Metamorphosis of Planar Parametric Curves via Curvature Interpolation

Tatiana Surazhsky* and Gershon Elber†
Technion, Israel Institute of Technology, Haifa 32000, Israel.

Abstract

This work considers the problem of metamorphosis interpolation between two freeform planar curves. Given two planar parametric curves, the curvature signature of the two curves is linearly blended, yielding a gradual change that is not only smooth but also employs intrinsic curvature shape properties, and hence is highly appealing. In order to be able to employ this curvature blending, we present a constructive scheme to derive curvature signatures of parameter curves. Additionally, we propose a scheme to reconstruct a curve from its curvature signature.

Keywords: freeform curves, metamorphosis, animation, curvature analysis.

1 Introduction

Morphing/metamorphing or the gradual and continuous transformation of one shape into another is a topic of great importance in computer graphics. Solutions to the metamorphosis problem have been presented under many domains, including two-dimensional images [2, 9, 20], planar polygons and polylines (i.e. piecewise linear curves) [10,14,15,17,18], polyhedra [1], free form curves [5,13,16], and even volumetric representations [6, 11].

The solution to the metamorphosis problem necessitates the resolution of two independent tasks. The first requires the derivation of a correspondence between the beginning and end objects. The second must deal with a proper continuous and appealing interpolation between the two shapes. In the context of piecewise linear planar curves (polylines or polygons), the correspondence problem is typically defined as a vertex correspondence problem. In other words, each vertex in the first key polygon is matched against one vertex (or more) in the second key polygon and vice versa (see [15] for an example). For higher order polynomial and rational planar curves, matching can be established as an optimization problem. For example, in [5] one of the two key curves is reparameterized so that the tangent fields of the two curves are made as parallel as possible.

The second problem, as yet lacking any better solution, is solved in many cases via a linear interpolation approach. This approach has a fundamental flaw that may result in intermediate results that are counter-intuitive, distorted, shrunken or even self-intersecting. In [14, 15], the bending and stretching energy is

*Center for Graphics and Geometric Computing, Applied Mathematics Department, E-mail: tess@cs.technion.ac.il
†Center for Graphics and Geometric Computing, Faculty of Computer Science, E-mail: gershon@cs.technion.ac.il

evaluated and employed to optimize the interpolation between piecewise linear curves. Unfortunately, and while typically achieving a more appealing metamorphosis sequence, the work of [14, 15] cannot guarantee self-intersecting free intermediate results.

The same approach is used in [16] for piecewise polynomial curves. The algorithm pesented in [16] linearly interpolates the B-spline curves that are matched by minimizing bending, stretching and kinking work functions in corresponding knot values of the key curves. The authors point to other interpolation options for the matched B-spline curves, such as [14].

As stated, linear interpolation of the control polygons and control meshes of freeform curves and surfaces is a common practice. Interestingly, linear interpolation of the derivatives of the two beginning and end freeforms yields no better results. Let $C_1(r)$ and $C_2(r)$ be two sufficiently differentiable parametric curves that share the same domain, and denote by $C^{(d)}$ and $\int^{(d)} C$ the d'th derivative and integral of C, respectively. Then, the intermediate curve $C(r)$ at time t that resulted from interpolating the d'th derivatives of $C_1(r)$ and $C_2(r)$ equals

$$\begin{aligned} C(r) &= \int^{(d)} C_1^{(d)}(r)(1-t) + C_2^{(d)}(r) t dt \\ &= (1-t)\int^{(d)} C_1^{(d)}(r) dt + t \int^{(d)} C_2^{(d)}(r) dt \\ &= (1-t)C_1(r) + tC_2(r), \end{aligned} \tag{1.1}$$

up to the integrations' constants. Specifically, interpolating the second order derivatives as an approximation to curvature interpolation yields little advantage and is identical to the interpolation of the curves themselves.

In this work, we seek a smooth and appealing interpolation scheme between two freeform parametric curves. By linearly interpolating intrinsic curvature properties, the result is far more appealing and intuitive than the regular linear interpolation scheme of the curve or its derivatives.

The fundamental theorem of differential geometry of curves shows us that the curvature field, $\kappa(s)$, of a planar curve parameterized by arc-length $C(s)$, fully prescribes it [3] up to a rigid motion transformation. In Section 2, we show how can one uniquely derive the curvature signature of a planar curve and more importantly, how can one reconstruct the curve $C(s)$ from its curvature signature $\kappa(s)$.

The rest of the paper is organized as follows. Following the curvature reconstruction scheme that is discussed in Section 2, we present in Section 3 a metamorphosis scheme that interpolates two curvature signatures. In Section 4 we present examples of metamorphosis sequences between freeform curves employing the proposed scheme, and finally, we conclude in Section 5.

2 Curve Reconstruction Using Curvature Signatures

Let $C(s)$ be an arc-length planar parametric curve. Then $C''(s) = \kappa(s)N(s)$, $N(s)$ being the unit normal field of $C(s)$. The fundamental theorem of differential geometry of curves [3] states that $\kappa(s)$ fully prescribes planar curve $C(s)$ up to a rigid motion transformation. That is, two planar curves share the same curvature signature $\kappa(s)$ if and only if one is a rigid motion transformation of the other.

In Section 2.1, we consider the derivation of $\kappa(s)$ from $C(s)$ while in Section 2.2, the reconstruction of $C(s)$ from $\kappa(s)$ is investigated.

2.1 Computation of the Curvature Signature

Given a non arc-length planar curve, $C(t) = (x(t), y(t))$, the derivation of the curvature field of $\kappa(t)$ is fairly simple [3]:

$$\kappa(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{(x'^2(t) + y'^2(t))^{3/2}}, \tag{2.1}$$

assuming $C(t)$ is regular or $||C'(t)|| \neq 0$.

Polynomial and rational forms cannot, in general, be reparameterized as arc-length parametric forms. Yet, one can arbitrary closely approximate $t(s)$, a reparameterization function from t to the arc-length parameter s. In [8], an algorithm is presented that constructs an arc-length polynomial approximation to a given polynomial curve. This algorithm can be made arbitrarily precise, using refinement [4].

Hence, methods to approximate $t(s)$ are both known and tractable. Once $t(s)$ is approximated to within a certain tolerance, $\kappa(s) = \kappa(t(s))$ can be evaluated as a composition [8] between piecewise polynomial functions, to yield

$$\kappa(s) = x'(s)y''(s) - x''(s)y'(s), \tag{2.2}$$

because $(x'^2(s) + y'^2(s)) = 1$. The composition of polynomials [8] can be computed analytically and has a fixed complexity that depends on the orders of the two curves and lengths of their control polygons.

2.2 Reconstruction of a Curve from its Curvature Signature

The more difficult question we are about to examine is how can one reconstruct $C(s)$, given $\kappa(s)$. From differential geometry, we know that

$$C'(s) = T(s), \qquad C''(s) = T'(s) = \kappa(s)N(s),$$

where $T(s)$ and $N(s)$ are the unit tangent and normal fields of $C(s)$. $T(s) = (x'(s), y'(s))$ is a unit size vector and hence is always on the unit circle. Let θ be the angle between $T(s)$ and the x-axis,

$$\theta(s) = tan^{-1}\left(\frac{y'(s)}{x'(s)}\right)$$

and consider $\theta'(s)$,

$$\begin{aligned}
\theta'(s) &= \left(tan^{-1}\left(\frac{y'(s)}{x'(s)}\right)\right)' \\
&= \frac{1}{1+\left(\frac{y'(s)}{x'(s)}\right)^2}\left(\frac{y'(s)}{x'(s)}\right)' \\
&= \frac{x'^2(s)}{x'^2(s)+y'^2(s)}\frac{x'(s)y''(s)-x''(s)y'(s)}{x'^2(s)} \\
&= \frac{x'(s)y''(s)-x''(s)y'(s)}{x'^2(s)+y'^2(s)} \\
&= x'(s)y''(s)-x''(s)y'(s),
\end{aligned}$$

again since $(x'^2(s) + y'^2(s)) = 1$. But then (compare with Equation (2.2)), $\theta'(s) = \kappa(s)$. Hence, assuming $\theta(0) = \varphi_0$, $\theta(s) = \int_0^s \kappa(\tilde{s})d\tilde{s} + \varphi_0$. The assumption of $\theta(0) = \varphi_0$ simply states that the reconstructed curve will be oriented so that its tangent's angle is φ_0 at $s = 0$, pinning down the degree of freedom of the orientation of the rigid motion.

$\kappa(s)$ is a scalar field. In order to derive $C(s)$, which is a vector-valued function, we need to introduce an arc-length unit circle with which $\kappa(s)$ will be composed. Let $Circ(s)$, $s \in [0,\ 2\pi]$ be an arc-length parameterized unit circle, $Circ(s) = (\cos(s), \sin(s))$. Then, $T(s) = Circ(\theta(s)) = Circ\left(\int_0^s \kappa(\tilde{s})d\tilde{s} + \varphi_0\right)$ prescribes the unit tangent field of $C(s)$. Hence, we can finally write

$$\begin{aligned}
C(s) &= \int_0^s T(\bar{s})d\bar{s} \\
&= \int_0^s Circ\left(\int_0^{\bar{s}} \kappa(\tilde{s})d\tilde{s} + \varphi_0\right) d\bar{s} + \left(r_x^0, r_y^0\right). \qquad (2.3)
\end{aligned}$$

This result is known (see, for example, page 24, question 9 of [3]). Yet, we now examine the computational aspects of Equation (2.3) starting with the two of the simplest possible cases of reconstructing the curve from its curvature signature, thus getting a grasp of the expected complexity. In Section 2.3, we consider the case of $\kappa(s) = 1$ and in Section 2.4 we look at $\kappa(s) = s$. Then, in Section 2.5, we will consider the computational aspects of evaluating Equation (2.3) for a general polynomial curvature signature.

2.3 Reconstructing the Curve of $\kappa(s) = 1$

Substituting 1 for $\kappa(\tilde{s})$ in Equation (2.3) one gets

$$C_1(s) = \int^s Circ\left(\int^{\bar{s}} 1\ d\tilde{s}\right) d\bar{s}$$

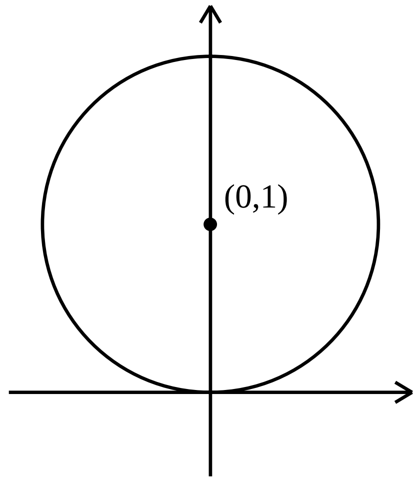

Figure 1: The curve of $\kappa(s) = 1$, $\varphi_0 = 0$ and $(r_x^0, r_y^0) = (0, 0)$.

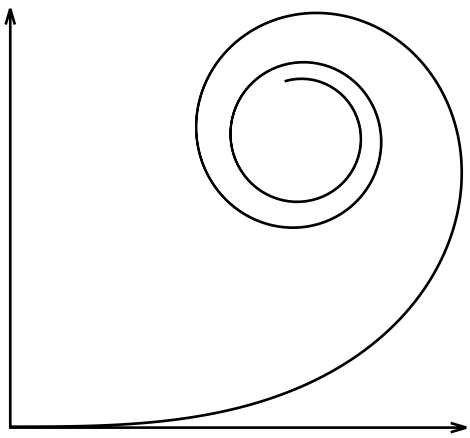

Figure 2: The curve of $\kappa(s) = s$, $\varphi_0 = 0$ and $(r_x^0, r_y^0) = (0, 0)$.

$$\begin{aligned} &= \int^s Circ\,(\bar{s} + \varphi_0)\, d\bar{s} \\ &= \int^s (\cos(\bar{s} + \varphi_0), \sin(\bar{s} + \varphi_0)) d\bar{s} \\ &= (\sin(s + \varphi_0) + r_x^0, -\cos(s + \varphi_0) + r_y^0). \end{aligned} \tag{2.4}$$

It is not surprising that the curve with a curvature of constant one turned out to be the unit circle itself (see Figure 1). φ_0 is the gained rotation degree of freedom. Moreover, any translation (r_x, r_y) of $C_1(r)$ in the plane is also a valid solution for $\kappa(s) = 1$.

2.4 Reconstructing the Curve of $\kappa(s) = s$

Substituting $\tilde{s}$ for $\kappa(\tilde{s})$ in Equation (2.3) one gets

$$C_s(s) = \int^s Circ\left(\int^{\bar{s}} \tilde{s}\; d\tilde{s}\right) d\tilde{s}$$

$$
\begin{aligned}
&= \int^s Circ\left(\frac{\bar{s}^2}{2} + \varphi_0\right) d\bar{s} \\
&= \int^s \left(\cos\left(\frac{\bar{s}^2}{2} + \varphi_0\right),\ \sin\left(\frac{\bar{s}^2}{2} + \varphi_0\right)\right) d\bar{s}.
\end{aligned} \tag{2.5}
$$

It is quite unfortunate but even for this simple case of $\kappa(s) = s$, the resulting curve (See Figure 2) has no close form solution. Indeed, no close form representation is known for the integrals of $C_s(s)$ in Equation (2.5).

The curve $C_s(s)$ in Equation (2.5) is known as the *Cornu Spiral* [19] curve and its shape is of a spiral of infinitely decreasing radius (increasing curvature).

2.5 Computational Aspect of General Curvature Signature, $\kappa(s)$

At this point it is obvious that no analytical solution to Equations (2.3) exists for a general $\kappa(s)$ function. Assume $\kappa(s)$ is represented as a piecewise polynomial form of degree d_κ. Then, $\theta(s) = \int_0^s \kappa(\tilde{s})d\tilde{s}$ is a piecewise polynomial function of one degree higher [4], $d_\kappa + 1$.

One can clearly represent the unit circle as a rational form. Yet, herein we are required to employ an *arc-length* parameterization of the unit circle or $Circ(s) = (\cos(s), \sin(s))$. Clearly, $Circ(s)$ is not rational. Let $\widehat{Circ}(s)$ be a polynomial approximation of degree d_c to $Circ(s)$, Then, $\widehat{Circ}(\theta(s))$ is a piecewise polynomial vector function that could be evaluated as a composition [8] operation between piecewise polynomial functions. $\widehat{Circ}(\theta(t))$ is of degree $d_c(d_\kappa + 1)$.

Finally, $\int_0^s \widehat{Circ}(\theta(\tilde{s}))d\tilde{s}$ is a piecewise polynomial vector function of one degree higher, $d_c(d_\kappa + 1) + 1$. We exploited the functionality of the IRIT [12] solid modeler to carry out all the necessary symbolic computations following [7], resulting in curve $C(s)$ being a B-spline curve approximating the reconstructed shape up to the accuracy of the arc-length approximation of $Circ(s)$.

To sum up, in Section 2.1, we have presented a tractable scheme to derive the curvature signature, $\kappa(s)$, of a given curve $C(t)$. Further, in Section 2.2, we have shown how to reconstruct $C(t)$ from $\kappa(s)$. In Section 3, we will employ these tools to perform an appealing metamorphosis blend between two parametric planar curves.

3 Metamorphosis Using Curvature Interpolation

The morphing problem can be defined as the continuous deformation $\mathcal{M}$:

$$
\begin{aligned}
&\mathcal{M}(t, \alpha(s_\alpha), \beta(s_\beta)), \qquad t \in [0, 1], \\
\text{such that} \quad &\mathcal{M}(0, \alpha(s_\alpha), \beta(s_\beta)) \quad = \alpha(s_\alpha),
\end{aligned}
$$

$$\mathcal{M}(1, \alpha(s_\alpha), \beta(s_\beta)) \quad = \beta(s_\beta),$$

where the first parameter t coincides with the *time* for fixing the intermediate curve, and $\alpha(s_\alpha)$ and $\beta(s_\beta)$ are the two given key curves, with arc-length parameterization s_α and s_β, respectively.

In this work, we present an algorithm to compute the metamorphosis of two-dimensional free-form parametric curves that exploits *intrinsic shape parameters*. That is, the shape of the intermediate curves, denoted by $\mathcal{M}(t, \alpha(s_\alpha), \beta(s_\beta))$, depends on the intrinsic geometric properties of the two given curves $\alpha(s_\alpha)$ and $\beta(s_\beta)$.

Let $r \in [0, 1]$ and denote by L_α and L_β the lengths of curves $\alpha(\cdot)$ and $\beta(\cdot)$, respectively. Further, let $s_\alpha = rL_\alpha$ and $s_\beta = rL_\beta$ be the arc-length parameters of the two curves that correspond to the former value r. Then, the idea of the presented approach is to linearly interpolate the curvature signatures of the curve $\mathcal{M}(t, \alpha(s_\alpha), \beta(s_\beta))$ from the two curvature signatures of $\alpha(s_\alpha)$ and $\beta(s_\beta)$, $\kappa_\alpha(s_\alpha)$ and $\kappa_\beta(s_\beta)$:

$$\kappa_t(s_t) = \kappa_t(rL_t) = (1-t)\kappa_\alpha(rL_\alpha) + t\kappa_\beta(rL_\beta) = (1-t)\kappa_\alpha(s_\alpha) + t\kappa_\beta(s_\beta), \tag{3.1}$$

where $L_t = (1-t)L_\alpha + tL_\beta$ is the length of the interpolated curve $\mathcal{M}(t, \alpha(s_\alpha), \beta(s_\beta))$ at time t. Note the length L_t of the intermediate curve is a monotone change of length function between length L_α and length L_β. Once the curvature, $\kappa_t(s_t)$, of the intermediate planar curve is known, we can recover the curve, following Equation (2.3). One may also apply rigid motion transformation to the interpolated result in order to set the curve's orientation and position properly (recall that the curvature is rotation and translation invariant).

3.1 Piecewise Interpolation

We should remember that, thus far, the above approach has not considered the correspondence between the parameterizations of the curves $\alpha(\cdot)$ and $\beta(\cdot)$.

Assume that we are given a set of N matched points, $\left\{s_i^\alpha, s_i^\beta\right\}_{i=1}^N$, along the two key curves, $\alpha(s)$ and $\beta(s)$. Subdivide curves $\alpha(s)$ and $\beta(s)$ at parameter values s_i^α and s_i^β, creating $N+1$ pairs of curve segments (assuming $\alpha(s)$ and $\beta(s)$ are open curves), and then use the curvature interpolation scheme for each pair of segments. The reconstruction of the whole intermediate curve from its segments necessitates the rotation and translation of each segment so as to match the end position and tangent of the previous curve segment, achieving G^2 continuity, as the curvature is matched by construction.

An important feature of this reconstruction scheme is identity preserving. That is, if the matched values are proportional in each pair $\left\{s_i^\alpha, s_i^\beta\right\}, \forall i$, or $s_i^\alpha = r_iL_\alpha$ and $s_i^\beta = r_iL_\beta$, for all $i \in \{1, \ldots, N\}$, then the metamorphosis sequence is identical to the one that is received via a simple curvature interpolation of the curves, as a whole.

3.2 Metamorphosis of Closed Curves

While interpolating between two, beginning and end, closed curves, the intermediate reconstructed curve $C_t(s)$ may end up open, i.e. $C_t(0) \neq C_t(L_t)$. The method to overcoming this difficulty, used in [14], is employed herein as well. Nonetheless, instead of the polygon's vertices we move the control points of the control polygon of $C_t(s)$. Let P_0 and P_n be the two end points of the control polygon of $C_t(s)$. We distribute the error $\vec{e} = P_n - P_0$ evenly, between the points of the control polygon of the curve:

$$\overline{P_i} = P_i - \frac{i}{n}\,\vec{e},$$

where $0 \leq i \leq n$. Now, the new control polygon $\left\{\overline{P_i}\right\}_{i=0}^{n}$ is closed, having $\overline{P_0} = \overline{P_n}$. Nevertheless, tangent continuity is still not guaranteed. The following conditions must be satisfied for C^1 continuity:

$$P_n = P_0 \quad \text{and} \quad P_0 - P_1 = \gamma\left(P_n - P_{n-1}\right), \quad \gamma > 0.$$

C^1 continuity could be accomplished as follows. Let $\vec{v} = P_{n-1} - (2P_0 - P_1)$. Then, assuming $\overline{P_n} = P_0$,

$$\overline{P_i} \;=\; P_i - \frac{i-1}{n-2}\,\vec{v}, \quad \text{where } 1 \leq i \leq n-1.$$

This process affects the G^2 continuity at the end. A similar propagation scheme of the end condition error can be defined using P_0, P_1, P_2 and P_n, P_{n-1}, P_{n-2} to update P_{n-2} and regain curvature continuity.

4 Examples

We now consider some illustrative examples of metamorphosis sequences that have been created using our scheme. The first example (see Figures 3 and 4) presents metamorphosis sequences between a Cornu spiral with curvature $\kappa_\alpha(s) = s$ (see Equation (2.4)) and a Cornu spiral with curvature $\kappa_\beta(s) = -0.5s$.

The sequence in Figure 3 is generated by our proposed algorithm. The curve is reconstructed as described in Section 2. Compare this metamorphosis sequence with Figure 4 where a regular linear interpolation between the control points of the spirals has been performed. The regular linear interpolation results in self-intersections of the intermediate curves and unnatural looking deformations. The respective curvature signatures of the metamorphosis sequences of Figures 3 and 4 are presented in Figures 5 and 6. Note the extreme curvature values that are reached when linear interpolation is used even for this very simple case.

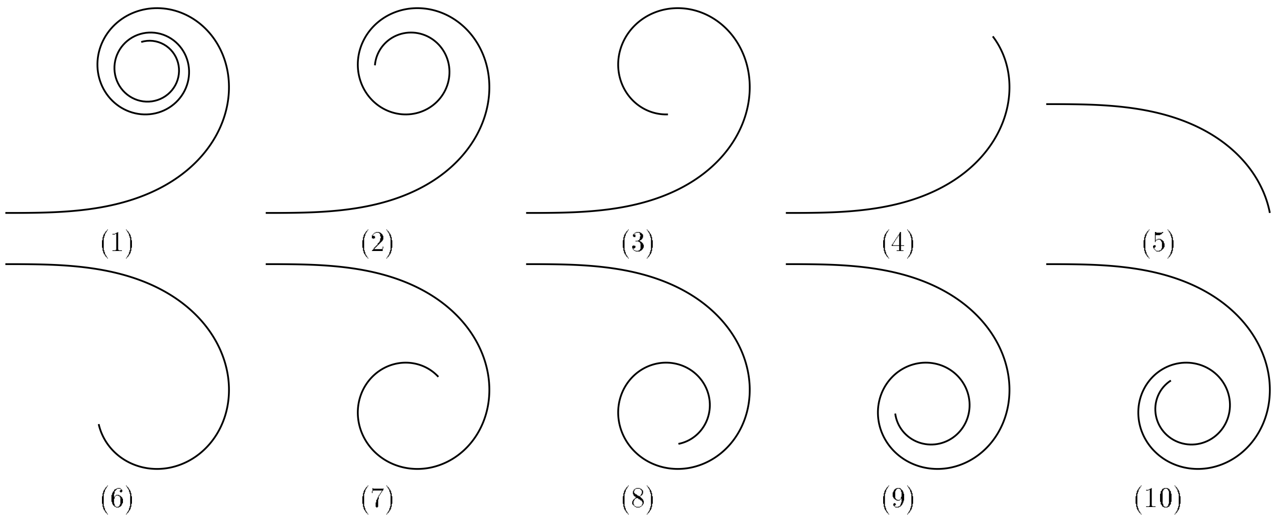

Figure 3: A metamorphosis sequence between two spirals with different linear curvature functions and signs, using curvature interpolation. Compare with Figure 4.

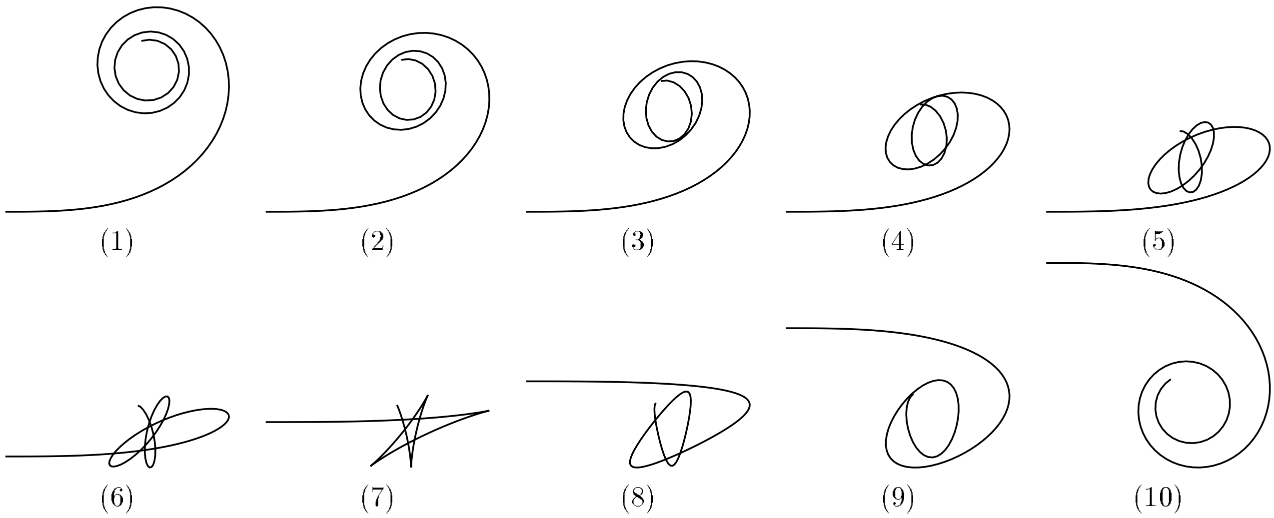

Figure 4: A metamorphosis sequence between two spirals by linear interpolation. Compare with Figure 3.

The next example (Figures 7 and 8) represents a metamorphosis sequence between the letters 'U' (1) and 'S' (10). As in the previous example, Figure 7 is the result of using our method, while Figure 8 has been created using linear interpolation between the two symbols. Clearly, Figure 7 looks far more natural.

A more interesting example of metamorphosis (between the two human shapes) is presented in Figures 9, 10 and 11. The end user (the artist designing the animation) can anticipate how the intermediate curves are to behave. Such a natural animation is produced by our algorithm (Figure 10), whereas the trivial linear interpolation (Figure 11) results in self-intersection in the arm region and distortions of the arms and legs of the human shapes. A global curvature interpolation has been used in Figure 9, while in Figure 10

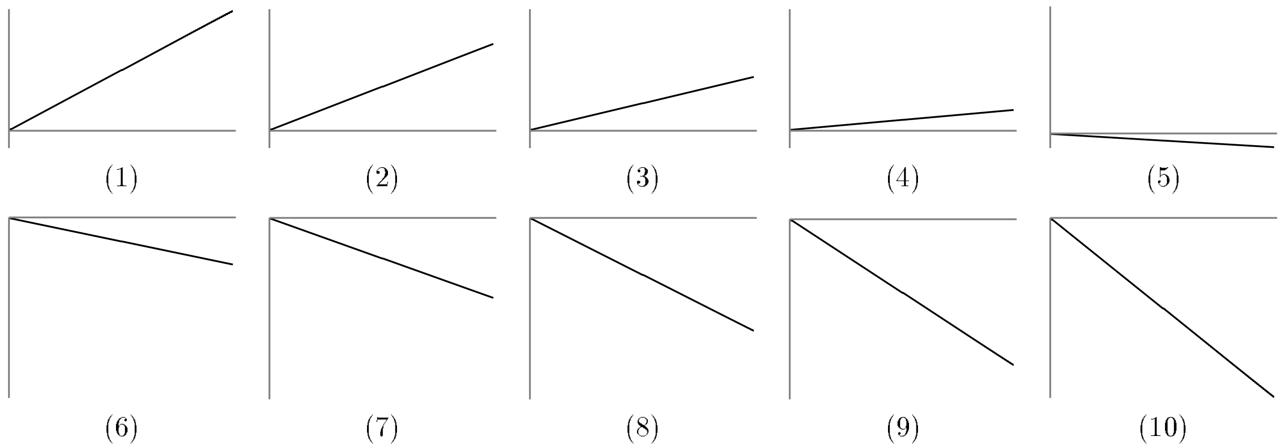

Figure 5: Curvature signature $\kappa_t(s)$ for the metamorphosis sequence between the two spirals (see Figure 3) with different curvature signs and different curvature change rates, using curvature interpolation. Compare with Figure 6.

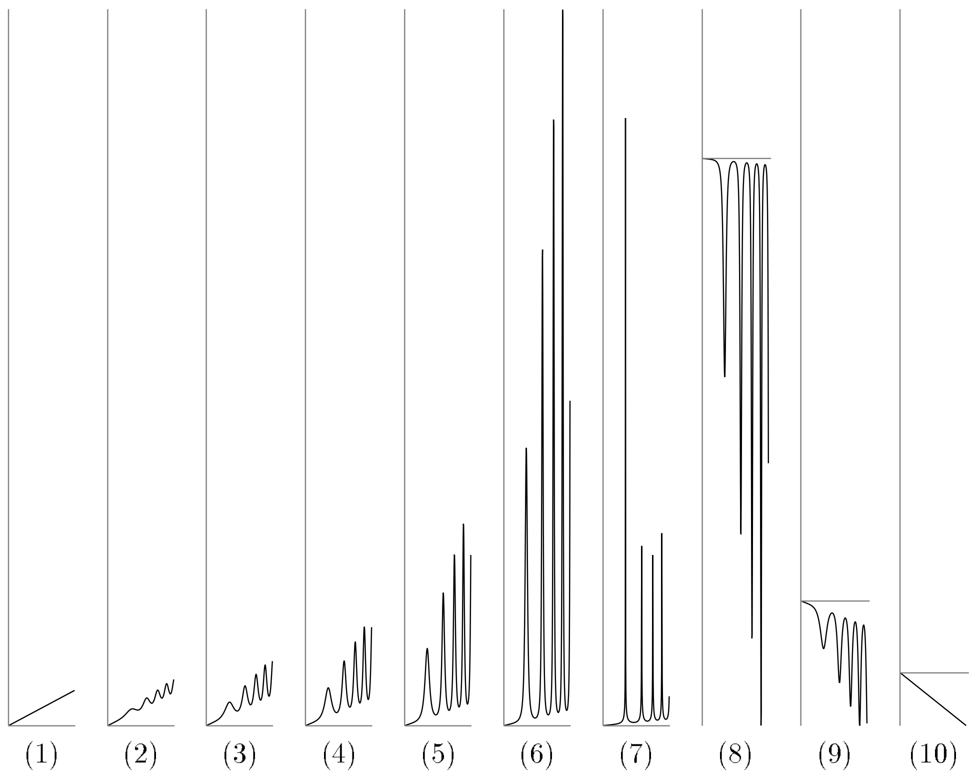

Figure 6: Curvature signature $\kappa_t(s)$ for the metamorphosis sequence between the two spirals (see Figure 4) by linear interpolation. Note the extreme curvature values and compare with Figure 5.

we subdivided the periodic curves into nine matched segments and performed the metamorphosis in a piecewise manner (see Section 3.1). The nine matched pairs of points have been specified manually for this specific example, whereas, for example, an automatic method that finds curvature extremum points might also be used.

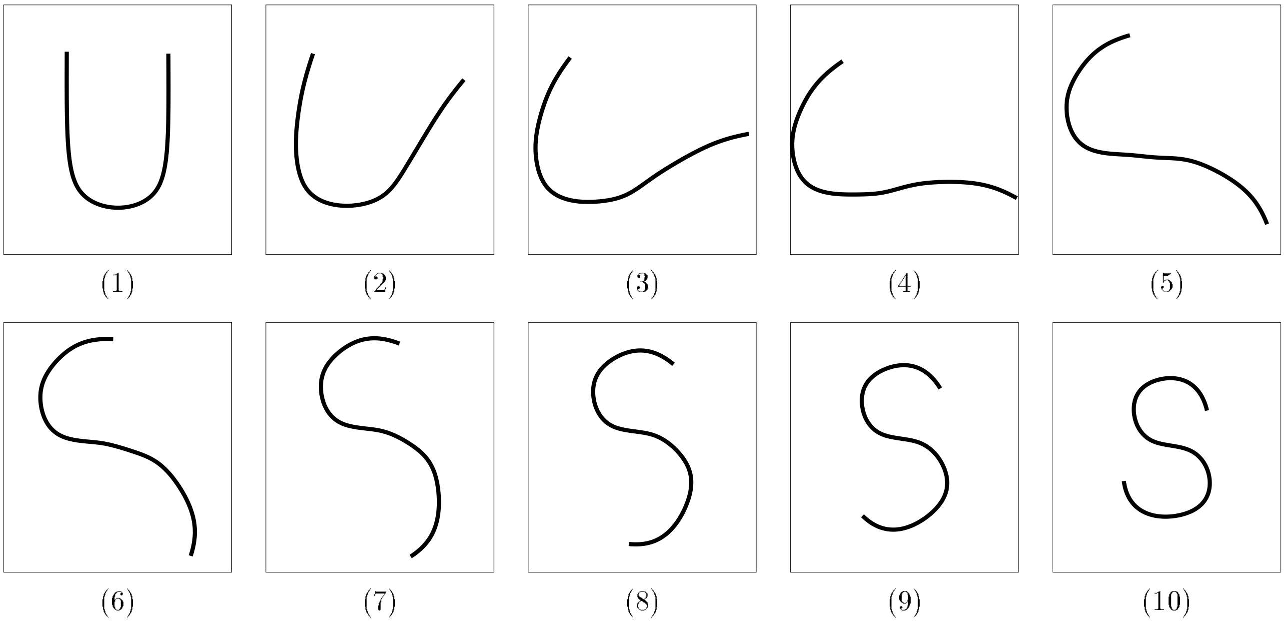

Figure 7: Metamorphosis sequence between 'U' (1) and 'S' (10) shapes using curvature interpolation. The length of the intermediate curves is changing monotonically. Compare with Figure 8.

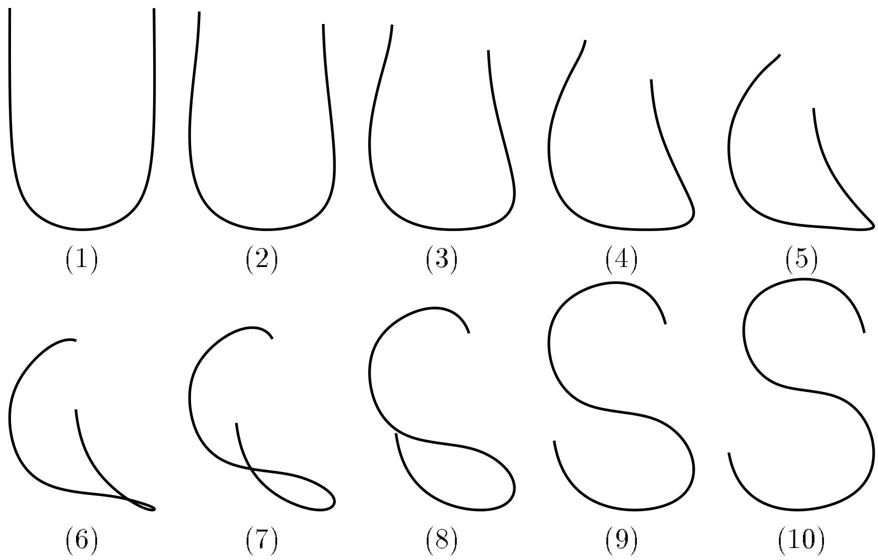

Figure 8: Metamorphosis sequence between 'U' (1) and 'S' (10) shapes using linear interpolation. Compare with Figure 7.

While the curvature interpolation scheme produces appealing results, it is also local and hence can easily handle non-simple curves as in Figure 9 (10), having self-intersecting curves as input.

Our final example consists of elephant shapes (see Figures 12 and 13). Again, the metamorphosis sequence in Figure 12 produced by our algorithm looks more appealing than the linear interpolation in Figure 13. The tail and the trunk should not shrink or grow wider during the animation (compare Figures 12

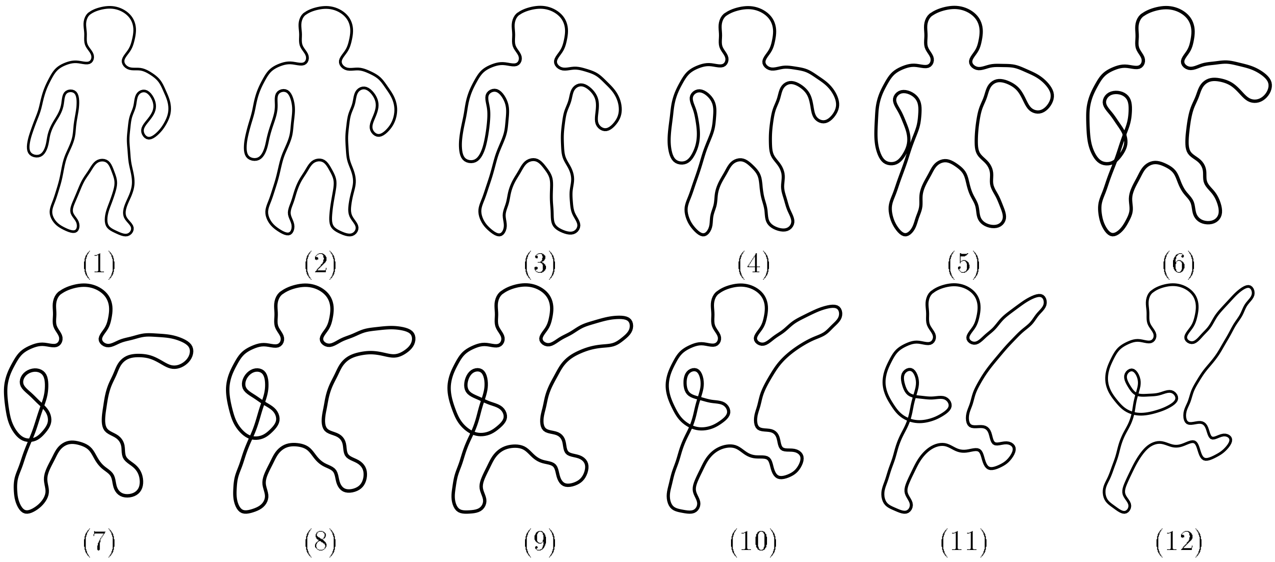

Figure 9: Metamorphosis sequence between two human shapes (1) and (12) using curvature interpolation. Note the blending of the arms. Compare with Figure 10 and Figure 11.

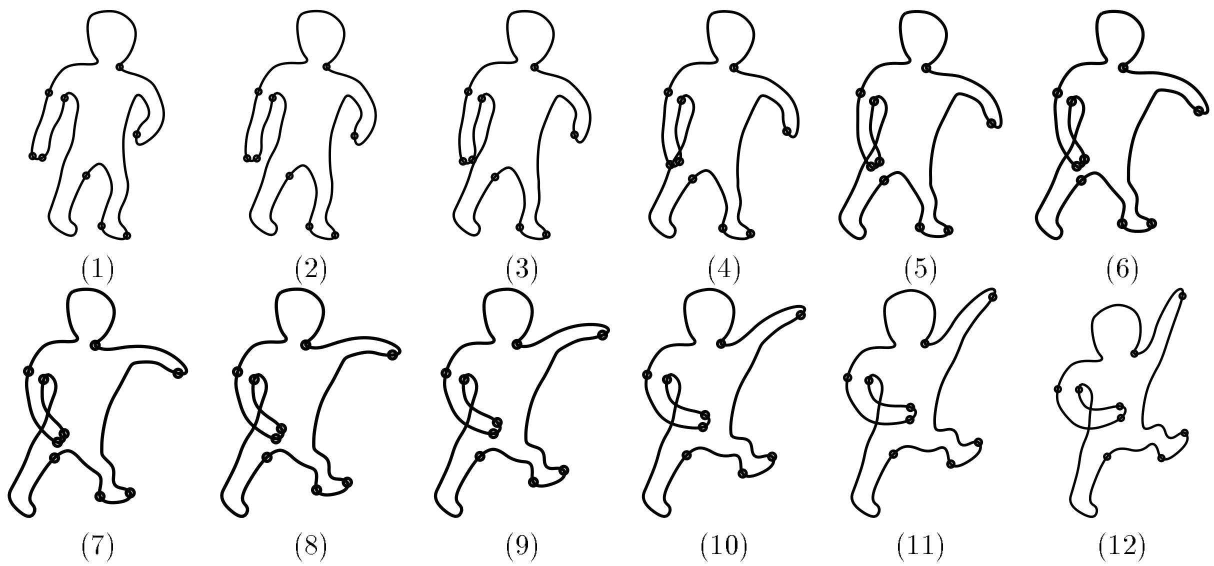

Figure 10: Metamorphosis sequence between two human shapes (1) and (12) using curvature interpolation and curve segmentation at the points that are labeled by circles. Compare with Figure 9 and Figure 11.

and 13). Both given curves have been subdivided into eleven segments and piecewise metamorphosis has been executed.

5 Conclusion

In this work, a new method to compute the metamorphosis of two-dimensional parametric curves has been introduced. The solution to the metamorphosis problem requires the solution of two independent tasks.

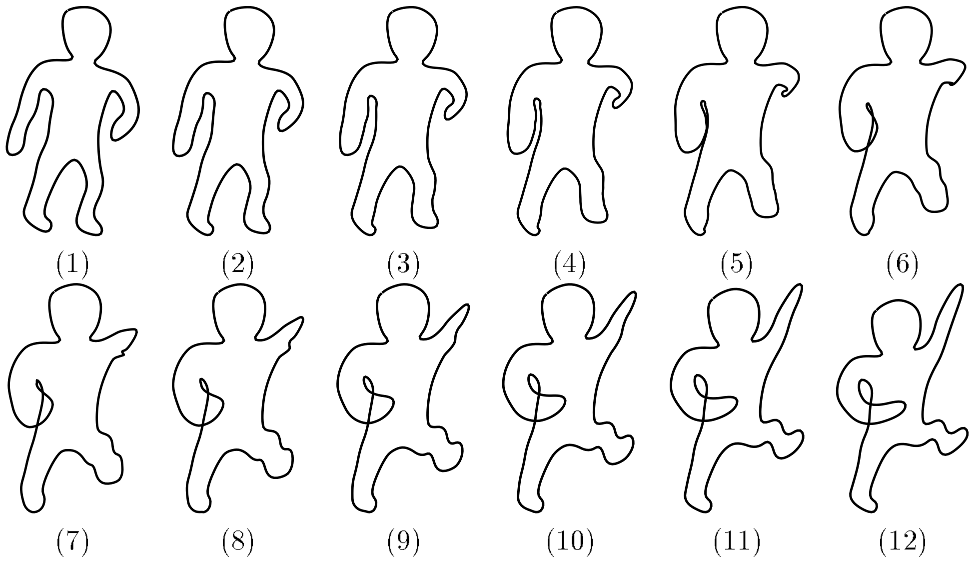

Figure 11: Metamorphosis sequence between two human shapes (1) and (12) by linear interpolation. Compare with Figure 9 and Figure 10.

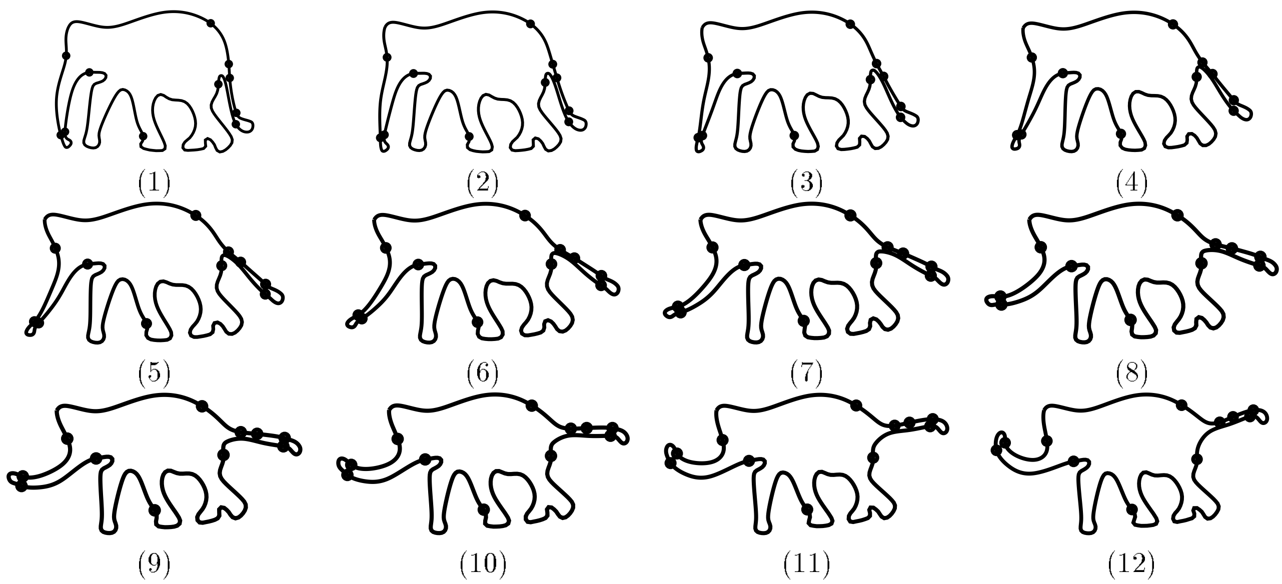

Figure 12: Metamorphosis sequence between two elephant shapes (1) and (12) using curvature interpolation and curve segmentation at the points that are labeled by circles. Compare with Figure 13.

The first derives a correspondence between the two objects. Our focus has been on the second subproblem that deals with a proper, continuous, appealing interpolation between the two shapes.

The proposed method interpolates linearly between the curvature signature functions of the two given key curves. When the sequence of the curvature signature functions is known, we are able to reconstruct the metamorphosis sequence itself.

The resulting animation sequences have a natural appearance without artificial shrinkage and distor-

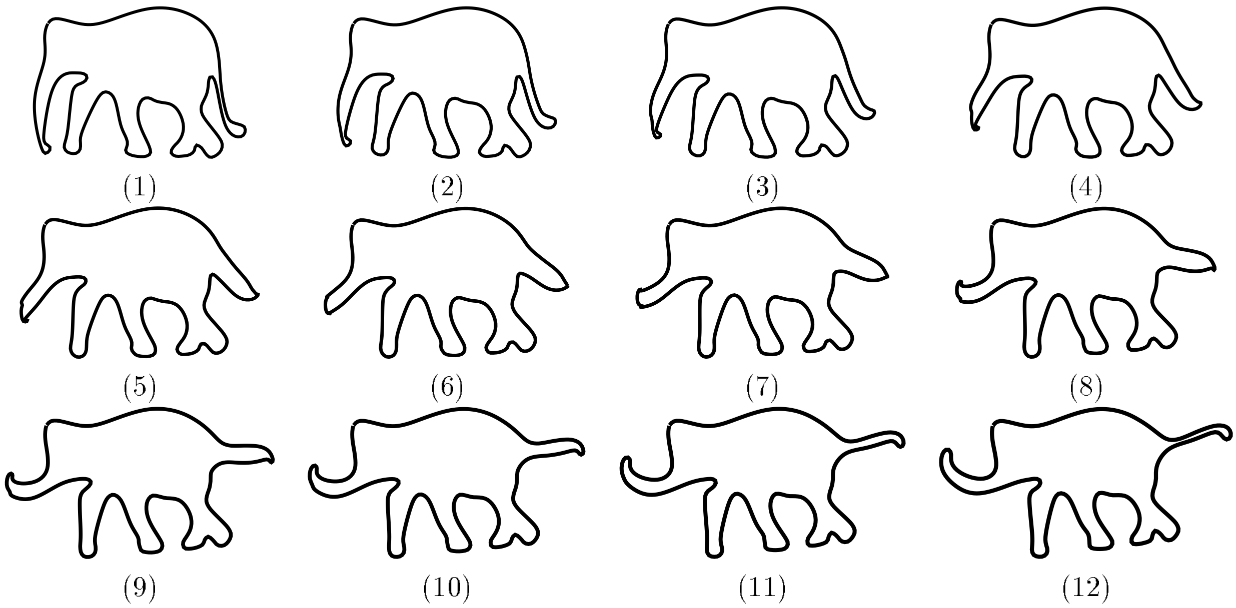

Figure 13: Metamorphosis sequence between two elephant shapes (1) and (12) by linear interpolation. Compare with Figure 12.

tions. The curvature signature can also provide us with global properties such as the winding number of the curve [3]. Hence, a metamorphosis of the two simple curves will neither introduce a double loop curve nor a self-intersecting eight shape curve. Regrettably, this new method does not guarantee self-intersection free intermediate curves even when the given two key curves are known to be simple. This remains an open question.

Finally, the extension of this work to free form surfaces is an obvious goal of high value and merit.

6 Acknowledgment

The authors would like to thank Alex Sherman for his reference to the Cornu Spiral.

This research was supported in part by the Fund for Promotion of Research at the Technion, IIT, Haifa, Israel.

References

[1] M. Alexa, D. Cohen-Or, and D. Levin, *As-rigid-as-possible shape interpolation*, Computer Graphics (Proceedings of SIGGRAPH 00), (July 2000), pp. 157– 164.

[2] N. Arad and D. Reisfeld, *Image warping using few anchor points and radial functions*, Computer Graphics *Forum*, 14 (1995), pp. 35 – 46.

[3] M. D. Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, 1976.

[4] E. Cohen, R. F. Riesenfeld, and G. Elber, *"Geometric Modeling with Splines - An Introduction"*, A. K. Peters. Natick, Massachusetts, 2001.

[5] S. Cohen, G. Elber, and R. Bar-Yehuda, *Matching of freeform curves*, CAD, 29 (1997), pp. 369 – 378. Also Center for Intelligent Systems Tech. Report, CIS 9527, Computer Science Department, Technion.

[6] D. Cohen-Or, D. Levin, and A. Solomovici, *Three-dimensional distance field metamorphosis*, ACM Transaction on Graphics, 17 (1998).

[7] G. Elber, *Free form surface analysis using a hybrid of symbolic and numeric computation*, PhD thesis, Computer Science Dept., University of Utah, 1992.

[8] G. Elber, *Symbolic and numeric computation in curve interrogation*, Computer Graphics Forum, 14 (1995), pp. 25 – 34.

[9] P. Gao and T. W. Sederberg, *A work minimization approach to image morphing*, The Visual Computer, 14 (1998), pp. 390 – 400.

[10] E. Goldstein and C. Gotsman, *Polygon morphing using a multiresolution representation*, Proceeding of Graphics Interface, (1995).

[11] J. Hughes, *Scheduled Fourier volume morphing*, Computer Graphics (SIGGRAPH '92), 26 (1992), pp. 43 – 46.

[12] IRIT, *Irit 7.0 User's Manual.*, Technion, Israel. http://www.cs.technion.ac.il/~irit/, Mar. 1997.

[13] T. Samoilov and G. Elber, *Self-intersection elimination in metamorphosis of two-dimensional curves*, The Visual Computer, 14 (1998), pp. 415 – 428.

[14] T. W. Sederberg, P. Gao, G. Wang, and H. Mu, *2D shape blending: an intrinsic solution to the vertex path problem*, Computer Graphics (SIGGRAPH '93), 27 (1993), pp. 15 – 18.

[15] T. W. Sederberg and E. Greenwood, *A physically based approach to 2D shape blending*, Computer Graphics (SIGGRAPH '92), 26 (1992), pp. 25 – 34.

[16] T. W. Sederberg and E. Greenwood, *Shape blending of 2-d piecewise curves*, Mathematical Methods in CAGD III, 26 (1995), pp. 1 – 3.

[17] M. Shapira and A. Rappoport, *Shape blending using the star-skeleton representation*, IEEE Trans. on Computer Graphics and Application, 15 (1995), pp. 44 – 51.

[18] V. Surazhsky and C. Gotsman, *Guaranteed intersection-free polygon morphing*, Computers and Graphics, (2001), pp. 67–75.

[19] E. Weisstein, *World of Mathematics*, A Wolfram Web Resource, "http://mathworld.wolfram.com/CornuSpiral.html".

[20] G. Wolberg, *Image morphing: a survey*, The Visual Computer, 14 (1998), pp. 360 – 372.