

Recent Advances in Image Morphing *

George Wolberg

Department of Computer Science
City College of New York / CUNY
New York, NY 10031

wolberg@cs-mail.engr.ccny.cuny.edu

Abstract

Image morphing has been the subject of much attention in recent years. It has proven to be a powerful visual effects tool in film and television, depicting the fluid transformation of one digital image into another. This paper reviews the growth of this field and describes recent advances in image morphing in terms of three areas: feature specification, warp generation methods, and transition control. These areas relate to the ease of use and quality of results. We will describe the role of radial basis functions, thin plate splines, energy minimization, and multilevel free-form deformations in advancing the state-of-the-art in image morphing. Recent work on a generalized framework for morphing among multiple images will be described.

1. Introduction

Image metamorphosis has proven to be a powerful visual effects tool. There are now many breathtaking examples in film and television depicting the fluid transformation of one digital image into another. This process, commonly known as *morphing*, is realized by coupling image warping with color interpolation. Image warping applies 2D geometric transformations on the images to retain geometric alignment between their features, while color interpolation blends their color.

Image metamorphosis between two images begins with an animator establishing their correspondence with pairs of feature primitives, e.g., mesh nodes, line segments, curves, or points. Each primitive specifies an image feature, or landmark. The feature correspondence is then used to compute mapping functions that define the spatial relationship between all points in both images. Since mapping functions are central to warping, we shall refer to them as warp

functions in this paper. They will be used to interpolate the positions of the features across the morph sequence. Once both images have been warped into alignment for intermediate feature positions, ordinary color interpolation (i.e., cross-dissolve) is performed to generate inbetween images.

Feature specification is the most tedious aspect of morphing. Although the choice of allowable primitives may vary, all morphing approaches require careful attention to the precise placement of primitives. Given feature correspondence constraints between both images, a warp function over the whole image plane must be derived. This process, which we refer to as warp generation, is essentially an interpolation problem. Another interesting problem in image morphing is transition control. If transition rates are allowed to vary locally across inbetween images, more interesting animations are possible.

The explosive growth of image morphing is due to the compelling and aesthetically pleasing effects possible through warping and color blending. The extent to which artists and animators can effectively use morphing tools is directly tied to solutions to the following three problems: feature specification, warp generation, and transition control. Together, they influence the ease and effectiveness in generating high-quality metamorphosis sequences. This paper describes recent advances in image morphing in terms of their role in addressing these three problems. Comparisons are given between various morphing techniques, including those based on mesh warping [14], field morphing [2], radial basis functions [1], thin plate splines [9, 6], energy minimization [7], and multilevel free-form deformations [8].

A tradeoff exists between the complexity of feature specification and warp generation. As feature specification becomes more convenient, warp generation becomes more formidable. The recent introduction of spline curves to feature specification raises a challenge to the warp generation process, making it the most critical component of morphing. It influences the smoothness of the transformation and dominates the computational cost of the morphing process.

*Presented at: Computer Graphics International '96, Pohang, Korea. June 1996



Figure 1. Cross-dissolve

We shall comment on these tradeoffs and describe their role in influencing recent progress in this field.

2. Morphing Algorithms

Before the development of morphing, image transitions were generally achieved through the use of cross-dissolves, e.g., linear interpolation to fade from one image to another. Fig. 1 depicts this process applied over five frames. The result is poor, owing to the double-exposure effect apparent in misaligned regions. This problem is particularly apparent in the middle frame, where both input images contribute equally to the output. Morphing achieves a fluid transformation by incorporating warping to maintain geometric alignment throughout the cross-dissolve process.

In this section we review several morphing algorithms, including those based on mesh warping, field morphing, radial basis functions, thin plate splines, energy minimization, and multilevel free-form deformations. This review is intended to motivate the discussion of progress in feature specification, warp generation, and transition control.

2.1. Mesh Warping

Mesh warping was pioneered at Industrial Light & Magic (ILM) by Douglas Smythe for use in the movie *Willow* in 1988. It has been successfully used in many subsequent motion pictures. To illustrate the 2-pass mesh warping algorithm, consider the image sequence shown in Fig. 2. The five frames in the middle row represent a metamorphosis (or morph) between the two faces at both ends of the row. We will refer to these two images as I_S and I_T , the source and the target images, respectively. The source image has mesh M_S associated with it that specifies the coordinates of control points, or landmarks. A second mesh, M_T , specifies their corresponding positions in the target image. Meshes M_S and M_T are respectively shown overlaid on I_S and I_T in the upper left and lower right images of the figure. Notice that landmarks such as the eyes, nose, and lips lie below corresponding grid lines in both meshes. Together, M_S and M_T are used to define the spatial transformation that maps

all points in I_S onto I_T . The meshes are constrained to be topologically equivalent, i.e., no folding or discontinuities are permitted. Therefore, the nodes in M_T may wander as far from M_S as necessary, as long as they do not cause self-intersection. Furthermore, for simplicity, the meshes are constrained to have frozen borders.

All intermediate frames in the morph sequence are the product of a 4-step process:

for each frame f do

 linearly interpolate mesh M , between M_S and M_T

 warp I_S to I_1 , using meshes M_S and M

 warp I_T to I_2 , using meshes M_T and M

 linearly interpolate image I_f , between I_1 and I_2

end

Fig. 2 depicts this process. In the top row of the figure, mesh M_S is shown deforming to mesh M_T , producing an intermediate mesh M for each frame f . Those meshes are used to warp I_S into increasingly deformed images, thereby deforming I_S from its original state to those defined by the intermediate meshes. The identical process is shown in reverse order in the bottom row of the figure, where I_T is shown deforming from its original state. The purpose of this procedure is to maintain the alignment of landmarks between I_S and I_T as they both deform to some intermediate state, producing the pairs of I_1 and I_2 images shown in the top and bottom rows, respectively. Only after this alignment is maintained does a cross-dissolve between successive pairs of I_1 and I_2 become meaningful, as shown in the morph sequence in the middle row. This sequence was produced by applying the weights $[1, .75, .5, .25, 0]$ and $[0, .25, .5, .75, 1]$ to the five images in the top and bottom rows, respectively, and adding the two sets together. This process demonstrates that morphing is simply a cross-dissolve applied to warped imagery. The important role that warping plays here is readily apparent by comparing the morph sequence in Fig. 2 with the cross-dissolve result in Fig. 1.

The use of meshes for feature specification facilitates a straightforward solution for warp generation: bicubic spline interpolation. The example above employed Catmull-Rom spline interpolation to determine the correspondence of all

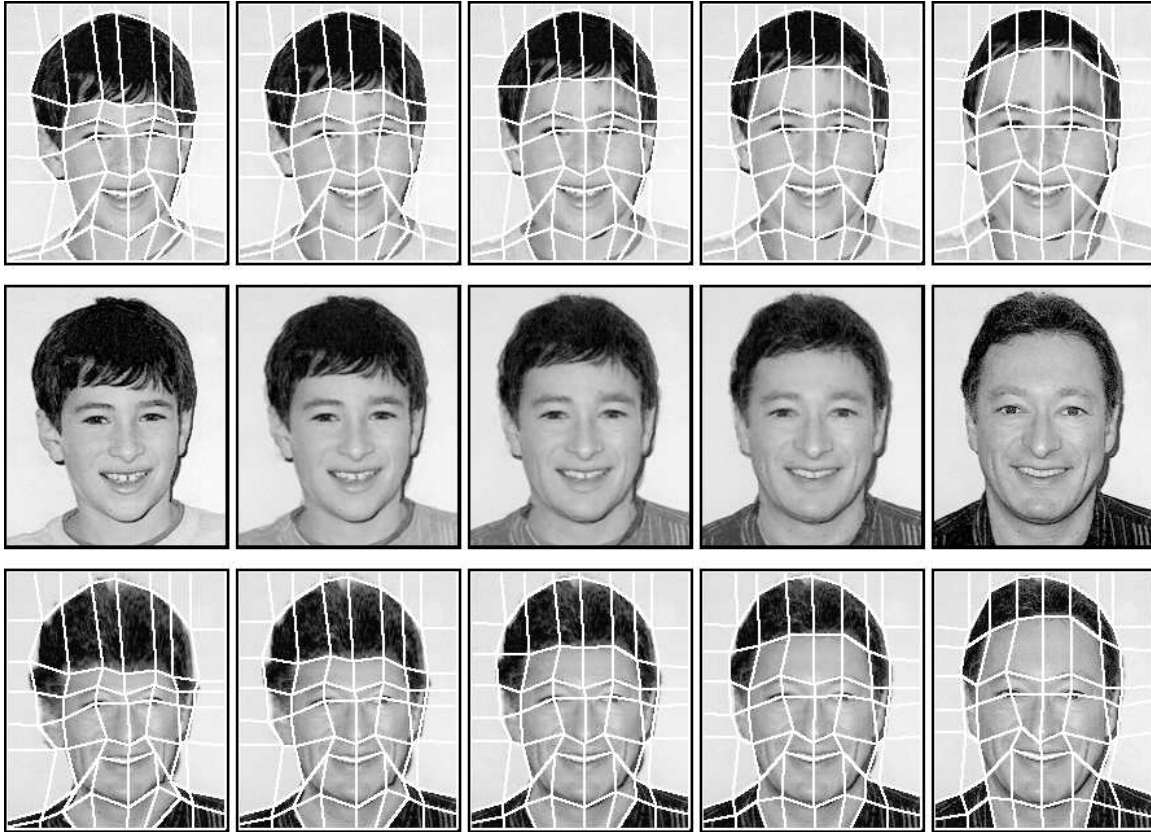


Figure 2. Mesh warping

pixels. Fant’s algorithm was used to resample the image in a separable implementation [4, 14].

2.2. Field Morphing

While meshes appear to be a convenient manner of specifying pairs of feature points, they are, however, sometimes cumbersome to use. The field morphing algorithm developed by Beier and Neely [2] at Pacific Data Images grew out of the desire to simplify the user interface to handle correspondence by means of line pairs. A pair of corresponding lines in the source and target images defines a coordinate mapping between the two images. In addition to the straightforward correspondence provided for all points along the lines, the mapping of points in the vicinity of the line can be determined by their distance from the line. Since multiple line pairs are usually given, the displacement of a point in the source image is actually a weighted sum of the mappings due to each line pair, with the weights attributed to distance and line length.

This approach has the benefit of being more expressive than mesh warping. For example, rather than requiring the correspondence points of Fig. 2 to all lie on a mesh, line pairs can be drawn along the mouth, nose, eyes, and cheeks

of the source and target images. Therefore only key feature points need be given.

Although this approach simplifies the specification of feature correspondence, it complicates warp generation. This is due to the fact that all line pairs must be considered before the mapping of each source point is known. This global algorithm is slower than mesh warping, which uses bicubic interpolation to determine the mapping of all points not lying on the mesh. A more serious difficulty, though, is that unexpected displacements may be generated after the influence of all line pairs are considered at a single point. Additional line pairs must sometimes be supplied to counter the ill-effects of a previous set. In the hands of talented animators, though, the mesh warping and field morphing algorithms have both been used to produce startling visual effects.

2.3. Radial Basis Functions / Thin Plate Splines

The most general form of feature specification permits the feature primitives to consist of points, lines, and curves. Since lines and curves can be point sampled, it is sufficient to consider the features on an image to be specified by a set of points. In that case, the x - and y -components of a warp

can be derived by constructing the surfaces that interpolate scattered points. Consider, for example, M feature points labeled (u_k, v_k) in the source image and (x_k, y_k) in the target image, where $1 \leq k \leq M$. Deriving warp functions that map points from the target image to the source image is equivalent to determining two smooth surfaces: one that passes through points (x_k, y_k, u_k) and the other that passes through (x_k, y_k, v_k) for $1 \leq k \leq M$.

This formulation permits us to draw upon a large body of work on scattered data interpolation to address the warp generation problem. All subsequent morphing algorithms have facilitated general feature specification by appealing to scattered data interpolation.

Warp generation by this approach was extensively surveyed in [11, 14]. Recently, two similar methods were independently proposed using the thin plate surface model [6, 9]. Another method using radial basis functions was described in [1]. These techniques generate smooth warps that exactly reflect the feature correspondence. Furthermore, they offer the most general form of feature specification since any primitive (e.g., spline curves) may be sampled into a set of points. Elastic Reality, a commercial morphing package from Avid Technology, uses curves to enhance feature specification. Their warp generation method, however, is unpublished.

2.4. Energy Minimization

All of the methods described above do not guarantee the one-to-one property of the generated warp functions. When a warp is applied to an image, the one-to-one property prevents the warped image from folding back upon itself. An energy minimization method has been proposed for deriving one-to-one warp functions in [7]. That method allows extensive feature specification primitives such as points, polylines, and curves. Internally, all primitives are sampled and reduced to a collection of points. These points are then used to generate a warp, interpreted as a 2D deformation of a rectangular plate. A deformation technique is provided to derive C^1 -continuous and one-to-one warps from the positional constraints. The requirements for a warp are represented by energy terms and satisfied by minimizing their sum. The technique generates natural warps since it is based on physically meaningful energy terms. The performance of that method, however, is hampered by its high computational cost.

2.5. Multilevel Free-Form Deformation

A new warp generation method was presented in [8] that is much simpler and faster than the related energy minimization method in [7]. Large performance gains are achieved by applying multilevel free-form deformation (MFFD) across

a hierarchy of control lattices to generate one-to-one and C^2 -continuous warp function. In particular, warps were derived from positional constraints by introducing the MFFD as an extension to free-form deformation (FFD) [12]. In that paper, the bivariate cubic B-spline tensor product was used to define the FFD function. A new direct manipulation technique for FFD, based on 2D B-spline approximation, was applied to a hierarchy of control lattices to exactly satisfy the positional constraints. To guarantee the one-to-one property of a warp, a sufficient condition for a 2D cubic B-spline surface to be one-to-one was presented. The MFFD generates C^2 -continuous and one-to-one warps which yield fluid image distortions. The MFFD algorithm was combined with the energy minimization method of [7] in a hybrid approach.

An example of MFFD-based morphing is given in Fig. 3. Notice that the morph sequence shown in the middle row of the figure is virtually identical to that produced using mesh warping in Fig. 2. The benefit of this approach, however, is that feature specification is more expressive and less cumbersome. Rather than editing a mesh, a small set of feature primitives are specified. To further assist the user, snakes are introduced to reduce the burden of feature specification. Snakes are energy minimizing splines that move under the influence of image and constraint forces. They were first adopted in computer vision as an active contour model [5]. Snakes streamline feature specification because primitives must only be positioned near the features. Image forces push snakes toward salient edges, thereby refining their final positions and making it possible to capture the exact position of a feature easily and precisely.

2.6. Discussion

The progression of morphing algorithms has been marked by more expressive and less cumbersome tools for feature specification. A significant step beyond meshes was made possible by the specification of line pairs in field morphing. The complications that this brought to warp generation, however, sometimes undermined the usefulness of the approach. For instance, the method sometimes demonstrated undesirable artifacts, referred to as *ghosts*, due to the computed warp function [2]. To counter these problems, the user is required to specify additional line pairs, beyond the minimal set that would otherwise be warranted. All subsequent algorithms, including those based on radial basis functions, thin plate splines, and energy minimization, formulated warp generation as a scattered data interpolation problem and sought to improve the quality (smoothness) of the computed warp function. They do so at relatively high computational cost. The newest approach, based on the MFFD algorithm, significantly improves matters by accelerating warp generation. The use of snakes further assists the user in reducing the burden of feature specification.

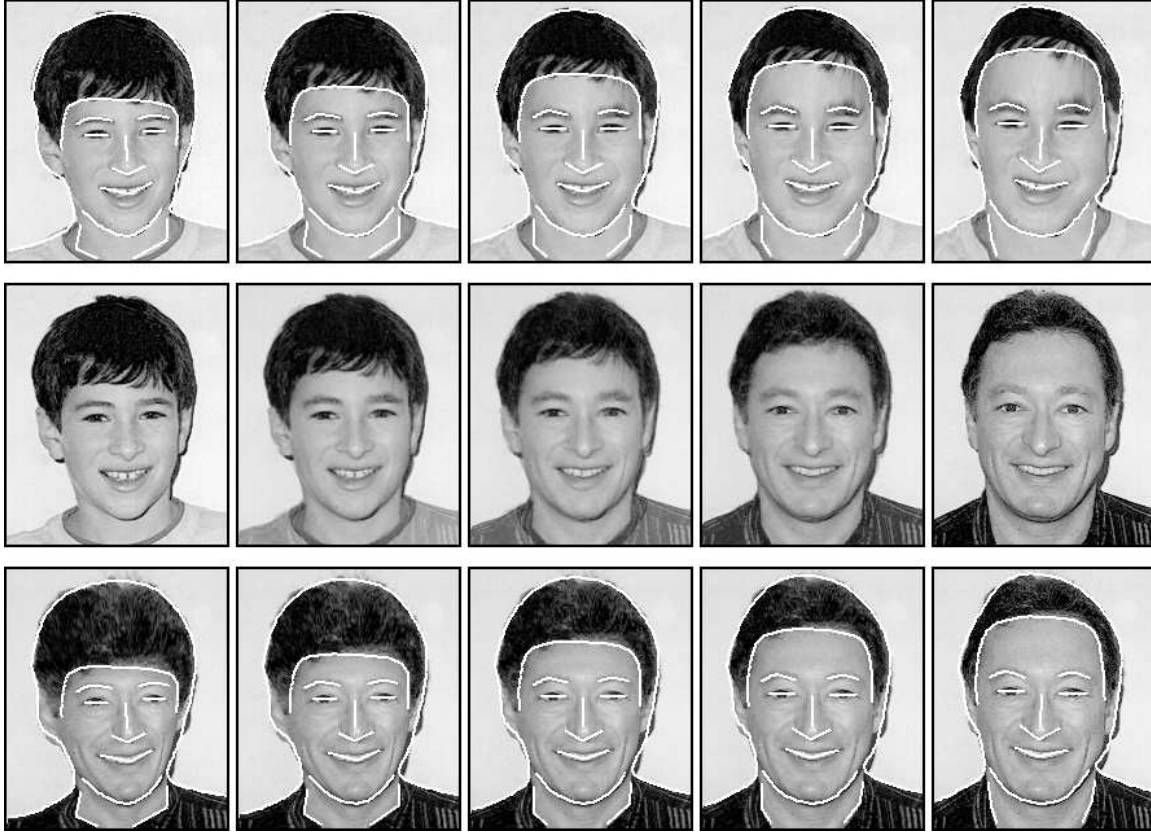


Figure 3. MFFD-based morphing

3. Transition Control

Transition control determines the rate of warping and color blending across the morph sequence. If transition rates differ from part to part in inbetween images, more interesting animations are possible. Such nonuniform transition functions can dramatically improve the visual content. Note that the examples shown thus far all used a uniform transition function, whereby the positions of the source features steadily moved to their corresponding target positions at a constant rate.

Figs. 4 and 5 show examples of the use of uniform and nonuniform transition functions, respectively. The upper left and lower right images of Fig. 4 are the source and target images, respectively. The features used to define the warp functions are shown overlaid on the two images. The top and bottom rows depict a uniform transition rate applied to the warping of the source and target images, respectively. Notice, for instance, that all points in the source and target images are moving at a uniform rate to their final positions. Those two rows of warped imagery are attenuated by the same transition functions and added together to yield the middle row of inbetween images. Note

that geometric alignment is maintained among the two sets of warped inbetween images before color blending merges them into the final morph sequence.

The example in Fig. 5 demonstrates the effects of a nonuniform transition function applied to the same source and target images. In this example, a transition function was defined that accelerated the deformation of the nose of the source image, while leaving the shape of the head intact for the first half of the sequence. The deformation of the head begins in the middle of the sequence and continues linearly to the end. The same transition function was used for the bottom row. Notice that this use of nonuniform transition functions is responsible for the dramatic improvement in the morph sequence.

Transition control in mesh-based techniques is achieved by assigning a transition curve to each mesh node. This proves tedious when complicated meshes are used to specify the features. Nishita *et al.* mentioned that the transition behavior can be controlled by a Bézier function defined on the mesh [10].

In the energy minimization method, transition functions are obtained by selecting a set of points on a given image and specifying a transition curve for each point. Although earlier

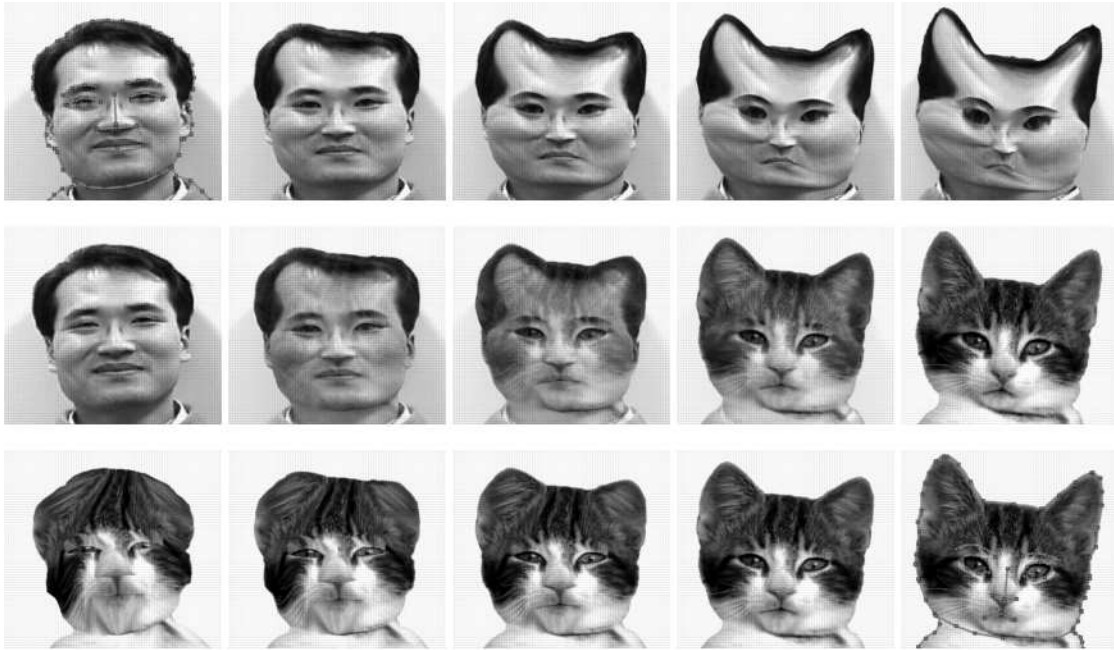


Figure 4. Uniform metamorphosis

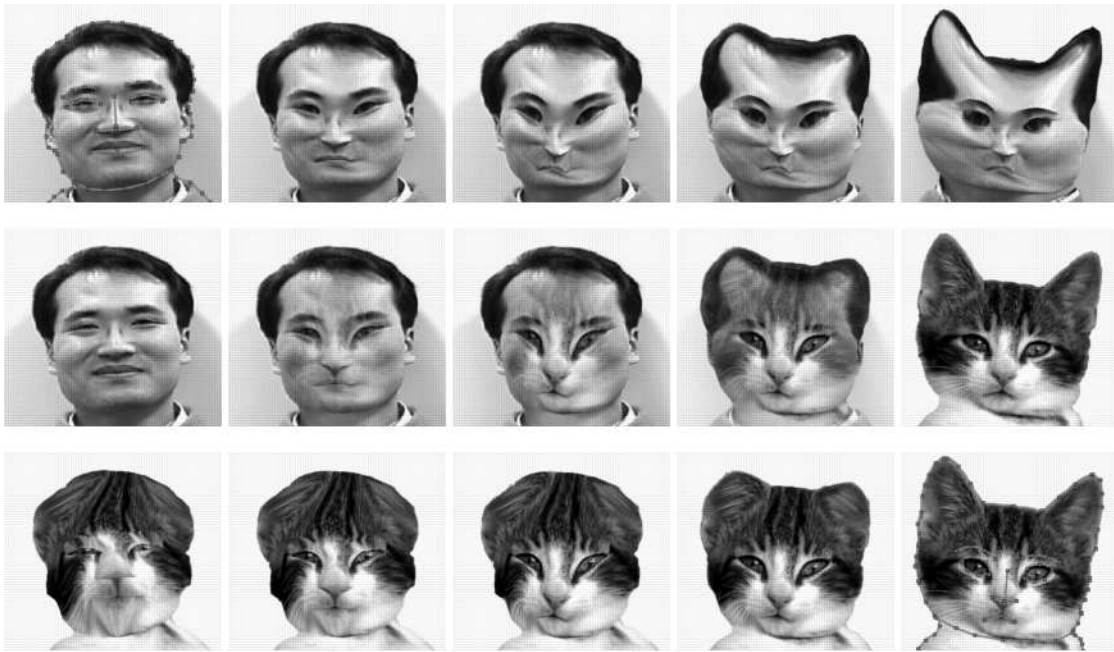


Figure 5. Nonuniform metamorphosis

morphing algorithms generally coupled the feature specification and transition control primitives, this method permits them to be decoupled. That is, the location of transition control primitives must not necessarily coincide with those of

the features. The transition curves determine the transition behavior of the selected points over time. For a given time, transition functions must have the values assigned by the transition curves at the selected points. Considering a tran-



Figure 6. Procedural transformation

sition rate as the vertical distance from a plane, transition functions are reduced to smooth surfaces that interpolate a set of scattered points. The thin plate surface model [13] was employed to obtain C^1 -continuous surfaces for transition functions.

In the MFFD-based approach, the MFFD technique for warp generation was simplified and applied to efficiently generate a C^2 -continuous surface for deriving transition functions. The examples in Figs. 4 and 5 were generated using the MFFD-based morphing algorithm.

Transition curves can be replaced with procedural transition functions [7, 8]. An example is depicted in Fig. 6, where a linear function varying in the vertical direction is applied to two input images. The result is a convincing transformation in which one input image varies into the other from top to bottom.

4. Future Work

The traditional formulation for image morphing considers only two input images at a time, i.e., the source and target images. In that case, morphing among multiple images is understood to mean a series of transformations from one image to another. This limits any morphed image to take on the features and colors blended from just two input images. Given the success of morphing using this paradigm, it is reasonable to consider the benefits possible from a blend of more than two images at a time. For instance, consider the generation of a facial image that is to have its eyes, ears, nose, and profile derived from four different input images. In this case, morphing among multiple images is understood to mean a seamless blend of several images at once.

Despite the explosive growth of morphing in recent years, the subject of morphing among multiple images has been neglected. In ongoing work conducted by the author and his colleagues, a general framework is being developed that extends the traditional image morphing paradigm applied to

two images. We formulate each input image to be a vertex of a regular convex polyhedron in $(n - 1)$ -dimensional space, where n is the number of input images. An inbetween (morphed) image is considered to be a point in the convex polyhedron. The barycentric coordinates of that point determine the weights used to blend the input images into the inbetween image.

Morphing among multiple images is ideally suited for image composition applications where elements are seamlessly blended from two or more images. A composite image is treated as a metamorphosis of selected regions in several input images. The regions seamlessly blend together with respect to geometry and color. In future work, we will determine the extent to which the technique produces high quality composites with considerably less effort than conventional image composition techniques. In this regard, the technique can bring to image composition what image warping has brought to cross-dissolve in deriving morphing: a richer and more sophisticated class of visual effects that are achieved with intuitive and minimal user interaction.

Future work in morphing will also address the automation of morphing among limited classes of images and video sequences. Consider a limited, but common, class of images such as facial images. It should be possible to use computer vision techniques to automatically register features between two images. As the examples given in Figs. 2 and 3 demonstrate, facial images require feature primitives to be specified along the eyes, nose, mouth, hair, and profile. Model-based vision should be able to exploit knowledge about the relative position of these features and automatically locate them for feature specification [3]. Currently, this is an active area of research, particularly for compression schemes designed for videoconference applications. The same automation applies to morphing among two video sequences, where time-varying features must be tracked. Interested readers may refer to the recent proceedings of the 1995 IEEE International Conference on Image Processing (Washington, D.C)

for several papers on facial image processing and motion tracking. These papers provide a good description of the state-of-the-art as well as future directions for these challenging problems.

5. Conclusions

This paper has reviewed the growth of image morphing and described recent advances in the field. Morphing algorithms all share the following components: feature specification, warp generation, and transition control. The ease with which an artist can effectively use morphing tools is determined by the manner in which these components are addressed. We compared various morphing techniques, including those based on mesh warping, field morphing, radial basis functions, thin plate splines, energy minimization, and multilevel free-form deformations.

The earliest morphing approach was based on mesh warping. It was motivated by a reasonably straightforward interface requiring meshes to mark features and bicubic spline interpolation to compute warp functions. The field morphing approach attempted to simplify feature specification with the use of line pairs to select landmarks. This added benefit demanded a more computationally expensive warp generation stage. Subsequent morphing algorithms have sought to maintain the use of curves and polylines to select features. Warp generation has consequently become formulated as a scattered data interpolation problem. Standard solutions such as radial basis functions and thin plate splines have been demonstrated.

The newest approach based on multilevel free-form deformations has further accelerated warp generation. That same approach demonstrated the use of snakes to assist the user in placing feature primitives, thereby reducing the burden in feature specification. Snakes are particularly useful when features lie along large intensity gradients.

Transition control determines the rate of warping and color blending across the morph sequence. If transition rates differ from part to part in inbetween images, more interesting animations are possible. The same techniques used to compute warp functions may be applied for transition control functions, thereby propagating that information everywhere across the image. Although early morphing algorithms generally coupled the feature specification and transition control primitives, more recent algorithms have permitted them to be decoupled.

Future work includes morphing among multiple images, and automating morphing among a class of images (e.g., facial images) and video sequences. The latter problem requires features to be tracked across video frames. Progress in this area is tied to that of the motion estimation research community.

Acknowledgements

This work was supported in part by an NSF Presidential Young Investigator award (IRI-9157260) and a PSC-CUNY grant (RF-666338). The author would like to thank Seung-Yong Lee for valuable discussions and help with some of the figures.

References

- [1] N. Arad, N. Dyn, D. Reissfeld, and Y. Yeshurun. Image warping by radial basis functions: Applications to facial expressions. *CVGIP: Graphical Models and Image Processing*, 56(2):161–172, March 1994.
- [2] T. Beier and S. Neely. Feature-based image metamorphosis. *Computer Graphics (Proc. SIGGRAPH '92)*, 26(2):35–42, 1992.
- [3] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.
- [4] K. M. Fant. A nonaliasing, real-time spatial transform technique. *IEEE Computer Graphics and Applications*, 6(1):71–80, January 1986.
- [5] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Intl. J. of Computer Vision*, pages 321–331, 1988.
- [6] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin. Image morphing using deformable surfaces. *Proc. Computer Animation '94*, pages 31–39, 1994. IEEE Computer Society Press.
- [7] S.-Y. Lee, K.-Y. Chwa, J. Hahn, and S. Y. Shin. Image morphing using deformation techniques. *J. Visualization and Computer Animation*, 7(1):3–23, 1996.
- [8] S.-Y. Lee, K.-Y. Chwa, S. Y. Shin, and G. Wolberg. Image metamorphosis using snakes and free-form deformations. *Computer Graphics (Proc. SIGGRAPH '95)*, pages 439–448, 1995.
- [9] P. Litwinowicz and L. Williams. Animating images with drawings. *Computer Graphics (Proc. SIGGRAPH '94)*, pages 409–412, 1994.
- [10] T. Nishita, T. Fujii, and E. Nakamae. Metamorphosis using Bézier clipping. *Proc. First Pacific Conf. on Computer Graphics and Applications*, pages 162–173, 1993. Seoul, Korea, World Scientific Publishing Co.
- [11] D. Ruprecht and H. Müller. Image warping with scattered data interpolation. *IEEE Computer Graphics and Applications*, 15(2):37–43, March 1995.
- [12] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. *Computer Graphics (Proc. SIGGRAPH '86)*, 20(4):151–160, 1986.
- [13] D. Terzopoulos. Multilevel computational processes for visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 24:52–96, 1983.
- [14] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.