Interactive Shape Metamorphosis

David T. Chen*, Andrei State* and David Banks‡

*Department of Computer Science
University of North Carolina at Chapel Hill

‡Institute for Computer Applications in Science and Engineering

1 INTRODUCTION

Image metamorphosis (morphing) is a powerful and easy-to-use tool for generating new 2D images from existing 2D images. In recent years morphing has become popular as an artistic tool and is used extensively in the entertainment industry. In this paper we describe a new technique for controlled, feature-based metamorphosis of certain types of surfaces in 3-space; it applies well-understood 2D methods to produce shape metamorphosis between 3D models in a 2D parametric space. We also describe an interactive implementation on a parallel graphics multicomputer, which allows the user to define, modify and examine the 3D morphing process in real time.

2 PREVIOUS WORK

Wolberg [4] described a point correspondence technique for morphing 2D images. Consider a pair of 2D source images, A and B. If a feature in image A is meant to match a feature in image B, the user chooses a point within the feature of each image. When the point morphs from A to B, so does a neighborhood surrounding it. By defining such pairs of points for all interesting features, the user can create a metamorphosis sequence between the two static images A and B.

Beier and Neely [1] described a segment correspondence technique for morphing 2D images. When a feature in image A is required to transform to a feature in image B, a line segment is drawn over the feature in each image. As the segment morphs from A to B, so does a neighborhood surrounding it. By judiciously creating line segments, the user can preserve all the important features throughout the morph. This technique is easier to use than the point correspondence method; usually fewer than half as many line segment pairs than point pairs are required to define a morph sequence between two static images.

2D methods provide simple user control for image-based morphing. However, since little or no information about actual 3D geometry is available, it is difficult to create "natural"-looking transformations; morphing animations created with 2D methods often exhibit a subtle "flattening" effect.

Kent, Carlson and Parent [3] described a method for morphing 3D polyhedral objects by merging the topologies of two 3D source polyhedra A and B. New vertices, edges, and faces are added to both A and B so that every polygon of A corresponds to a polygon of B. To morph between them one interpolates between corresponding vertices. The user can exercise some control over how the correspondences are established, but only very indirectly, by selecting a specific method of mapping the two source objects onto a common intermediate mapping surface used for topology merging (for example, a sphere). Kent concludes:

> ... techniques that provide a finer level of control over the transformation are needed. One possibility is to add a warping step ... before the topologies are merged.

We implemented Beier's technique as that warping step for the special case of cylindrical mapping surfaces, warping the model's 2D parameter space instead of a (projected) 2D image.

3 CONTROLLED 2D-3D MORPHING

Our method consists of morphing the common intermediate mapping surface or 2D parameter space of a pair of surface models. We use Beier's techniques to establish correspondences and accomplish the warping. The 2D nature of the process makes interaction easy. While defining correspondences, the user can simultaneously inspect the two parametric images as well as the resulting surface in 3-space .

We begin with a pair of surface models A and B (Figure 1) which have been meshed over some parameter space. Models in other formats (like polygon-lists, NURBS, or implicit surfaces) must be resampled and meshed so that they have similar parameter spaces. This may seem like a relatively harsh restriction,

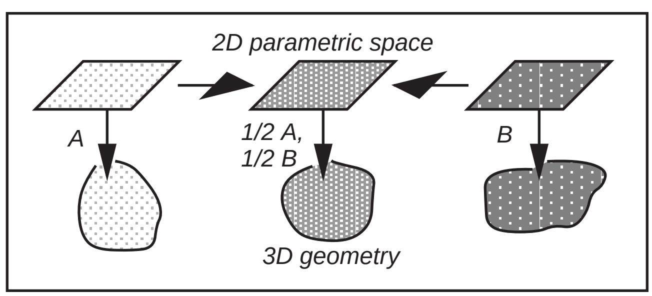

Figure 1. Object A is morphed into Object B. The objects are parametric surfaces. To interpolate between the geometries, interpolate between the 2D parameter spaces.

*Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, North Carolina, 27599-3175. (chen|state)@cs.unc.edu.

‡Institute for Computer Applications in Science and Engineering, Mail Stop 132C, NASA Langley Research Center, Hampton, Virginia, 23681. banks@icase.edu.

making the technique applicable only to convex or star-shaped objects. However, there are physically-based and model-specific projection techniques [3] that can be applied to more complex geometries.

All the surface attributes of the source models must be available in the 2D parameter space so that they may be interpolated. There are map-parameters attached to each sample as well. For example, in the case of a spherically-projected apple, the map-parameter is the radius at each sample point. Knowing the radius, one can reconstruct the surface of the original apple and attach the sample's surface attributes to it.

The surface attributes are interpolated as well as the samples' map-parameters. The interpolated map-parameters serve to construct the morphed target model from a morphed image in the 2D parametric space. The 3D target model is derived from this image by applying the mappings; in doing so, we use the "morphed" values of the map-parameters at each sample point to construct the surface of the target.

4 INTERACTIVE IMPLEMENTATION

We have implemented a prototype system on the Pixel-Planes 5 graphics multicomputer, a heterogeneous system consisting of over 30 Intel i860-based MIMD nodes and a massively parallel array of SIMD pixel processors [2]. We chose Beier's technique for its easy and intuitive control methods. We demonstrate our method on 3D models of human heads generated by a 3D scanner (Cyberware™). These models are represented in cylindrical coordinates (with the mantle of the cylinder serving as the 2D parameter space for the morphing process). Our samples contain the surface attribute color and the map-parameter radius. Traditional morphing between 2D images operates on color as a function of 2D pixel coordinates; here we operate on color and radius as functions of the 2D parametric coordinates angle and height.

The software design of the system is straightforward: the entire 2D parameter space of each of the two source models with surface attributes and map-parameters is replicated on all MIMD nodes. Each node generates a subset of the morphed parametric image. The nodes then apply the morphed colors and map-parameters to generate colored polygons from the morphed parametric image (Plate 1).

Plate 2a shows a pair of 2D parametric images on which a user has marked features. The background images show the color intensity of the models in the parameter space of cylindrical coordinates. Plate 2b shows the radii (essentially height functions) in cylindrical coordinates, mapped to gray intensity values. Note the pairs of line segments: they establish correspondences between various features of the two source models in the Beier-Neely technique. These features may be chosen simply by their similar color (like matching the red regions of lips in a 2D image), but they may also be chosen by their similar 3D geometry (like matching the pointed tip of each nose). This latter ability is crucial for matching features in regions of constant color. These regions are prominent in profile, but not in the general projected views. It would be inefficient to search for corresponding features by continually rotating the objects until their features are identifiable by their colors alone.

Plate 3 shows a sequence of shape metamorphosis images generated by our system. Mapped onto the surfaces of the 3D models, the line segments become surface-following curves. The face rotates as it is morphed to demonstrate how the geometric features are preserved during the interpolation. Notice, for example, how the lips spread open as the morphing progresses. Notice also that one of the eyes is obscured in the left image. Pure image-based morphing cannot interpolate between features when one of them is obscured under a particular viewing projection.

The entire process of matching features and warping between the surfaces in Plate 3 takes only a few minutes for a trained user. The 274-by-222 surface mesh with 33 pairs of line segments for correspondence definition is morphed and rendered on Pixel-Planes 5 at 20 frames per second (4-by-4 decimation) or at 1 frame per second (full resolution).

5 FUTURE WORK

We are currently working on a true 3D interface for our system. This will allow the user to specify correspondence areas directly on the 3D source objects, while continuing to use 2D parametric space morphing techniques. In the future we plan to add support for other types of parametric spaces besides cylindrical projections. Then our system could allow controlled shape metamorphosis for many of the classes of 3D objects described in [3].

6 CONCLUSIONS

We have described how to apply image-based metamorphosis to parametrically defined 3D surfaces or arbitrary surfaces that can be expressed parametrically using projection techniques described in [3]. For such surfaces our method is superior to ordinary image-based warping: the warp is defined only once (rather than frame-by-frame) for an entire animation and can be accomplished in a short interactive session. Since our method provides local correspondence definition, it is superior to previous techniques that automatically map between surfaces in a global manner. The technique is easily parallelizable; on our prototype system the interpolating surfaces can be constructed and displayed at interactive frame rates.

Finally, the animation sequences produced with our method do not exhibit the "flattening" effect typical for image-based morphing. Our sequence has an intuitively three-dimensional "look," noticeable not only in high-resolution animations, but also at lower resolutions, during interactive operation.

We would like to acknowledge the contributions of Andrew Brandt, Anselmo Lastra, Greg Turk and David Addleman.

REFERENCES

1. Beier, Thaddeus, and Shawn Neely. "Feature-Based Image Metamorphosis." Proceedings of SIGGRAPH '92 (Chicago, Illinois, July 26-31, 1992). In Computer Graphics 26, 2 (July 1992), pp 35-42.

2. Fuchs, Henry et. al. "Pixel-Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor Enhanced Memories." Proceedings of SIGGRAPH '91 (Las Vegas, Nevada, July 29-August 2, 1991). In Computer Graphics 25, 4 (July 1991), pp 79-88.

3. Kent, James R., Wayne E. Carlson, and Richard E. Parent. "Shape Transformation for Polyhedral Objects." Proceedings of SIGGRAPH '92 (Chicago, Illinois, July 26-31, 1992). In Computer Graphics 26, 2 (July 1992), pp 47-54.

4. Wolberg, G. Digital Image Warping. IEEE Computer Society Press, 1990.

Plate 1. Parallel 2D-3D morphing. Each stripe is processed on a different computing node.

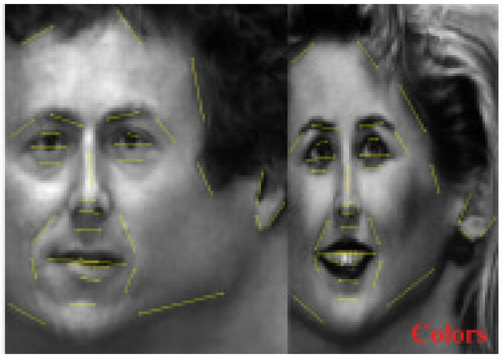

Plate 2a. Line segments define similar features in two parametric models of human heads. The grayscale background image represents surface color.

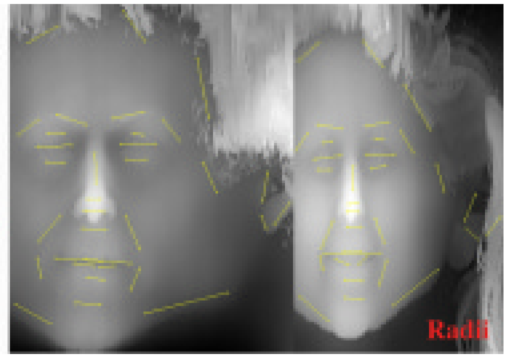

Plate 2b. The line segments are overlayed over grayscale images representing radial distance in cylindrical coordinates.

Plate 3. Metamorphosis of surface shapes in 3-space: 0%, 33%, 67%, and 100% versions of the David -> Heidi sequence.