Behavior Research Methods, Instruments, & Computers
1999, 31 (2), 359-369

Morphing techniques for manipulating face images

MARK STEYVERS
Indiana University, Bloomington, Indiana

This paper describes morphing techniques to manipulate two-dimensional human face images and three-dimensional models of the human head. Applications of these techniques show how to generate composite faces based on any number of component faces, how to change only local aspects of a face, and how to generate caricatures and anticaricatures of faces. These techniques are potentially useful for many psychological studies because they permit realistic images to be generated with precise control.

Much empirical work has focused on the perception and processing of human faces. For many experiments, researchers have generated artificial faces using a variety of techniques. In this article, I explain how morphing techniques can be used to generate and manipulate faces. For psychological studies, there are two main uses of the morphing technique. First, one can generate faces that are composites of other faces. For example, Rhodes and Tremewan (1996) have used composite faces to investigate whether composite faces are more attractive than the component faces used to create them. Several studies have already exploited morphing techniques to generate the composites (e.g., Perret, May, & Yoshikawa, 1994). Also, Beale and Keil (1995) have used composite faces to investigate whether there is categorical perception (a heightened sensitivity for stimuli at the category boundary of these stimuli) for faces. Another application of morphing is the generation of caricatures of a face (Benson & Perret, 1991). In a caricature of a face, the distinctive facial features are exaggerated. Caricatures are useful in studies investigating the effects of the distinctiveness or typicality of a face on its recognition (e.g., Mauro & Kubovy, 1992; Rhodes, Brennan, & Carey, 1987). In these kind of studies, morphing techniques have been applied to generate the caricatures (Levin, 1996).

I will explain one particular morphing algorithm, "field morphing," in detail. Beier and Neely (1992) originally introduced the basic technique for two-dimensional (2-D) images. Chen and Banks (1996) have applied the same basic technique to three-dimensional (3-D) models of the human head. I will show various applications of the technique for both 2-D images and 3-D models. These applications are all potentially useful for psychological studies involving not only faces but also just about any set of objects that share a part configuration.

There are many commercial software packages available that perform the basic morphing operations that are discussed in this paper. The majority of these packages are very user-friendly, and a morphing sequence between two images can be generated in a very short time. In this paper, I will discuss the morphing algorithm that is used in some software packages and extend these algorithms to handle problems that most software packages cannot handle—for example, to morph only parts of images while keeping other parts constant or to morph multiple images simultaneously. Also, I will discuss how to extend of the basic morphing algorithm to handle 3-D models, a feature absent in most available commercial packages.

ALTERNATE METHODS TO MANIPULATE FACE IMAGES

Many previous studies have utilized the Identi-Kit facial reproduction system to prepare facial stimuli for experiments. In the original Identi-Kit method, plastic templates, each representing a facial characteristic (eyes, nose, lips, etc.), are overlayed to form a composite face. More recent Identi-Kit methods (e.g., see Jensen, 1987) use digitized versions of the facial templates that can be manipulated electronically in the composition process. For example, the appearance, size, and position of features can be changed, and these changes can be specified parametrically. The disadvantage of this method is that the resulting faces usually look schematic.

In the method by Brennan (1985), a face is represented as a line drawing. The face image is converted to a line drawing by presenting the face on a computer screen. A set of control points is overlayed on the face on the screen, and the control points are joined by lines. The control points are moved to specific locations on the face. The lines that join the control points define the line drawing and represent the face. The same number of control points and lines are used for several different faces, and the points have specified, aligned positions on the faces.

The author thanks Rob Goldstone and Tom Busey for all their support and insights for preparing this manuscript. The author also thanks Dan Levin and Satoru Suzuki for numerous helpful comments on an earlier draft and Alice O'Toole for providing the 3-D laser scans. Correspondence should be addressed to M. Steyvers, Psychology Department, Indiana University, Bloomington, IN 47405 (e-mail: msteyver@indiana.edu).

 Copyright 1999 Psychonomic Society, Inc.

With these line drawings of faces, there are several possibilities to create new line drawings. For example, for line drawings of two faces, the corresponding control point positions can be interpolated, and the lines can be redrawn through this new set of points to generate a line drawing that resembles both faces. Another possibility is to calculate a norm or average face based on numerous line drawings of different faces. With this norm face, the extrapolation of the positions of the points of a particular face away from the norm face will yield a caricature in which the deviations from the norm are exaggerated. With this technique, continuous parametric control is possible, however, like the Identi-Kit technique, the resulting face stimuli lack realism.

MORPHING

Typically, morphing has been used as an animation tool for cinematic special effects (Wolberg, 1990). The term *morphing* is used to describe the image processing technique in which a metamorphosis is specified that transforms one image into another. For example, if there are two images, A and B, new images can be generated that transform A to B, and vice versa. I will also show how to generate an image that is a transformation based on more than two images. The morphing algorithm can generate an image anywhere along the continuum from A to B, and the position on this continuum is specified parametrically. The parametrization is a useful feature to generate images for psychological studies because it allows specification of the proportion of "A-ness" and "B-ness" in the generated image such that the similarity of the image to A and B can be controlled.

In general, there are two distinct phases in the calculation of a morph between A and B: *warping* and *cross-dissolving*. First, the image A is warped to A′ and the image B is warped to B′, which means that the shapes A and B are distorted toward A′ and B′ such that A′ and B′ have similar shapes. Then, A′ and B′ are cross-dissolved, meaning that the colors or grayscale values of images A′ and B′ are combined to form a new image that can share both shape and color aspects of A and B. There are two techniques that are widely used to warp images: *field morphing* and *mesh warping*. Mesh warping will be briefly addressed in a later section.

Field Morphing

Field morphing was introduced by Beier and Neely (1992). The user has to provide the algorithm with a set of control line segments. The segments serve to align the features of images A and B. There is no minimum or maximum number of line segments that can be used. The more control line segments there are, the more control the user has over these shape changes. For faces, for example, the line segments can be drawn over the contour line segments of the head, the ears, the eyes, the eyebrows, the lips, the nose, and so on. In Figure 1a, 127 segments are drawn over the contours and features of the face. In the warping phase of field morphing, correspondences are calculated between the pixels of the images to be morphed. The algorithm is called field morphing because every line segment excerts a field of influence on the alignment such that pixels near a segment in one image will tend to be aligned with pixels near the corresponding segment in other images. Under a certain parameter setting, the algorithm guarantees that pixels on a segment in one image will be aligned with the pixels on the corresponding segment in the other images. Finally, in the cross-dissolving phase, the colors or grayscale values of corresponding pixels are blended to create the colors or grayscale values of the morphed image. The simulations in this paper are based on the field morphing technique.

I will now first explain the field morphing method with just a single control line segment and then explain the method with any number of control line segments.

Single Control Line Segment. Let us first discuss the case in which there is only a single control line segment in two images. Let l_i represent the control line segment in image i. Let i = 0 to refer to the destination image, and i = 1 and i = 2 refer to the two source images, respectively. The control line segments are defined in terms of a pair of vectors, one representing its starting location, $\mathbf{n}_i$ (which goes from the origin of the coordinate system to the starting location of l_i), and the other representing its orientation and length, $\mathbf{m}_i$ (which goes from the starting location to the ending location of l_i). The distinction between the start and the end of a line segment is needed only to align the appropriate ends of corresponding control line segments. The first step in field morphing is to calculate the coordinates for the line segment in the destination image on the basis of the coordinates of the line segments from the source images. The following equations transform the segments l_1 and l_2 in the source images into the segment l_0 in the destination image by transforming $\mathbf{m}_1$ and $\mathbf{m}_2$ into $\mathbf{m}_0$ and by transforming $\mathbf{n}_1$ and $\mathbf{n}_2$ into $\mathbf{n}_0$:

$$\begin{aligned} \mathbf{m}_0 &= \mathbf{m}_1 + \alpha(\mathbf{m}_2 - \mathbf{m}_1) \\ \mathbf{n}_0 &= \mathbf{n}_1 + \alpha(\mathbf{n}_2 - \mathbf{n}_1). \end{aligned} \tag{1}$$

The parameter α determines the relative influence of the segments of the first and the second image. It usually varies between 0 and 1, although values outside this range will be used for generating caricatures. In Figure 2a, the destination image has one line segment whose vectors $\mathbf{m}_0$ and $\mathbf{n}_0$ are determined from $\mathbf{m}_1$, $\mathbf{n}_1$, $\mathbf{m}_2$, and $\mathbf{n}_2$, with $\alpha = ½$.

The second step is the calculation of the distortions of the two images toward the new line segments. There are two ways of doing this: a forward mapping and a reverse mapping. In the forward mapping, every pixel of the source image is mapped to a pixel in the destination image. In the reverse mapping, every pixel in the destination image is mapped to a pixel in each source image. This last method is discussed because it guarantees that

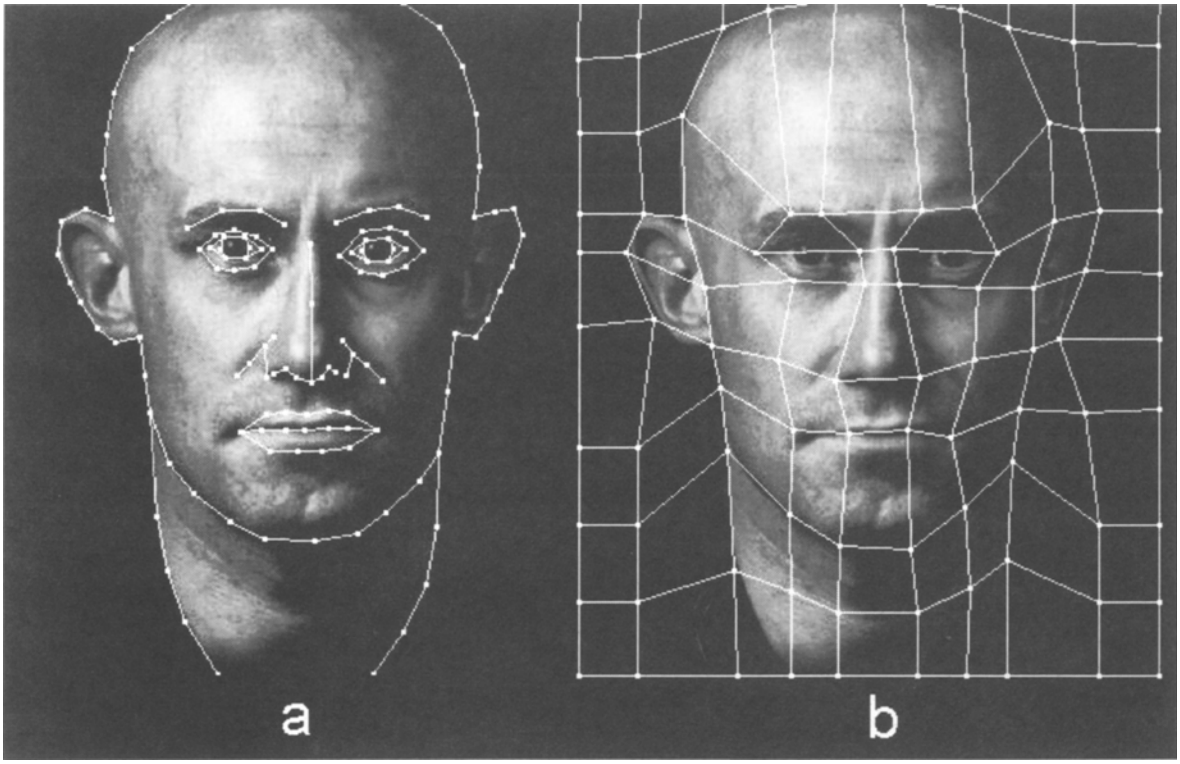

Figure 1. (a) The control line segments for a face as used for field morphing. (b) An example mesh of control points that can be used for mesh warping. The method used in panel a of specifying line segments is more intuitive and provides more control over the morphing process than the method used in panel b by aligning a grid of points to key feature locations.

every pixel in the destination image will be assigned a color or grayscale value.

The geometry of the reverse mapping algorithm for a single segment is shown in Figure 2a. For notational convenience, I will refer to pixels as *position vectors*. Let $\mathbf{p}_0$ be an arbitrary position vector in the destination image. In the reverse mapping, the corresponding position vector $\mathbf{p}_i$ has to be found for every source image i. Let $\mathbf{w}_0$ be the vector from $\mathbf{n}_0$ to $\mathbf{p}_0$. For every position vector $\mathbf{p}_0$ in the resulting image, the vectors $\mathbf{u}_0$ and $\mathbf{v}_0$ are calculated. These vectors define a relation of $\mathbf{p}_0$ to the line segment l_0: $\mathbf{u}_0$ is the projection of $\mathbf{w}_0$ onto $\mathbf{m}_0$, and $\mathbf{v}_0$ is the projection of $\mathbf{w}_0$ onto the perpendicular of $\mathbf{m}_0$. The idea behind the field morphing algorithm is that $\mathbf{p}_i$ will have a similar geometric relation to the segment l_i as $\mathbf{p}_0$ has to l_0. The implementation of this idea is to let the length of $\mathbf{u}_i$ relative to $\mathbf{m}_i$ be the same as the length of $\mathbf{u}_0$ relative to $\mathbf{m}_0$. The length of $\mathbf{v}_i$ is set to the same length as $\mathbf{v}_0$ (the same algorithm is used as was used by Beier & Neely, 1992, who found that scaling of $\mathbf{u}_i$ relative to $\mathbf{m}_i$ is useful but that scaling for $\mathbf{v}_i$ is not). Then, for source image i, $\mathbf{u}_i$ is calculated according to

$$\mathbf{u}_i = \mathbf{m}_i(\mathbf{w}_0 \cdot \mathbf{m}_0)/\|\mathbf{m}_0\|^2. \tag{2}$$

Then, $\mathbf{v}_i$ is calculated by

$$\mathbf{v}_i = [\vartheta_i/\|\vartheta_i\|]\,\|\mathbf{v}_0\|, \tag{3}$$

where $\vartheta_i/\|\vartheta_i\|$ is the unit vector perpendicular to $\mathbf{m}_i$ and ϑ_i is calculated by

$$\vartheta_i = \mathbf{w}_i - (\mathbf{w}_i \cdot \mathbf{m}_i)\mathbf{m}_i/(\mathbf{m}_i \cdot \mathbf{m}_i), \tag{4}$$

and $\|\mathbf{v}_0\|$ is given by

$$\|\mathbf{v}_0\| = \det(\mathbf{m}_0,\mathbf{w}_0)/\|\mathbf{m}_0\|. \tag{5}$$

The position vector $\mathbf{p}_i$ corresponding to $\mathbf{p}_0$ is then calculated as

$$\mathbf{p}_i = \mathbf{n}_i + \mathbf{u}_i + \mathbf{v}_i. \tag{6}$$

With these correspondences calculated for the two source images, there are two new images generated: the distortion of the first image toward the destination image and the distortion of second image toward the destination image.

The last step that is left in the morphing process is the combination of the two distortions into a single image by means of cross-dissolving. If it is assumed that the images are in grayscale with single grayscale values $C(\mathbf{p}_0)$ for each position vector $\mathbf{p}_0$, then $C(\mathbf{p}_0)$ can be obtained by combining $C(\mathbf{p}_1)$ and $C(\mathbf{p}_2)$ by the weighting rule:

$$C(\mathbf{p}_0) = C(\mathbf{p}_1) + \beta\{C(\mathbf{p}_2) - C(\mathbf{p}_1)\}, \tag{7}$$

where β is a parameter similar to α in Equation 1 to determine the relative influence of the source images. In

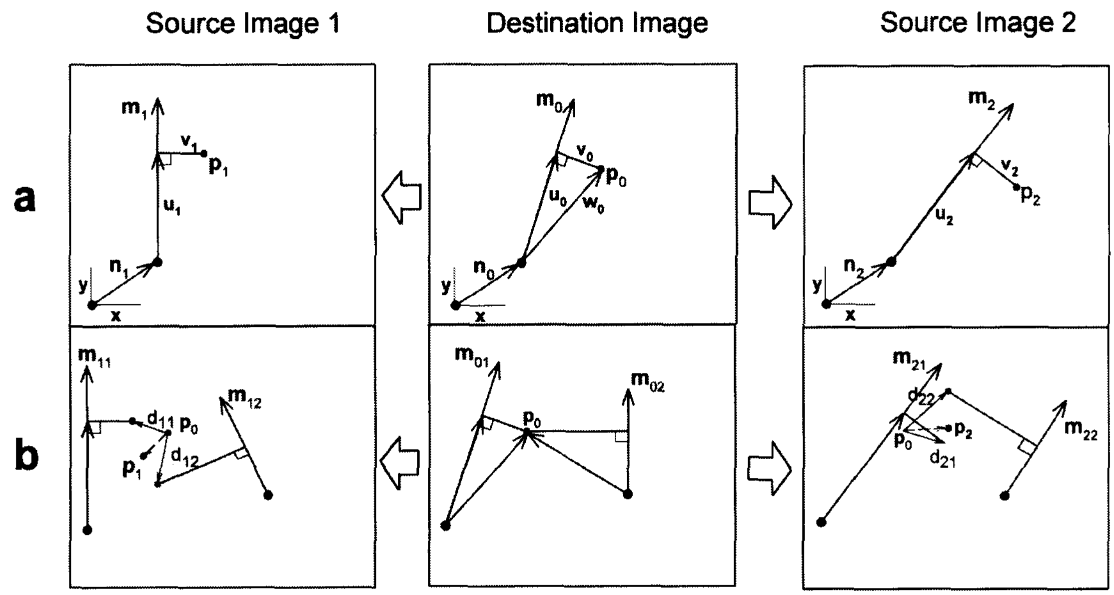

Figure 2. Geometrical example of the warping process with a single control line (a) or two control line segments (b). (a) A single control line segment of image i is defined by vectors n_i (from origin of coordinate system to start of segment) and m_i (from start to end of segment). The control line segments of the source images are determined manually, and the control line segment of the destination image is determined by a weighted average of the control line segments of the source images. In the reverse mapping procedure, the field morphing algorithm calculates for each position vector (pixel coordinate) p_0 in the destination image and the corresponding position vectors p_1 and p_2 in the two source images. The colors or grayscale values of corresponding pixels are blended together in the cross-dissolving process. The idea behind field morphing is that corresponding position vectors p_1 and p_2 will have geometrically similar positions with respect to the control line segments, as the chosen position vector p_0 has with respect to the control line segment in the destination image (see text for details). (b) Each image now has two control line segments. For every position vector p_0, calculations for each segment in each of the two source images lead to possibly different position vectors. A unique position vector p_i is calculated for image i by combining all individual position vectors in a weighted average. Various labels are omitted for reasons of clarity.

most applications, $\alpha = \beta$, so that the relative influence of both source images will be the same with respect to shape and grayscale information. In a later section, I will discuss the case of $\alpha \neq \beta$, which is desirable when generating caricatures.

Multiple control line segments. The problem of dealing with multiple control line segments will now be addressed. Given multiple segments, the application of Equations 2–6 will give a (possibly different) position vector $\mathbf{p}_i$ for each individual line segment. Therefore, each segment suggests, for each destination coordinate, a source coordinate. Each suggested source coordinate will be denoted as $\mathbf{p}_{ij}$, where j indexes the control line segment. Beier and Neely's algorithm weights the contributions of each segment to obtain a single source coordinate $\mathbf{p}_i$. Specifically, a displacement vector $\mathbf{d}_{ij}$ is calculated, which is a displacement from $\mathbf{p}_0$ to $\mathbf{p}_{ij}$. The individual displacements $\mathbf{d}_{ij}$ are summed in a weighted average to obtain the single displacement toward $\mathbf{p}_i$. Each weight is a function of the distance from $\mathbf{p}_0$ to the segment j and the length of segment j. The functions used are

$$\mathbf{W}_{ij} = \left[\frac{\left\| \mathbf{m}_{0j} \right\|^{c}}{\left(a + \left\| \mathbf{v}_{ij} \right\| \right)} \right]^{b} \tag{8}$$

$$\mathbf{p}_i = \mathbf{p}_0 + \left(\Sigma \mathbf{d}_{ij} \mathrm{w}_{ij} \right) / \left(\Sigma \mathrm{w}_{ij} \right), \tag{9}$$

where a, b, and c are parameters. In our simulations, c is set to 0, giving the length of the control line segment no influence on the weighting process. The parameter a is set to a small number (0.001) to avoid divide-by-zero errors and to ensure that if the distance from the pixel to a segment is zero, then the weight will be so large that only this segment will exert influence on the distortion of this pixel. By this method, it is guaranteed that pixels on a segment in the destination image are mapped to pixels on the corresponding segment in the source images. This fact can be used to force the alignment of particular contours in different images by tracing those contours with control segments. The parameter b determines how the relative strength of the different control line segments falls off with distance (it is set to 1). Figure 2b illustrates part of the geometry involved in computing the position vectors for the source images. For both source images, each individual segment suggests a different position vector. The position vectors are weighted according to their inverse distance to the segments. With this algorithm, pixels in a region near a segment of the first source image will correspond to pixels in a region near the segment in the other source image.

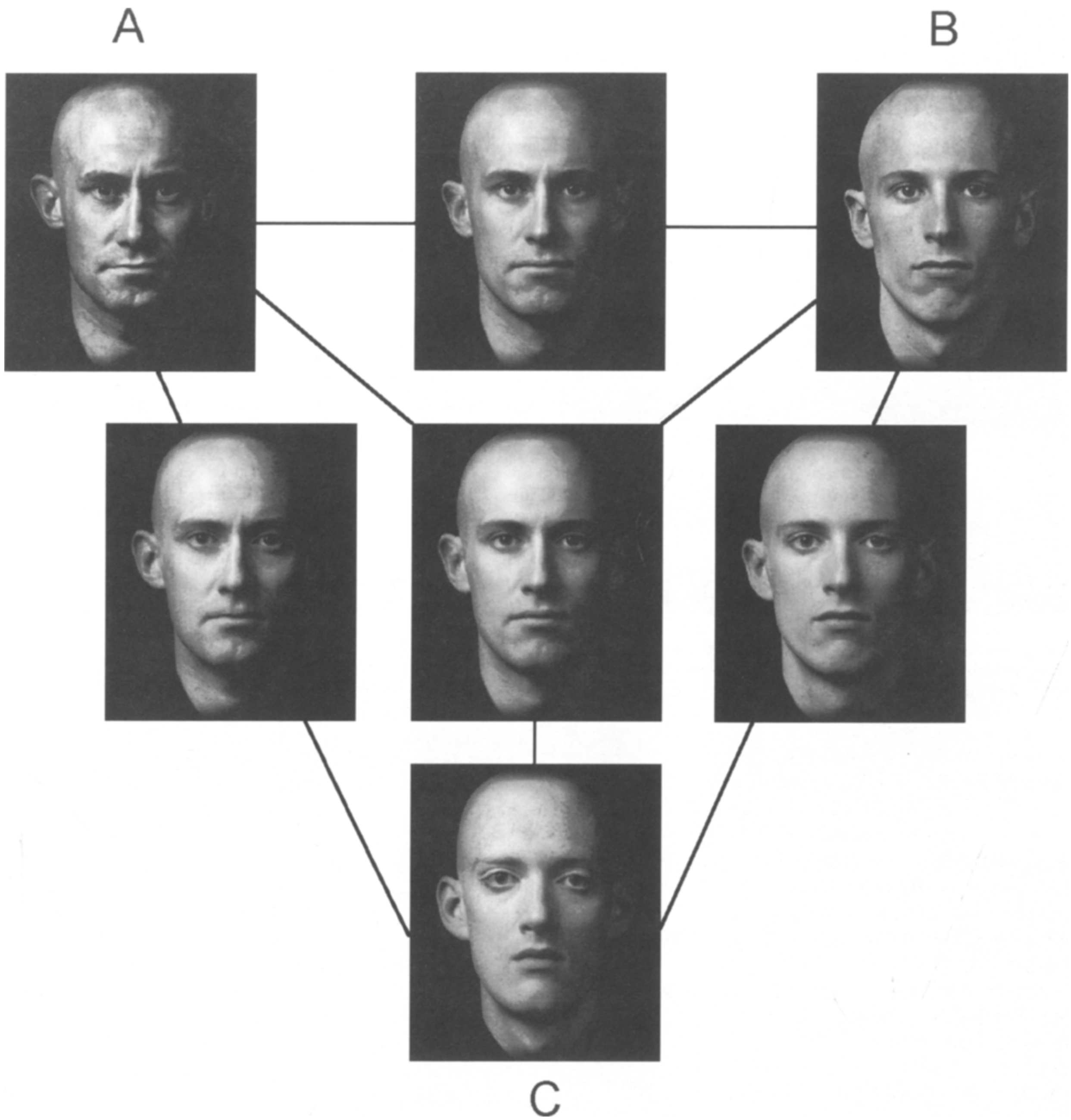

Figure 3. Of three original face images, A, B, and C are shown on the corners of the triangle. On the sides of the triangle are shown the morphs of any pair of faces. The center of the triangle shows the morph of all three faces.

Mesh Warping

An alternative is to warp images with a mesh warping algorithm (e.g., Smythe, 1990; Wolberg, 1990). Briefly, the user aligns a rectangular grid of points to key feature locations of the images. In Figure 1b, a 10 × 10 grid of points is shown aligned to the same face as in Figure 1a. In the warping phase, a destination grid is calculated with coordinates that are weighted averages of the corresponding coordinates of the source grids. For the warping algorithm, the coordinates of the source and destination grids are fitted with piecewise continuous mapping functions that are used for calculating the correspondences between the source and destination images. The warping phase then proceeds in two passes. In the first pass, an intermediate destination image is generated in which every row of the image is warped independently of the other rows. Then, in the second pass, the final destination image is created by warping every column of the intermediate image.

The advantage of the field morphing method over the mesh warping method lies in the ease of aligning the line segment to features present in an image. In the mesh

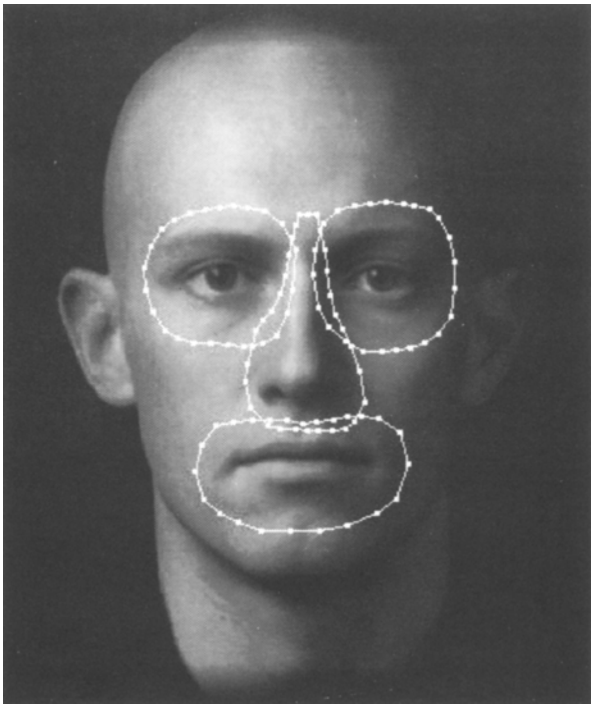

Figure 4. The lines for face A that designate parts of the faces that can be morphed separately. Four areas are shown: left and right eyes, nose, and mouth.

warping method, when diagonal contours are present in the image, one can pick either an initially vertical or an initially horizontal edge of the grid and align it to part of the diagonal contour. Sometimes, it can be difficult to make these choices when the image contains contours that change orientation (such as the contour of a face). On the other hand, it is easy to align control line segments to contours of any orientation. The disadvantage of the field morphing method is that, in general, it takes more time to compute a morph than with the mesh warping method.

CREATING COMPOSITES BASED ON MULTIPLE FACES

The basic field morphing algorithm morphs two images, and the parameter α determines the relative "A-ness" to "B-ness" of images A and B in the resulting image. It is also possible to generate images that are composites of any number of images. The general scheme for weighting the control line segments and the colors or grayscale values of the source images is

$$\mathbf{m}_{0j} = (\alpha_1 \mathbf{m}_{1j} + \alpha_2 \mathbf{m}_{2j} + \ldots + \alpha_n \mathbf{m}_{nj}) / \Sigma\alpha_i$$
$$\mathbf{n}_{0j} = (\alpha_1 \mathbf{n}_{1j} + \alpha_2 \mathbf{n}_{2j} + \ldots + \alpha_n \mathbf{n}_{nj}) / \Sigma\alpha_i \quad (10)$$

$$C(\mathbf{p}_0) = \{\alpha_1 C(\mathbf{p}_1) + \alpha_2 C(\mathbf{p}_2) + \ldots + \alpha_n C(\mathbf{p}_n)\} / \Sigma\alpha_i \quad (11)$$

When n = 2, Equations 10 and 11 reduce to Equations 1 and 7, respectively. The weights α_i are the parameters that determine the relative influences of the source images on the resulting image. This scheme was applied to three faces. For all three faces, 127 control line segments were used that were placed on and around the face, as in Figure 1. Three original faces, A, B, and C, are shown in Figure 3. They are positioned on a triangle. On the sides of the triangle, the resulting morphs of any pair of images are shown (n = 2, αs are equal). In the center of the triangle, the face with equal contributions of all three images is shown (n = 3, αs are equal). This image was used in the caricature generation process discussed in a later section. While judging the quality of the images generated is a very subjective process, a comparison of these images with the original faces they were based on shows that they are remarkably realistic.

CREATING COMPOSITES WITH ONLY PARTS OF THE FACE BEING MORPHED

It is also possible to change only certain aspects of a face and leave other features intact. For example, it might be desirable in some studies to change only the eyes or the mouth of a face while keeping the rest of the face constant. In order to do this, areas need to be specified in which the changes occur. In Figure 4, areas are shown bounded by segments that delineate the left eye, the right eye, the nose, and the mouth area. In Figure 5, two examples are shown in which either the eyes of face A are replaced by the eyes of face B or the mouth of face A is replaced by the mouth of face B. These images were created by a program that slightly varies from the basic morphing program. If the eyes have to change from A to B and the rest of the face remains A, then the destination control line segments that trace the eye features are a weighted average between A and B (as before) and all other segments to be equal to the segments of A. Furthermore, in the areas of the eyes defined by the bounded areas around the eyes in Figure 4, the grayscale values of A and B are weighted averages. Outside the eye areas, the grayscale values are copied from face A. To avoid seams at the boundary of the eye area, the cross-dissolving equation (Equation 7) was augmented to allow for a gradual transition at the boundary of the eye region:

$$C(\mathbf{p}_o) = \begin{cases} \eta\left\{(1-\alpha)C(\mathbf{p}_1) + \alpha C(\mathbf{p}_2)\right\} + (1-\eta)C(\mathbf{p}_1) & \text{if } \mathbf{p}_0 \text{ in eyes area,} \\ C(\mathbf{p}_1) & \text{otherwise.} \end{cases} \quad (12)$$

Image 1 and 2 correspond to images A and B, respectively. The parameter α (between 0 and 1) denotes the contribution of the eyes of A relative to B. The variable η depends on the distance d_{min} of the pixel $\mathbf{p}_0$ to the nearest segment of the segments that define the eye area:

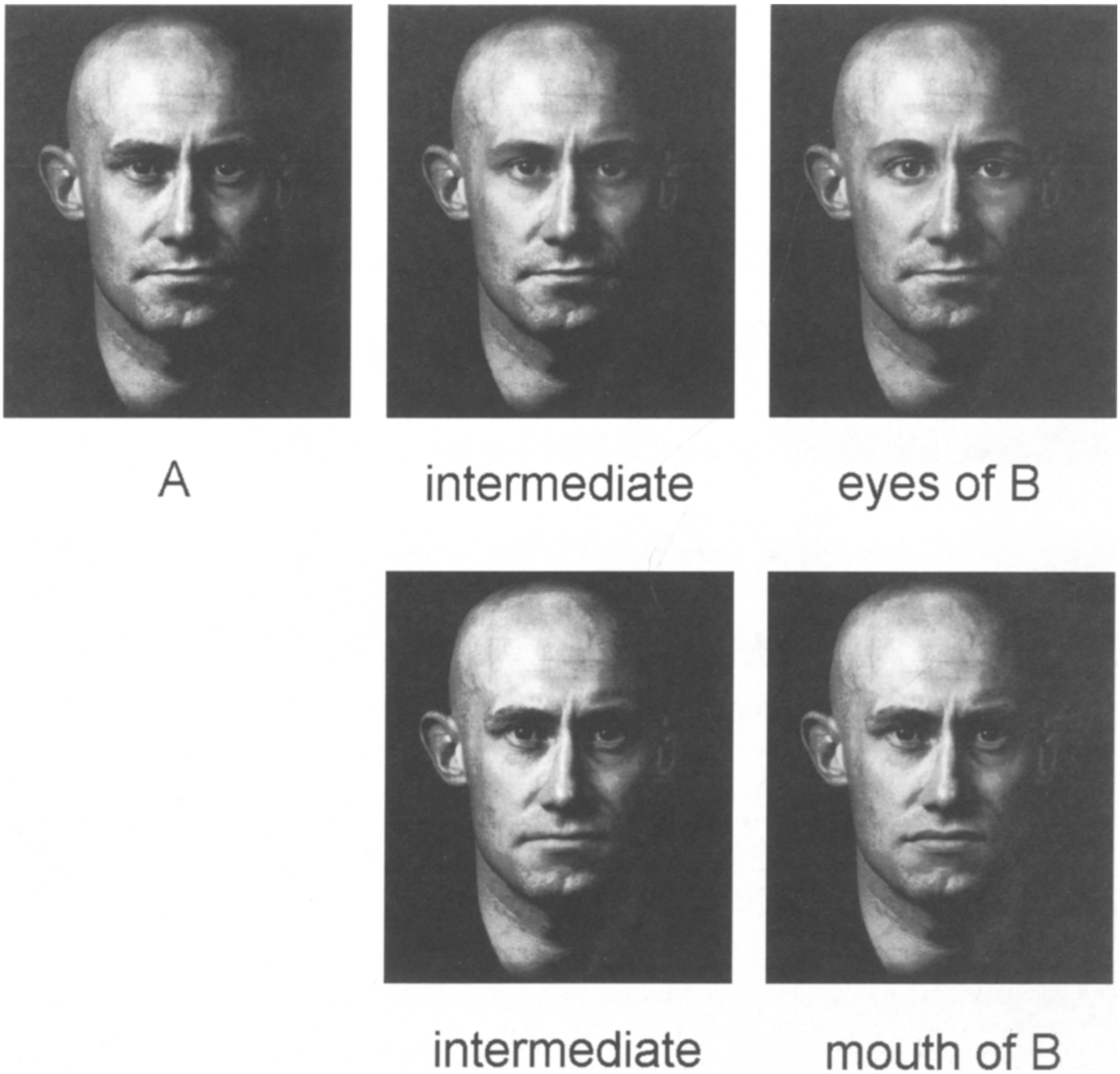

Figure 5. Face A (upper left) has eyes replaced with the eyes of face B (upper right) or the mouth of B (lower right). In the middle upper figure, both face A and B contribute to the eyes, whereas only face A contributes for the rest of the face. Similarly, in the middle lower figure, both face A and B contribute to the mouth, whereas the rest of the face comes from face A.

$$\eta = 1 - \exp(-d_{min}/e), \tag{13}$$

where the parameter e determines how fast the cross-dissolving changes from just depending on face A to depending on a weighted average of A and B.

This method of changing parts of faces has an important advantage over other approaches. With digitized Identi-Kit materials, one can vary parts of faces by pasting the parts on the image of the face and then smoothing and blurring the image at the seams so that the seams disappear or are less visible. In this blurring process, the texture is significantly altered and texture is added that does not correspond to either the background face or the individual face parts. Here, the method does incorporate a smoothing process but only to smooth the changes in the cross-dissolving process; all grayscale values originate either from the source face or from a weighted average of two source images.

CREATING FACE CARICATURES

In Brennan's caricature generator, a caricature of a face is generated by extrapolating the positions of the control line segments of a face away from the positions of the control line segments of a norm face (the norm face is a composite of several faces). In this way, all deviations from the norm are exaggerated, and the distinctive features be-

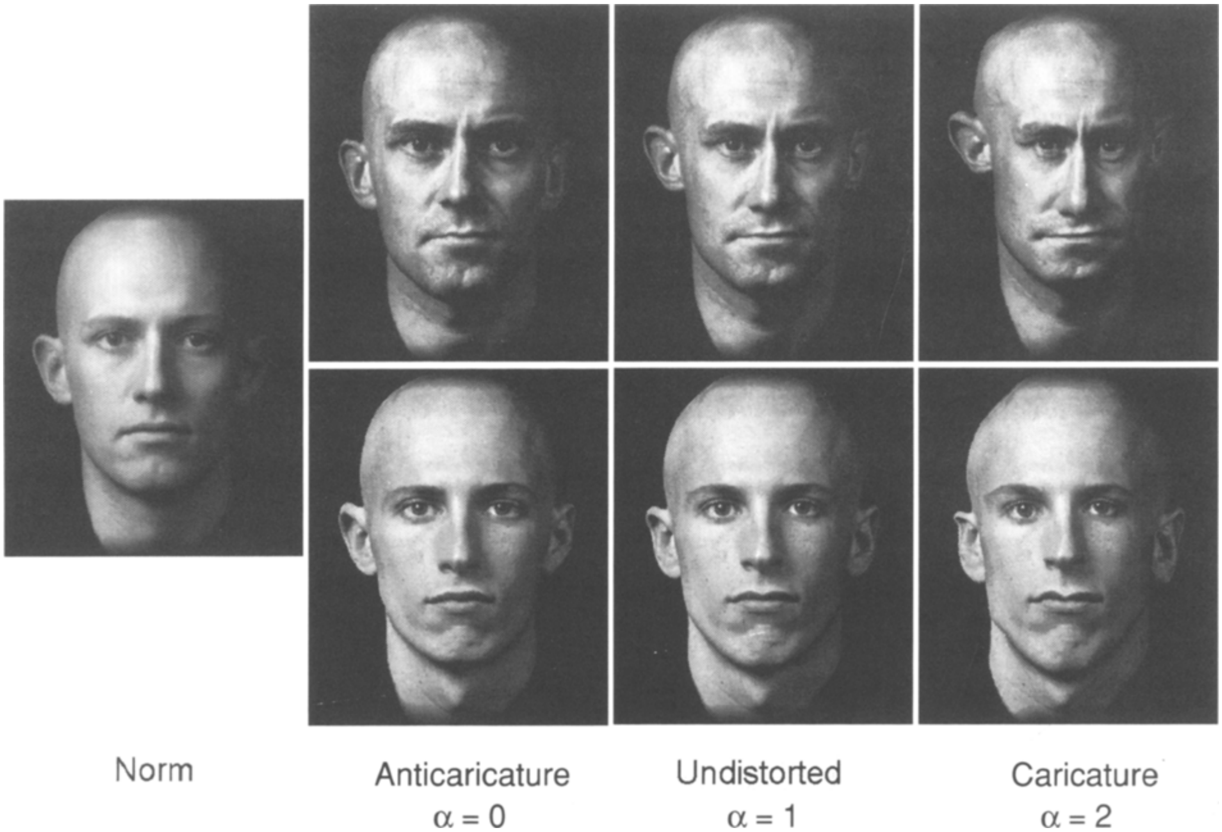

Figure 6. On the left, the norm face of faces A, B, C, and D is shown. From left to right, the anticaricatures, the undistorted images, and the caricatures are shown for face A (upper row) and face B (lower row). For both the anticaricatures and the caricatures, the grayscale values originate solely from the undistorted image. Therefore, there is no smoothing or blending of the colors of the undistorted image with the norm image. The caricatures are generated by extrapolating the metric differences of the control points of the undistorted image away from the norm. The anticaricature is generated by interpolating the metric differences toward the norm.

come more apparent. An anticaricature is generated by interpolating the control points of the face and the norm face. The anticaricature of a face has its distinctive features deemphasized. Numerous studies have utilized this method to create caricatures (Benson & Perret, 1991; Levin, 1996).

In the approach in this paper, extrapolation amounts to setting α in Equation 1 to values higher than 1. A similar parameter, β, occurs in Equation 7, where the grayscale colors of the source images are blended. While it is possible to set β in Equation 7 to values higher than 1 to extrapolate the colors from one source image away from the colors of another source image, this may not be desirable in some studies. The problem can be seen in Figure 6, where the norm face and the original undistorted faces A and B are shown. Note that the norm was obtained by averaging the control line segment coordinates of the three faces and by blending the grayscale colors of the three faces. The resulting norm face looks more symmetrical than any of the three faces it was based on because the asymmetries in the designated facial features are averaged out in the norm face. The resulting norm face also looks smoother in skin composition. This is because many irregularities in the skin and the variations in skin texture are averaged out in the blending process. If grayscale colors are allowed to be extrapolated, then, in the caricature, all irregularities in the skin and the texture will be emphasized, which can lead to bizarre results. The interpolation (or extrapolation) of the colors of a face with another face may lead to a confounding factor in experiments. For example, in an experiment by Langlois and Roggman (1990), the attractiveness of faces was judged by subjects for both average faces and original faces on which the average faces were based. The results indicated that average faces were judged to be more attractive than the original faces. As pointed out by Rhodes and Tremewan (1996), the confound is that the average faces look more attractive because all irregularities of the skin are blurred and smoothed by the blending process—this factor alone making the face more attractive, so that the method could have contributed to the results obtained. One solution (Rhodes & Tremewan, 1996) is to use line drawings instead of composite pictures to circumvent the smoothing problem. Another is to sharpen the composite picture with image processing software to remove some of the averaging-induced smoothness.

The smoothing problem can also be circumvented in another way. If the grayscale colors for the caricature and

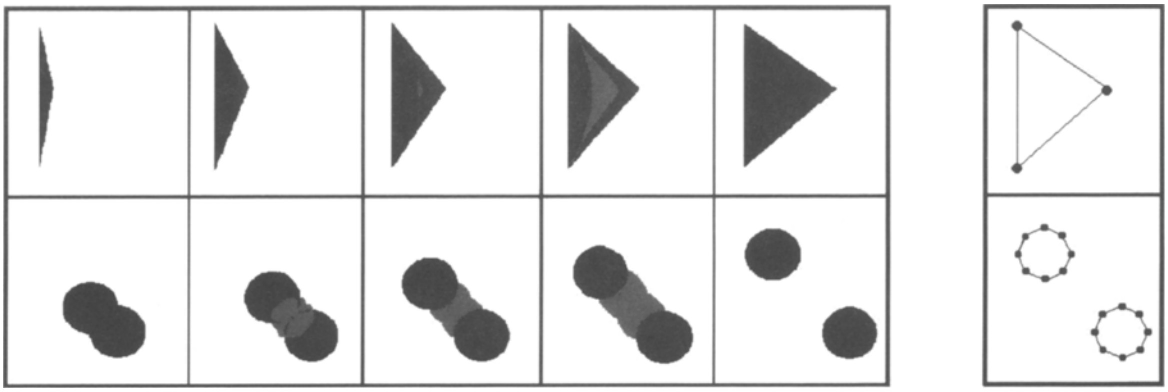

Figure 7. The field morphing algorithm can lead to undesirable results on some very simple objects. In the top row, the left triangle is morphed with the triangle shown on the right. In the bottom row, two overlapping circles are morphed with two separated circles. The control line segments that were used are shown in the rightmost figure. For the triangles, the two warped source images did not always align perfectly, resulting in gray areas where pixels inside one triangle are blended with pixels outside the other triangle. Even though the line segments enclose the triangles in both source images, the field morphing algorithm does not guarantee that pixels within one enclosed area are aligned uniquely with pixels within the other enclosed area. For the circles, pixels outside the enclosed areas are aligned with pixels that originate from within an enclosed area.

the anticaricatures originate only from the undistorted source image, then the grayscale colors do not participate in the interpolation or extrapolation. Only the control line coordinates are then subject to interpolation and extrapolation. If the norm image is indexed with i = 1 and the undistorted source image with i = 2, then the parameter setting $\beta = 1$ is kept and α is varied between 0 (anticaricature) and 2 (caricature). If, in addition to setting α to 0, β would also be set to 0, then the resulting image would be identical to the original norm face. However, since $\beta = 1$, the anticaricatures have the same grayscale information as the undistorted image but have their features aligned to positions identical to the feature positions in the norm image. The caricatures also have the same grayscale information as the undistorted image but have their features aligned to positions that are extrapolations of the feature positions of the undistorted image away from the feature positions of the norm image. The results of this approach are shown in Figure 6.

UNDESIRABLE RESULTS

Sometimes, the application of the field morphing algorithm leads to undesirable results, even with very simple geometrical figures. In the top row of Figure 7, an example is given in which the user attempts to morph two different triangles. The leftmost panel shows the first source image, and the panel second to the right shows the second source image. In the rightmost panel, the control line segments are shown that were used for the second source image. The user expected that by putting three control line segments around the edges of the two trian-

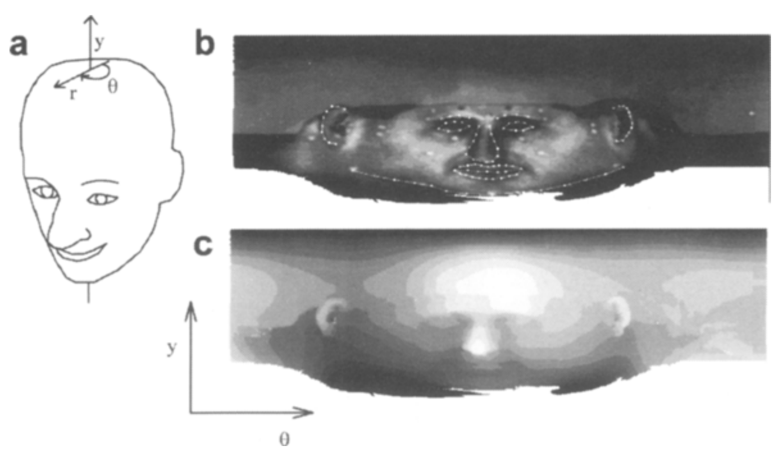

Figure 8. (a) The cylindrical coordinates *r*, θ, and *y* are used as a coordinate system for the head. (b) The grayscale map of one laser scan. (c) The corresponding range map, where the distance from the central axis, *r*, is shown as a function of *y* and θ. Lighter gray in the range map indicates a higher value of *r*.

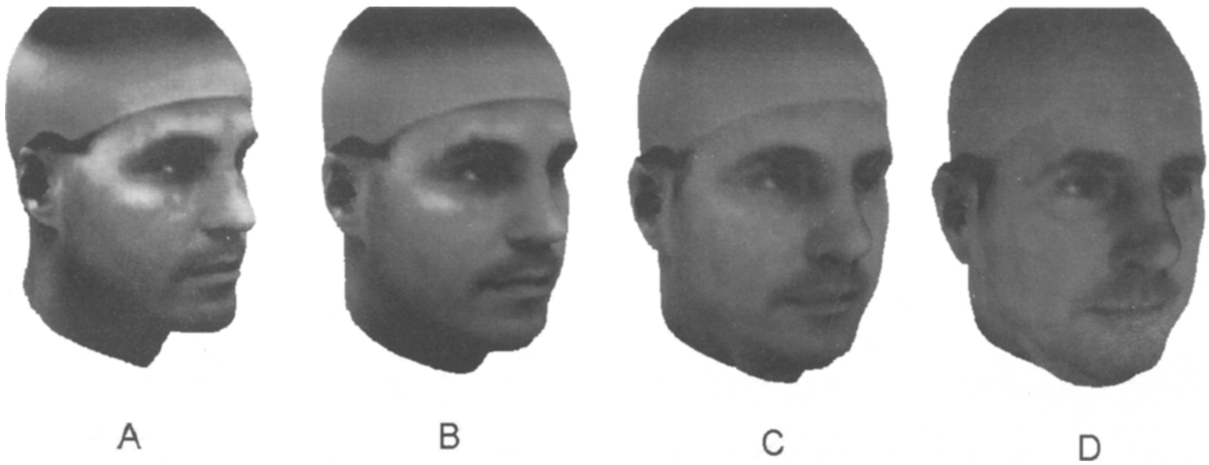

Figure 9. Heads A and D are original heads displayed under different rotations. Head B is a morph that is combination of 66% A and 33% D, and head C is a morph that is a combination of 33% A and 66% D.

gles, the field morphing algorithm would align points falling within one triangle, with points falling within the other triangle, and points falling outside one triangle, with points falling outside the other triangle. The three intermediate panels in Figure 7 show the morphs that were produced by the field morphing algorithm. Two intermediate morphs show unexpected and undesirable gray areas within the triangle. These are areas where pixels from within a triangle of one source image are blended with pixels outside the triangle of the other source image. This means that the algorithm produced correspondences between pixels that were unintended by the user.

While the field morphing algorithm can guarantee that points that lie on a control line segment in one source image are aligned with points that lie on the corresponding control line segment of the other source image, the algorithm does not guarantee that points within an area enclosed by control line segments from one source image are aligned only with points falling within the corresponding area from another source image. One way to alleviate this problem is to add extra control line segments so that the area that one wants to enclose is divided into smaller areas. Sometimes, it also helps to tinker with parameters a, b, and/or c in Equation 8. If possible, the user has to avoid using shapes that require acute angles between neighboring line segments, but this does not help in all circumstances, as illustrated in the bottom row of Figure 7. Here, the user attempts to morph two overlapping circles with two separated circles. The user expects that the intermediate morphs will be images showing two circles with varying degrees of separation. For the circles that overlap, the control line segments intersect each other at two points. In the resulting morphs, undesirable gray areas show up between the two circles. In these areas, pixels from within one circle from one source image are blended with pixels outside the circle in the other source image. This can be avoided by not allowing intersections between line segments to occur. In general, the user should not expect the algorithm to work for every kind of image. Images with many rounded features, such as faces, are easier to morph with this algorithm than are images that have sharply angled features. The algorithm also works better with images where the configuration of the parts in one image is not much different from the configuration of the parts in the other image.

EXTENDING THE BASIC ALGORITHM TO 3-D FACE MODELS

Chen and Banks (1996) applied the field morphing algorithm to 3-D models of the human head. The approach is relatively simple: convert the 3-D model to two 2-D maps, then apply the basic morphing algorithms to the two 2-D maps, and, finally, convert the morphed 2-D maps back to a 3-D model. I will demonstrate the procedure on laser scans (Cyberware) of human heads. The laser scans provide two data files that represent the "wrap-around images" of the head. The first data file contains the grayscale or color data of the head. This data can be displayed on a map with on the two axes: the cylindrical coordinates, y and θ, of the head. The second data file contains the range data—the distances of the points on the head to the central axis that runs through the center of the bottom and the top of the head. This data can also be displayed as a map with cylindrical coordinates y and theta. The two maps have, at every matched point, (y,θ), the grayscale or color information of that point on the head and the distance to the central axis. Figure 8 shows the grayscale map and the range map of a face and the line segments that were used for the morphing algorithm.

As for the basic field morphing algorithm, a set of line segments has to be aligned with the features of each head (e.g., eyes, nose, mouth, ears, chin). Then, field morphing can be applied to the grayscale (or color) map with corresponding line segment set to create a new grayscale (or color) map. The algorithm is also applied on the range maps with the same corresponding sets of line segments. This creates a new range map. The two created maps represent the morph of the head that are used in the morphing process. The maps can then be combined to display a 3-D version of the morph. This is shown in Figure 9, in which heads A and D are original heads and displayed under slightly different rotations. Head B is a morph that was created by mixing in 66% of A and 33% of D. Similarly, head C is a morph that was created by mixing in 33% of A and 66% of D. The morphs B and C are displayed in Figure 9, with viewing angles between those of A and D. The nice feature of a 3-D morph calculated in this way is that the morphed head can be viewed from any angle.

CONCLUSION

Although numerous software packages are now available that allow the user to morph images, these packages lack some options that are potentially useful for researchers who want to create and manipulate images for behavioral experiments. The present article shows how a basic morphing algorithm can be extended to morph multiple images simultaneously, how to morph only parts of an image, how to create face caricatures, and how to morph 3-D models of the human head. Although, in all examples in this article, human faces were used, the techniques should work equally well on other objects that share a part configuration.

Software Specifics

All source code that was used in this paper to generate the figures is available from the author. The source code for the field morphing algorithm was written in standard C and is virtually platform independent since it does not use graphics. The source code for the program that interactively edits the line segments that need to be aligned with the image of the face was written in Turbo Pascal and can run on an IBM-compatible machine with a standard VGA graphics mode 320 × 200 with 256 colors.

REFERENCES

Beale, J. M., & Keil, F. C. (1995). Categorical effects in the perception of faces. *Cognition*, **57**, 217-239.

Beier, T., & Neely, S. (1992). Feature-based image metamorphosis. *Computer Graphics*, **26**, 35-42.

Benson, P. J., & Perret, D. I. (1991). Synthesizing continuous-tone caricatures. *Image & Vision Computing*, **9**, 123-129.

Brennan, S. E. (1985). The caricature generator. *Leonardo*, **18**, 170-178.

Chen, D. T., & Banks, D. (1996). *Interactive shape metamorphosis*. Unpublished manuscript.

Jensen, D. G. (1987). Facial perception studies using the Macintosh. *Behavior Research Methods, Instruments, & Computers*, **19**, 252-256.

Langlois, J. H., & Roggman, L. A. (1990). Attractive faces are only average. *Psychological Science*, **1**, 115-121.

Levin, D. T. (1996). Classifying faces by race: The structure of face categories. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, **22**, 1364-1382.

Mauro, R., & Kubovy, M. (1992). Caricature and face recognition. *Memory & Cognition*, **20**, 433-440.

Perret, D. I., May, K. A., & Yoshikawa, S. (1994). Facial shape and judgments of female attractiveness. *Nature*, **368**, 239-242.

Rhodes, G., Brennan, S., & Carey, S. (1987). Identification and ratings of caricatures: Implications for mental representations of faces. *Cognitive Psychology*, **19**, 473-497.

Rhodes, G., & Tremewan, T. (1996). Averageness, exaggeration, and facial attractiveness. *Psychological Science*, **7**, 105-110.

Smythe, D. B. (1990). *A two-pass mesh warping algorithm for object transformation and image interpolation* (ILM Technical Memo No. 1030). San Rafael, CA: Lucasfilm.

Wolberg, G. (1990). *Digital image warping*. Los Alamitos, CA: IEEE Computer Society Press.

(Manuscript received February 18, 1997;
revision accepted for publication April 2, 1998.)