Interactively Morphing Irregularly Shaped Images Employing Subdivision Techniques

Jan Van den Bergh Fabian Di Fiore Johan Claes Frank Van Reeth

Expertise Centre for Digital Media - Limburg University Centre
Wetenschapspark 2, B-3590 Diepenbeek
Belgium

{jan.vandenbergh, fabian.difiore, johan.claes, frank.vanreeth}@luc.ac.be

Abstract

The morphing or metamorphosis of images is often used to generate special effects in films and animation. In many cases only a part of an image will be affected by the morph, and often that part has an irregular shape. This work presents a new interactive technique for performing morphs based on locally interpolating subdivision surface schemes. The technique allows the definition of morph problems between irregular objects with equal genus containing holes and protruding limbs, and their execution in a relatively short time, without losing local control in the interior of objects or their borders. Furthermore, applying subdivision surface methods allows large variations in control point density.

Keywords

morphing, image metamorphosis, subdivision surfaces, animation, shape blending, interpolation

1. INTRODUCTION

For a long time animators needed means to make one object change smoothly into another, a process known as morphing. The classical approach involves changing the appearance of the physical object or by a cross-dissolve of the images. Here smooth transitions cannot be achieved because the features of the objects do not map directly onto each other.

Various image-morphing techniques [Johan00, Wolberg98] offer a solution to this problem by letting the user create a mapping between features in the source and features in the destination image. Once a good mapping is made, a smooth transition can be generated. Besides image morphing, which works on rectangular images, there is another two-dimensional morphing technique, which creates a smooth transition between two objects and is known as object-space morphing or shape blending [Sederberg93]. Combining these techniques allows to create a pleasing morph between irregularly shaped objects extracted from images.

Before the transformation between images or parts of them can be calculated, the features in the source and destination need to be associated with each other in order to define the mapping (the so-called correspondence problem). This mapping requires a method to, (i) mark the features, and (ii) calculate the correspondences between them. The difficulties in defining such a mapping include placing the large number of corresponding points or lines accurately enough to generate a good morph. With current methods, this is a cumbersome and time consuming task when done by hand, while an automated algorithm often produces undesired effects.

The technique we propose employs a locally interpolating subdivision surface to approximate the shape of the subject, and this is used to define and generate the morph. This approach means that we only need to use a limited number of points to have full control over a subject with an irregular outline. This model enables a natural handling of features such as eyes in a head, or even holes inside the outline (such as inside the handle of a cup, as in figure 1). Our technique, in short, provides a solution to the outstanding problems of shape interpolation management identified here. The use of the technique will be further explained in section 3, while section 4 discusses details of the core subdivision technique. Results obtained by our method are shown in section 5, followed by a discussion in section 6. Finally, our conclusions and future work can be found in section 7.

2. PREVIOUS WORK

Mesh based warping in its most basic form, as described by Wolberg [Wolberg90], starts from a uniform mesh of points in the source image. This technique uses a grid of control points that are uniformly spread over the image. The animator has to move these control points to specify the correspondence between the images. This approach has the disadvantage that, when local control is desired

Figure 1. Example of a morph created by our system: morphing between objects with a hole.

somewhere in the image, this is affected by individually moving each one of a dense mesh of points to their correct position. This is a tedious and error-prone job, urging the development of more appropriate techniques.

Nishita et al. [Nishita93] describe a method that uses a non-uniform mesh to refine the location of the features. They use active nets to ease the positioning of the points. This is an adaptive image processing technique, which places a mesh to fit over features. Active nets can substantially reduce the work required to specify features but the technique depends heavily on the colors used in the image, and therefore it does not deliver good results when there are many features close together, or when there is insufficient color differentiation.

Field morphing [Beier92, Lee98] uses pairs of directed line-segments to mark the features. Lee et al. [Lee96] let the user specify pairs of polylines. These polylines are uniformly sampled to extract points for the actual morph. They also propose the use of snakes to get the drawn polylines to converge on image features.

The method of Arad and Reisfeld [Arad95] only needs a few points to specify correspondences. Others [Johan00, Tal99] ask the user to specify interactively defined curves on outlines, or to delimit features in objects. Both methods use least squares fitting of respectively cubic Bézier chains or B-spline curves to reduce the number of generated points, and automated methods to calculate the correspondence.

The correspondence problem is very important in object-space morphing and researchers described many useful approaches. Although it would be possible to automatically establish correspondences [Carmel97, Kent92, Tal99], we encountered too many artifacts for practical input images. Therefore, as Shapira and Rappoport [Shapira95], we let the user specify them manually. Still we will try to minimize the effort that is needed to specify this correspondence. Details are given in section 3 and later sections.

Once the mapping between objects is established, the actual calculation of the in-between objects can start. Wolberg [Wolberg98] gives an excellent overview and discussion of the techniques current in 1998. We refer to his work for a detailed discussion of the techniques, which include mesh warping, field morphing and techniques based on radial basis functions, thin plate splines, energy minimization, multilevel free-form deformation and work minimization.

Various algorithms for object-space morphing use linear vertex interpolation to compute the intermediate objects. This interpolation scheme can result in distortions when the morphed objects, or parts of them don't have the same orientation. Several techniques are developed to avoid these distortions.

Some techniques involve a special representation of the object. Shapira and Rappoport [Shapira95], for example, use a star-skeleton representation. In this representation, the morphed objects are split into several star-shaped polygons, which are represented by the edge points and an extra star-point that is used to connect the different star-shaped polygons. Goldstein and Gotsman [Goldstein95] use a multi-resolution representation, in which the surface is represented at different levels of detail with the lower-resolution representation being smoother than the higher resolution version. At the lowest resolution the polygon is convex. Their approach leads to pleasing results for objects such as stick figures, where the relative thickness of the object parts has to be conserved during the morphing animation.

Other techniques to morph polygonal objects keep the original representation, and employ a special purpose interpolation scheme. Sederberg and Greenwood [Sederberg93] interpolate the length of the edges of the polygon and the angles between them, while Alexa et al. [Alexa00] use Delaunay triangulated polygons, and instead of the outline, transform the triangles of the resulting mesh. For image morphing, the use of triangles has the drawback that the border between the triangles causes non-$\mathbf{C}^1$ deformations.

Most of the morphing algorithms referenced here require the specification of a large number of reference points. The precise placement of these reference points is crucial to the success of the method. In many cases the user gets some help from computer vision techniques, but if the resulting placement is not satisfactory, the user is forced back to the manual methods of position specification. It is especially annoying that when locally a denser control mesh is needed, control points need to be added everywhere. Subdivision surfaces on the contrary, allow high variations in control point density, freeing the user of dealing with too many control points in regions that he is less interested in.

3. OUR APPROACH

The technique we propose is targeted at morphing objects in image space although the process itself is carried out in object space, followed by a projection step. This has the advantage that only the object that the animator wants to morph is affected and that the background and possible other objects in the scene are unaffected. As Tal and Elber [Tal99] mention, this also allows to insert the created morph in animations, websites, etc. as an animated clipart.

As input for our approach we use two objects represented as images, for example the extreme images of figure 1). The first step of the algorithm is positioning the vertices of a coarse triangulated mesh at strategic points near the features of the objects in both images. The mesh can have an arbitrary form and may contain holes. There are no special requirements for the triangulation, however the amount of skinny triangles should be kept to the bare minimum, in order to prevent distortions.

When the mesh points are roughly on the right position, the mesh can be refined globally as well as locally. The global refinement is reached through the use of a user-determined number of steps from the approximating subdivision scheme proposed by Loop [Loop87]. For the local refinement, Loop's subdivision scheme is extended to allow interpolation around selected points. The details of how we apply the subdivision scheme and the extension that was used are given in the next section.

4. LOCALLY INTERPOLATING SUBDIVISION SURFACES

Since the publication of Catmull and Clark's subdivision surface scheme [Catmull78] in 1978, numerous researchers have been putting these schemes and their variants to use for many different purposes. For our research, we are opting for Charles Loop's scheme [Loop87], as it generates smoothly joined triangles starting from a rather coarse polygon mesh. We refer the interested reader to the Siggraph 2000 notes [Zorin00] for an in depth overview of the state of the art.

In the standard Loop scheme, the polygon mesh is recursively subdivided by adding new points (and the edges needed to integrate them) in the middle of every edge of the mesh and then averaging the location of every point of the newly generated mesh. The rule for adding a new edge point E on the interior edge between V_1 and V_2 and with immediate neighbors Q_1 and Q_2 (see figure 2(a)) is:

$$E = \frac{3}{8}(V_1 + V_2) + \frac{1}{8}(Q_1 + Q_2) \qquad (1)$$

An interior point V_0, surrounded by k vertices Q_1 to Q_k

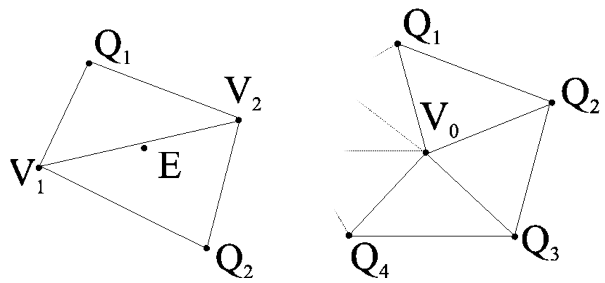


Figure 2. a) Situation for insertion of an interior edge point E. b) Situation around an interior vertex V_0.

(see figure 2(b)) is averaged using equation 2.

$$V_0' = \sum_{i=1}^{k} \beta Q_i + (1 - k\beta) V_0 \qquad (2)$$

$$with \quad \beta = \frac{1}{k}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{k}\right)^2\right)$$

We extended Loop's scheme to better fulfill the needs of our application. The details of these techniques are described in [Claes01]; here we are giving a brief overview of the parts of [Claes01] that are needed to understand our implementation. The extension provides local interpolation either at the border or at interior vertices of the surface. We obtained this local interpolation without changing the standard uniform and stationary subdivision rules. For an interior vertex V_0, we observed that by making sure that V_0 is equal to the mean of the surrounding vertices (equation 3), the subsequent iterations of the subdivision scheme keep this vertex constant at its location. When a vertex V_0 with surrounding vertices V_1 to V_k is marked as interpolating, new points Q_1 to Q_k are inserted into the mesh on the edges between V_0 and V_i with i between 1 and k in such way that equation 3 is valid. The exact position of a point Q_i is determined by the position of V_i and a tension parameter that can be manipulated by the user. This process may introduce polygons with more than three vertices, which are converted to triangles by inserting a point in their center.

$$V_0 = \frac{1}{k}\sum_{i=1}^{k} Q_i \qquad (3)$$

Equation 4 shows that the condition of 3 ensures that V_0 will keep its position after a subdivision step.

$$\begin{aligned} V_0' &= \beta k \frac{1}{k}\sum_{i=1}^{k} Q_i + (1 - k\beta) V_0 \\ &= V_0 \end{aligned} \qquad (4)$$

Equation 5 verifies that equation 3 recursively holds again for the newly generated edge points. These edge points will form the surrounding vertices for the next subdivision step. Note that here the mod-operator is supposed to put the index between 1 and k whenever necessary.

$$\begin{aligned}\frac{1}{k}\sum_{i=1}^{k} E_i &= \frac{1}{k}\sum_{i=1}^{k}\Bigg(\frac{3}{8}(V_0+Q_i)+ \\ &\quad \frac{1}{8}(Q_{(i+1)mod\,k}+Q_{(i-1)mod\,k})\Bigg) \\ &= \frac{1}{k}\sum_{i=1}^{k}\frac{3}{8}V_0+\frac{1}{k}\sum_{i=1}^{k}\frac{5}{8}Q_i \\ &= \frac{3}{8}V_0+\frac{5}{8}\left(\frac{1}{k}\sum_{i=1}^{k}Q_i\right) \\ &= V_0 \qquad (5)\end{aligned}$$

Furthermore an additional degree of freedom is given by the observation that the configuration of ghost points can be scaled without loosing the property of equation 3. This degree of freedom provides a handy tension parameter, that helps in marking specific features.

For the vertices at the border of the surface, a similar arrangement can be constructed. More details of this approach can be found in [Claes00]. In brief, the idea is to add a ghost point on every side of the point to be interpolated, both at the same distance and placed on the desired tangent line. This is sufficient as the subdivision rules for the edge do not take the internal points into account.

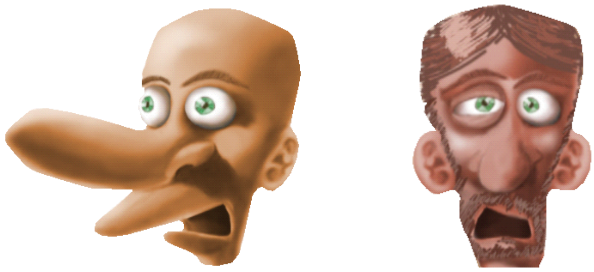
As our method only inserts new points and keeps the underlying Loop subdivision scheme intact, its continuity properties are preserved. Therefore, provided no control vertices coincide, the scheme stays $\mathbf{C}^2$ at vertices with a regular valence and at least $\mathbf{C}^1$ at the extraordinary vertices. It can even be proven that at our locally interpolating vertices, the the surface is always $\mathbf{C}^2$. This makes our subdivision based approach a very suitable technique to achieve fluent morphs without visual artifacts.

5. IMPLEMENTATION AND RESULTS

In the current prototype implementation, we start from two two-dimensional objects O_0 and O_1 that will be morphed towards each other. They are represented by two-dimensional images, I_0 (figure 3(a)) and I_1 (figure 3(b)), each in a different position in time.

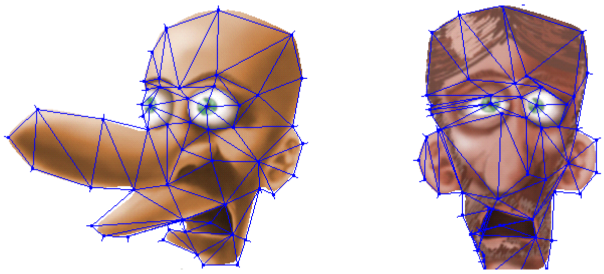
First a mesh M is created whose interior vertices identify the features and whose outline roughly matches the outline of O_0 in I_0. Then M is laid over I_1 and the user moves its vertices so that they roughly match the outline and the features of O_1. This results in the representations in object-space S_{R0} and S_{R1}, shown in figure 4.

In the next step, S_{R0} and S_{R1} can be subdivided, so that they get more fine grained and globally smoother. The effect of one subdivision step on S_{R0} is shown in figure 5. We observed that no more than one or two subdivision steps suffice in most cases to achieve pleasing results.



(a) (b)

Figure 3. a) The initial and b) target image.



(a) (b)

Figure 4. a) The initial and b) target image with overlaid rough mesh.

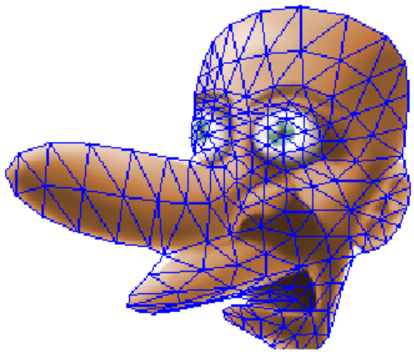


Figure 5. Once subdivided mesh over the initial image.

In order to improve local control, the user can decide to make some of the vertices interpolating. An interpolating point is especially useful in specific circumstances e.g.:

- an isomorphic nested feature can be marked using only one vertex in the coarse mesh (e.g. the pupil of an eye);
- curvatures in the outline or in features can be marked using a very limited number of features.

This is possible because the tension in those points — and thus the newly added vertices — can be manipulated (see

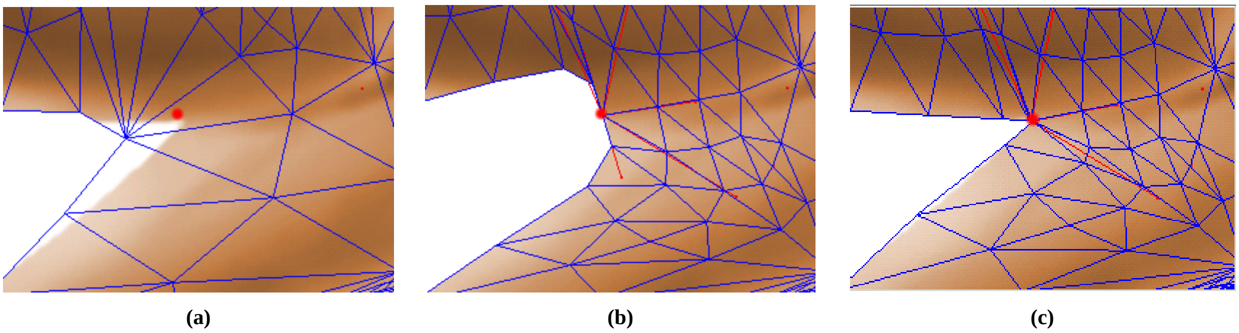


Figure 6. Zoom of the mesh on the initial image. a) Without interpolating point, b) with interpolating point and excessive tension, c) with interpolating point and good tension. The big dot indicates the position of the interpolating point.

figure 6). For separating the objects we want to morph from their background, the border of the control mesh is placed to match the border of the objects. Normally we work with images having a fully transparent background, making life easier for the animators, as they do not have to follow the borders too precisely. Either way, the user should prevent that the background would show up as part of the deformed objects. The local interpolation ensures high-precision control at specific points.

When the animator is satisfied with the object-space representation of the objects (S_{S0} and S_{S1}) he created, the points of the mesh are fixed to the image. If the user isn't satisfied with the shape of an object, he can use the same techniques to deform it as is done in Claes et al. [Claes00].

The default calculation of an intermediate image at time t is done by linearly interpolating the positions of the vertices in S_{R0} and S_{R1}, resulting in a rough mesh S_{Rt}. Based on the vertices of S_{Rt} and the subdivision and interpolated tension information from the interpolating points in S_{S0} and S_{S1}, the positions of the vertices of S_{St} are determined (see figure 7).

This interpolation scheme has the drawback that the result can contain some unnatural images [Alexa00, Shapira95], especially when objects O_0 and O_1 don't have the same orientation. We solve this potential problem by allowing the user to correct this behavior by specifying spline paths for the control points of the mesh by locally manipulating the tension, continuity and bias of spline curves. We opted for the splines described by Kochanek and Bartels [Kochanek84].

Our implementation in OpenGL has the ability to display the effects of the morph in real-time, allowing the animator to flexibly change the behavior until its quality matches his artistic needs.

6. DISCUSSION

Our technique exhibits the following advantages:

- The vertex correspondence problem is handled in an intuitive and easy way. The user explicitly positions the points of the rough mesh in the right place, establishing the correspondence. The process of subdivision is performed on the common mesh and therefore it has no influence on the correspondence problem.
- The subdivision surface paradigm ensures that the interior of the object is deformed in at least a $\mathbf{C}^1$ way (usually even $\mathbf{C}^2$). Furthermore subdivision surfaces exhibit an arbitrary topology. This allows for the creation of holes, but, even more important, this also enables explicit discontinuities. Nearby control points do not necessarily need to be connected, allowing them to move independently.
- Another important feature of subdivision is that control points can be densely distributed in some regions, while being very coarse in other regions.
- By manipulating the tension, one can specify sharp edges in the subdivision surface (as can be seen in figure 6) without sacrificing smoothness in other parts of the surface.
- As the base mesh we use is polygonal, its edges could be used as input to shape-blending techniques such as those defined by Shapira and Rappoport [Shapira95] and Sederberg and Greenwood [Sederberg93]. This would improve these automatic techniques, as the deformations now can be smooth, both near the borders and at the interior of the images.
- The same framework can be used to incorporate physics based methods, making sure local areas and lengths of individual parts morph in a smooth way.
- Our method involves an optional step in which the user can interactively manipulate trajectory paths of the morph. As the automatic generation has no artistic knowledge, the generated paths can be improved by letting the user modify the representing curves.

To be complete, we also list some limitations of our approach:

- The work described here, only works with 2D images as input. It would be interesting to have a similar approach for 2D or for 3D objects.

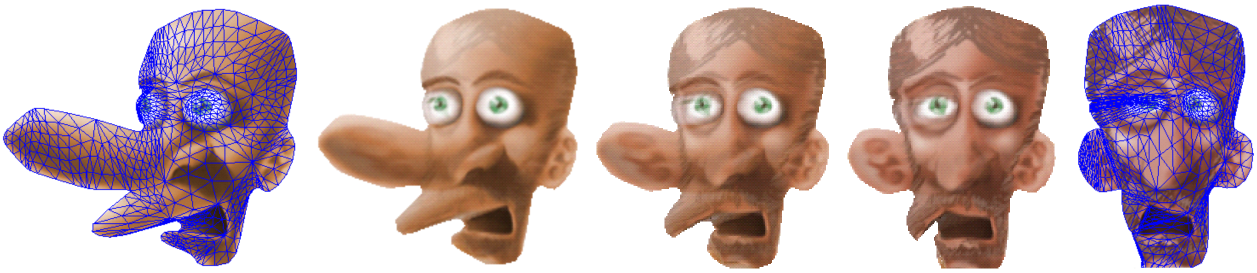


Figure 7. O_0 **and** O_1 **overlaid with once subdivided meshes (having 14 interpolating mesh points) and in-between images at times 0.25, 0.50 and 0.75.**

- Although the interface is easy and intuitive, there still is quite some user interaction needed. Some kind of interaction will always be necessary, as fully automated processes can not predict the artistic wishes of the animators. A possible improvement would be to add physics based techniques to create more intelligent default behavior.
- For some kinds of input, it would be possible that the intermediate meshes get folded somewhere. In our current approach, the user can modify the transition curve of individual control points to work away these artifacts. Again, a physics based approach could make it possible to eliminate this folding altogether. However, the subdivision process itself does not create extra foldings.

7. CONCLUSIONS AND FUTURE WORK

In this paper we presented an interactive morphing technique, combining the advantages of both image and object space morphing. Our technique allows to specify — and when appropriate adapt — the features of two-dimensional objects, which can have a very irregular shape, possibly including holes and thin limbs sticking out. These objects serve as input to the morphing process, where the animator can choose between a fully automatic morph or controlling the facets he wants to change. Furthermore the morphing can be successfully combined with a unified methodology that also handles deformations.

Our approach offers a lot of control by only modifying a limited set of vertices, while the underlying subdivision surface scheme gently smoothes out the deformations. Our extensions to the subdivision scheme allow local control both to the interior and the exterior vertices, together with a handy tension parameter.

The current implementation offers real-time displaying and editing of the results of the morphing process, maximizing the comfort of the animator who wishes to adapt the morphing to his artistic needs.

Future integration of physics based techniques will further reduce the work of the animator, because this way better default morphs could be produced in certain cases. Also via an adequate constraint mechanism, editing the behavior of the actual morph can be made even more powerful.

8. ACKNOWLEDGEMENTS

The authors are very pleased to acknowledge that this work has been partially funded by the European Fund for Regional Development and the Flemish Government. Furthermore, we want to thank John Patterson for his invaluable suggestions and advise about the language and content of this paper. Also the creative input from Joan Cabot is highly appreciated.

9. REFERENCES

[Alexa00] M. Alexa, D. Cohen-Or, and D. Levin. As–rigid–as–possible shape interpolation. In *Proceedings of SIGGRAPH 2000*, pages 157–164. ACM, ACM Computer Graphics, 2000.

[Arad95] N. Arad and D. Reisfeld. Image warping using few anchor points. *Computer Graphics Forum*, 14:35–46, 1995.

[Beier92] T. Beier and S. Neely. Feature–based image metamorphosis. In *Proceedings of SIGGRAPH 1992*, volume 26, pages 35–42. ACM Computer Graphics, 1992.

[Carmel97] E. Carmel and D. Cohen-Or. Warp–guided object space morphing. *The Visual Computer*, 13:465–478, 1997.

[Catmull78] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer–Aided Design*, 10:350–355, September 1978.

[Claes00] Johan Claes, Frank Van Reeth, and Marc Ramaekers. Locally interpolating subdivision surfaces supporting free–form 2D deformations. In *Proceedings of Deform 2000*, pages 59–71, 2000.

[Claes01] Johan Claes, Marc Ramaekers, and Frank Van Reeth. Providing local interpolation, tension and normal control in the manipulation of loop subdivision surfaces. In *CGI 2001*, pages 299–305, Hong Kong, July 2001.

[Goldstein95] E. Goldstein and C. Gotsman. Polygon morphing using a multiresolution representation. In *Proceedings of Graphics Interface 1995*, pages 247–254, 1995.

[Johan00] H. Johan, Y. Koiso, and T. Nishita. Morphing using curves and shape interpolation techniques. In *Proceedings of Pacific Graphics 2000, 8th Pacific Conference on Computer Graphics and Applications*, pages 348–358, 2000.

[Kent92] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In *Proceedings of SIGGRAPH 1992*, volume 26(2), pages 47–54, 1992.

[Kochanek84] D. Kochanek and R. Bartels. Interpolating splines with local tension, continuity and bias control. In *Proceedings of SIGGRAPH 1984*, volume 18, pages 33–41, July 1984.

[Lee96] S. Lee, G. Wolberg, K.-Y. Chwa, and S. Y. Shin. Image metamorphosis with scattered feature constraints. *IEEE Transactions on Visualization and Computer Graphics*, 2(14):337–354, 1996.

[Lee98] S. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin. Fast feature-based volume metamorphosis and operator design. In *Computer Graphics Forum (Eurographics 1998)*, volume 17(3), pages 15–22, 1998.

[Loop87] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.

[Nishita93] T. Nishita, T. Fujii, and E. Nakamae. Metamorphosis using Bézier clipping. In *Proceedings of 1st Pacific Conference on Computer Graphics and Applications*, pages 162–173, 1993.

[Sederberg93] Thomas W. Sederberg and E. Greenwood. A physically based approach to 2D shape blending. *Computer Graphics*, 26:25–34, 1993.

[Shapira95] M. Shapira and A. Rappoport. Shape blending using a star–skeleton representation. *IEEE Computer Graphics and Applications*, 15:44–51, 1995.

[Tal99] A. Tal and G. Elber. Image morphing with feature preserving texture. In *Proceedings of Eurographics 1996*, volume 18(3), pages 339–348, 1999.

[Wolberg90] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, ISBN: 0–8186–8944–7, 1990.

[Wolberg98] G. Wolberg. Image morphing survey. *The Visual Computer*, 14(8/9):360–372, 1998.

[Zorin00] D. Zorin, P. Schröder, A. Levin, L. Kobbelt, W. Sweldens, and T. DeRose. Subdivison for modelling and animation. *SIGGRAPH 2000 Course notes 23*, July 2000.