metamorphosis of planar parametric curves via curvature interpolation. *International Journal of Shape Modeling*, 8(2):201–216, 2002.

[25] L. Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *Proc. SIGGRAPH '00*, pages 479–488, 2000.

[26] G. Wolberg. Image morphing: a survey. *The Visual Computer*, 14(8-9):360–372, 1998.

[27] Y. Zhang. A fuzzy approach to digital image warping. *IEEE Computer Graphics and Applications*, 16(2):33–41, 1996.

[28] P. Zhu and P. Chirlian. On critical point detection of digital shapes. *IEEE Trans. on Pattern Recognition and Machine Intelligence*, 17(8):737–748, 1995.

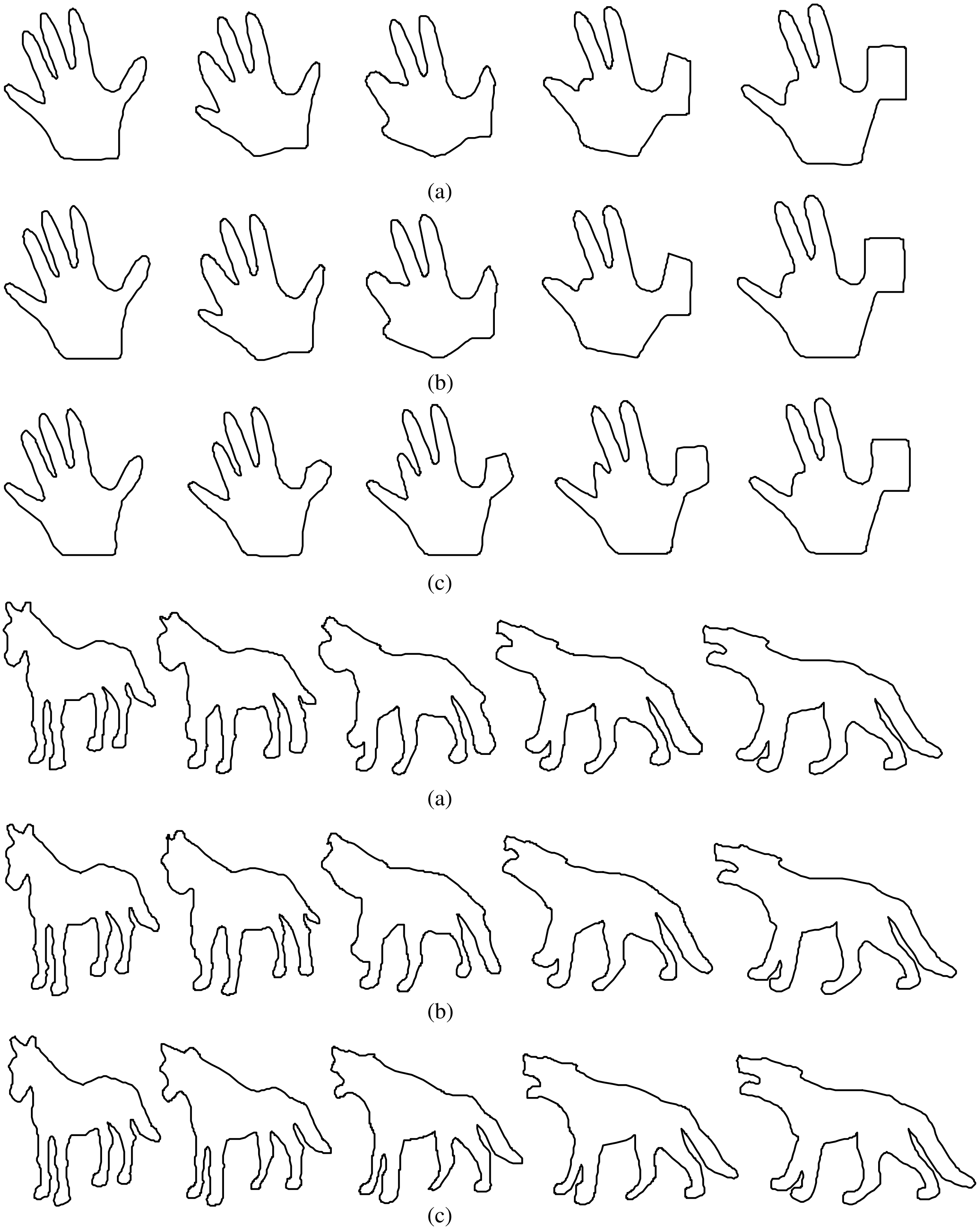


Figure 7. Comparative results of morphing approaches. (a) Approach of [5]; (b) Approach of [20]; (c) Ours.