As-Rigid-As-Possible Shape Interpolation

Marc Alexa
Darmstadt University of Technology

Daniel Cohen-Or
Tel Aviv University

David Levin
Tel Aviv University

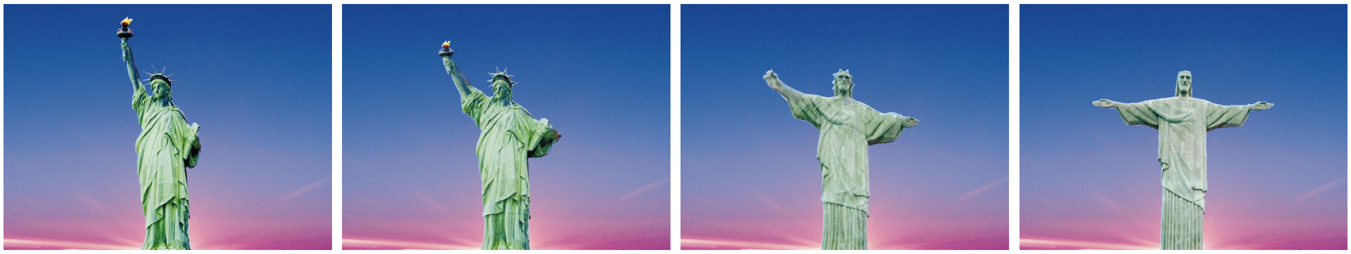


Figure 1: The Statue of Liberty becomes the Christ statue on the Corcovado.

Abstract

We present an object-space morphing technique that blends the interiors of given two- or three-dimensional shapes rather than their boundaries. The morph is rigid in the sense that local volumes are least-distorting as they vary from their source to target configurations. Given a boundary vertex correspondence, the source and target shapes are decomposed into isomorphic simplicial complexes. For the simplicial complexes, we find a closed-form expression allocating the paths of both boundary and interior vertices from source to target locations as a function of time. Key points are the identification of the optimal simplex morphing and the appropriate definition of an error functional whose minimization defines the paths of the vertices. Each pair of corresponding simplices defines an affine transformation, which is factored into a rotation and a stretching transformation. These local transformations are naturally interpolated over time and serve as the basis for composing a global coherent least-distorting transformation.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; I.4.7 [Image Processing]: Feature Measurement—Size and shape

Keywords: shape blending, vertex path problem, compatible triangulation

1 Introduction

Techniques that transform one graphical object into another have gained widespread use in recent years. These techniques, known as *morphing* or *blending* techniques, involve the creation of a smooth transition from an initial object to a target object. They have proven to be powerful for visual effects, and there are now many breathtaking examples in film and television. In most cases, these morphing effects are generated using image-based techniques [3, 21, 30] where the geometry of the source and target object are somewhat similar, or the process requires extensive user labor. A primary challenge in this area is to devise algorithms to blend two given objects of significantly different shape, with minimal user interaction, so that the in-between objects retain their original appearance and properties as much as possible. The morph example in Figure 1 is difficult to achieve with an image-based technique, because foreground and background behave differently.

Object-space morphing treats explicit representations of the objects [14, 20] instead of discretizations of space (images, volumes). Assuming the objects are either polygons (in 2D) or polyhedra (in 3D), the morphing process consists of generating a correspondence between the geometric features of the representation, known as the *vertex correspondence* problem [23, 19, 6, 18, 14, 1, 20], and then interpolating the positions of the boundary representations along predetermined paths, known as the *vertex path* problem [24, 25, 13, 8].

Most of the cited object-space morphing techniques are concerned with the correspondence problem while simply linearly interpolating the corresponding vertices, not taking into account that the blended shapes are implicitly representing meta-physical entities. Generally speaking, aesthetic and intuitive shape blending should aim at treating the objects as rigidly as possible and avoid superfluous global or local deformations. In [24], Sederberg intro-

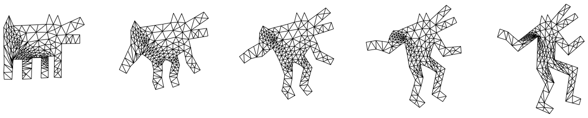
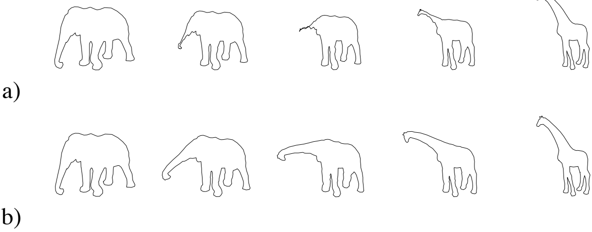


Figure 2: Morph sequence of Haring-like figures showing their homeomorphic dissections


a)
b)

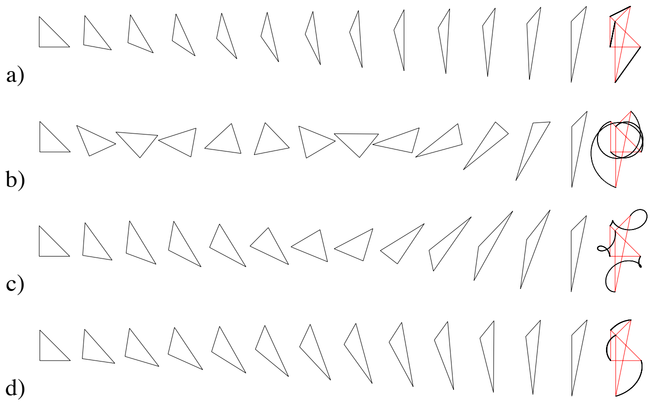


Figure 5: Transformations of a single triangle. (a) Linear vertex interpolation. (b-d) An affine map from the source to the target triangle is computed and factored into rotational and scale-shear parts. Intermediate triangles are constructed by linearly interpolating the angle(s) of rotation, the scaling factors, and the shear parameter. (b) corresponds to Equation 4; (c) shows the results of reducing the overall angle of (b) by subtracting 2π from one of the angles; (d) corresponds to Equation 5 and represents the method of our choice. The last column in all rows shows plots of the vertex paths.

We do not take the translation $\vec{l}$ into account for shape interpolation since it does not describe any property of the shape itself except for its placing in the scene. Rather, we want to describe intermediate shapes by varying the rotational and scaling parts comprising A, over time. Note that the coefficients of A are linear in the coordinates of the target shape.

Intermediate shapes $V(t) = (\vec{v_1}(t), \vec{v_2}(t), \vec{v_3}(t))$ are described as $V(t) = A(t)P$. 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 with respect to t.
- The rotational angle(s) and scale should change linearly.
- The triangle should keep its orientation, i.e. should not be reflected.
- The resulting vertices' 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. A natural candidate is the singular value decomposition (SVD) [12], since the resulting decomposition is symmetric and well-defined for arbitrary dimensions:

$$A = R_\alpha D R_\beta = R_\alpha \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} R_\beta, \quad s_x, s_y > 0 \qquad (2)$$

However, through experimentation, we have found a decomposition into a single rotation and a symmetric matrix, to yield the visually-best transformations. This result is supported by Shoemake in [27] for mathematical, as well as psychological, reasons. The decomposition can be deduced from the SVD as follows:

$$\begin{aligned} A &= R_\alpha D R_\beta = R_\alpha (R_\beta R_\beta^T) D R_\beta = \\ &\quad (R_\alpha R_\beta)(R_\beta^T D R_\beta) = R_\gamma S = R_\gamma \begin{pmatrix} s_x & s_h \\ s_h & s_y \end{pmatrix} \end{aligned} \qquad (3)$$

with $s_x, s_y > 0$. Based on the decomposition, $A(t)$ is computed by linearly interpolating the free parameters in the factorizations in (2) and (3) , i.e.

$$A_{\alpha,\beta}(t) = R_{t\alpha}((1-t)I + tD)R_{t\beta} \qquad (4)$$

and respectively,

$$A_\gamma(t) = R_{t\gamma}((1-t)I + tS) \qquad (5)$$

Figure 5 illustrates the resulting transformations from a source to a target triangle. For comparison, 5(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 5(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 1(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 5(d)). It avoids unnecessary rotation or shear compared to the SVD and is usually more symmetric than a QR decomposition-based approach. Note that the rotation of the triangle does not contribute to its shape. However, this is no longer true for more than a single triangle, as we shall see in the next section, which discusses the generalization to more than one triangle.

2.2 Closed-Form Vertex Paths for a Triangulation

We now consider a triangulation $\mathcal{T} = \{T_{\{i,j,k\}}\}$ rather than a single triangle. Each of the source triangles $P_{\{i,j,k\}} = (\vec{p_i}, \vec{p_j}, \vec{p_k})$ corresponds to a target triangle $Q_{\{i,j,k\}} = (\vec{q_i}, \vec{q_j}, \vec{q_k})$. For each pair of triangles, we compute a mapping $A_{\{i,j,k\}}$, which can be factored by Eq. 5 to determine $A_{\{i,j,k\}}(t)$. Since most of the vertices correspond to more than one triangle, a mapping of all vertices could not (in general) be conforming with all the individual ideal transformations $A_{\{i,j,k\}}(t)$.

Let

$$V(t) = (\vec{v_1}(t), \ldots, \vec{v_n}(t)), t \in [0,1] \qquad (6)$$

be the desired paths of the vertices, satisfying

$$\begin{aligned} V(0) &= (\vec{p_1}, \ldots, \vec{p_n}) \\ V(1) &= (\vec{q_1}, \ldots, \vec{q_n}). \end{aligned}$$

We define $B_{\{i,j,k\}}(t)$ to be the matrix in the affine transformation from $P_{\{i,j,k\}}$ to $\vec{v_i}(t), \vec{v_j}(t), \vec{v_k}(t)$, i.e.

$$B_{\{i,j,k\}}(t)\vec{p_f} + \vec{l} = \vec{v_f}(t), \quad f \in \{i,j,k\} \qquad (7)$$

Note that the coefficients of $B_{\{i,j,k\}}(t)$ are linear in $\vec{v_i}(t), \vec{v_j}(t), \vec{v_k}(t)$. We define an intermediate shape $V(t)$ as the vertex configuration which minimizes the quadratic error between the actual matrices $B_{\{i,j,k\}}(t)$ and the desired ones $A_{\{i,j,k\}}(t)$. This quadratic error functional is expressed as

$$E_{V(t)} = \sum_{\{i,j,k\} \in \mathcal{T}} \left\| A_{\{i,j,k\}}(t) - B_{\{i,j,k\}}(t) \right\|^2, \qquad (8)$$

where $\|\cdot\|$ is the Frobenius norm. In this expression, the $A_{\{i,j,k\}}(t)$ are known for a fixed time t and each $B_{\{i,j,k\}}$ is linear in the $\vec{v_i}(t), \vec{v_j}(t), \vec{v_k}(t)$. Thus, $E_{V(t)}$ is a positive quadratic form in the elements of $V(t)$. In order to have a unique minimizer to $E_{V(t)}$, we should predetermine the location of one vertex, say $v_{1_x}(t), v_{1_y}(t)$, for example, by linear interpolation.

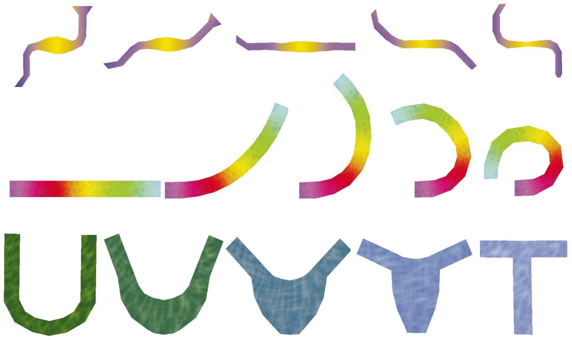


Figure 6: Transformations of different shapes representing solid objects. Note that parts of the shapes transform rigidly whenever possible. The lowest row shows an example where the shapes have no obvious common skeleton.

The functional $E_{V(t)}$ can be expressed in matrix form, setting $u^T = (1, v_{2_x}(t), v_{2_y}(t), \ldots, v_{n_x}(t), v_{n_y}(t))$ as

$$E_{V(t)} = u^T \begin{pmatrix} c & G^T \\ G & H \end{pmatrix} u, \tag{9}$$

where $c \in \mathbb{R}$ represents the constant, $G \in \mathbb{R}^{2n \times 1}$ the linear, and $H \in \mathbb{R}^{2n \times 2n}$ the mixed and pure quadratic coefficients of the quadratic form $E_{V(t)}$. The minimization problem is solved by setting the gradient $\nabla E_{V(t)}$ over the free variables to zero:

$$H \begin{pmatrix} v_{2_x}(t) \\ v_{2_y}(t) \\ \vdots \end{pmatrix} = -G \tag{10}$$

Note that H is independent of t. This means we can invert H and find solutions for time t by computing the corresponding $G(t)$ and a single matrix multiplication:

$$V(t) = -H^{-1} G(t) \tag{11}$$

In practice, we compute the LU decomposition of H and find $V(t)$ by back substitution. Furthermore, the computations are separable and are performed independently for the two coordinates. Note that only G depends on the dimension, while H is the same for the x and y components. Thus, H is effectively of size $n - 1 \times n - 1$, which means the dominating factor of the computation is independent of the dimension.

The above definition has the following notable properties:

- For a given t, the solution is unique.
- The solution requires only one matrix inversion for a specific source and target shape. Every intermediate shape is found by multiplying the inverted matrix by a vector.
- The vertex path is infinitely smooth, starts exactly in the source shape, and ends exactly in the target shape. These are properties typically difficult to achieve in physically-based simulations.

Figure 6 shows transformations of some simple shapes produced with the described method.

2.3 Symmetric Solutions

While we were satisfied with the degree of symmetry the previosly explained approach exhibited in most of our test cases, a symmetric solution can be advantageous – in particular, if the corresponding triangles in the source and target shapes have largely differing area. We can make the solution symmetric by simply blending the optimization problems from both directions. Let $A_f^{\rightarrow}(t)$ be the affine transformation of triangle f from source to intermediate shape at time t, and $A_f^{\leftarrow}(t)$ the respective transformation coming from the target shape. Similarly, we define $B_f^{\rightarrow}(t)$ and $B_f^{\leftarrow}(t)$. We define intermediate $E_{V(t)}$, the vertex configuration at time t, by

$$E_{V(t)} = (1-t) E_{V(t)}^{\rightarrow} + t E_{V(t)}^{\leftarrow} \tag{12}$$

where

$$\begin{aligned} E_{V(t)}^{\rightarrow} &= \sum_{f \in \text{Tri}} \left\| A_f^{\rightarrow}(t) - B_f^{\rightarrow}(t) \right\|^2 & (13) \\ E_{V(t)}^{\leftarrow} &= \sum_{f \in \text{Tri}} \left\| A_f^{\leftarrow}(1-t) - B_f^{\leftarrow}(1-t) \right\|^2 & (14) \end{aligned}$$

With this definition, we lose the advantage of only one matrix inversion for given source and target shapes. Instead, every time t requires the solution of a linear system of equations. Whether the computation times are acceptable depends on the shapes and the desired application.

3 Isomorphic Dissections of Shapes

In this section, we construct isomorphic dissections given two shapes in boundary representation. We assume that the correspondence of the boundaries has been established, i.e. a bijective map between boundary vertices is given. For polygons, reasonable correspondence can be found automatically [23, 7]. In difficult cases, few correspondences could be specified manually and the remaining vertices are matched automatically. For polyhedral objects, several techniques exist, which are based on topological merging introduced by Kent et al. [19]. Recent work [14, 20] also allows the specification of corresponding features which seems sufficient to produce acceptable results for a variety of polyhedral models.

3.1 Polygons

The problem of constructing a common triangulation for two given polygons is discussed in the literature as *compatible triangulation* [2]. Triangulating a single polygon π is possible using only the vertices of the polygon (e.g. [5]). However, this is usually not possible for two different polygons. Aronov et al. [2] show how to triangulate two polygons in a compatible way if at most $O(n^2)$ additional vertices (so-called Steiner points) are allowed. The general scheme [2] is to first triangulate each polygon independently. Then, both polygons are mapped to a regular n-gon so that corresponding boundary vertices coincide. The compatible triangulation is established by overlaying the two edge sets in the convex n-gon. The resulting new interior vertices are then mapped back into the original polygons, yielding compatible triangulations of the source and target shapes.

We would like to stress that the quality of the blend, in terms of the quality of the in-between shapes, strongly depends on the shape of the simplices. In particular, skinny triangles (or tetrahedra in 3D) cause numerical problems. Thus, in the following, we describe how this scheme can be enhanced to yield compatible triangulations with a significantly better triangle shape.

First, we apply Delaunay triangulations (see any textbook on Computational Geometry, e.g. [4]) as the initial triangulation since

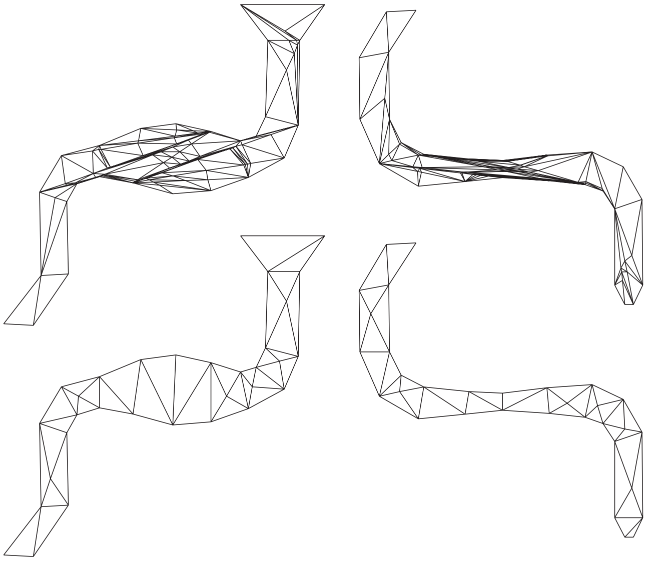


Figure 7: A comparison of compatible triangulations. The upper row shows triangulations generated from using ear-capping for the initial triangulation step. Initial triangulations are overlaid on a convex domain to produce compatible triangulations. The triangulations in the lower row were generated with the same general procedure, but using initial Delaunay triangulations. Far fewer triangles are induced, since Delaunay triangulations yield similar partitioning for similar regions.

Delaunay triangulations maximize the minimum interior angle and, thus, avoid skinny triangles. Of course, any skinny triangle in the independent triangulations is inherited by the merged triangulation. Moreover, Delaunay triangulations are unique, and similar regions in the shapes will result in similar triangulations. Thus, skinny triangles resulting from the overlay process can be avoided.

Nevertheless, the merged triangulations still have skinny triangles, and further enhancement is required to avoid numerical problems. We optimize the triangulations by further maximizing the minimum interior angle, which is known to be a reasonable triangulation quality criterion (see e.g. [4]). We use two independent operations:

1. Moving interior vertices. Freitag et al. [11] show how to find vertex positions which maximize the minimum angle for a given triangulation.

2. Flipping interior edges simultaneously in both triangulations. This procedure follows the edge flip criteria used in Delaunay triangulation. Given that an edge flip is legal in both triangulations, it is performed if the operation increases the overall minimum angle.

The above two operations are applied in turn until no valid flips are necessary. Convergence is assured since each step can only increase the minimum angle. We call this procedure *compatible mesh smoothing*. The smoothing step optimizes the compatible triangulations without changing the vertex count.

However, we also consider changing the vertex count by means of splitting edges. The split operation is well-defined in terms of topology, if it is applied to both triangulations simultaneously, the isomorphy remains. The idea is to split long edges to avoid long skinny triangles. Splitting edges according to their lengths does not guarantee an increase in triangle quality. In practice, smaller triangles are more likely to be improved by the smoothing step. After each edge split, the triangulations are smoothed. This avoids the generation of edges in regions where the smoothing operation would produce nicely-shaped triangles. Figure 8 illustrates the results of splitting edges, as well as of the smoothing process.

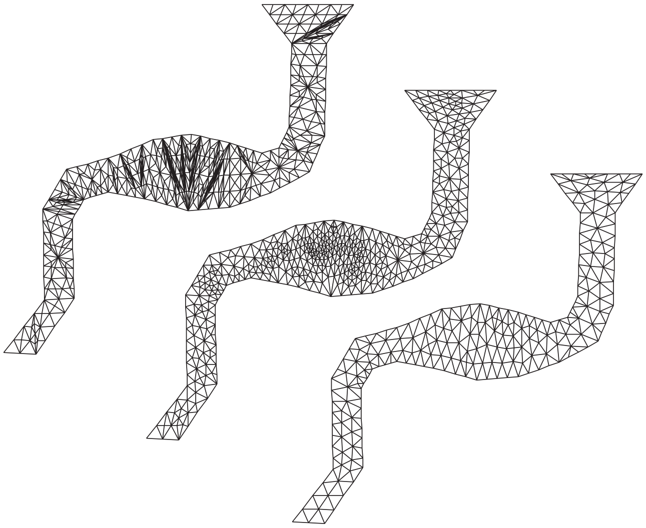


Figure 8: The mesh refinement process. In the first row, the merged Delaunay triangulations from Figure 7 are refined by edge splits until all edge lengths are bounded. The second row shows the result of compatible mesh smoothing on this triangulation. The third row shows the actual technique, where splitting and smoothing is performed concurrently. Note that the edge length bound is the same in the first and third row.

3.2 Polyhedra

To the best of our knowledge, the three-dimensional analog to compatible triangulations has not been discussed in the literature. Work has been done to dissect polyhedra into simplicial complexes, a process referred to as tetrahedralization. However, the work of Aronov et al. [2] can be extended to genus 0 polyhedra. First, the source and target polyhedra are tetrahedralized independently using common techniques, e.g. Barry Joe's Geompack [17]. Then, the tetrahedralizations are mapped to a corresponding convex shape. Alexa [1] as well as Shapiro and Tal [26] describe methods to map an arbitrary genus 0 polyhedron to a convex shape. Since the source and target polyhedra are assumed to have the same vertex-edge topology and the convexification process is deterministic, the polyhedra are mapped to the same convex shape. The interior vertices of their tetrahedralizations are mapped using barycentric coordinates. The fact that vertices are mapped to a convex shape using barycentric coordinates for interior vertices assures that no tetrahedra will be flipped. Then, an overlay of the two tetrahedralizations is computed, where faces are cut against faces, resulting in new edges. Note that the intersection of two tetrahedra results in four-, five-, or six-sided convex shapes, which are easy to tetrahedralize. The resulting structure is mapped back into original polyhedra. In case the source and target shapes are not genus 0, they have to be cut into genus 0 pieces which are independently treated as explained above.

4 Results and Conclusion

We have applied the techniques explained above to various inputs. The two-dimensional shapes are generated by extracting a contour out of an image. For the correspondence of contours, we defined manually several vertex-to-vertex correspondences, while the re-

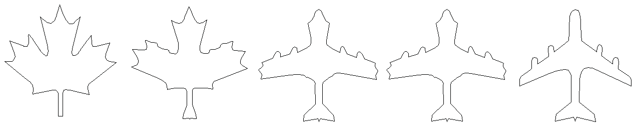


Figure 9: The contour of a maple leaf blended with a plane using as-rigid-as-possible shape interpolation. Note that the features of the plane grow out of contour according to the current direction of wings and not their final position.

maining vertices were automatically aligned. The resulting polygons were dissected as described above. In Figures 10, 11, 12, and 15, the triangulations were used to map a texture to the shape (as was suggested by Tal and Elber in [28]). Textures were extracted with the contours from the source images. More elaborate techniques for space-time control (e.g. [21]) could be easily integrated in our work to give the user more control as to what is transformed and when. Also note that the techniques are not restricted to simple polygons.

Since our technique interpolates shapes "naturally" in the sense that it preserves parts that just change relative position or orientation, it could be also used to extrapolate beyond the source and target shapes. Figure 13 demonstrates this with the example of Leonardo DaVinci's studies on proportions (see Figure 15 for the interpolation). We can generate shapes for time values -0.5 and 1.5 while preserving the proportions of the human figure.

We have also applied the interpolation technique to three-dimensional models. The examples in Figure 14 were generated by using deformed versions of a polyhedral model. Note the difference between linear vertex interpolation (upper row) and as-rigid-as-possible interpolation (lower row). In Figure 16, morphable polyhedral models were generated using topological merging. As in the two-dimensional case, the vertex paths result from defining transformations for each pair of corresponding tetrahedra by factoring the affine transform into rotational and stretching components and, then, minimizing the deviation from these ideal transformations.

The current implementation seems to be robust and fast. The most time-consuming step is optimizing triangle shape. Without optimizing triangle shape numerical problems are likely to occur. In all our examples no simplex changed orientation (i.e. flipped), however, we have not been able to prove this to be a property of our approach.

The examples clearly demonstrate the superior quality of our approach compared to plain linear vertex interpolation. Additionally, it offers the possibility to texture the shapes, so that shape blending becomes applicable to images. In turn, traditional image morphing techniques could serve to generate the homeomorphic dissections of the shapes and, thus, make use of more advanced vertex/pixel interpolation technique(s). However, the quality of a morph lies in the eye of the beholder. Nevertheless, there is a clear consensus that - lacking other information - the geometry along the morph sequence should change monotonically with no superfluous distortions. The idea of as-rigid-as-possible shape interpolation is to avoid distortions as much as possible and let angles and scales change linearly. We believe that this captures the notion of the above-mentioned consensus.

Despite this, shape blending is always an aesthetic problem and no automatic method will meet the needs that arise in different applications. Consequently, user interaction will always be necessary to produce the desired results. Nevertheless, we believe that more elaborate methods for shape blending simplify and minimize the involvement of the designer.

Finally, we want to mention that dissections of shapes seem to extend the concept of skeletons while fully capturing their information. Dissections are more powerful in representing the mechanics of shapes as they allow fine grained analysis of local behaviour. In many cases, shapes naturally have no skeleton or their metamorphoses could not be described in terms of a skeleton. These benefits come along with easier and less ambiguous computation of dissections as compared to skeletons.

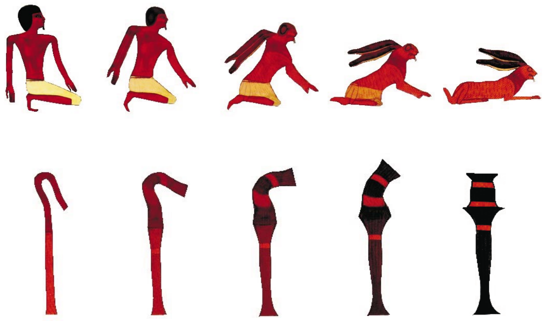


Figure 10: Morphs between Egyptian art pieces using textures from the original images. Contours are blended using as-rigid-as-possible shape interpolation and texture colors are linearly interpolated.

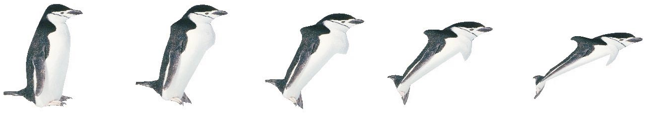


Figure 11: Contour blend of a penguin and a dolphin using only the texture of the penguin.

Acknowledgements

We thank Kai Kreuzer for helping with the implementation and many examples. Craig Gotsman, George Wolberg, Thomas Sederberg, Herbert Edelsbrunner, and the anonymous reviewers provided useful advice and suggestions. Thanks to Wolfgang Müller and Jose Encarnaçaõ for their support. Work on this paper has been partially supported by the Hermann Minkowski – Minerva Center for Geometry at Tel Aviv University and by The Israel Academy of Sciences.

References

[1] M. Alexa. Merging Polyhedral Shapes with Scattered Features. The Visual Computer, 16, 1, 2000

[2] B. Aronov, R. Seidel, and D. Souvaine. On compatible triangulations of simple polygons. Computational Geometry: Theory and Applications 3, pp. 27-35, 1993

[3] T. Beier and S. Neely. Feature-based Image Metamorphosis. SIGGRAPH '92 Proceedings, pp. 35-42, 1992

[4] M. de Berg, M. van Krefeld, M. Overmars, and O. Schwarzkopf. Computational Geometry - Algorithms and Applications. Springer, Berlin, 1997

[5] B. Chazelle. Triangulating a simple polygon in linear time. Proc/ 31st Symp. on Foundations of Computer Science (FOCS), pp. 220-230, 1990

[6] E. Carmel, D. Cohen-Or. Warp-guided Object Space Morphing. The Visual Computer, 13, 1997

[7] S. Cohen, G. Elber, R. Bar Yehuda. Matching of freeform curves. CAD, 19, 5, pp. 369-378, 1997

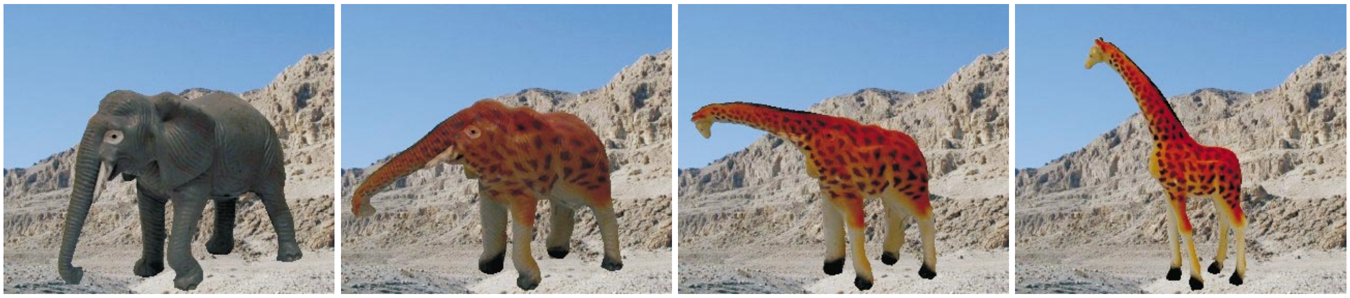


Figure 12: Morph between photographs of an elephant and a giraffe.

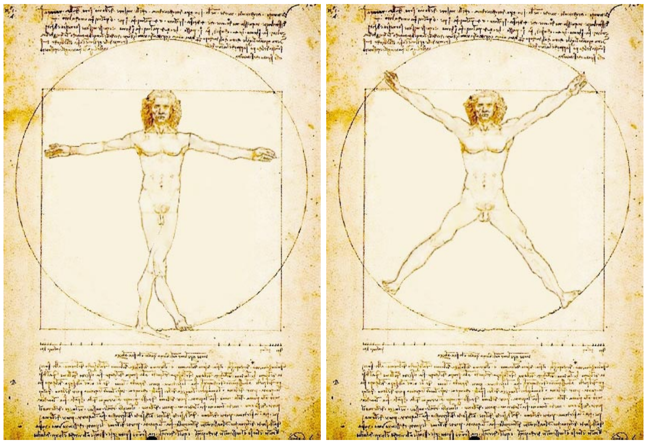


Figure 13: Shape extrapolation. Using as-rigid-as-possible shape interpolation, shapes can be naturally extrapolated beyond the source and target shapes. The images show the human figure of Leonardo DaVinci's proportions at time values -0.5 and 1.5.

[8] D. Cohen-Or, D. Levin, and A. Solomovici. Three dimensional distance field metamorphosis. ACM Transactions on Graphics, 1998

[9] M. Etzion and A. Rappoprt. On Compatible Star Decompositions of Simple Polygons. IEEE Transactions on Visualization and Computer Graphics, 3, 1, pp. 87-95, 1997

[10] M. S. Floater and C. Gotsman. How to Morph Tilings Injectively. J. Comp. Appl. Math., 101, pp. 117-129, 1999

[11] L.A. Freitag, M.T. Jones, and P.E. Plassmann. An efficient parallel algorithm for mesh smoothing. 4th Int. Meshing Roundtable, pp. 47-58, 1995

[12] G.H. Golub and C.F. van Loan. Matrix Computations. The Johns Hopkins University Press, Baltimore, 1983

[13] E. Goldstein and C. Gotsman. Polygon Morphing using a Multiresolution Representation. Graphics Interface '95, pp. 247-254, 1995

[14] A. Greogory, A. State, M. Lin, D. Manocha, and M. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. Proceedings of Computer Animation '98, pp. 64-71, 1998

[15] T. He, S. Wang, and A. Kaufman. Wavelet-based Volume Morphing. Proceedings of Visualization, IEEE Computer Society, pp. 85-91, 1994

[16] J.F. Hughes. Scheduled Fourier Volume Morphing. Computer Graphics (SIGGRAPH '92 Proceedings), 26, 2, pp. 43-46, 1992

[17] B. Joe. Geompack. ftp://ftp.cs.ualberta.ca/pub/geompack

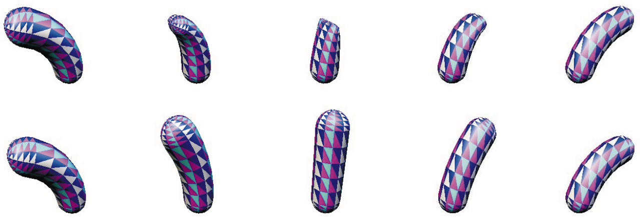


Figure 14: A simple example of three-dimensional objects. The difference of linear and as-rigid-as-possible vertex interpolation is demonstrated on a bent cigar-like shape.

[18] T. Kanai, H Suzuki, and F. Kimura. 3D geometric metamorphosis based on harmonic maps. Proceedings of Pacific Graphics '97, pp. 97-104, 1997

[19] J.R. Kent, W.E. Carlson, and R.E. Parent. Shape Transformation for polyhedral objects. Computer Graphics, 26, pp. 47-54, 1992

[20] A.W.F. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution Mesh Morphing. SIGGRAPH '99 Proceedings, pp. 343-350, 1999

[21] S.Y. Lee, K.Y. Chwa, S.Y. Shin, and G. Wolberg. Image Metamorphosis Using Snakes and Free-Form Deformations. SIGGRAPH '95 Proceedings, pp. 439-448, 1995

[22] A. Lerios, C.D. Garfinkle, and M. Levoy. Feature-Based Volume Metamorphis. SIGGRAPH '95 Proceedings, pp. 449-456, 1995

[23] T.W. Sederberg and E. Greenwood. A physically based approach to 2D shape blending. Computer Graphics, 26, pp. 25-34, 1992

[24] T.W. Sederberg, P. Gao, G. Wang, and H. Mu. 2-D shape blending: An intrinsic solution to the vertex-path problem. Computer Graphics, 27, pp. 15-18, 1993

[25] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. IEEE CG&A, 15, pp. 44-51, 1993

[26] A. Shapiro and A. Tal. Polyhedron realization for shape transformation. The Visual Computer, 14, 8/9, 1998

[27] K. Shoemake and T. Duff. Matrix Animation and Polar Decomposition. Proceedings of Graphics Interface '92, pp. 258-264, 1992

[28] A. Tal and G. Elber. Image Morphing with Feature Preserving Texture. Computer Graphics Forum (Eurographics '99 Proceedings), 18, 3, pp. 339-348, 1999

[29] G. Wolberg. Digital Image Morphing. IEEE Computer Society Press, 1990

[30] G. Wolberg. Image Morphing Survey. The Visual Computer, 14, 8/9, 1998

[31] Y. Zhang. A Fuzzy Approach to Digital Image Warping. IEEE Computer Graphics and Applications, pp. 33-41, 1996

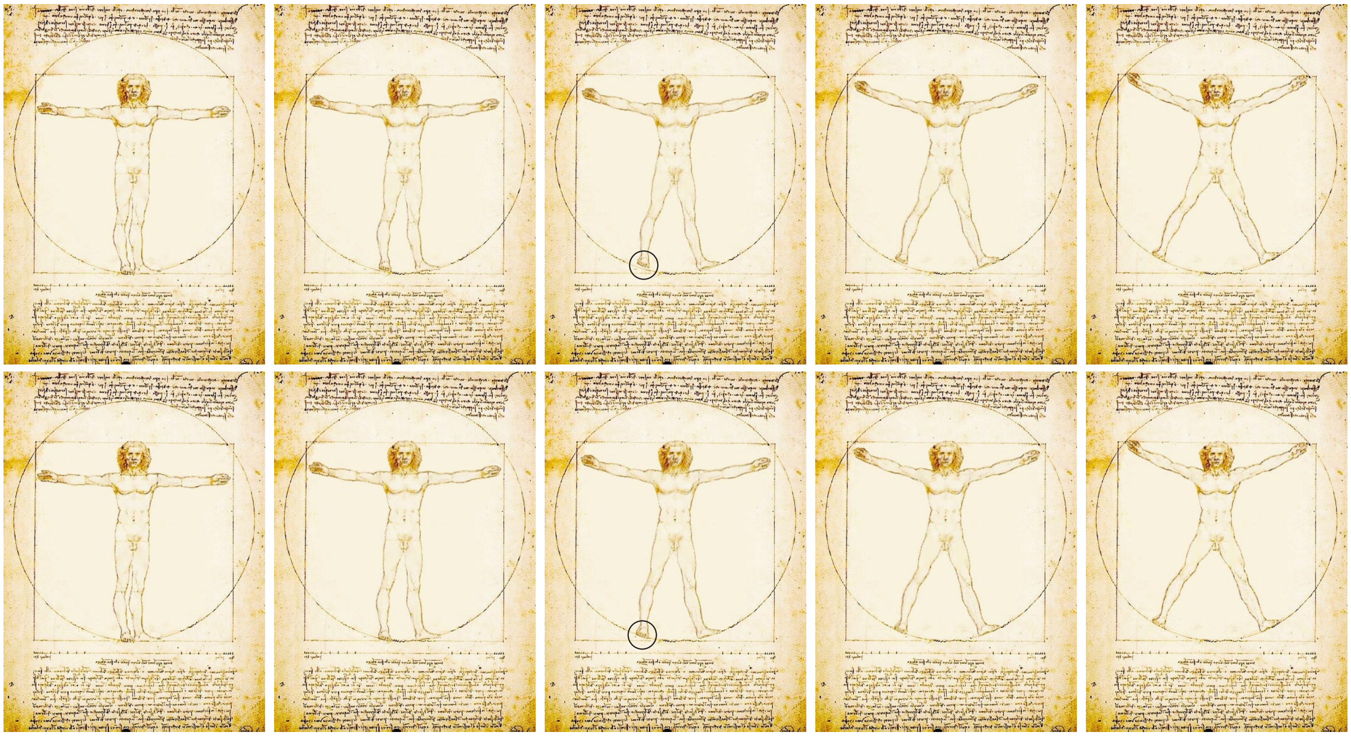


Figure 15: Leonardo DaVinci's studies on proportions. The two rows contrast linear and as-rigid-as-possible interpolation of the figure. The difference is subtle, but Leonardo proves our method right (see the feet leaving the circle in the linear interpolation).

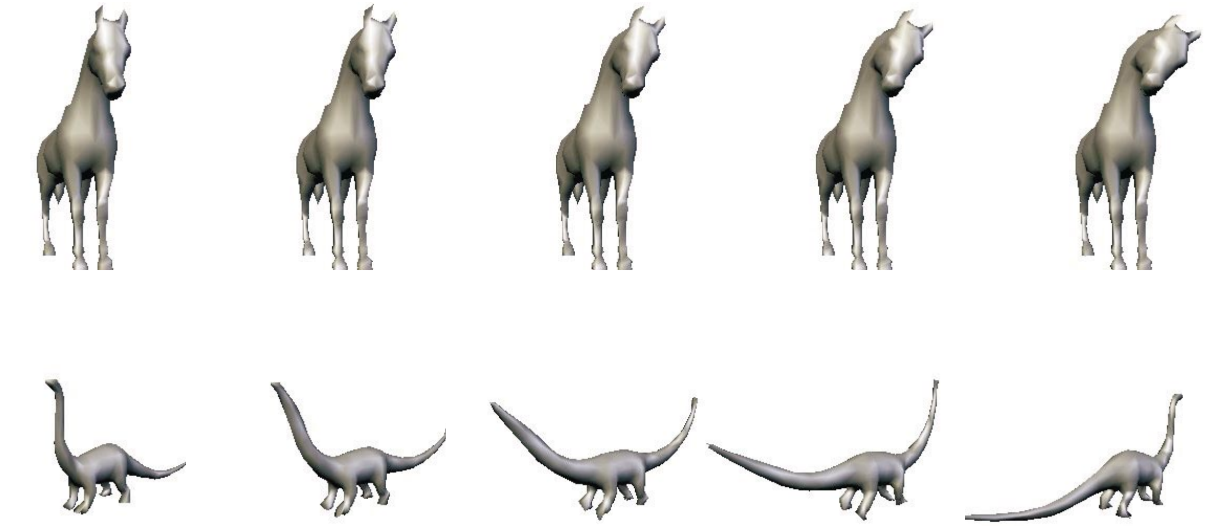


Figure 16: Our technique is also useful to mimic motions of articulated three-dimensional objects in case the underlying skeleton is missing, as demonstrated for a horse turning its head. The example in the lower row was produced using a polyhedral morphing technique (facilitating topological merging). Note that the lengths of the tails/necks are preserved.