Computer Graphics and Applications
Proceedings of the First Pacific Conference on
Computer Graphics and Applications — Pacific Graphics '93, Vol. 1
Seoul, Korea, Aug. 30–Sep. 2, 1993. ©World Scientific Publishing Co.

METAMORPHOSIS USING BÉZIER CLIPPING

TOMOYUKI NISHITA TOSHIHISA FUJII
Department of Electronic & Electrical Eng., Fukuyama University, Higashimura-cho, Fukuyama, 729-02 Japan

and

EIHACHIRO NAKAMAE
Department of Information Science, Hiroshima Prefectural University, Nanatuka-cho, Shoubara City, Hiroshima Prefecture, 727 Japan

Abstract

Recently, in the field of entertainment, the metamorphosis of 2-D images, as it is an attractive technique, has come to be used in movies and television commercials. This technique is referred to as *morphing*. In the proposed method, Bézier or B-spline nets are overlapped onto digital images, and their deformation is performed by moving the control points of the nets on the images. The method proposed here has the following advantages: 1st order derivative continuity is preserved after deformation, any kind of affine transformation is available, and the nets are automatically set by extracting the contours of images.

1 Introduction

Computer graphics have been being used in various fields, and have recently come into wide use in arts and entertainment such as animation. In entertainment, free-form deformation of images, i.e., metamorphosis or *morphing* (short for metamorphosis), is an image processing technique for digital or photographed images which has attracted a lot of attention, and is widely used in motion pictures and television commercials because of its effect and easy handling.

The morphing technique requires the following conditions: (a) smooth images (there are no discontinuous parts), (b) various types of transformation such as affine transformations, (c) easy specification of deformation, and (d) the ability to control the deforming speed of shapes and colors.

In the proposed method in order to deal with the free-form deformation of digital images, free-form deformation is performed by moving some mesh points which are overlaid onto digital images: the mesh points are considered as the control points of

bicubic Bézier or B-spline patches in order to realize continuous deformation. Therefore, we can get a continuous image with 1st derivative continuity after deformation.

For simple image deformation, affine transformations, such as transforming, scaling, rotation, shearing, are used in special effects(video effects). Previous methods of morphing cannot attain all affine transformations, even though they can handle free-form deformation. The method proposed here can handle all types of affine transformation. When we use rational Bézier patches, perspective transformation can be realized.

In animations, the effect of the colors of certain parts changing gradually and in some order may be desired: the proposed method can attain this.

The method proposed here is based on the FFD method[2] and a scanline algorithm for Bézier patches[3]. In a morphing operation, the expression of the relation between the two images, the source and destination images, is significant. Considering the parametric coordinates of the mesh overlapped on the images, two points whose parametric values are equivalent correspond to each other: the parametric coordinates system used in FFD is used here (a continuous mapping function expressed by Bézier functions). It is required to get one parametric value for each pixel: this technique is called *inverse mapping*. Inverse mapping is performed by a similar technique to that used in the scanline-based hidden surface removal for curved surfaces[3]. As an efficient solver of intersection points between Bézier patches and rays, one of the authors have developed the *Bézier Clipping Method*[1] for ray tracing. By expanding this method, inverse mapping can be achieved, and animations with smooth deformation can be realized.

2 Previous Work

There are two types of metamorphosis, distortion of a single image, and transformation from one image into another. Examples of the former examples include the moving of a flag and the deformation of clouds or fire. Those of the latter are the transformation of a tiger to a woman, and a man-to-mummy morph, as seen in recent movie pictures such as *Willow*, and *Terminator 2*. So, the former deformation means from A to A', and the latter one from A to B. As images to be deformed, computer-generated images or photographed images are usually used.

There are two major previous methods: The *mesh warping technique*[4] and *field morphing* (or feature-based approach)[5]. In the former, B-spline nets are overlaid onto the digital images, and the deformation is performed by moving the control points of the nets(see Fig. 1). The latter method is morphing based on fields of influence surrounding two-dimensional control primitives: A pair of lines(one defined relative to the source image, the other defined relative to the destination image) defines the mapping from one image into another (see Fig. 2).

In the former, as all four edges of the meshes are frozen, transformation around the boundary of the screen is limited: This means that not all affine transformations are possible. This technique uses the two-pass algorithm[6,7] which creates distortion when used for a large rotational angle. In the latter, the operation is easy because of

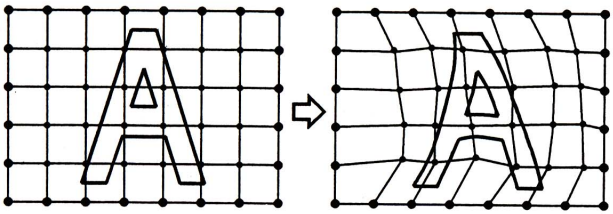


Figure 1: Mesh Warping

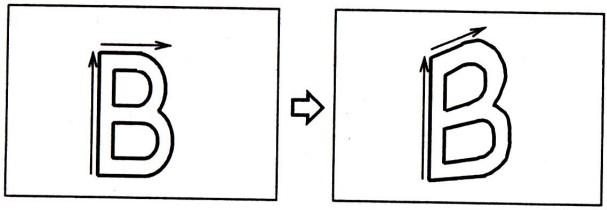


Figure 2: Feature–based image metamorphosis.

having to handle only lines, but sometimes unexpected interpolations are generated. And not all affine transformations are possible here either. In particular, it is not possible to specify uniform scales and shears, while the method proposed here can handle all affine transformations when a bilinear Bézier patch is used.

For deformation of 3-D geometric models(not for digital images), Sederberg developed the FFD(*Free Form Deformation*) Method[2]. In his method, Bernstein polynomials are employed. Transformation is achieved by moving the control points of the Bézier patches overlaid onto the objects to be deformed. This operation is applied not only to 3-D objects but also to 2-D images, and is limited to A to A' transformation : the method can obtain the coordinates after transformation (i.e., forward mapping), but can not obtain the coordinate on the original image corresponding to pixels on the screen (i.e., inverse mapping). The method proposed here can solve this problem, too.

3 The Basic Idea of the Proposed Method

The method proposed here can specify any mesh point position, can handle all affine transformations, and can realize deformed images with 1st-derivative continuity because the mesh points correspond to the Bézier nets. In the case of A to B transformation (e.g., adult face to child face), mesh points overlaid on images A and B are specified onto feature points(in general contours of objects) by using a mouse. To get a smooth transformation, the mesh is treated as a set of Bézier nets.

Basically, transformation is specified by using a mouse. So if images are compli-

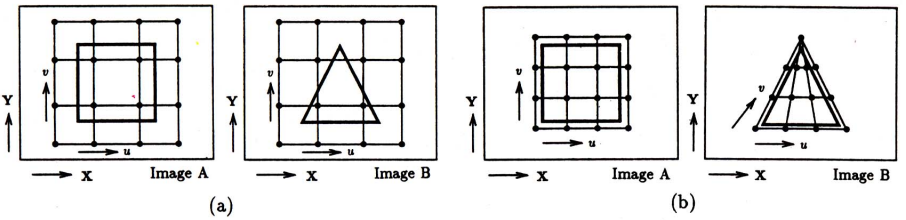


Figure 3: Manipulation of meshes to specify morphing.

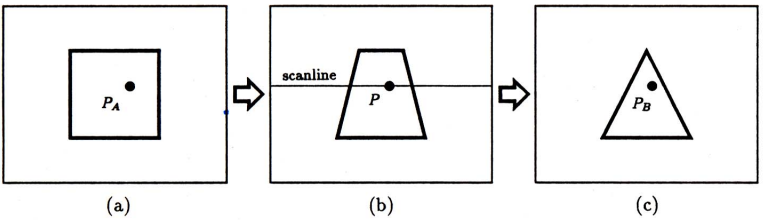


Figure 4: Calculation of color by corresponding colors of two images.

cated, the user may be forced into having to do annoying work. The major operation for the user may be tracing of the contour lines of the objects. Furthermore, the mesh warping has the following drawback: when control points placed on the contour of an image, some other control points must be positioned in the interior regions. In order to overcome this problem, we employ an image processing technique for contour extraction. For this process, the *active contour model*(called *snakes*)[8] and *active net*[9] have been developed. The latter is useful in our system because its output is meshes while that of the former is polyline. *Active net* is a simulated two dimensional elastic network model which minimizes the energy functional. Its energy functional consists of the internal strain energy of the net and the image energy which attracts the net to features in the image. The net deforms to wrap the region as the energy functional is minimized: the solution can be obtained by iteration. Note that the boundary of the initial net has to be set outside of the contour of the target image. In this method, every control point in the boundary mesh can be positioned in the interior regions of the contour of the image automatically.

By this process, the extent of the operation to specify transformation is reduced.

Let's explain the procedure of generating an image at a given step by using an example, the transformation of a rectangular to a triangle, as shown in Figs. 3 and 4, where the coordinates systems on the screen are (x, y), and the parametric coordinates of the meshes are (u, v)(see Eq.(1)). We call the meshes for image A and image B as a source mesh and a destination mesh, respectively.

1) Overlap a mesh onto each image(see Fig. 3(a)).

2) Move the mesh points of the source mesh onto the contour of image A, and move the mesh points of the destination mesh onto the corresponding contour of image B (Fig. 3(b))

3) Calculate the position of each mesh point at the given step by means of linear interpolation between the source mesh and the destination mesh, and obtain Bézier patches from those mesh points.

4) Scan the Bézier patches, and

(1) calculate (u, v) coordinates corresponding to point $P(x, y)$ on a scanline(Fig. 4(b)).

(2) calculate both (x, y) coordinates on image A and B from (u, v) (P_A and P_B in Fig. (a) and (c)), obtain the colors at those points, and then calculate the color at P by means of interpolation between those colors.

In step 3), a smooth connection between Bézier patches is not easy, so after specifying each of the mesh points as a control point of B-spline, the mesh is converted into Bézier patches[10]. Steps 3) and 4) are repeated. We discuss mainly step 4) in the following.

4 Metamorphosis using Meshes and Inverse Mapping

As mentioned above, the method proposed here is based on the FFD method, let's describe it briefly prior to discussing how to solve inverse mapping, which can not be solved through FFD.

4.1 The FFD Method

Figure 5 shows an example of 2-dimensional FFD. The mesh points correspond to the control points of the Bézier patches. Let's Consider $(n + 1) \times (m + 1)$ mesh, coordinates (x_{ij}, y_{ij}) of mesh point P_{ij} is given by

$$P_{ij} = \mathbf{O} + (i/n)\mathbf{U} + (j/m)\mathbf{V} \quad (i = 0, 1, .., n, j = 0, 1, 2, .., m), \tag{1}$$

where $\mathbf{O}$ is the origin of the mesh, $\mathbf{U}$ and $\mathbf{V}$ are basic vectors generally parallel to X and Y-axes, respectively(see Fig. 5(a)). Using parameters (u, v), point $P(x, y)$ within the mesh is expressed by

$$P = \mathbf{O} + u\mathbf{U} + v\mathbf{V} \quad (0 \leq u \leq 1, 0 \leq v \leq 1). \tag{2}$$

Let's denote the transformed mesh point as P'_{ij}, then transformed point P' is obtained by

$$P'(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} P'_{ij} B_i^n(u) B_j^m(v), \tag{3}$$

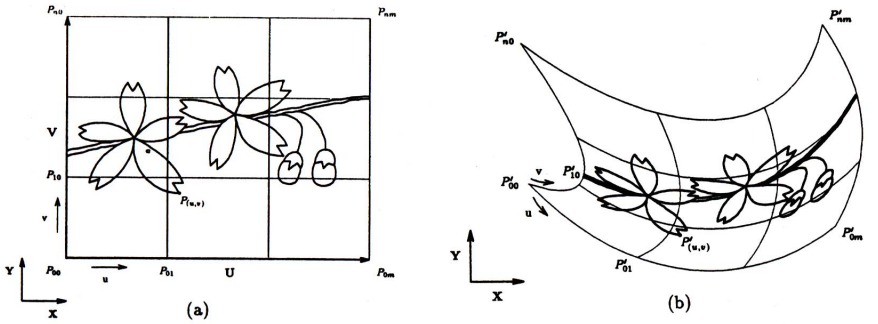


Figure 5: Free Form Deformation.

where $P'_{ij}(x_{ij}, y_{ij})$ are the coordinates of the control points of the Bézier patches (see Fig. 5(b)). B is the Bernstein polynomial, and is given by

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}. \tag{4}$$

In FFD, transformed coordinates (x', y') are obtained by substituting (u, v) into Eq. (3), after calculating (u, v) corresponding to $P(x, y)$ by using Eq. (2) (this process is simple because a liner equation is used). In this method shapes are approximated by polygons(or a set of piecewise segments) and displayed: This method is acceptable only for forward mapping of polygons, and is suited for use with all polygonal renderers.

4.2 Inverse Mapping

This paper deals with bicubic Bézier patches in order to realize a deformed image with 1st order derivative continuity (i.e., C^1 continuity): To get the kth order of continuity we can consider Bézier patches of degree $(k+2)$. When deformation is performed locally, this method maintains continuity around the boundary of the deformed region: C^1 continuity is guaranteed not only at its boundary but everywhere.

An inverse mapping technique is generally required for hidden-surface removal of curved surfaces and texture mapping. Therefore, for 3-D models the parameters corresponding to a point on the projection plane are obtained, (x, y, z) coordinates are calculated by using those parameters, and a visibility test is executed by using its coordinates. In some case, texture coordinates to be mapped are calculated by using the parameters. In the same manner, inverse mapping for 2-D images is divided into the following methods.

4.2.1 Categories of Inverse Mapping

1) Inverse mapping using approximated bi-linear patches

This mapping is similar to the idea of hidden-surface removal using polygonal approx-

imation; that is, the curved surfaces are subdivided into 4-sided polygons(or bi-linear patches), the boundaries of which are linear. The (u, v) coordinates are obtained by solving a quadratic equation[11]. However, the continuity at the boundary between the patches is C^1 continuity; i.e., the image obtained is discontinuous at its boundary. When the number of subdivisions increases, the degree of discontinuity decreases. In this case, however, memory requirement increases and computation time for the inclusion test, that is, in which patch calculation points are included, also increases.

2) Inverse mapping using ray tracing

Solutions to the ray/patch intersection problem can be categorized as being based on subdivision and numerical approaches. The representative method in the former is Whitted's method[12]. His method recursively subdivides a patch to get intersections, so a large number of subdivisions are required to solve this problem precisely. The representative method in the latter is Kajiya's method[13]. His method is based on an algebraic technique. His algorithm reduces the problem of intersecting a bicubic patch with a ray into one of finding the real root of an univariate polynomial of degree 18. Though many algorithms have been developed, an efficient algorithm was recently developed by one of the authors, the ray tracing method for rational Bézier patches, in which the root is obtained by a linear equation iterative method (this method is called *Bézier Clipping*[1]). Even though, in general, the ray tracing method gives precise solutions, it is computationally expensive.

3) Inverse mapping using the scanline algorithm

Lane[14] rendered curved surfaces by means of the subdivision of the surfaces into small polygons. That is, the curved surfaces are subdivided into subpatches until they are flat enough. In his method, the surfaces are dynamically subdivided for each scanline. His method has some disadvantages in that gaps arise between approximated polygons. To overcome this problem, Nishita's method[3] has been proposed. In this method, curved surfaces are subdivided into subpatches with curved boundaries. Parameters (u, v) are obtained by linear interpolation between the intersections of the boundaries of the subpatches and the scanline. The scanline algorithm is generally faster than the raytracing algorithm.

4.2.2 Proposed algorithm

In this paper, the scanline algorithm is used. Parameter values at each pixel on a scanline are obtained by means of the interpolation of sampled values.

Let's consider inverse mapping from (x, y) coordinates on a screen to a parametric space. Even though many Bézier patches are overlaid onto the images to be deformed, we consider a single bicubic Bézier patch here. (x, y) coordinates of point P are expressed by Eq.(3).

Let's denote y_s as y coordinate of the scanline, and then the equation of the scanline is expressed by $y - y_s = 0$. The intersection between the curve and the scanline is obtained by substituting y-component of Eq. (3) in this line equation,

$$\sum_{i=0}^{3}\sum_{j=0}^{3} y_{ij} B_i^3(u) B_j^3(v) - y_s = 0. \tag{5}$$

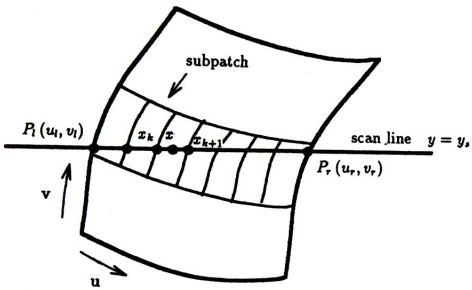


Figure 6: Inverse Mapping: intersection between scanline and Bézier curve.

Intervals (u, v) satisfying Eq.(5) can be obtained by Bézier clipping[3], and anything outside of the interval of the curved surface is clipped away. After extracting the subpatches on a scanline, the intersection points between the boundaries of the subpatches and the scanline are calculated(see Fig. 6). Where there is no intersection between the generated boundaries of the subpatch when clipped away, the intersection test is done only for the other boundaries (generally 2 of them).

Let's denote the coordinates of the intersection points as x_l and x_r, and their parameters as (u_l, v_l) and (u_r, v_r). Then the larger component,$(u_r - u_l)$ or $(v_r - v_l)$, is subdivided evenly. For example, if u-component is larger than v-component, iso-parametric curve $u = u_k$ $(=u_{k-1} + du)$ is calculate, and the intersections among these curves and the scanline are calculated, where du is determined by $(x_r - x_l)$ so that the sampling interval becomes to 2 or 3 pixels. As an iso-parametric curve is a bicubic Bézier curve, the value v at the intersection between the iso-parametric curve and the scanline can be calculated by

$$\sum_{j=0}^{3} d_j B_j^3(v) = 0, \tag{6}$$

where $d_j = y_j - y_s$, and y_j is y coordinate of the control point of the iso-parametric curve. The intersection points of a scanline and iso-parametric curves can be solved efficiently by using the Bézier clipping method. As the subpatches are smaller than the original patch, the iso-parametric curve is close to a straight line. Therefore, the solution is obtained with few iterations(e.g., 2.2 or 2.4 iterations for the examples shown in Fig. 8- 10). Using the value of v, x coordinate is calculated by using Eq.(3). Let's denote these intersections as v_k and x_k (see Fig. 6). Then the parameters (u, v) at x can be obtained by the following linear interpolation.

$$\begin{aligned} u &= u_k + (u_{k+1} - u_k)t, \\ v &= v_k + (v_{k+1} - v_k)t, \end{aligned} \tag{7}$$

where $t = (x - x_k)/(x_{k+1} - x_k), (0 \leq t \leq 1)$.

After calculation of (u, v) for every sampling point, the values of the parameters at each pixel between the sampling points are linearly interpolated. Subdivision is

accomplished by the well-known de Casteljau method. This method reduces the problem from degree 18 to degree 3.

5 Animation

To get a smooth animation, deformation is performed by using several steps between the source image and destination image. For the case of the distortion of an image(from A to A'), only the shape is transformed, while for the transformation from one into another(from A to B) colors should be also changed.

For shape transformation, the position of each mesh point at a midway image is obtained by linear interpolation between the meshes of the source image and those of the destination image. The color at each pixel is obtained by a blending of the corresponding pixel colors(their parametric values are the same). That is, the source image is gradually distorted and faded out, while the destination image is gradually distorted toward the first one and faded in.

Even though the lotus of the mesh points is linear, the speed of transformation is not uniform over the whole screen. We can control the speed: the transformation proceeded from some part of the screen. For example, it may be effective if distortion proceeded from the top to the bottom of the face. As the mesh has (u, v) parameters, the distortion direction can be followed by u-component(or v-component). In our system, the speed of transformation is specified by a Bézier function of (u, v).

In general, within an image, only some objects are transformed and the background does not change. These objects are extracted from the image by a masking process. After the morphing operation, the deformed image and the background image are composited[15]. So, in our system, three images, a source image, a destination image, and a background, are required.

6 Example

Figure 7 shows a basic example of the distortion of a image: (a) is the original image. (b) is the distorted image by using the Bézier nets of (c) and (d): (c) is overlaid on the original image, and the inner four points of (d) are moved. (e) and (f) are the partially distorted image by using the set of B-spline nets, (g) and (h), (i) and (j), respectively: (g) and (i) are overlaid on the original image (a), and the upper points of (h) and (j) are moved while the lower parts of the nets are fixed. By changing the some points of the nets, we can control the extent of continuity at the border between the distorted and frozen parts: even though the border in (f) looks discontinuous, 1st order derivative continuity is preserved after deformation. As shown in these figures, continuous deformation is realized.

Figure 8 shows an example of the distortion of a single image: a flag waving in the wind. (a) is the source image (the flag of Fukuyama University). Both (b) and (c) are composited images; a distorted image and a background image.

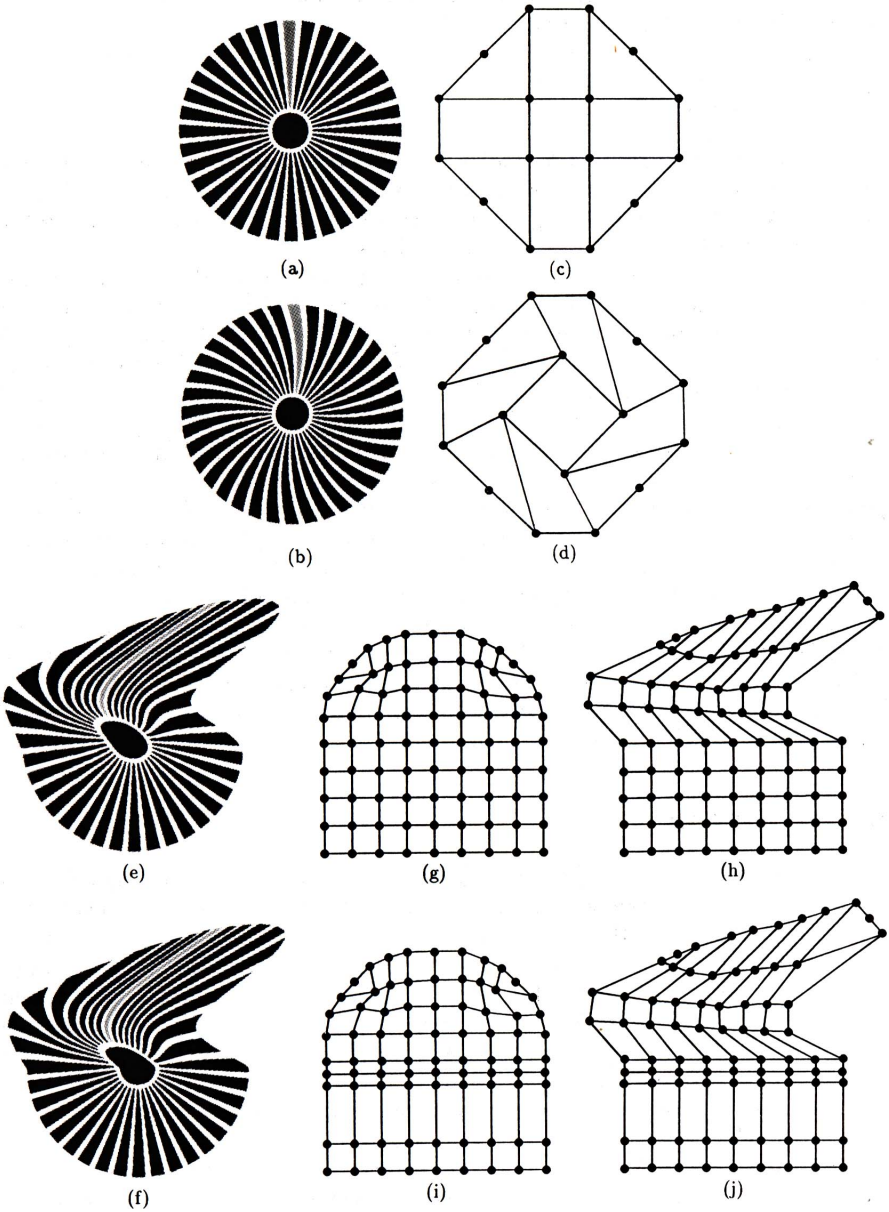


Figure 7: An example of distortion of a image.

Figure 9 is an example of morphing from one to another: a vacuum cleaner to an automobile. (a) and (d) are the source and the destination images, respectively: these images are generated by our scanline algorithm for Bézier patches[3].

Figure 10 shows the morphing of a horse to a tiger. The transformation is done gradually from head to tail.

These examples are cuts from animations. We used workstation SGI Indigo(R3000). The computation time for Fig. 8, Fig. 9 and Fig. 10 are 20., 18.0 and 22.0 seconds, respectively (the image size is 500 × 400).

7 Conclusion

We have proposed a morphing algorithm using Bézier clipping as a powerful animation tool.

The advantages of the proposed method are as follows:

(1) 1st order derivative continuity is preserved after deformation by using Bézier nets.

(2) Any kind of affine transformation is available.

(3) Inverse mapping is solved by using *Bézier Clipping*, and easy manipulation of mesh setting by the automatic contour extraction of images can be realized.

References

1 T. Nishita, T. Sederberg, M. Kakimoto, "Raytracing Trimmed Rational Surface Patches," *Computer Graphics*, Vol.**24**, No.4, (1990) pp.337-345.

2 T. Sederberg, S. Parry, "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol.**20**, No.4, (1986) pp.151-160.

3 T. Nishita, K. Kaneda, E. Nakamae, "A Scanline Algorithm for Displaying Trimmed Surfaces by using B'ezier Clipping," *The Visual Computer*, Vol.**7**, No.5, (1991) pp.269-279.

4 G. Wolberg, "Digital Image Warping," *IEEE Computer Press*(1990)

5 T. Beier, S. Neely, " Feature-Based Image Meta-morphosis, " *Computer Graphics*, Vol.**26**, No.2, (1992) pp.35-39.

6 E. Catmull, A. Smith, " 3-D Transformation of Images in Scanline Order," *Computer Graphics*, Vol.**14**, No.2, (1980) pp.279-285.

7 A. Smith, " Planar 2-Pass Texture Mapping and Warping," *Computer Graphics*, Vol.**21**, No.4, (1987) pp.263-272.

8 M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision* (1988) pp.321-331.

9 K. Sakaue, K. Yamamoto, "Active Net model and Its Application to region Extraction," *Journal of TV*, Vol.**45**, No.10, (1991) pp.1155-1163(in Japanese).

10 G. Farin, "Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, 2nd ed.," *Academic Press, Inc.* (1990)

11 A. Glassner, "An Introduction to Ray Tracing," *Academic Press*, (1989) p.59-64.

12 T. Whitted, "An Improved Illumination Model for Shaded Display," *CACM*, Vol.**23**, No.6, (1980) pp.96-102.

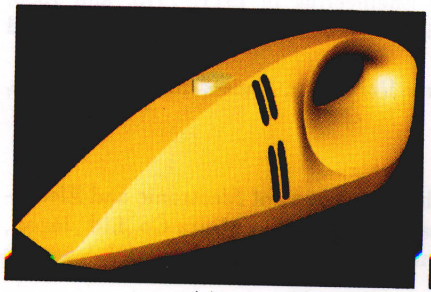
13 J. Kajiya, "Ray tracing Parametric Patches," *Computer Graphics*, Vol.**16**, No.3, (1982) pp.245-254.

14 J. Lane, L. Carpenter, T. Whitted, "Scan line Methods for Displaying Parametrically Defined Surfaces," *CACM*, Vol.**23**, No.1, (1980) pp.23-34.

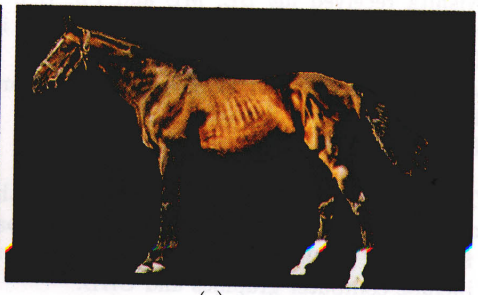
15 E. Nakamae, K. Harada, T. Ishizaki, T. Nishita, "Montage: The Overlaying of The Computer Generated Image onto a Background Photograph," *Computer Graphics*, Vol.**20**, No.3, (1986) pp.207-214.

(a) (b) (c)

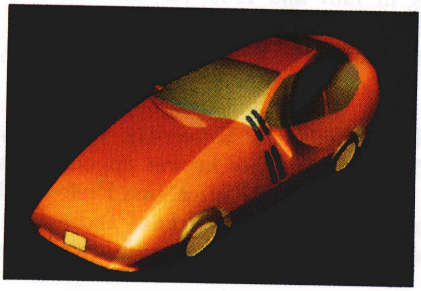
Figure 8: An example of distortion of a image: a flag waiving in a wind.



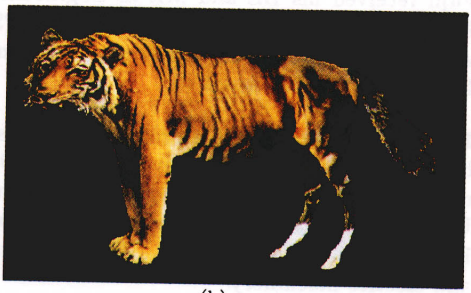
(a)



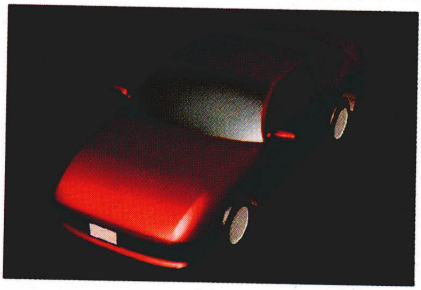
(a)



(b)

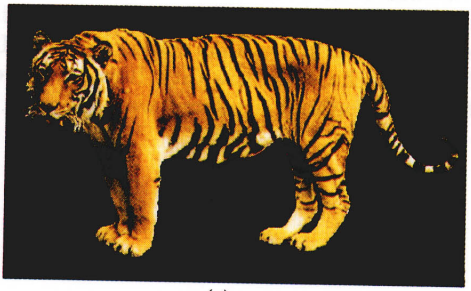


(b)



(c)

Figure 9: An example of morphing from a vacuum cleaner to a car.



(c)

Figure 10: An example of morphing from a horse to a tiger.

Author's Biographies

Tomoyuki Nishita is a professor in the department of Electronic and Electrical Engineering at Fukuyama University, Japan. He was on the research staff at Mazda from 1973 to 1979 and worked on design and development of computer-controlled vehicle system. He joined Fukuyama University in 1979. He was an associate researcher in the Engineering Computer Graphics Laboratory at Brigham Young University from 1988 to the end of March, 1989. His research interests involve computer graphics including lighting model, hidden-surface removal, and antialiasing.
Nishita received his BE, ME and Ph. D in Engineering in 1971, 1973, and 1985, respectively, from Hiroshima University. He is a member of ACM, IPS of Japan and IEE of Japan.
Address: Faculty of Engineering, Fukuyama University, Sanzo, Higashimura-cho, Fukuyama, 729-02 Japan.
E-mail: nis@eml.hiroshima-u.ac.jp

Toshihisa Fujii is an research associate in the department of Electronic and Electrical Engineering at Fukuyama University, Japan. He worked at Miura Co.,Ltd.,Japan, from 1989 to 1992. He joined Fukuyama University in 1992. His research interests include computer graphics and CAD.
Fujii received his BE and ME in Engineering in 1987 and 1989, respectively, from Fukuyama University. He is a member of IPS of Japan and IEE of Japan.
Address: Faculty of Engineering, Fukuyama University, Sanzo, Higashimura-cho, Fukuyama, 729-02 Japan.

Eihachiro Nakamae is a professor at Hiroshima Prefectural University. Previously he worked at Hiroshima University from 1956 to 1992, where he was appointed as research associate in 1956 and a professor in 1968. He joined Hiroshima Prefectural University in 1992. He was an associate researcher at Clarkson College of Technology, Potsdam, N. Y., from 1973 to 1974. His research interests include computer graphics and electric machinery.
Nakamae received the BE, ME, and DE degrees in 1954, 1956, and 1967 from Waseda University. He is a member of IEEE, IEE of Japan, IPS of Japan and IEICE of Japan.
Address: Faculty of Information Science, Hiroshima Prefectural University, Nanatuka-cho, Shoubara City, Hiroshima Prefecture, 727 Japan.
E-mail: naka@eml.hiroshima-u.ac.jp