Morphing Rational B-spline Curves and Surfaces Using Mass Distributions

Tao Ju[1] and Ron Goldman[1]

[1] Department of Computer Science, Rice University, Houston, Texas, USA

Abstract

A rational B-spline curve or surface is a collection of points associated with a mass (weight) distribution. These mass distributions can be used to exert local control over the morph between two rational B-spline curves or surfaces. Here we propose a technique for designing customized morphs by attaching appropriate mass distributions to target B-spline curves and surfaces. We also develop a user interface for this morphing method that is easy to use and requires no knowledge of B-splines on the part of the designer.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Hierarchy and geometric transformations

1. Introduction

Morphing transforms one target shape into another. Besides being popular in movies and TV commercials, morphing has also found applications in various aspects of computer graphics, visualization, and design. Research on morphing techniques has centered around two tasks: establishing a proper mapping between target shapes, and creating a path between corresponding vertices.

While solutions to these problems have been proposed in the domain of polygonal meshes (Alexa[2] gives an excellent review), here we consider morphing between target shapes that are represented by rational curves or surfaces. The advantage of using a parametric representation is that the mapping between two targets is established by the given parameterizations. Hence in our discussion we assume the existence of a correspondence between the targets. We are particularly interested in rational B-splines, which are widely used in computer-aided design and computer graphics for modelling smooth geometry[7]. We will focus on the problem of interpolation between pairs of points with the same parameter on two rational B-spline curves or surfaces, although our technique is suitable for any rational representation.

One simple way to perform this interpolation is by linear averaging between the spacial locations of the two points[12, 18]. Unfortunately, direct application of linear averaging often produces unsatisfactory results, such as self-intersection or undesirable shrinking, and alternatives have been explored in the domain of solid representations[11], polygonal meshes[16, 17, 3, 10] and polynomial forms[15].

The unique property of a rational representation, unlike dis-

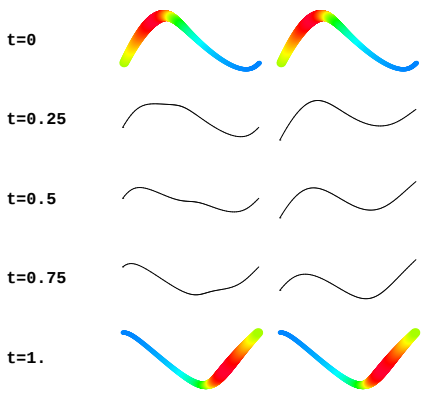

Figure 1: *The morph between two rational B-spline curves by linear averaging (left) and weighted averaging (right). Red regions on the targets have greater mass and blue regions have smaller mass. Notice the flattening and the small wriggles in the middle transition curve on the left.*

crete meshes or polynomial representations, is that each point on the curve or surface is associated with a weight (or *mass*). The variation of masses in a rational representation acts as an additional control on the shape of the curve or surface. Linear averaging of spatial locations alone ignores the relative mass distributions on the targets, and may incur artifacts such as unnatural flattening or wriggles on the transitions (see figure 1 left and figure 2 top). Therefore interpolation in affine space is inappropriate for morphing between rational representations.

In our work, we propose to perform interpolation in *Grassmann space*, a vector space in which operators such as addition and scalar multiplication model physical properties by considering the mass associated with each point. The vertex path between two points is computed as the non-linear *weighted average* of their spatial locations, often resulting in a more natural transition between two rational curves or surfaces (see figure 1 right and figure 2 bottom).

Although attempts have been made to quantify the quality of a morph[3, 6], we have found it difficult to measure the "naturalness" of a transition sequence, which in any event is often subject to user interpretation. Instead, we let the user judge the quality of the morph and improve it interactively by adjusting unsatisfactory regions on the targets. In this way, a designer can produce special effects such as in the two sequences at the bottom of figure 6.

This idea of gradual improvement of the morph through interaction, rather than one-step optimization, is relatively unexplored. In fact, most existing morphing methods do not allow local adjustment of the morph once the vertex paths are computed from the target shapes. Rossignac et al.[14] suggest using additional polyhedra as the control points of a Bezier curve to adjust the transition shapes. Since the Minkowski sum is used in place of vector addition, their method does not extend to morphing between non-solid representations, such as curves and surfaces. Recently, Alexa[1] proposed using a relative representation of vertices in a polygonal mesh in place of absolute coordinates, and then exerting local morph control by linear averaging using different transition states at each vertex. However, the computation of the vertex positions on each transition shape involves solving a large system of linear equations; therefore this method is not suitable for editing smooth morphing sequences in real-time.

In our method, each rational B-spline curve or surface can be augmented with a user-defined mass distribution. Designers can conveniently adjust the morphing behavior of local regions on the targets interactively by varying the masses on the targets; the entire morphing sequence is updated at trivial computational cost. Best of all, the user can accomplish the design with no knowledge of the mass distributions and no understanding of B-splines.

We begin with a brief introduction to rational B-splines and their mass distributions. Weighted averaging is presented next, followed by a detailed discussion of non-uniform modification of masses under user control. A user interface is presented that allows easy, real-time design of surface morphing using the proposed method. We conclude with some possible applications.

2. Rational B-splines and their Associated Mass Distributions

A rational B-spline curve of degree n is typically written as

$$P(u) = \frac{\sum_{k=0}^{p} w_k P_k N_k^n(u)}{\sum_{k=0}^{p} w_k N_k^n(u)} \tag{1}$$

where P_k is a collection of control points, w_k are scalar weights associated with the corresponding control points, and $N_k^n(u)$ are the B-spline basis functions defined over some knot vector $\{u_0, u_1, ..., u_{p+n-1}\}$. Similarly a rational tensor product B-spline surface of bi-degree (m,n) can be written as

$$P(u,v) = \frac{\sum_{j=0}^{p}\sum_{k=0}^{q} w_{jk} P_{jk} N_j^m(u) N_k^n(v)}{\sum_{j=0}^{p}\sum_{k=0}^{q} w_{jk} N_j^m(u) N_k^n(v)} \tag{2}$$

where P_{jk} is a two-dimensional array of control points with scalar weights w_{jk}, and $N_j^m(u)$ and $N_k^n(u)$ are B-spline basis functions defined over knot vectors $\{u_0, u_1, ..., u_{p+m-1}\}$ and $\{v_0, v_1, ..., v_{q+n-1}\}$. The weight or mass w_k (w_{jk}) associated with each control point P_k (P_{jk}) acts as a tension parameter, controlling the shape of the curve (surface) near that control point. A larger mass pulls the curve (surface) closer to the control point; a smaller mass pushes the curve (surface) further away from the control point.

The control structure of a rational B-spline curve or surface consists of mass-points[8]. Recall that a *mass-point* consists of a non-zero scalar mass $m \neq 0$ attached to a point P in affine space. These mass-points reside in a vector space, called *Grassmann Space*, in which operations such as addition and scalar multiplication model physical properties. (Grassmann space also contains vectors, mass-points where the mass is zero.) In the following definitions, we adopt the notation mP/m to denote both the mass-point (i.e., the mass m located at the point P) and the affine point P (i.e, the quotient)[9].

1. Scalar multiplication. Multiplying the mass by a scalar leaves the position of the affine point unchanged.

$$c \otimes \frac{mP}{m} = \frac{cmP}{cm}$$

2. Addition. To add two mass-points, we sum their masses and position this sum at their center of mass.

$$\frac{m_1 P_1}{m_1} \oplus \frac{m_2 P_2}{m_2} = \frac{m_1 P_1 + m_2 P_2}{m_1 + m_2}$$

Note that standard homogeneous coordinates will not work here since projective space is not a vector space, hence operations such as addition and scalar multiplication are not

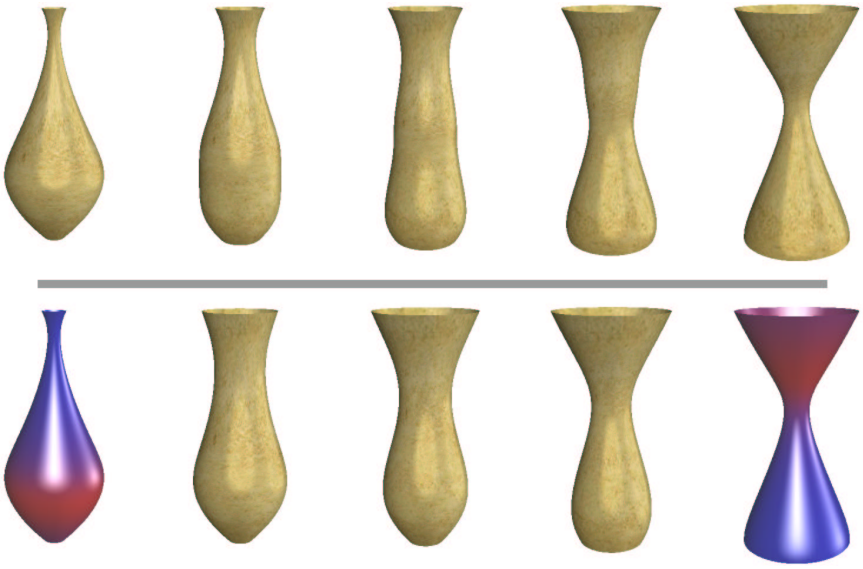

Figure 2: *The morph between rational B-spline surfaces by linear averaging (top) and weighted averaging (bottom). Red regions on the targets have greater mass and blue regions have smaller mass. Notice the flattening and the small wriggles on the boundary of the middle vase on the top.*

defined[9]. Using the Grassmann space operations $\otimes$ and $\oplus$, we can reformulate the representations in (1) as

$$P(u) = \sum_{k=0}^{p} \left(\frac{w_k P_k}{w_k} \right) \otimes N_k^n(u)$$

where $\sum_{k=0}^{p}$ denotes applying the $\oplus$ operator p times. Similarly, a rational tensor product B-spline surface can be represented as

$$P(u,v) = \sum_{j=0}^{p} \sum_{k=0}^{q} \left(\frac{w_{jk} P_{jk}}{w_{jk}} \right) \otimes N_j^m(u) N_k^n(v)$$

The reformulated representations suggest that each point on a rational B-spline curve or surface is actually a mass-point. The *mass distribution* $m_P(u)$ on a curve $P(u)$ or $m_P(u,v)$ on a surface $P(u,v)$ is the denominator in the basis function representations in (1) or (2).

3. Morphing By Averaging

Here we only consider morphing between two rational B-spline curves or surfaces with the same degree and the same knot vectors. This requirement can always be enforced by applying degree elevation[13] and knot insertion techniques[5, 4].

3.1. Morphing by linear averaging

An easy way to morph between two points P and Q is simply to average their geometric positions using a time parameter t $(0 \le t \le 1)$

$$M_{aff}(t) = (1-t)P + t\,Q$$

This morphing function represents a linear transition from P to Q in affine space. The corresponding morph between points on two degree n rational B-spline curves $P(u)$ and $Q(u)$ can be expressed as

$$M_{aff}(u,t) = (1-t)P(u) + tQ(u)$$

Similarly, the morph between two bi-degree (m,n) rational B-spline tensor-product surfaces $P(u,v)$ and $Q(u,v)$ can be written as

$$M_{aff}(u,v,t) = (1-t)P(u,v) + t\,Q(u,v)$$

Linear averaging is simple to understand and easy to implement. However, the method often leads to undesirable morphs. An example illustrating morphing between two rational B-spline curves by linear averaging is shown in figure 1 left. In contrast with the wavy target curves, the transitional curves exhibit unexpected flattening and minute wriggles towards the middle of the morph. Similar defects can be observed on the transitional vases in the surface example at the top of figure 2. These defects occur, in part, because linear averaging is based only on the geometric positions of the targets, and ignores the associated mass distributions.

3.2. Morphing by weighted averaging

If the target points have mass, we can perform the linear averaging operation in Grassmann space first and then

project onto affine space. The morph between two mass-points m_1P/m_1 and m_2Q/m_2 using Grassmann operators can be expressed as

$$M_{mass}(t) = \left((1-t)\otimes\frac{m_1P}{m_1}\right)\oplus\left(t\otimes\frac{m_2Q}{m_2}\right) \quad (3)$$

$$= \frac{(1-t)m_1P+tm_2Q}{(1-t)m_1+tm_2} \quad (4)$$

Unlike linear averaging in affine space, this new averaging scheme takes into account the masses of the points. Expression (4) can be rewritten as an affine combination of P and Q,

$$M_{mass}(t) = (1-D(t))P + D(t)Q,$$

where $D(t) = \frac{t\,m_2}{(1-t)m_1+t\,m_2}$. Thus the morph is now computed as a *weighted average* of the geometric positions of the targets. This construction has the following notable properties:

1. $D(t)$ is a continuous, non-negative, monotonic function of t on the interval $[0,1]$. Hence the resulting vertex path between P and Q is infinitely smooth, monotonic, and bounded between the two end points. These properties are typically difficult to achieve with other transition functions.

2. The morphing behavior is affected locally by the *ratio* of the masses at the two target points. If both targets have the same mass ($m_1 = m_2$), $D(t) = t$ and weighted averaging reduces to linear averaging. But, if $m_1 > m_2$, then $D(t) < t$; hence $M_{mass}(t)$ always stays closer than $M_{aff}(t)$ to the first target P during the morph. In fact, as $\frac{m_1}{m_2}$ increases (decreases), $D(t)$ moves closer to 0 (1). Consequently, at any time during the morph the transition point generated by weighted averaging is always "attracted" to the target with the larger mass. This attractive force increases as the ratio between the target masses increases.

Morphing by weighted averaging between the mass-points on two degree n rational B-spline curves $P(u)$ and $Q(u)$ with corresponding mass distributions $m_P(u)$ and $m_Q(u)$ is represented as

$$M_{mass}(u,t) = \frac{(1-t)m_P(u)P(u)+tm_Q(u)Q(u)}{(1-t)m_P(u)+tm_Q(u)}$$

Similarly, two bi-degree (m,n) rational B-spline surfaces $P(u,v)$ and $Q(u,v)$ with mass distributions $m_P(u,v)$ and $m_Q(u,v)$ can be transformed by

$$M_{mass}(u,v,t) = \frac{(1-t)m_P(u,v)P(u,v)+tm_Q(u,v)Q(u,v)}{(1-t)m_P(u,v)+tm_Q(u,v)}$$

As the ratio of the masses between corresponding mass-points changes along the targets, the transitional curves (surfaces) will be drawn towards the first or second target at different rates in different regions. Note that weighted averaging also keeps the degree lower than linear averaging, since there is no need to compute a common denominator between $P(u)$ and $Q(u)$ or $P(u,v)$ and $Q(u,v)$.

In figure 1, the target curves on the left are transformed again using weighted averaging on the right. Notice that the transitional curves are drawn to $P(u)$ in the first part of the curve, where the mass-points on $P(u)$ contain larger masses, and drawn to $Q(u)$ in the second part, where the mass-points on $Q(u)$ are heavier. The stretching forms a nice wavy shape that eliminates the flattening and wriggling exhibited by linear averaging. In figure 2, the target vases at the top are transformed again using weighted averaging at the bottom. The damping effect in the morphing example using linear averaging disappears because the targets contain higher masses in different regions and affect the morph accordingly.

4. Controlled Morphing Using Mass Distributions

Since the mass distributions we have been using so far are uniquely determined by the weights associated with the control points, our power to adjust the morph is limited. In fact, it is not hard to generate examples in which weighted averaging using the inherent mass distributions performs no better than linear averaging (for instance, when both targets exhibit the same mass variation). We need to be able to modify the masses without changing the shapes or positions of the associated curves or surfaces to produce user-desired morphs.

4.1. Uniform scaling of masses

Modifying the weights associated with the control points normally alters the shape of the corresponding rational B-spline curve or surface. One straightforward alternative, however, is to scale the mass associated with each control point on one target by the same amount. The position of each point on the curve (surface) will not change, but the associated mass distribution will be scaled uniformly. The weighted average between two curves $P(u)$ and $Q(u)$, with the masses of $P(u)$ scaled by S, can be expressed as

$$M_{scaled}(S,u,t) = \frac{S(1-t)m_P(u)P(u)+tm_Q(u)Q(u)}{S(1-t)m_P(u)+tm_Q(u)}$$

$$= \frac{(1-t')m_P(u)P(u)+t'm_Q(u)Q(u)}{(1-t')m_P(u)+t'm_Q(u)} = M_{mass}(u,t')$$

where $t' = \frac{t}{((1-t)S+t)}$. Therefore uniform scaling has the effect of advancing ($S<1$) or delaying ($S>1$) the morph of the whole curve. The same effect can be observed on rational B-spline surfaces. However, in order to control the transformation of local regions on the target, we need to be able to apply a non-uniform modification to the mass distribution.

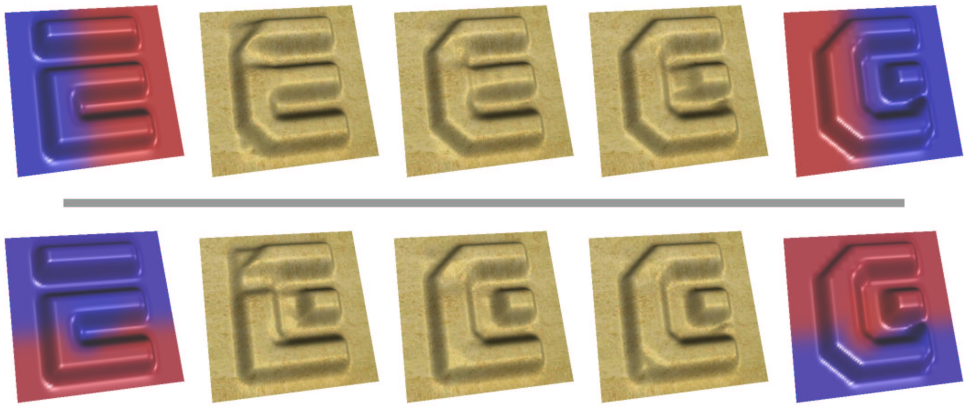

Figure 3: *Two morphs between letters **E** and **G** after non-uniform modification of masses. Red regions on the targets have greater masses and blue regions have smaller masses.*

4.2. Non-uniform modification of masses

We can modify the representations of the target degree n rational B-spline curves $P(u)$ and $Q(u)$ by setting

$$\overline{P}(u) = \frac{\overline{m}_P(u) \cdot P(u)}{\overline{m}_P(u)} \text{ and } \overline{Q}(u) = \frac{\overline{m}_Q(u) \cdot Q(u)}{\overline{m}_Q(u)}$$

where $\overline{m}_P(u)$ and $\overline{m}_Q(u)$ are new mass distribution functions defined by

$$\overline{m}_P(u) = \sum_{k=0}^{p} \overline{w}_k N_k^n(u) \text{ and } \overline{m}_Q(u) = \sum_{k=0}^{p} \overline{v}_k N_k^n(u) \qquad (5)$$

Here $\overline{w}_k$ and $\overline{v}_k$ are additional positive masses attached to each control point of $P(u)$ and $Q(u)$. The modified curves $\overline{P}(u)$ and $\overline{Q}(u)$ maintain the same shapes and positions as the original curves $P(u)$ and $Q(u)$, but exhibit new mass distributions controlled by $\overline{w}_k$ and $\overline{v}_k$. The weighted averaging between $\overline{P}(u)$ and $\overline{Q}(u)$ is now expressed as

$$M_{new}(u,t) = \frac{(1-t)\overline{m}_P(u)P(u) + t\overline{m}_Q(u)Q(u)}{(1-t)\overline{m}_P(u) + t\overline{m}_Q(u)} \qquad (6)$$

Notice that by choosing different control weights $\overline{w}_k$ and $\overline{v}_k$, we can reproduce our previous morphs between $P(u)$ and $Q(u)$ using weighted averages of $\overline{P}(u)$ and $\overline{Q}(u)$. For instance, if the control weights are set to unit mass, (6) reduces to linear averaging. If instead we use the weights associated with the control points of $P(u)$ and $Q(u)$, we get the same morph induced by weighted averages of $P(u)$ and $Q(u)$. Also, the morph generated by uniformly scaling the masses of $P(u)$ can be reproduced by setting $\overline{w}_k$ equal to the scaled weights associated with the control points of $P(u)$.

This new representation allows us to manipulate the morphing behavior of different regions on the targets through a small set of scalar parameters. The B-spline form of the new mass distribution also guarantees smooth variation of masses, so that no abrupt changes in the attracting forces will occur during the morph.

Similarly, we can now represent two bi-degree (m,n) rational B-spline surfaces $P(u,v)$ and $Q(u,v)$ as

$$\overline{P}(u,v) = \frac{\overline{m}_P(u,v) \cdot P(u,v)}{\overline{m}_P(u,v)} \text{ and } \overline{Q}(u,v) = \frac{\overline{m}_Q(u,v) \cdot Q(u,v)}{\overline{m}_Q(u,v)}$$

where $\overline{m}_P(u,v)$ and $\overline{m}_Q(u,v)$ are new mass distribution functions defined by

$$\overline{m}_P(u,v) = \sum_{j=0}^{p}\sum_{k=0}^{q} \overline{w}_{jk} N_j^m(u) N_k^n(v), \text{ and}$$

$$\overline{m}_Q(u,v) = \sum_{j=0}^{p}\sum_{k=0}^{q} \overline{v}_{jk} N_j^m(u) N_k^n(v).$$

Here $\overline{w}_{jk}$ and $\overline{v}_{jk}$ are positive weights attached to each control point of $P(u,v)$ and $Q(u,v)$ that determine the new mass distributions. The morph is then computed as the weighted average between $\overline{P}(u,v)$ and $\overline{Q}(u,v)$. An example is shown in figure 3, where new control weights are assigned to the control points of the targets representing the letters **E** and **G** from the Eurographics logo. In the top sequence, the first target expresses higher masses towards the right, while the second target expresses higher masses towards the left. Hence the transition shape in the middle is attracted to the letter **E** on its right portion and to the letter **G** on its left portion, creating a connected **E**. In the bottom sequence, control weights are higher at the bottom of the first target and at the top of the second target, pulling the middle shape in a different manner and generating a letter **G** with a right-angled corner.

4.3. User-controlled morphing

The freedom to choose $\overline{w}_k$ and $\overline{v}_k$ ($\overline{w}_{jk}$ and $\overline{v}_{jk}$) certainly gives us the power of control. However, we need an intuitive way to compute these control weights so that the transition

curves (surfaces) are drawn to the *desired* target to the *desired* extent. We start by investigating the influence of the control weights on the shape of the morph. In analogy with our analysis of the effect of masses on weighted averaging, we can rewrite $M_{new}(u,t)$ in the form of an affine combination of $P(u)$ and $Q(u)$ by setting

$$M_{new}(u,t) = (1-D(u,t))P(u) + D(u,t)Q(u)$$

where

$$D(u,t) = \frac{t\overline{m}_Q(u)}{(1-t)\overline{m}_P(u)+t\overline{m}_Q(u)} \tag{7}$$

The value $D(u,t) \in [0,1]$ describes the *normalized* distance of a point on the transition curve from the first target at time t. Substituting (5) into (7), we get

$$D(u,t) = \frac{\sum_{k=0}^{p} W_k R_k N_k^n(u)}{\sum_{k=0}^{p} W_k N_k^n(u)}$$

where

$$R_k = \frac{t\bar{v}_k}{W_k} \;\; and \;\; W_k = (1-t)\overline{w}_k + t\bar{v}_k$$

Therefore the normalized distances at a given time form a rational B-spline curve controlled by the scalars R_k with weights W_k, both determined by the control weights on the targets. Conversely, we can express $\overline{w}_k$ and $\bar{v}_k$ in terms of the scalars R_k and weights W_k at time t as

$$\overline{w}_k = \frac{W_k(1-R_k)}{1-t} \; and \; \bar{v}_k = \frac{W_k R_k}{t} \tag{8}$$

Since R_k and W_k act as B-spline control points and weights on the shape of the transition curve (with respect to the two targets), users can specify these values in a way similar to modelling a rational B-spline curve. The control weights on the targets can be computed by equation (8) to produce a morph that interpolates the desired transition curve at any time.

Now morph design is an interactive process as illustrated in figure 4. On the left, we start with an existing morph with pre-chosen $\overline{w}_k$ and $\bar{v}_k$ (such as the original weights associated with the control points). Then we select a transition curve at some time t, shown in the middle, and adjust the corresponding normalized distance curve (by moving the scalars R_k between 0 and 1 or changing the associated weights W_k). The new control weights are computed automatically so that the transition curve at time t follows the new normalized distances, as seen on the right. This process can be repeated at different values of t until a desirable morph is generated.

Notice from (8) that for any $R_k \in (0,1)$ and positive W_k at time $t \in (0,1)$, the control weights $\overline{w}_k$ and $\bar{v}_k$ are always uniquely determined. The modified morph using weighted averaging always represents a smooth, monotonic transformation from the first target to the second.

We can customize the morph between two rational B-spline surfaces $\overline{P}(u,v)$ and $\overline{Q}(u,v)$ in a similar fashion. The normalized distances of the points on the transition surface at time t can be expressed by

$$D(u,v,t) = \frac{\sum_{j=0}^{p}\sum_{k=0}^{q} W_{jk} R_{jk} N_j^m(u) N_k^n(v)}{\sum_{j=0}^{p}\sum_{k=0}^{q} W_{jk} N_j^m(u) N_k^n(v)}$$

where

$$R_{jk} = \frac{t\bar{v}_{jk}}{W_{jk}} \;\; and \;\; W_{jk} = (1-t)\overline{w}_{jk} + t\bar{v}_{jk}$$

These normalized distances describe a rational B-spline surface defined by the scalars R_{jk} with weights W_{jk}. The control weights $\overline{w}_{jk}$ and $\bar{v}_{jk}$ on the targets can be expressed in terms of R_{jk} and W_{jk} at time t by

$$\overline{w}_{jk} = \frac{W_{jk}(1-R_{jk})}{1-t} \; and \; \bar{v}_{jk} = \frac{W_{jk}R_{jk}}{t} \tag{9}$$

Users can design the control weights (and thus the resulting morph) by manipulating the scalars and weights that control the normalized distances of a transition surface. An example is shown in figure 6, where different morphs between two target faces are generated using this design paradigm. For comparison, the targets are first morphed using linear averaging, as shown at the top. In the next two morphs, weighted averaging is used with different control weights attached to the targets. These control weights are computed from the user-defined R_{jk} and W_{jk} that generate the normalized distances $D(u,v,t)$ at $t = 0.5$, shown on the left of figure 5. The green regions on the normalized distance surfaces indicate where the attracting force from the second target is stronger, as reflected in the bulging nose (figure 5 top-right) and cheeks (figure 5 bottom-right) of the corresponding tran-

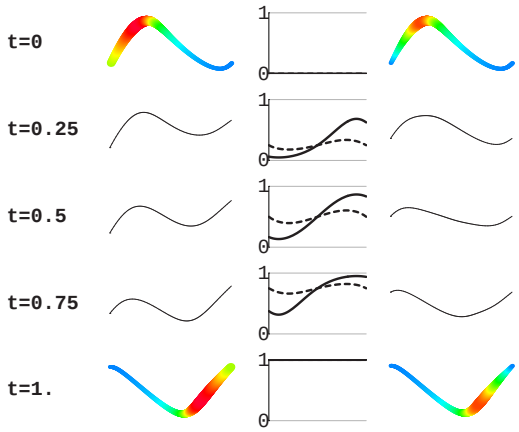

Figure 4: *Designing a morph (right) from an existing morph (left). The normalized distances are plotted as solid lines for the first morph and dotted lines for the second morph.*

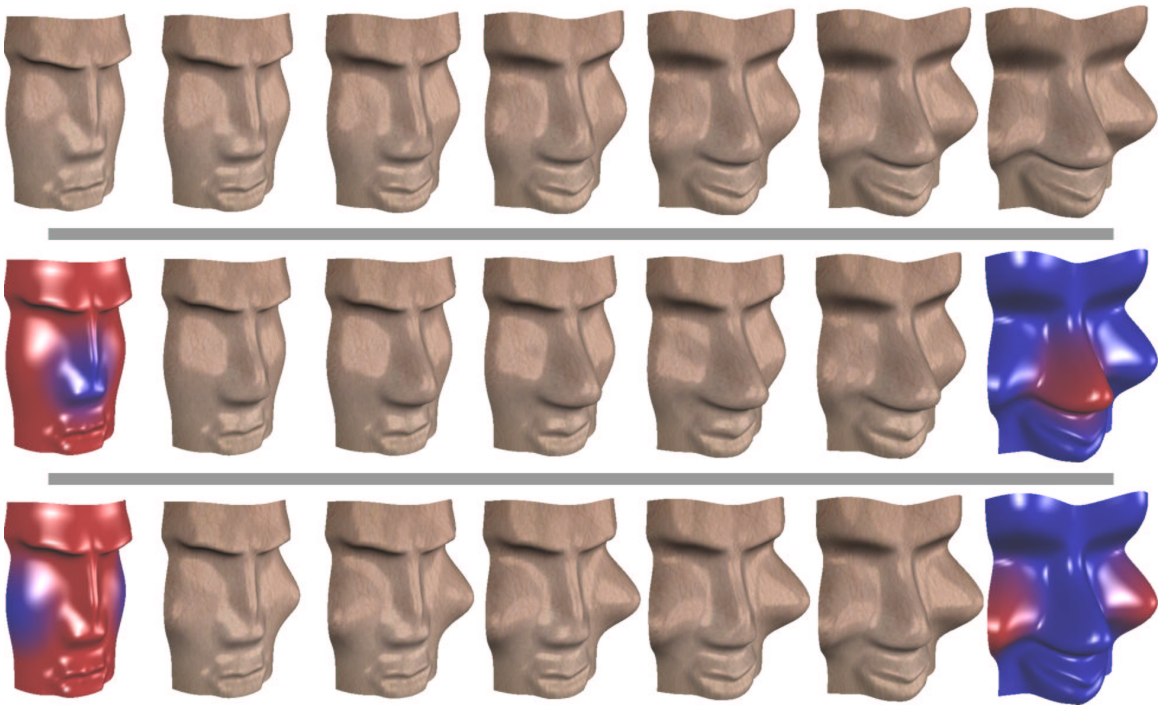

Figure 6: *Morph between two faces using linear averaging (top) and weighted averaging after applying different non-uniform mass modifications (middle and bottom).*

sition surfaces at $t = 0.5$. The resulting morphs thus produce the effect that either the nose comes out faster (figure 6 middle) or the cheeks bulge first (figure 6 bottom).

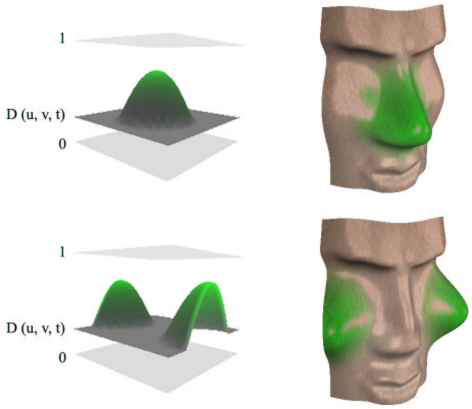

Figure 5: *Normalized distances (left) and corresponding transition surfaces (right) at $t = 0.5$ in the second and third morphs from figure 6.*

4.4. User interface design

Our morphing examples were created using a graphical interface (shown in figure 7) that we implemented to facilitate the design of surface morphs. The interface consists of a main window $W1$ for viewing the transition surfaces, a sub window $W2$ showing the corresponding normalized distance plane, and a panel $W3$ for selecting control scalars.

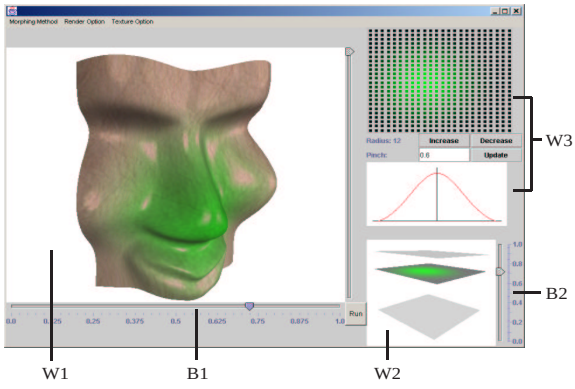

Figure 7: *The user interface for designing surface morphs.*

The user can pick a transition surface at any time parameter t (by sliding bar $B1$), and modify the corresponding normalized distance plane by selecting control scalars R_{jk} (in $W3$) and moving them between 0 and 1 (using bar $B2$); the weights W_{jk} are left unchanged. The regions affected by the selected control scalars are colored green on both the transition surface and the normalized distance plane. To facilitate dealing with a large number of control scalars, the interface allows the user to define an area of impact in the parameter space (shown at the top of $W3$). The user moves the control scalars only at the center of the selected area; the remaining scalars will be modified automatically according to a user-designed falloff function (shown at the bottom of $W3$).

As the normalized distances are being modified in $W2$, the changes are reflected immediately on the transition surface in $W1$, and the entire morphing sequence is updated on the fly by computing the new control weights $\overline{w}_{jk}$ and $\overline{v}_{jk}$ as in (9). Once a desirable morph is generated, the result can be conveniently stored by these control weights on the targets.

Although the morph is represented internally by weighted averages using mass distributions, the user requires no knowledge of masses or B-splines to accomplish morph design using this interface.

5. Applications

Computer animation: Morphing by weighted averaging can be applied directly to computer animations involving rational parametric curves and surfaces. As seen in figure 6, by varying the mass distributions on the targets, we can produce a variety of animations in which different regions on the surface animate at different rates. The power of local control offered by the mass distributions can also augment existing techniques, such as key-frame interpolation, to produce more vivacious animations.

Model design: Weighted averaging is also useful for designing models from existing samples. By choosing appropriate mass variations on the samples (i.e., target shapes), the model (i.e., transition shape) will resemble the samples at regions with larger mass. Therefore we can exert local control over the similarity of the new model to existing samples at different regions. This technique can be extended without difficulty to morphing multiple targets, which allows this design paradigm to be performed on more than two sample models.

6. Conclusion

We have presented a framework for smooth, non-uniform morphing of rational B-spline curves and surfaces by weighted averaging using associated mass distributions. Weighted averaging reduces undesired flatness and wiggles, and has the added advantage over linear averaging that it gives the user local control over the morph in different regions on the targets. We also provide the user with an intuitive way to execute this control by manipulating the transition curves (surfaces); no knowledge of B-splines is required. The only computations involved in calculating the mass changes and the resulting vertex paths from user inputs are simple algebraic operations, which makes possible a real-time, locally controlled morph editing environment.

References

1. M. Alexa. Local Control for Mesh Morphing. *Shape Modeling International '01 Proceedings*, pp. 209–215 2001. 2
2. M. Alexa. Mesh Morphing. *Computer Graphics Forum*, **21**(2): 173-196, 2002. 1
3. M. Alexa, D. Cohen-Or, and D. Levin. As Rigid as Possible Shape Interpolation. *Proceedings of SIGGRAPH 2000*, pp. 157–164, 2000. 1, 2
4. W. Boehm. Inserting New Knots into B-spline Curves. *Computer-Aided Design*, **12**(4): 199-201, 1988. 3
5. E. Cohen, T. Lyche, R. Riesenfeld. Discrete B-splines and Subdivision Techniques in Computer Aided Geometric Design and Computer Graphics. *Computer Graphics and Image Processing*, **14**(2): 87-111, 1980. 3
6. A. Efrat, L.J. Guibas, S. Har-Peled and T.M. Murali. Morphing between Polylines. *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 680-689, 2001. 2
7. G. Farin. *Curves and Surfaces for CAGD: A Practical Guide* . Academic Press, 5th edition, 2001. 1
8. J.C. Fiorot and P. Jeanin. *Rational Curves and Surfaces: Applications to CAD*. Wiley & Sons, Chileston, England, 1992. 2
9. R.N. Goldman. The Ambient Spaces of Computer Graphics and Geometric Modeling. *IEEE Computer Graphics & Applications*, **20**: 76–84, 2000. 2, 3
10. C. Gotsman and V. Surazhsky. Guaranteed Intersection-free Polygon Morphing. *Computers & Graphics*, **25**(1): 67–75, 2001. 1
11. A. Kaul and J. Rossignac. Solid-Interpolating Deformations: Construction and Animation of PIPS. *Computers & Graphics*, **16**(1): 107-115, Sept 1992. 1
12. J. Kent, W.E. Carlson, and R.R. Parent. Shape Transformation for Polyhedral Objects. *Computer Graphics (Proceedings of SIGGRAPH 1992)*, **26**(2): 47–54, 1992. 1
13. H. Prautzsch and B. Piper. A Fast Algorithm To Raise the Degree of B-spline Curves. *Computer Aided Geometric Design*, **8**(4): 253–266, 1991. 3
14. J. Rossignac and A. Kaul. AGRELs and BIPs: Metamorphosis as a Bezier curve in the space of polyhedra. *Computer Graphics Forum*, **13**(3): 179-184, Sept 1994. 2
15. T.W. Sederberg and E. Greenwood. Shape Blending of 2-D Piecewise Curves. *Mathematical Methods for Curves and Surfaces*, pp. 497–506, 1995. 1
16. T.W. Sederberg, P. Gao, G. Wang, and H. Mu. 2D Shape Blending: An Intrinsic Solution to the Vertex Path Problem. *Proceedings of SIGGRAPH 1993*, pp. 15–18, 1993. 1
17. M. Shapira and A. Rappoport. Shape Blending Using the Star-skeleton Representation. *IEEE Computer Graphics & Applications*, **15**(2): 44–50, 1995. 1
18. M. Zockler, D. Stalling and H.C. Hege. Fast and Intuitive Generation of Geometric Shape Transitions. *The Visual Computer*, **16**(5): 241–253, 2000. 1