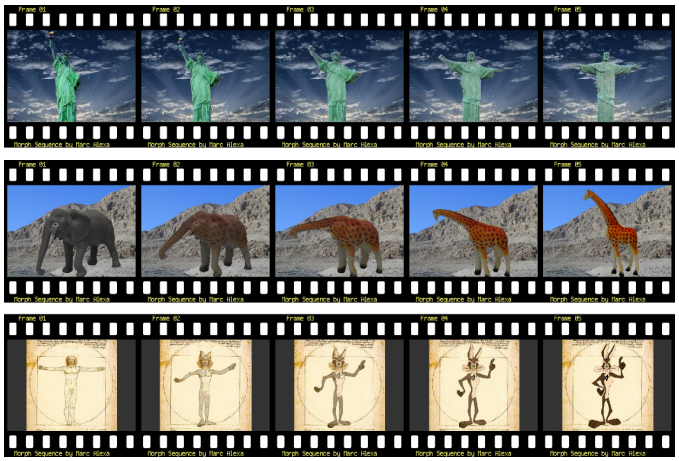
Chapter 4

Interpolating corresponding shapes



Using the methods presented in the previous chapters, it is possible to generate one topological shape representation for several shapes. Topology, in this context, refers to the number and structure of primitives used to describe the shape, e.g. the number of line segments in a polygon or the connectivity information of a mesh. In any case, the shape is represented as the coordinates of vertices.

A morph sequence contains several states of an object transforming from one state to another. The natural idea to generate morphs is to interpolate representations of shapes. The easiest way is to interpolate coordinates of vertices as they are readily available. However, as with corresponding features of the shapes, the human observer has certain expectations regarding interpolated shapes. It is difficult to define a set of rules which have to be followed. Different approaches to the interpolation of shapes are characterized by different conditions, which are believed to describe natural shape interpolation.

The coefficients describing such linear interpolation are sometimes called transition parameters. In classical morphing applications, the transition parameter can be represented by a single scalar ranging from 0 (representing the source shape) to 1 (representing the target shape) and is mentally connected to time values in an animated sequence.

4.1 Linear Interpolation of Boundary and/or Orientation

The easiest way to produce blends of corresponding shapes is to interpolate the coordinates of vertices. Assume a set of n shapes is represented by a topological structure $\mathcal{S}$ and vectors V_i containing vertex coordinates. $\mathcal{S}$ might be polygon, a skeleton, a triangulation, a mesh, a tetrahedralization, or some other structural description of – at least – the shape's boundary. The V_i's just contain real numbers.

Given a transition parameter t_i the coordinates of an interpolated shape are computed by

$$V = \sum_i t_i V_i \tag{4.1}$$

This is the easiest way of computing interpolated shapes. It produces good results if the shapes have the same orientation and are somewhat similar. Figures 4.1, 4.2, and 4.3 show morph sequences obtained by linear interpolation.

Different orientation could lead to displeasing results. Imagine two squares that are rotated by 180 degrees against each other. If simple vertex interpolation is applied in this configuration, the interpolated shapes will shrink until the shape is collapsed to one point and then grow again. This is not the desired result in most applications. It is advisable to interpolate the orientation separately from the vertex coordinates.

4.1.1 Interpolation of orientation

Several ways exist to compute a relative orientation of two shapes. Note that it is difficult to interpolate the orientation of more than two shapes in 3D so the following discussion will be restricted to two shapes.

As a first step, the shapes are usually translated so that their centers of mass coincide with the origin. Then, a rotation [Cohen-Or & Carmel 1998; Cohen-Or et al. 1998] or an affine transform [Alexa 2000] is computed seprating the rigid/affine part from the elastic part of the morph. A way of defining the rigid/affine part is to minimize the squared distances of corresponding vertices using the corresponding transform. The minimization problem of finding an affine transform can be solved using the pseudo inverse of the coordinate vector. Let the vertex vectors be

arranged as a $n \times 3$ matrix

$$V = \begin{pmatrix} v_{1_x} & v_{1_y} & v_{1_z} \\ v_{2_x} & v_{2_y} & v_{2_z} \\ v_{3_x} & v_{3_y} & v_{3_z} \\ \dots & & \end{pmatrix}.$$

Then the squared distance of coordinates under an affine transform A is

$$(V(0)A - V(1))^2 \tag{4.2}$$

and has to be minimized. This leads to linear system of equations, which can be solved using pseudo inverse $V(0)^+$:

$$A = V(0)^+ V(1) = \left(V(0)^T V(0)\right)^{-1} V(0)^T V(1) \tag{4.3}$$

Alternatively, the least squares solution (or, the pseudo inverse) could be computed using the SVD, which allows explicit control over the sensitivity to near rank deficiencies [Golub & Van Loan 1989].

Intermediate shapes $V(t) = \{\mathbf{v_1}(t), \mathbf{v_2}(t), \dots\}$ are described as $V(t) = A(t)V(0)$. The question is how to define $A(t)$ reasonably? The simplest solution would be: $A(t) = (1-t)I + tA$. However, some properties of $A(t)$ seem to be desirable, calling for a more elaborate approach:

- The transformation should be symmetric.
- The rotational angle(s) and scale should change monotonic.
- The transform should not reflect.
- The resulting paths should be simple.

The basic idea is to factor A into rotations (orthogonal matrices) and scale-shear parts with positive scaling components. We have examined several decompositions [Alexa et al. 2000]. Through experimentation, wehave found a decomposition into a single rotation and a symmetric matrix (i.e. the polar decomposition), to yield the visually-best transformations. This result is supported by Shoemake & Duff [1992] for mathematical, as well as psychological, reasons. The decomposition can be deduced from the SVD as follows

$$\begin{aligned} A &= R_\alpha D R_\beta = R_\alpha (R_\beta R_\beta^T) D R_\beta = \\ &\quad (R_\alpha R_\beta)(R_\beta^T D R_\beta) = R_\gamma S \end{aligned} \tag{4.4}$$

however, there are computationally cheaper alternatives [Shoemake & Duff 1992]. Based on the decomposition, $A(t)$ is computed by linearly interpolating the free parameters in the factorizations in (4.4) , i.e.

$$A_\gamma(t) = R_{t\gamma}((1-t)I + tS). \tag{4.5}$$

Figure 4.4 illustrates the resulting transformations for a triangle. For comparison, 4.4(a) shows linear interpolation of vertex coordinates. The transformation resulting from a singular value decomposition and linear interpolation $A_{\alpha,\beta}(t)$ is depicted in 4.4(b). Note that the result is symmetric and linear in the rotation angle but still unsatisfactory, since a rotation of more than π is unnecessary. However, if we subtract 2π from one of the angles (depicted in 4.4(c)) the result is even more displeasing. We have found that decomposing A into one rotation and a symmetric matrix and using $A_\gamma(t)$ yields the best results (Figure 4.4(d)). It avoids unnecessary rotation or shear compared to the SVD and is usually more symmetric than a QR decomposition-based approach.

4.2 Interpolation of intrinsic boundary representation

Linear interpolation of vertices can lead to undesirable effects such as shortening of parts of the boundary during the transition. To avoid such problems, Sederberg et al. [1993] propose to interpolate an intrinsic representation of the boundary. For polygons, such an intrinsic representation is edge length an interior angles. Unfortunately, there is no simple analogue in 3D. An attempt was made to extend the ideas of to polyhedra [Sun et al. 1997] but the methods are computationally expensive and unreliable. Here, only the 2D case will be explained.

Assume the polygons P and Q are described by their vertex positions $\mathbf{p_i}, \mathbf{q_i}$. Let $\theta_{p_i}, \theta_{Q_i}$ be the interior angles around $\mathbf{p_i}, \mathbf{q_i}$ and

$$L_{P_i} = |\mathbf{p_{i+1}} - \mathbf{p_i}|, L_{Q_i} = |\mathbf{q_{i+1}} - \mathbf{q_i}| \tag{4.6}$$

the length of the i-th edge. Additionally, let $\alpha_{P_i}, \alpha_{Q_i}$ be the angles between the i-th edge and a fixed axis. An intermediate polygon is represented by

$$L_i(t) = (1-t)L_{P_i} + L_{Q_i} \tag{4.7}$$
$$\theta_i(t) = (1-t)\theta_{P_i} + \theta_{Q_i} \tag{4.8}$$
$$\alpha_i(t) = (1-t)\alpha_{P_i} + \alpha_{Q_i} \tag{4.9}$$

However, this description will lead to an open polygon in the general case.

4.2.1 Closing the polygon

The idea is to close the polygon by small changes of the defining parameters. Since the edge length has to be changed in some cases and, on the other hand, it is easier to change only one of three parameters, only the edge lengths will be changed to close the polygon.

The interpolated edge becomes

$$L_i(t) = (1-t)L_{P_i} + L_{Q_i} + S_i \tag{4.10}$$

To uniquely determine the S_i, the squared relative length change

$$g(S_0, \ldots, S_n) = \sum_i \frac{S_i^2}{|L_{P_i} - L_{Q_i}|^2 + \epsilon}, \quad \epsilon > 0 \tag{4.11}$$

will be minimized under the constraint that the polygon has to be closed. Closure of the polygon can be formulated by the following necessary conditions:

$$\phi_1(S_0, \ldots, S_n) = \sum_i ((1-t)L_{P_i} + L_{Q_i} + S_i) \cos\alpha_i = 0 \tag{4.12}$$

$$\phi_2(S_0, \ldots, S_n) = \sum_i ((1-t)L_{P_i} + L_{Q_i} + S_i) \sin\alpha_i = 0 \tag{4.13}$$

This kind of minimization problem can be solved using Lagrange multipliers.

$$\Phi(\lambda_1, \lambda_2, S_0, \ldots, S_n) = g + \lambda_1\phi_1 + \lambda_2\phi_2 \tag{4.14}$$

This leads to the surprsingly compact solution

$$S_i = -\frac{1}{2} |L_{P_i} - L_{Q_i}|^2 (\lambda_1 \cos\alpha_i + \lambda_2 \sin\alpha_i) \tag{4.15}$$

4.3 Interpolation of differential boundary representation

In classical morphing applications the transition parameter can be represented by a single scalar ranging from 0 (representing the source shape) to one (representing the target shape) and is mentally connected to time values in an animated sequence.

Now we want to locally morph certain features or regions of interest, i.e. the transition parameters are different for different vertices. We will call the set of transition parameters for vertices the *transition state*. A major problem when morphing only locally arises from the fact that corresponding features might not have the same position in space and, thus, interpolation of absolute coordinate could lead to undesirable effects. This problem is illustrated in Figure 4.5 The shapes in a) and b) are source and target geometry of one mesh. The idea is to locally change the geometry of the baby's face so that the nose takes the shape of the boy's. Locally interpolating vertex coordinate leads to the shape depicted in c), which is clearly not usable. Note that the faces are overall aligned in space and that the misalignment of the noses results from different relative positions in the faces.

We could ease the problem of misalignment by assigning an affine transform to a local morph. However, this leads to problems when features overlap. More generally, a shape should be defined by the transition state of its vertices. In that way, the transition states is representative for the shape of a morphable object. This could be a very compact way of representing deforming or animated objects.

The main idea to overcome the mentioned problems is to represent vertex coordinates with respect to their neighbors in the mesh. Given a vertex and its one-neighborhood ring (see Figure 2 a), the position should be described relative to

the positions of vertices in the neighborhood ring. Further, the representation of a vertex should be linear in the absolute coordinates. Non-linear functions tend to be numerically difficult to handle and many morphable meshes have sliver triangles, which, together, leads to unpredictable results.

The relative representation aims at making the shape of the mesh invariant to translation or, ideally, invariant under affine transforms. If a vertex were represented in the affine space of its neighbors invariance under affine transforms would trivially follow. Floater and Gotsman have shown how to use such representations to morph planar triangulations [Floater & Gotsman 1999]. The extension to triangulations in $\mathbb{R}^3$is difficult because vertices of a the neighborhood are not necessarily affinely independent.

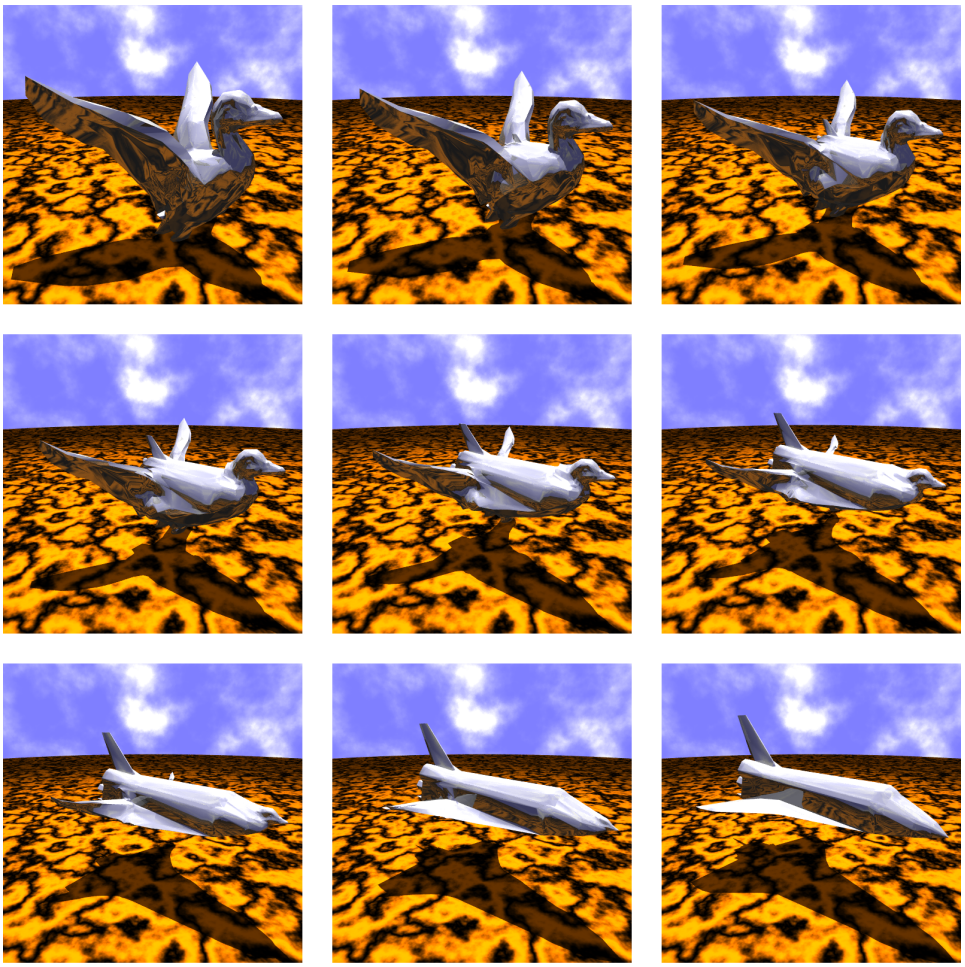


Figure 4.1: A morph sequence obtained by linear interpolation of merged embeddings on the sphere.

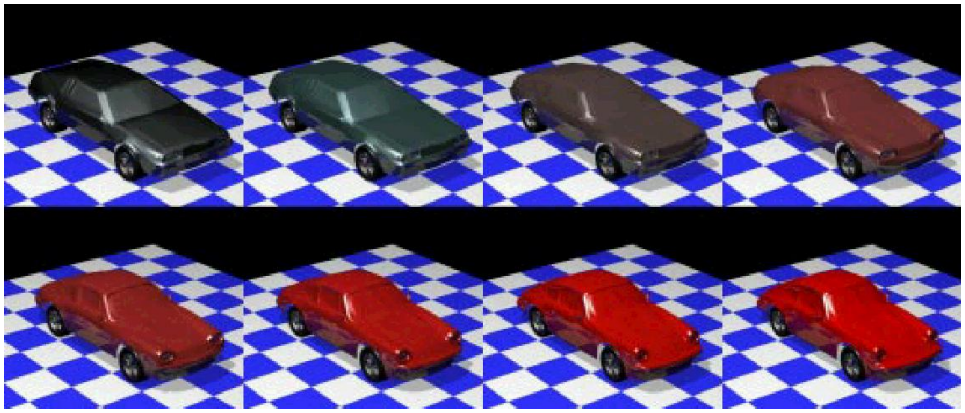


Figure 4.2: A morph sequence obtained by linear interpolation using the base domains depicted in Figure 4.3. Reprinted from Kanai et al. [2000].

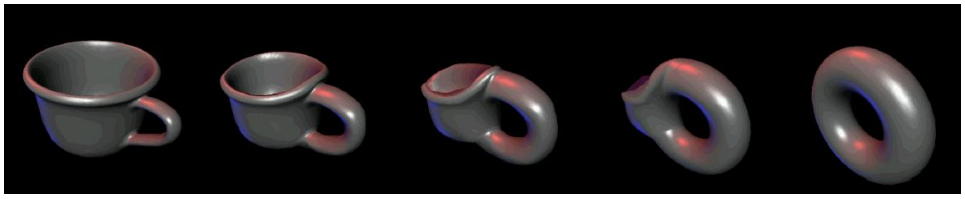


Figure 4.3: A morph of objects with genus higher than zero. Reprinted from Lee et al. [1999].

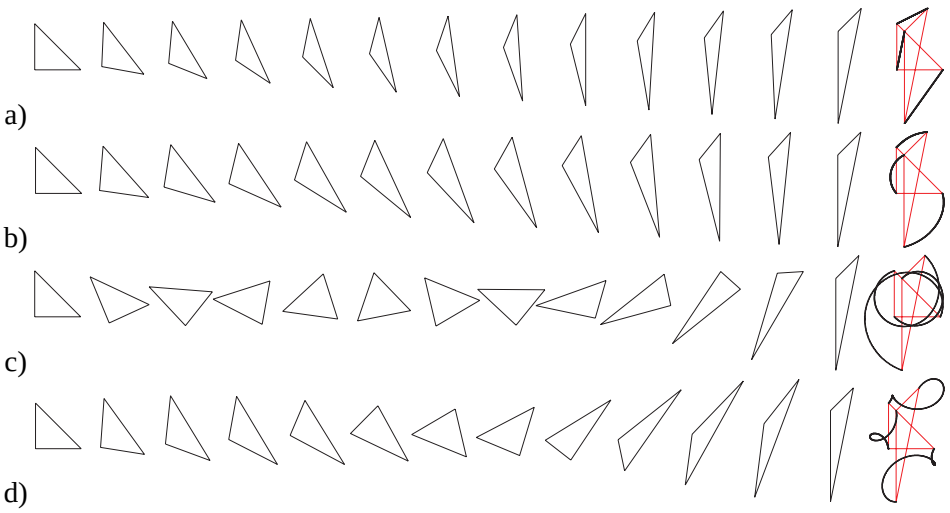


Figure 4.4: Transformations of a single triangle. (a) Linear vertex interpolation. (b-d) An affine map from the source to the target triangle is computed and factored into rotational and scale-shear parts. Intermediate triangles are constructed by linearly interpolating the angle(s) of rotation, the scaling factors, and the shear parameter. (b) is generated using the SVD; (c) shows the results of reducing the overall angle of (b) by subtracting 2π from one of the angles; (d) corresponds to Equation 4.5 and represents the method of our choice. The last column in all rows shows plots of the vertex paths.

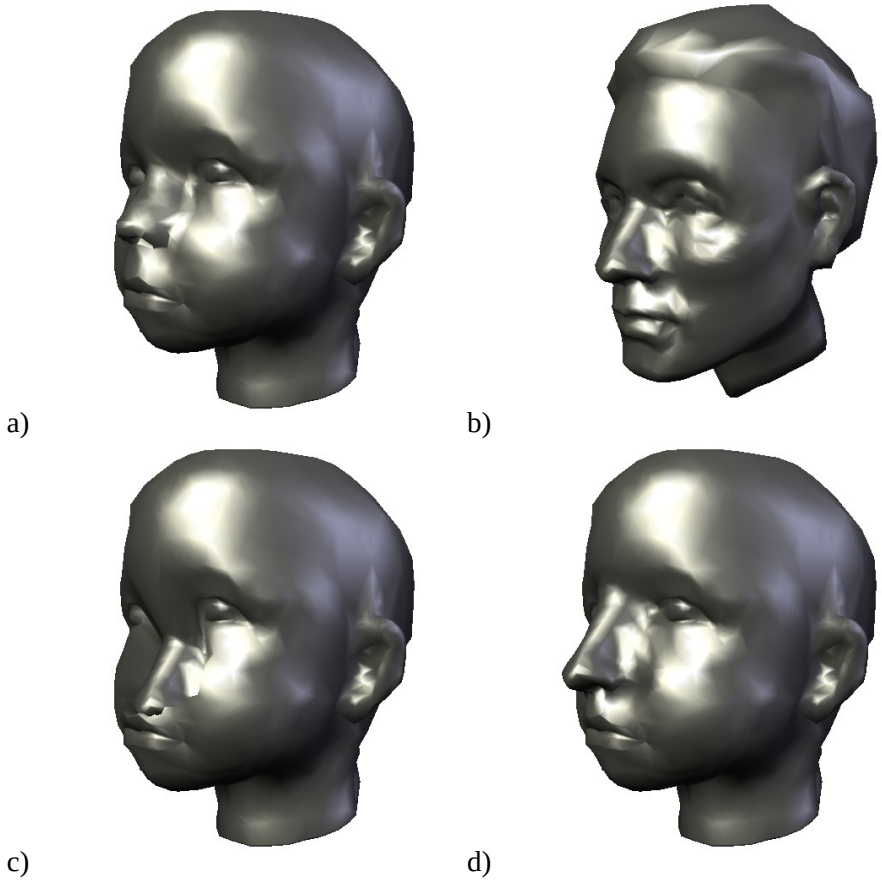


Figure 4.5: Given a mesh with two geometries a) and b) so that corresponding features (eyes, ears, nose, mouth, etc.) are represented by the same vertices in both geometries. If one feature (in this example the nose) is morphed towards the target geometry in absolute coordinates, different positions in space lead to undiserable effects shown in c). The shape in d) shows a more pleasing result achieved by interpolating a differential encoding of the vertices.