B-morphs between b-compatible curves

Brian Whited
Georgia Institute of Technology
Atlanta, GA 30332
brian.whited@gatech.edu

Jarek Rossignac
Georgia Institute of Technology
Atlanta, GA 30332
jarek@cc.gatech.edu

ABSTRACT

We define *b-compatibility* for planar curves and propose three ball morphing techniques (*b-morphs*) between pairs of *b-compatible* curves. *B-morphs* use the automatic ball-map correspondence, proposed by Chazel et al. [11], from which they derive vertex trajectories (*Linear*, *Circular*, *Parabolic*). All are symmetric, meeting both curves with the same angle, which is a right angle for the *Circular* and *Parabolic*. We provide simple constructions for these *b-morphs* using the maximal disks in the finite region bounded by the two curves. We compare the *b-morphs* to each other and to other simple morphs (*Linear Interpolation* (*LI*), *Closest Projection* (*CP*), *Curvature Interpolation* (*CI*), *Laplace Blending* (*LB*), *Heat Propagation* (*HP*)) using seven measures of quality deficiency (*travel distance*, *distortion*, *stretch*, *local acceleration*, *surface area*, *average curvature*, *maximal curvature*). We conclude that the ratios of these measures depends heavily on the test case, especially for *LI*, *CI*, and *LB* which compute correspondence from a uniform geodesic parameterization. Nevertheless, we found that the *Linear b-morph* has consistently the shortest *travel distance* and that the *Circular b-morph* has the least amount of *distortion*.

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Two-Dimensional Graphics —*Animation*

Keywords

Morphing, Curve Interpolation, Medial Axis, Curve Averaging, Surface Reconstruction from Slices, Ball map

1. INTRODUCTION

1.1 Problem statement

A variety of techniques have been proposed for computing automatically a morph between two curves P and Q in the plane (see [28] and [2] for examples). In this paper, we present a new family of morphs, which we call the *b-morphs*, and discuss two related issues: (1) How to compare different morphing solutions and (2) How do the *b-morphs* introduced here compare to other approaches.

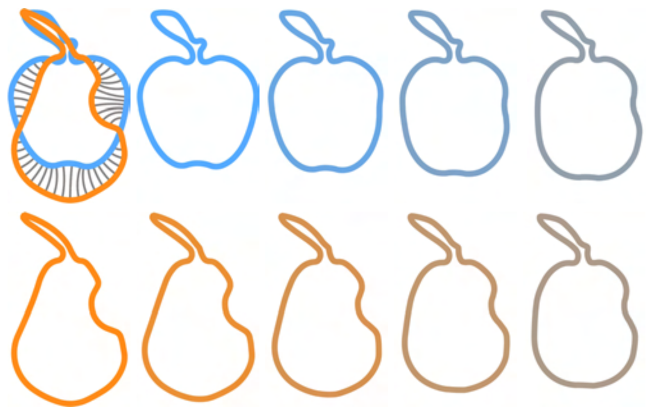

Figure 1: A morph between an apple and pear along *Circular b-morph* trajectories (top left).

1.2 Motivation and applications

Morphing is a fundamental tool in animation design where *in-between* [10] frames are produces from a sparse set of keyframes that are often designed by lead artists [48]. Although several successful attempts at automating the construction of in-between frames have been proposed [33], the artist responsible for in-betweening like to have control over correspondence and the trajectories selected landmarks or stroke end-points. These specifications are difficult to automate because they involve aesthetic judgement, style guidelines, and context semantics about the relative 3D motions of the strokes and their mutual occlusions.

Once these matching and control trajectories are given, the overall problem is naturally broken into a series of tight inbetweening taskss [38]. These are viewed as tedious and hence are a prime candidate for artist-supervised automation. In most of such tight in-between tasks, the goal is to generate a small number of intermediate frames between two reasonably simple and aligned curve segments.

At this point, it is unreasonable to ask the artist to identify a good set of candidate techniques that promise to generate acceptable morphs. Then, rather than offloading upon the artist the burden of choosing the best one in each case, one may want to compare these techniques to better assess the strength of each. This paper is a modest–although we hope useful–step in this direction. It may not be the final answer to tight in-betweening for several reasons: (1) The quantitative quality measures that we use may not reflect artistic concerns. (2) For practical reasons, we compare the proposed *b-morphs* to our simple and un-optimized implementations of candidate techniques, and not to state of the art solutions. (3) We do not take into account the broader context of the whole animation, but instead focus on interpolating only the instances of the same stroke in two consecutive key-frames. Nevertheless, we feel that the experiments described here are useful and that the conclusions we draw from them about the specific benefits of the *b-morphs* will help the reader appreciate their potential.

Furthermore, the problem (encountered in the segmentation of medical scans) of constructing a surface in 3D that interpolates between each pair of consecutive planar *cross-sections* may be solved [14] using the morphing between the projection, onto the same plane, of the two cross-section curves. This problem of surface reconstruction has been studied extensively [16][6][1][25][17].

As it was the case for tight in-betweening, our investigation of the benefit of *b-morphs* to the problem of cross-section interpolation has limitations. For example, it only considers two consecutive slices, instead of building a smooth surface through the whole series, as proposed in [7]. However, because the *b-morph* reach the interpolated contours at right angles, the projection of these trajectories on the slice plane in C^1. We expect that this property may help researches devise solutions that smooth connect surface sections generated by *b-morphs*.

Furthermore, the approach is limited to *b-compatible* curves and hence is not suited for dealing with topological changes, as discussed for example in [26].

In both applications, the quality of the morph is important as one typically favors a solution where the animation or interpolating surface is smooth and free from self-intersections [20] and unnecessary distortions. We show that when the curves are *b-compatible*, the *b-morph* always satisfies these properties.

1.3 Contributions

We propose a family of three new morphing techniques (that we call *b-morphs*) for which the correspondence and the vertex trajectories are both derived from the maximal disks and their tangential contact points with the curves.

We propose seven measures of quality inadequacy: *travel distance*, *distortion*, *stretch*, *local acceleration*, *surface area*, *average curvature*, and *maximal curvature*.

We use these measures to compare the *b-morphs* to each other and also five simple morphing techniques which we have implemented (*Linear Interpolation* (*LI*), *Closest Projection* (*CP*), *Curvature Interpolation* (*CI*), *Laplace Blending* (*LB*), *Heat Propagation* (*HP*)).

1.4 Limitation

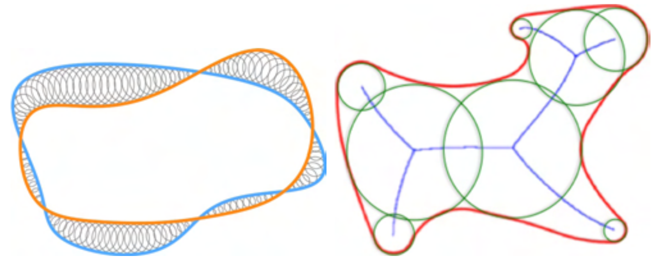

Figure 2: Maximal disks (left) and medial axis (right) with bifurcation disks shown in green

Our *b-morph* constructions assume that the two curves have been registered and are sufficient similar. We provide a formal definition of compatibility that captures these assumptions for the two situations considered here:

1. P and Q are each a simple closed loop.

2. P and Q are open curve segments and share the same two end-points.

Loosely speaking, our compatibility conditions require that each maximal disk [46] in the finite region bounded by the union of the two curves have exactly one contact point with each curve (see Fig. 2).

Where P and Q are similar but not properly registered, one may consider combining a *b-morph* with the animation of a rigid or non-rigid registration [50] or of a smooth space warp [8], as was done for image morphs [9]. Numerous solutions to the automatic registration problem have been proposed using ICP [21], automatically identified landmarks [36] [27] [37], or distortion minimizing parameterization [51] [42].

1.5 Structure of the paper

Section 2 briefly reviews prior art in curve morphing and slice interpolation. Section 3 provides a precise definition of *b-compatibility* and contrasts it with a previous discussed notion of normal compatibility. Section 4 presents our three *b-morphs* and compares their properties with the closest projection morphs. Section 5 defines the seven measures we compute and explains our strategy for sampling and for a fair integration of these measures over the set of all trajectories. Section 6 discusses our results.

2. PRIOR ART

A large variety of techniques have been investigated for the automatic generation of in-betweening frames or animations that morph between two planar curves.

We only discuss techniques that are appropriate to the tight in-betweening problem addressed here. Hence, we do not discuss the problems of registration or landmark (salient feature) identification.

First, we consider techniques that assume that the correspondence between vertices or samples on both curves is either given by the artist or computed automatically using

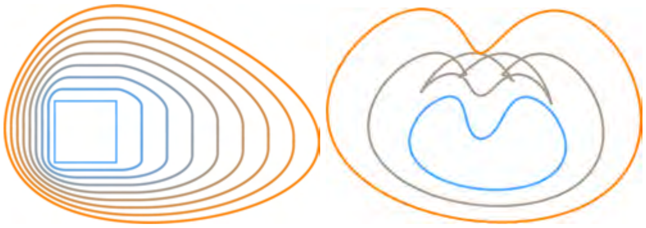

Figure 3: Example Minkowski morphs between convex (left) and non-convex (right) shapes.

uniform geodesic sampling, minimization of area or travel [25], curvature-sensitive sample [19], or optimization of matching to affine transformations extracted from an example morph [47].

If the correspondence is given, the simplest approach is to use a linear interpolation between corresponding pairs. Linear trajectories are computed between these pairs of points points on P and Q in order to produce morph curves with vertices $v_i = p_i + t(q_i - p_i)$. We include this *Linear Interpolation* (*LI*) in our benchmark set of approaches that we compare to the *b-morphs.* This naïve approach may lead to unpleasant artifacts, such as self-intersections in the intermediate frames (as for example pointed out by [44]). The *Linear Interpolation* fails to take into account the relative orientation and curvature of the curves at the corresponding points. A Poisson equation method [54] may be used to produce better vertex trajectories.

To take these into account, a popular morphing technique proposed by [43] for polygonal curves interpolates the lengths of corresponding edges and the angles at corresponding vertices and use optimization to ensure that the curve closes properly. We include a simple version of this approach, which we call *Curvature Interpolation* (*CI*) in our benchmark set. When it is applied to open curve segments, we ensure that the interpolating frames meet at end-point constraints by retrofitting them through a trivial similarity transformation (rotation, scaling, and translation).

A different approach that takes into account the relative orientation and curvature of the two curves at the corresponding samples is to compute the local coordinates of each vertex in the coordinate system defined by its neighbors on each curve. Then, the corresponding local coordinates are average linearly to produce a *desired* set of local coordinates for a given frame. Iterative techniques may be used to construct a curve that satisfies the two endpoint constraints and minimizes the discrepancy between the actual and *desired* local coordinates. Variations of these techniques have been successfully used [24]. We include a simple version of this approach, which we call *Laplace Blending* (*LB*), in our benchmark set.

Vertex trajectories and correspondences may also be solved by solving a PDE or by computing a gradient field that interpolates the two contours and then following the steepest gradient to obtain the trajectory of each point or equivalently, the in-between frames may be obtained as iso-contours of that field. A heat propagation formulation may be used to characterize the desired field[18]. We include a simple version of this approach, which we call *Heat Propagation* (*HP*), in our benchmark set.

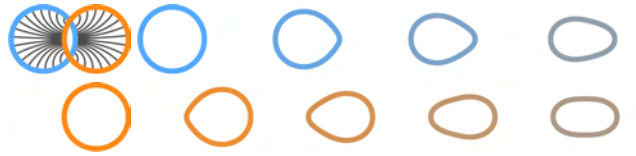

Figure 4: A morph between two offset circles along *Circular b-morph* trajectories (top left).

Several approaches for morphing closed curves use compatible triangulations [4] of their interior [52][29][2] or compatible skeletons to ensure rigidity [45][13]. Other approaches blend distance fields to both surfaces [32][17].

We separate the approaches that establish correspondence using a direct geometric criterion (as opposed to a global optimization or feature recognition as discussed above) into three categories: (1) Proximity-based, (2) Orientation-based, and (3) both Proximity- and Orientation-based.

The popular distance-based approach is the closest point projection, which to each point p on P maps a point q on Q that minimizes the distance to p. Variations of this approach are used for Iterative Closest Point (ICP) registration [15]. We include a simple version of this approach, which we call *Closest Projection* (*CP*) in our benchmark set.

An orientation-based approach is the Minkowski morph [39] and yields satisfying results, even when the shapes are not aligned (see Fig. 3 left). For smooth curves, the approach establishes a correspondence between points with the same normal. Unfortunately, as shown in Fig. 3 (right), the approach may yield surprising and self-intersecting frames when the two curves are not convex. Hence, we do not include it in our benchmark.

The *b-morphs* proposed here take into account both proximity and orientation. We discuss them in detail between and compare then against our benchmark set.

3. COMPATIBILITY

In this section, we define compatibility between curves.

A curve is *simple* if it is planar, connected, free from self-intersections, and if it is topologically closed. A simple curve is either a *loop* (no end-points) or a *stroke* (two endpoints). We consider two simple curves, P and Q.

Given a closed and regularized [49] set X, following [46], we say that a disk in X is *maximal* if it is not contained in any other disk in X and we define the *medial axis* as the closure of the union of the maximal disks in X. The *closest projection* of a point m onto a curve P is a set of points $p \in P$ for which $\|m - p\| = distance(m, P)$.

Definition 1. B-compatibility: Let P and Q be simple curves, such that $P \cap Q$ is zero-dimensional, P and Q are

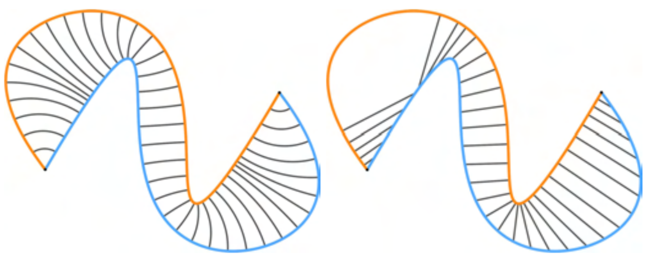

Figure 5: Two curves that are *b-compatible*, but not *c-compatible.*

b-compatible if and only if the following conditions are satisfied:

1. P and Q are either both loops or both strokes.

2. If P and Q are strokes, the endpoints of Q are identical to the endpoints of P.

3. The medial axis M of the finite and closed set X bounded by $P \cup Q$ is simple.

4. Each point $m \in M$ has a unique closest projection p on P and a unique closest projection q on Q.

We contrast *b-compatibility* with the following definition of *c-compatibility.*

Definition 2. C-compatibility: Two curves, P and Q are *c-compatible* if for every point p on any one of them, the closest projection onto the other is a single point.

A more precise definition of *c-compatibility* is discussed in [12].

The term *B-compatibility* stands for "ball-compatibility" and *c-compatibility* stands for "closest-point compatibility".

Two curves are *c-compatible* when each one can be expressed as the normal offset of the other. Two curves are defined as *b-compatible* when each can be expressed as the ball-offset of the other, i.e., as the envelope swept by a variable radius disk as it rolls on the other curve.

As shown in Fig. 5, one may easily find cases where two strokes are *b-compatible*, but not *c-compatible*. In such cases, the *b-morphs* proposed here will work, while the *Closest Projection* (*CP*) morph may not.

Let h be the *Hausdorff distance* [30] between P and Q. Recall that h is the smallest r for which $P \subset Q^r$ and $Q \subset P^r$, where the offset [41] X^r is the set of points at distance r or less from X.

We use the term *minimum feature size* (or *reach* [22]) as the set X defined as the largest r for which $F_r(X) = X$, where the *mortar* [53] $F_r(X)$ of X is the set not reachable by an open ball of radius r that does not intersect X.

Chazel et al. [11] show that if P and Q are smooth loops and their Hausdorff distance is less than the minimum feature size (mfs) of both P and Q, then P and Q are b-compatible. Note that this is a sufficient, but not necessary condition. In contrast, Chazel et al. [12] show that two curves are *c-compatible* when their minimum feature size f and their Hausdorff distance h satisfy the following relation: $h < (2 - \sqrt{2})f$.

Finally, it has been shown [11] that when two curves are *b-compatible*, their Hausdorff distance and their Fréchet distance [3] are identical.

Through the rest of the paper, we assume that P and Q are *b-compatible* and that M is their medial axis.

4. B-MORPHS

In this section, we describe the correspondence used for our *b-morphs* and present the various options for *b-morph* trajectories between corresponding pairs of points. The definitions are independent of the nature of the two curves and of their representation. We have implemented these techniques for two domains: (1) smooth (C^1) piecewise circular curves [40], and (2) relatively smooth polygons (such as those obtained through smoothing or subdivision). Our implementation on pairs of b-compatible piecewise-circular curves is numerically precise and yields the theoretically correct *b-morph*. Clearly, the implementation for polygons is not theoretically correct. Indeed, the polygonized versions of two b-compatible curves are not b-compatible, because the region they bound must have convex vertices and hence bifurcations in its medial axis. Nevertheless, when the polygonal curves are reasonably smooth and densely sampled, our polygonal algorithm computes *b-morphs* that closely approximate the *b-morphs* of the original smooth curves and are acceptable for animation or surface reconstruction. Because most other morphing schemes to which we compare our *b-morphs* work on polygonal curves, we use the polygonal *b-morph* implementation to ensure consistency.

4.1 Ball-map correspondence

Consider the maximal disk centered at point $m \in M$. The *ball-map* [11] establishes the correspondence between the closest projection p of m onto P and the closest projection of q of m onto Q. The maximal disk D centered at m touches P at p and Q at q, as shown in Fig. 6. The ball-map may be viewed as a continuous version of an approach proposed by [32] for establishing correspondences between surfaces by considering their distance fields.

A uniform sampling of the ball-map correspondence may be computed in several ways: (1) By initially computing M as the medial axis of the symmetric difference between the two curves using efficient medial axis construction techniques [23][55] and then generating the closest projections p and q for a set of uniformly spaced sample points $m \in M$; (2) By computing the radii of the maximal disks that touch p at a set of uniformly spaced samples p; or (3) By simultaneously advancing the corresponding points, p and q, until one of them has travelled from the previous sample on its curve by a prescribed geodesic distance along the corresponding curve. To ensure a fair comparison with techniques that lack the symmetry of the *b-morphs*, we will use the second (asymmetric) approach, although the first one yields the best results.

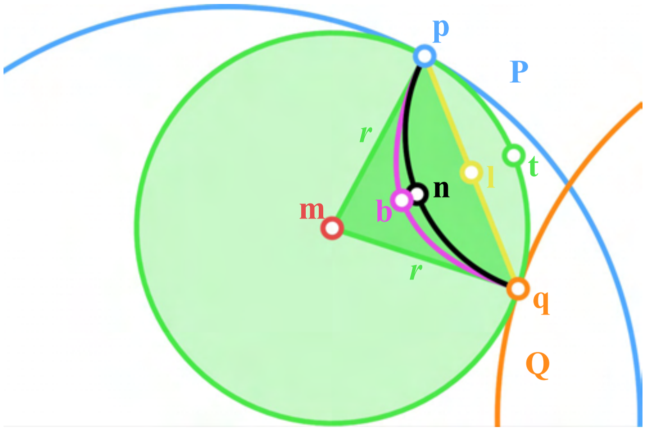

Figure 6: To obtain the point q on Q that corresponds, through the ball-map, to point p on P, we compute the smallest positive r such that $m = p+rN_p(q)$ is at distance r from Q and return its closest projection q on Q. Point m is on the median M (red) and defines the center of the circle tangent at both p and q. The *Circular* (black) and *Parabolic* (purple) *b-morph* trajectories are defined by the inscribing isosceles triangle (p, m, q). The *Linear b-morph* trajectory is the line segment (p, q).

The details of the construction of this mapping for the case when P and Q are piecewise-circular and when they are polygonal approximations of smooth curves are provided in the Appendix.

4.2 B-morph trajectories

For each maximal disk, we consider five *paths* (curve segments) from p to q (Fig 2):

***Hat*:** The broken line segment from p to m to q (Fig 6 green).

***Linear*:** The straight line segment from p to q (Fig 6 yellow).

***Tangent*:** The shorter of the two circular arc segments of the boundary of D that joins p and q (Fig 6 green).

***Circular*:** The circular arc segment that is orthogonal to P at p and to Q at q (Fig 6 black).

***Parabolic*:** The parabolic arc segment that is orthogonal to P at p and to Q at q (Fig 6 violet).

The *Circular* and *Parabolic* paths are trivially defined by their enclosing isosceles triangle (Δpmq). For example, the *Parabolic* path is the quadratic Bézier curve with control vertices p, m and q and the center of the circle supporting the *Circular* path is the intersection of the tangent to P at p and the tangent to Q at q.

All paths, including the *Linear* path, are symmetric in that the angles where they meet P and Q are the same. Swapping the role of P and Q does not affect these segments. Hence, the *b-morphs* derived here are *symmetric* and may be inverted easily by swapping the role of P and Q and reversing time.

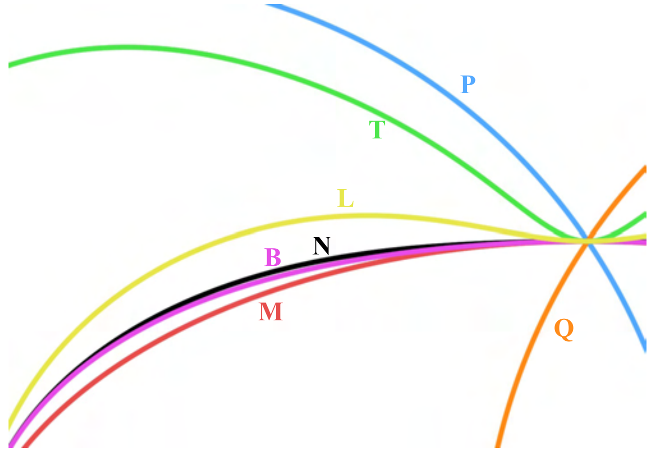

Figure 7: The associated average curves for the various *b-morph* constructions.

Let l be the midpoint of the *Linear* path and let L be the set of all points l. Note that L is the midpoint locus proposed by Asada and Brady [5]. Let t be the midpoint of the *tangent* path and T be the set of all points t. Note that T is the PISA proposed by Layton [34] as a variation of the medial axis. Let n be the midpoint of the *Circular* path and N the set of all points n. Let b be the midpoint of the *Parabolic* path (quadratic B-spline) and B be the set of all points b. The construction of these 4 points, along with m is illustrated in Fig. 6.

The curves M, L, T, N and B usually differ from one another, but may all be viewed as *averages* of P and Q. The are shown superimposed in Fig. 7.

A *b-morph* advances each point p according to uniform arc-length parameterization along one of the aforementioned paths. A result, sampled at 7 intermediate points along each path of the *Circular b-morph* is shown in Fig. 4.

5. MEASURES

We first discuss how we sample space and time. Then, we provide details of the measures used here to compare morphs.

5.1 Measure normalization

Three of the studied morphs (*Linear Interpolation*, *Curvature Interpolation*, and *Laplace Blending*) assume a given correspondence. For simplicity, we use a uniform arc-length sampling to produce the same number of uniformly distributed samples on each curve. The three *b-morphs* use the ball-map correspondence. The other morphs compute their own correspondence. This sampling disparity makes it difficult to compute measures for a fair comparison.

Consider for example the problem of measuring the average *travel distance*. This should be the integral of travel distances. The problem is how to fairly select the integration element. If for example we use *LI* morph, then the average distance measured for a set of uniformly distributed samples will depend on whether we start form P or Q. Since the average *travel distance* is a property of the mapping, and not the sampling, a measure that so blatantly depends on the sampling is clearly incorrect.

To overcome this problem, each reported measures is the average of two measures, one computed by sampling P and one computed by sampling Q. For the first measure, we sample the departure curve P using a dense set of samples that are uniformly distributed on each curve so as to be separated by a prescribed geodesic distance u. For each sample p_i on P, we compute the corresponding point q_i on the arrival curve Q so that q_i is the image of p_i by the mapping associated with the particular morphing scheme. We compute a measure m_i associated with the trajectory from p_i to q_i and the associated weight $w_i = (d(p_i-1, p_i) + d(q_{i-1}, q_i) + d(p_i, p_{i+1}) + d(q_i, q_{i+1}))/4$. Then, we report the normalized weighted average $(\Sigma w_i a_i)/(\Sigma w_i)$. For the second measure, we sample the arrival curve Q as before using the same geodesic distance u. For each sample q_i on Q, we compute the corresponding point p_i on the departure curve P, so that q_i is the image of p_i by the mapping associated with the particular morphing scheme. Then, we proceed as above.

5.2 Morph Measures

We have implemented the following seven measures.

Travel distance. For each sample p_i, we measure m_i as the arc length of the trajectory to the corresponding point q_i. Then, as explained in Section 5.1, we report the weighted average of these from P to Q and vice-versa.

Stretch. We define stretch $S(P,Q)$ as the average of the integral over time of the stretch factor for an infinitesimal portion of the curve. We compute its discrete approximation as follows. Let p and p' be consecutive samples on P. Let $L(p,t)$ be the length of the segment of $P(t)$ between $p(t)$ and $p'(t)$. We compute $S(P,Q)$ as

$$\begin{aligned} S(P,Q) = \quad & \textstyle\sum_{t\in[0,1-\epsilon]} \left(\sum_{p\in P} |L(p,t+\epsilon) - L(p,t)|\right) \\ + \quad & \textstyle\sum_{t\in[0,1-\epsilon]} \left(\sum_{q\in Q} |L(q,t+\epsilon) - L(q,t)|\right) \end{aligned}$$

Acceleration. *Acceleration*, or unsteadiness [50], is defined as the derivative of the expression of velocity in the local, time-evolving frame, and which measures the lack of steadiness of the motion.

To compute *acceleration*, let p_t denote the position of a sample p at a time t. We approximate the instantaneous velocity of p_t by the vector $p_t p_{t+\epsilon}$. For each such velocity on a morph trajectory, we compute two barycentric coordinate vectors $B_L(p_t p_{t+\epsilon})$ and $B_R(p_t p_{t+\epsilon})$ relative to the left and right neighboring triangles L_t and R_t as shown in Fig. 8.

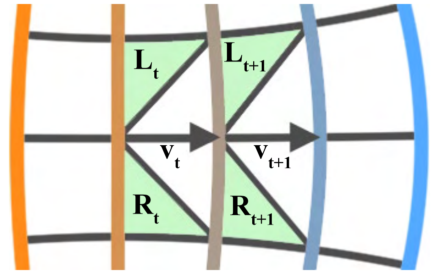

Figure 8: Computation of *acceleration* (steadiness) for a given vector v of the morph trajectory is computed relative to the neighboring triangles (green).

The steadiness at a point p_t is then computed as:

$$\begin{aligned} g_t = \quad & \tfrac{1}{2}\|B_L(p_{t-\epsilon}p_t) - B_R(p_{t-\epsilon}p_t)\| \\ + \quad & \tfrac{1}{2}\|B_L(p_t p_{t+\epsilon}) - B_R(p_t p_{t+\epsilon})\| \end{aligned}$$

We compute the acceleration measure m_i as the sum of the g_i terms over the trajectory of each point p_i and report their weighted average, as described above.

Distortion. At each point along the evolving curve and at each time, the amount of distortion is proportional to $1/cos\theta$, where θ is the angle between the direction of travel and the normal to the evolving curve.

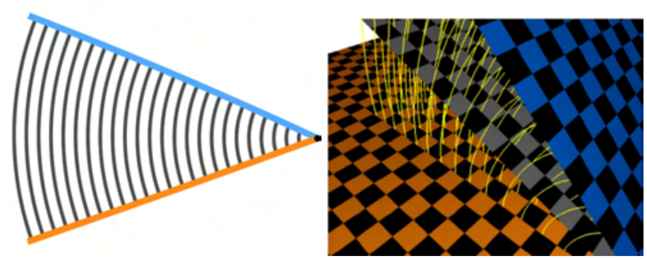

Figure 9: The *b-morph* produces a pure rotation with zero distortion between linear segments in 2D (left) and planar segments (in 3D).

Let p and p' be consecutive samples on P and $L(p,t)$ define the length of the segment pp'. Let $V(p,t)$ define the length of the segment $p_t p_{t+\epsilon}$. We compute

$$R = \sum_{t\in[0,1-\epsilon]} \left(\frac{\frac{1}{2}(L(p,t) + L(p,t+\epsilon)) \cdot \frac{1}{2}(V(p,t) + V(p',t)}{Area(p_t p'_t p'_{t+\epsilon} p_{t+\epsilon})} \right)$$

It was shown in [11] that (1) the *b-morph* is a $C^{k-1} isotopy$ when the inputs are C^k for $k \geq 2$ and (2) that the *Circular b-morph* is free from distortion when morphing between line segments (in 2D) or planar portions (in 3D) of P and Q (Fig. 9).

5.3 Mesh measures

In addition to the 2D measures, we also present results of 3D measures of surface area and also average and maximum squared mean curvature [35] of the resulting triangle mesh

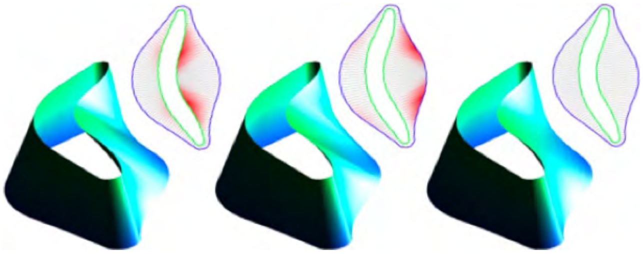

Figure 10: We show the slice-interpolating surface reconstructed using a *Closest Projection* morph from-green-to-blue (left), the reverse *Closest Projection* morph from-blue-to-green (center), and the symmetric *Circular b-morph* (right) which appears smoother. The amount of local *distortion* is shown in red on the 2D drawings.

surfaces. In applications of surface reconstruction from 2D planar contours, minimizing the surface area and smoothness of the resulting reconstruction is often desirable (see Fig. 10).

6. RESULTS

We first compare the *b-morphs* to our benchmark set‘ using two different test cases, as shown in Figures 11 and 12. Then, we compare two of the *b-morphs* to the best two other morphs (Laplace and Heat) on a test case between an apple and a pear. The *CP* morph is not used since it is incompatible for the cases shown here (Fig. 5).

The first test case (Fig. 11) shows a morph between two offset circles. The *b-morphs* work even if the two curves intersect. Some of the benchmark morphs do not. Hence, to accommodate their limitations, we have split the curves at their intersections and performed morphs independently on corresponding strokes.

Our experiments demonstrate that the average *travel distance* is the shortest when using the *Linear b-morph* and that the *Circular b-morph* has the least amount of distortion. The *HP* morph is the closest in terms of appearance and measure to the *Circular b-morph*.

The second test case (Fig. 12) shows a set of symmetric ‘S’ shaped curves. This example highlights the strength of the morphs which compute their own correspondence (*HP*, *b-morphs*). The other results, which define correspondence through uniform arc-length parameterization exhibit extreme distortion and travel lengths and produce self-intersections with the original curves. Again, the *HP* morph is closest to the family of *b-morphs* in terms of measure and appearance.

The final test case (Fig. 13) uses contours representing an apple and a pear. We show the best four morphs (*Linear b-morph*, *Circular b-morph*, *Heat Propagation* and *Laplace Blending*) and compare their measures. The measures for these four are quite similar. Travel distance and distortion are still minimized for the *Linear* and *Circular b-morphs*, respectively. The *LB* approach comes wins in terms of *acceleration*, *stretch* and *curvature*.

7. ACKNOWLEDGEMENTS

This work has been partially supported by NSF grant 0811485 and by the Walt Disney Corporation.

8. CONCLUSION

We have proposed a family of morphs between curves which are *b-compatible*. All are based on variations of the medial axis construction. We have compared them to one another and to several other simple morphs. We used four measures of morph quality in our comparison, as well as surface measures for comparing them as surface reconstruction techniques.

Although the Heat morph produces very similar results to the *b-morph*, it has the disadvantage of requiring several steps, including rasterization to a grid, PDE solve, and point tracking. Due to the rasterization, it would also be sensitive to inputs which intersect or self-intersect. This method would be desirable for more extreme cases that are not *b-compatible*.

We conclude that for the cases of *b-compatible* shapes, the *b-morphs* offer a precise and desirable result in terms of *distortion*, *travel distance*, as well as *curvature*. The *Circular b-morph* is guaranteed to produce morph curves of C^{k-1} continuity for inputs of C^k for $k \geq 2$.

9. REFERENCES

[1] S. Akkouche and E. Galin. Implicit surface reconstruction from contours. *Vis. Comput.*, 20(6):392–401, 2004.

[2] M. Alexa, D. Cohen-Or, and D. Levin. As-rigid-as-possible shape interpolation. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 157–164, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[3] H. Alt, C. Knauer, and C. Wenk. Bounding the fréchet distance by the hausdorff distance. In *Proc. 17th European Workshop on Computational Geometry*, pages 166–169, 2001.

[4] B. Aronov, R. Seidel, and D. Souvaine. On compatible triangulations of simple polygons. *Comput. Geom. Theory Appl.*, 3(1):27–35, 1993.

[5] H. Asada and M. Brady. The curvature primal sketch. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(1):2–14, 1986.

[6] G. Barequet and M. Sharir. Piecewise-linear interpolation between polygonal slices. *Comput. Vis. Image Underst.*, 63(2):251–272, 1996.

[7] G. Barequet and A. Vaxman. Nonlinear interpolation between slices. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 97–107, New York, NY, USA, 2007. ACM.

[8] A. H. Barr. Global and local deformations of solid primitives. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 21–30, New York, NY, USA, 1984. ACM.

[9] T. Beier and S. Neely. Feature-based image metamorphosis. *SIGGRAPH Computer Graphics*, 26(2):35–42, 1992.

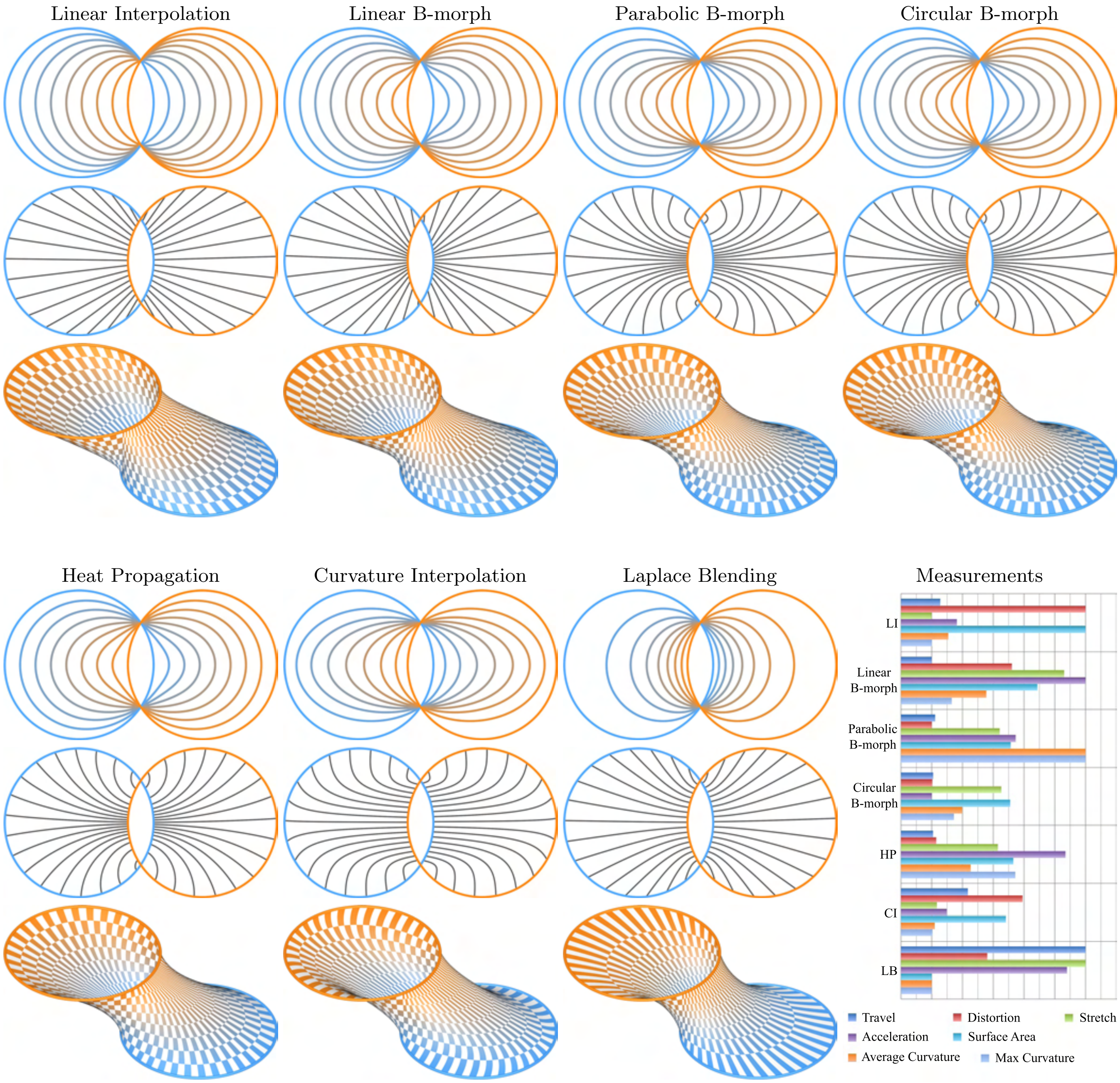

Figure 11: Morph results for offset circles, showing the morph curves (top), the morph trajectories (middle) and the surface created by linearly interpolating the morph curves along the z-axis (bottom). Also displayed are the measures for each (bottom-right). Each measure is normalized independently for easy comparison.

[10] E. Catmull. The problems of computer-assisted animation. *SIGGRAPH Comput. Graph.*, 12(3):348–353, 1978.

[11] F. Chazal, A. Lieutier, J. Rossignac, and B. Whited. Ball-map: Homeomorphism between compatible surfaces. *International Journal of Computational Geometry and Applications*, 2009 (to appear).

[12] F. Chazel, A. Lieutier, and J. Rossignac. Orthomap: Homeomorphism-guaranteeing normal-projection map between surfaces. Technical Report GIT-GVU-04-28, Georgia Institute of Technology, Graphics, Visibility, and Usability Center, 2004.

[13] W. Che, X. Yang, and G. Wang. Skeleton-driven 2d distance field metamorphosis using intrinsic shape parameters. *Graphical Models*, 66(2):102–126, 2004.

[14] S. E. Chen and R. E. Parent. Shape averaging and it's applications to industrial design. *IEEE Comput. Graph. Appl.*, 9(1):47–54, 1989.

[15] X. Chen, M. R. Varley, L.-K. Shark, G. S. Shentall, and M. C. Kirby. An extension of iterative closest

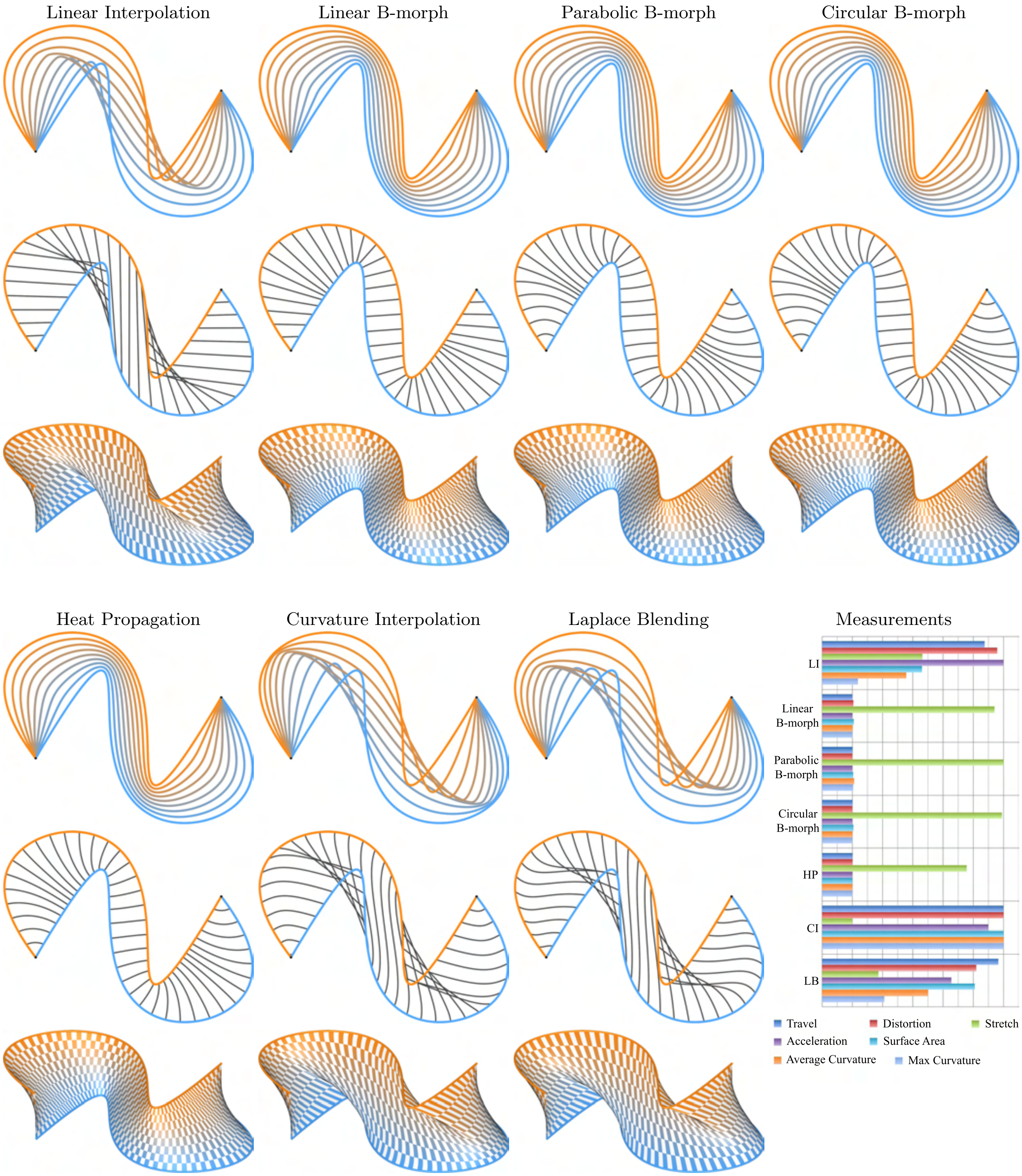

Figure 12: Morph results for a set of 'S'-shaped curves, showing the morph curves (top), the morph trajectories (middle) and the surface created by linearly interpolating the morph curves along the z-axis (bottom). Also displayed are the measures for each (bottom-right). Each measure is normalized independently for easy comparison. Note that some of the morphs do not remain within the bounds of the inputs.

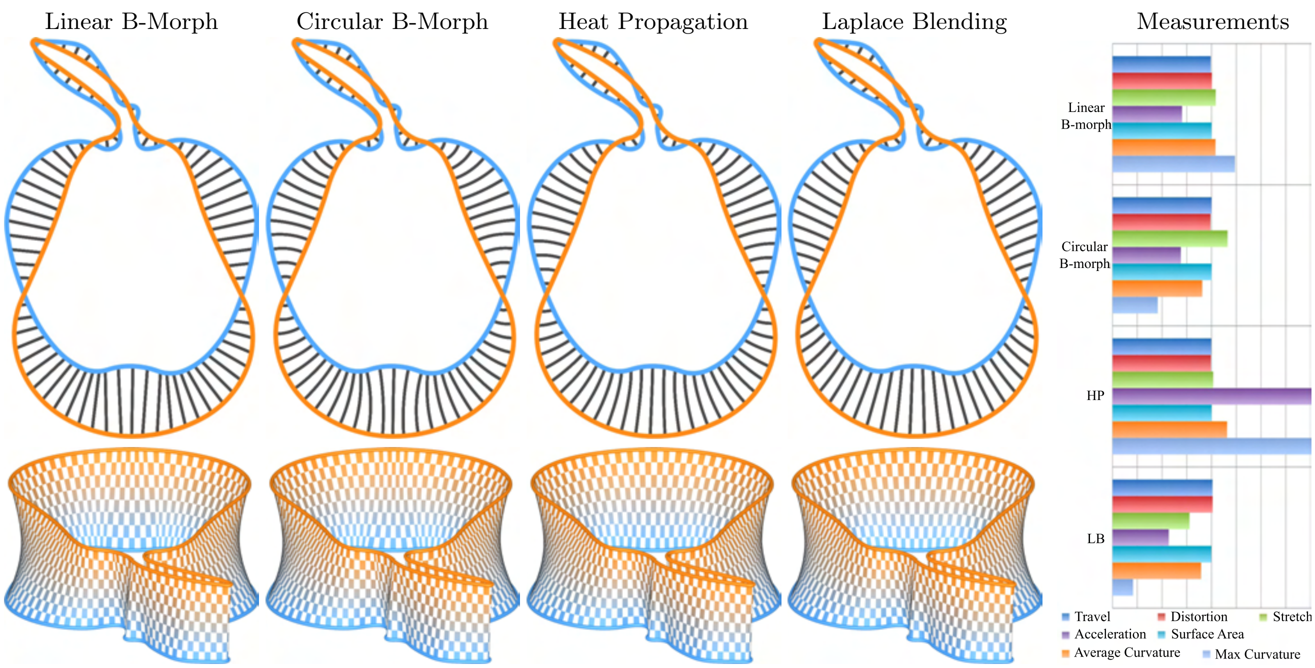

Figure 13: Morph results for a set of apple and pear shaped curves, showing the morph curves (top), the morph trajectories (middle) and the surface created by linearly interpolating the morph curves along the z-axis (bottom). Also displayed are the measures for each (bottom-right). Each measure is scaled independently for easy comparison.

point algorithm for 3d-2d registration for pre-treatment validation in radiotherapy. In *MEDIVIS '06: Proceedings of the International Conference on Medical Information Visualisation–BioMedical Visualisation*, pages 3–8, Washington, DC, USA, 2006. IEEE Computer Society.

[16] S.-W. Cheng and T. K. Dey. Improved constructions of delaunay based contour surfaces. In *SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 322–323, New York, NY, USA, 1999. ACM.

[17] D. Cohen-Or, A. Solomovic, and D. Levin. Three-dimensional distance field metamorphosis. *ACM Trans. Graph.*, 17(2):116–141, 1998.

[18] G. Cong and B. Parvin. A new regularized approach for contour morphing. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, pages 458–463 vol.1, 2000.

[19] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan. Curve matching for open 2d curves. *Pattern Recogn. Lett.*, 30(1):1–10, 2009.

[20] A. Efrat, S. Har-Peled, L. J. Guibas, and T. M. Murali. Morphing between polylines. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 680–689, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[21] E. Ezra, M. Sharir, and A. Efrat. On the icp algorithm. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 95–104, New York, NY, USA, 2006. ACM.

[22] H. Federer. *Geometric Measure Theory*. Springer Verlag, 1969.

[23] M. Foskey, M. Lin, and D. Manocha. Efficient computation of a simplified medial axis. In *ACM Symposium on Solid Modeling and Applications*, pages 96–107, 2003.

[24] H. Fu, C.-L. Tai, and O. K. Au. Morphing with laplacian coordinates and spatial temporal texture. In *Proc. Pacific Graphics '05*, pages 100–102, 2005.

[25] H. Fuchs, Z. M. Kedem, and S. P. Uselton. Optimal surface reconstruction from planar contours. *Commun. ACM*, 20(10):693–702, 1977.

[26] N. C. Gabrielides, A. I. Ginnis, P. D. Kaklis, and M. I. Karavelas. G1-smooth branching surface construction from cross sections. *Comput. Aided Des.*, 39(8):639–651, 2007.

[27] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann. Robust global registration. In *SGP '05: Proceedings of the third Eurographics symposium on Geometry processing*, page 197, Aire-la-Ville, Switzerland, Switzerland, 2005. Eurographics Association.

[28] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping and morphing of graphical objects*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

[29] H. Guo, X. Fu, F. Chen, H. Yang, Y. Wang, and H. Li. As-rigid-as-possible shape deformation and interpolation. *J. Vis. Comun. Image Represent.*, 19(4):245–255, 2008.

[30] M. Guthe, P. Borodin, and R. Klein. Fast and accurate hausdorff distance calculation between meshes.

Journal of WSCG, 13(2):41–48, February 2005.
[31] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *Visualization and Computer Graphics, IEEE Transactions on*, 6(2):181–189, Apr-Jun 2000.
[32] R. Klein, A. Schilling, and W. Straβer. Reconstruction and simplification of surfaces from contours. In *PG '99: Proceedings of the 7th Pacific Conference on Computer Graphics and Applications*, page 198, Washington, DC, USA, 1999. IEEE Computer Society.
[33] A. Kort. Computer aided inbetweening. In *NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*, pages 125–132, New York, NY, USA, 2002. ACM.
[34] M. Leyton. *Symmetry, Causality, Mind.* MIT Press, 1992.
[35] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. Technical report, CalTech.
[36] F. Mokhtarian, S. Abbasi, and J. Kittler. Efficient and robust retrieval by shape content through curvature scale space. In *Proc. International Workshop IDB-MMSÕ96*, pages 35–42, 1996.
[37] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(11):1832–1837, 2005.
[38] W. T. Reeves. Inbetweening for computer animation utilizing moving point constraints. *SIGGRAPH Comput. Graph.*, 15(3):263–269, 1981.
[39] J. Rossignac and A. Kaul. Agrels and bips: Metamorphosis as a bézier curve in the space of polyhedra. *Comput. Graph. Forum*, 13(3):179–184, 1994.
[40] J. Rossignac and A. A. G. Requicha. Piecewise-circular curves for geometric modeling. *IBM J. Res. Dev.*, 31(3):296–313, 1987.
[41] J. R. Rossignac and A. A. Requicha. Offsetting Operations in Solid Modelling. *Computer Aided Geometric Design*, 3:129–148, 1986.
[42] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 870–877, New York, NY, USA, 2004. ACM.
[43] T. W. Sederberg, P. Gao, G. Wang, and H. Mu. 2-d shape blending: an intrinsic solution to the vertex path problem. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 15–18, New York, NY, USA, 1993. ACM.
[44] T. W. Sederberg and E. Greenwood. A physically based approach to 2–d shape blending. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 25–34, New York, NY, USA, 1992. ACM.
[45] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. *IEEE Comput. Graph. Appl.*, 15(2):44–50, 1995.
[46] K. Siddiqi and S. Pizer. *Medial Representations: Mathematics, Algorithms and Applications.* Springer, 2008. 450pp. In Press.
[47] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-based inverse kinematics. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 488–495, New York, NY, USA, 2005. ACM.
[48] F. Thomas and O. Johnston. *The Illusion of Life: Disney Animation.* Disney Editions, revised edition, 1995.
[49] R. Tilove. Set membership classification: A unified approach to geometric intersection problems. *Computers, IEEE Transactions on*, C-29(10):874–883, Oct. 1980.
[50] A. Vinacua and J. Rossignac. Sam: Steady affine morph. *Submitted for review.*, 2009.
[51] S. Wang, Y. Wang, M. Jin, X. Gu, and D. Samaras. 3d surface matching and recognition using conformal geometry. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2453–2460, Washington, DC, USA, 2006. IEEE Computer Society.
[52] Y. Weng, W. Xu, Y. Wu, K. Zhou, and B. Guo. 2d shape deformation using nonlinear least squares optimization. *Vis. Comput.*, 22(9):653–660, 2006.
[53] J. Williams and J. Rossignac. Tightening: curvature-limiting morphological simplification. In *ACM Symposium on Solid and Physical Modeling*, pages 107–112, 2005.
[54] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 267–274, New York, NY, USA, 2005. ACM.
[55] Y. Yang, O. Brock, and R. N. Moll. Efficient and robust computation of an approximated medial axis. In *SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 15–24, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

APPENDIX

A. ADDITIONAL DETAILS

A.1 Case of partly overlapping curves

In Section 3, we state that one of the requirements for *b-compatibility* is that the $P \cap Q$ is zero dimensional. This constraint is overcome by removing sections of the curves that overlap, breaking them into sub-curves and computing morphs on them. This is because the overlapping sections of the curves would be static through time and the correspondence is already explicitly defined.

A.2 Details of the B-morph construction

The contributions reported here are primarily theoretical and independent of dimension, representation, and implementation. However, to convince the reader that the *b-morph* provides a practical solution, we include here the details of an exact implementation (Expect for numerical round-off errors) for the case of piecewise-circular curves [40] in 2D, where P and Q are each a series of smoothly connected circular-arc edges.

Now assume $p \notin Q$. We compute the corresponding point q from p as follows. Consider the parameterized offset point $m = rN_P(p)$, whose distance from p is defined by the parameter r. Here, we have oriented $N_P(p)$ so that it points

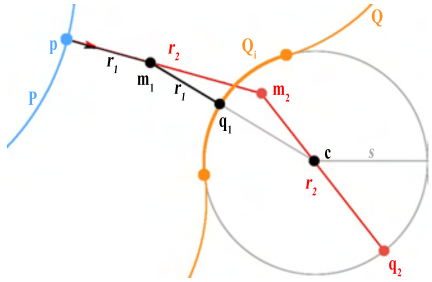

Figure 14: Computing r, m and q from p for a circular arc Q_i.

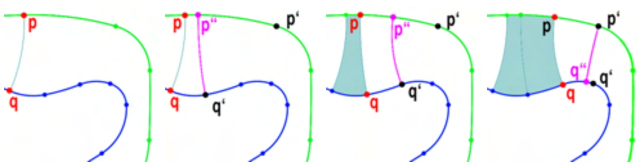

Figure 15: Lacing steps that divide the gap into slabs

towards the interior of the gap. m is the center of a circle of radius r that is tangent to P at p. We want to compute the smallest positive r for which m is at distance r from Q, and hence for which the circle is tangent to Q. Note that when P and Q are *b-compatible*, to each p corresponds a unique point q. The set of points m is the *median* of the two shapes.

First consider a circular edge Q_i of Q with center c and radius s (Fig. 14). We compute r_1 and r_2 as the roots $(s^2 - cp^2)/(2N_P(p) \cdot cp \pm 2s)$ of $cm^2 = (r \pm s)^2$ and keep the smallest positive solution for which the ball-map (q_1 or q_2) lies in Q_i.

We apply the above approach to all edges Q_i of Q. We compute the r-value for a circle supporting each edge, compute the corresponding candidate point q on the circle, discard it if it falls outside of the edge, and select amongst the retained (r, q) pairs with the smallest r-value.

We assume that P and Q are *b-compatible*. There is exactly one (r, q) pair for each point $p \in P$. The above process computes the *b-morph* correspondence for any desired sampling of P or Q.

To accelerate the computation of the *b-morph* and produce a sampling-independent representation from which different samplings densities can be quickly derived, we perform a "lacing" process (Fig. 15), to split the gap into *slabs*, each bounded by 4 circular arcs: one being a segment of an edge of P, one being an edge-segment of Q, and two being *Circular b-morph* trajectories from a vertex of P or Q to its image on the other curve.

To perform the lacing, we first pick a vertex $p \in P$, where two edges of P meet and compute its image $q \in Q$ as described above. Then, we perform a synchronized walk to "lace" the gap, one vertex of P or Q at a time. At each step, p is the start of an edge-segment P_i of P not yet laced and q is the start of an edge-segment Q_k of Q not yet laced. Let p' be the end of P_i and q' be the end of Q_k. Let q'' be the corresponding point for p' and p'' be the corresponding point for q'. If p'' falls on P_i, we record that the edge-segment $[q, q']$ of Q_k maps to the edge-segment $[p, p'']$ of P_i, close the current slab with the trajectory from q' to p'', and set p to p'' and q to q' to continue the lacing process, as shown in Fig. 15.

This lacing process splits the edges of P and Q into edge-segments and establishes a bijective mapping between edge-segments of P and edge-segments of Q that bound the same slab. The cost of this pre-computation is $O(n)$ in the number of edges in P and Q. It can be performed in real-time, as the curves are edited, which is convenient for the interactive design of a 2D morph.

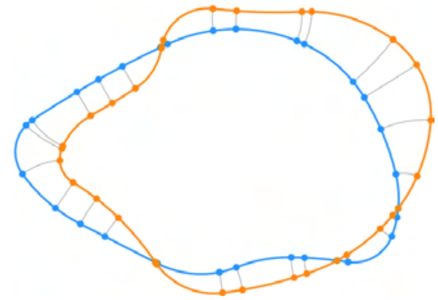

Figure 16: Lacing splits the gap into slabs. Each one is bounded by 4 circular arcs (2 edge-segments and 2 trajectories) and defines a *b-morph* from an edge-segment of P to an edge segment of Q.

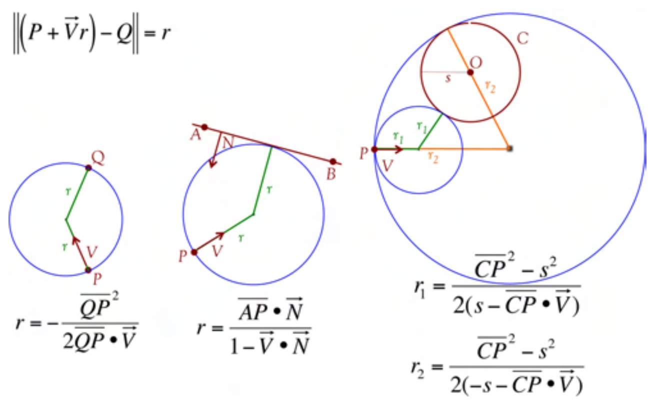

Figure 17: The equations for computing the ball-map radius r given an initial point P and direction (normal) V for mapping to a point Q (left), an edge AB (center), and a circle C (right).