Hierarchical Feature-Based Volume Morphing

Hao-Ren Ke
Library of
National Chiao-Tung
University, Hsinchu, Taiwan,
R.O.C
claven@cc.nctu.edu.tw

Yu-Pao Tsai
Department of Computer
and Information Science,
National Chiao-Tung
University, Hsinchu, Taiwan,
R.O.C.
gis86529@cis.nctu.edu.tw

Zen-Chun Shih
Department of Computer
and Information Science,
National Chiao-Tung
University, Hsinchu, Taiwan,
R.O.C.
zcshih@cis.nctu.edu.tw

Abstract

Morphing techniques have been broadly applied in many fields to create astounding visual effects. In this paper, a hierarchical feature-based volume morphing algorithm is presented. A new type of feature element, the cylinder field, is proposed to facilitate the task of feature specification and corresponding. In order to interpolate object shapes smoothly, Distance Field Interpolation (DFI) is employed in our algorithm. Furthermore, the idea of feature-element hierarchy is leveraged to improve the morphing effect significantly.

Keyword: feature-based volume morphing, distance field interpolation, cylinder fields, feature-element hierarchy

1 Introduction

Morphing is a group of techniques that create a series of transformation from one object to another object, and it is broadly used in many applications.

In the past, there have been many researches on image morphing [1][7][11][12]. Image morphing produces a series of intermediate images, which represents a gradual transformation from a source image to a destination image.

If the source and destination in image morphing are rendered from 3D models, image morphing may reveal a few disadvantages. First of all, if a feature of the original 3D model is not visible in the rendered 2D image, it will not appear during morphing, even though it ought to. Furthermore, inaccurate shadows and highlights may be generated during morphing. By contrast, 3D morphing directly produces a series of intermediate 3D models, and then morphing images are constructed by rendering these intermediate 3D models. Intermediate 3D models are created once, but can be rendered with various camera angles and lighting conditions.

3D models used in 3D morphing can be either geometric primitives (***geometric morphing***) [4][5][13] or volumetric data (***volume morphing***) [2][3][8]. With the rapid development of volume graphics [6], researches on volume morphing attract more and more attention in recent years.

One popular method for volume morphing is ***feature-based volume morphing***. In feature-based volume morphing, source *S* and destination *D* volumes are associated with pairs of feature elements, which specify the correspondence of key features between *S* and *D*. A sequence of in-between volumes V_1, V_2,..., V_l is then generated as follows. For the *i*th in-between volume V_i, warp *S* and *D* to create two distorted volumes S_i and D_i, respectively. The warping process is controlled and influenced by the feature elements associated with *S* and *D*. Then, V_i is constructed by smoothly interpolating S_i and D_i. Two popular approaches to interpolating S_i and D_i are ***Gray-Level Interpolation*** and ***Distance Field Interpolation*** (DFI) [3][9]. The method of DFI achieves better interpolation for object shapes by interpolating distance values to the nearest object shape rather than gray values.

In this paper, a method for feature-based volume morphing using a new kind of feature elements, ***cylinder fields***, and DFI is presented. We also enhance feature-based volume morphing by organizing feature elements into hierarchies. The remainder of this paper is organized as follows. We give an overview of our method in Section 2. The concepts of cylinder fields, together with a volume warping algorithm using cylinder fields, are described in Section 3. In section 4, we propose the idea of feature-element hierarchies to improve feature-based volume morphing. Section 5 shows our implementation and results. Finally, conclusions and future directions are described in Section 6.

2 Method Overview

This section presents an overview of our feature-based volume morphing algorithm, which integrates cylinder fields, DFI, and hierarchical feature elements.

Beier and Neely [1] proposed feature-based image metamorphosis, and Lerios et al. [8] extended their method to three-dimension. Our algorithm is based on the two

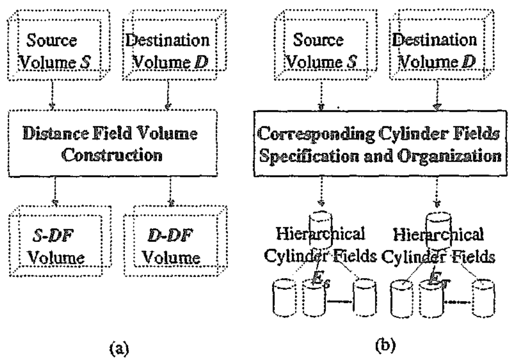


Figure 1 The preprocessing stage of our feature-based volume morphing algorithm. (a) Construct distance field volumes. (b) Specify and organize cylinder fields.

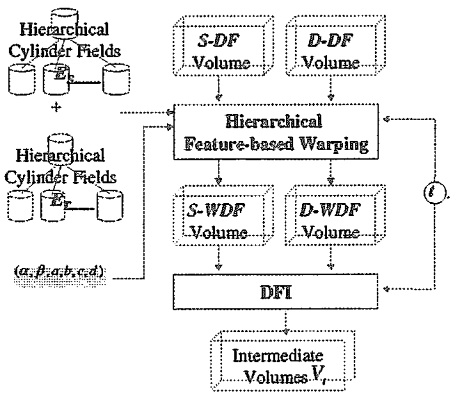


Figure 2 The morphing stage of our feature-based volume morphing algorithm. The α, β, *a*, *b*, *c*, and *d* (described in Section 3) are controlling parameters specified by users, and *t* is the interpolation parameter.

approaches with the following modification:

(1) Instead of using the feature elements proposed in the two approaches, we design a new kind of feature, the cylinder field.

(2) The two approaches employed gray-level interpolation; however, we use distance field interpolation.

(3) The two approaches operate feature elements independently, which may produce dissatisfactory morphing effects. We propose a feature-element hierarchy to enhance morphing effects.

The entire morphing process of our algorithm is divided into a *preprocessing* stage and a *morphing* stage. The flow charts of the two stages are shown in Figure 1 and Figure 2.

Given the source *S* and destination *D* volumes, there are two tasks in the preprocessing stage. The first task is to construct the distance field volumes *S-DF* and *D-DF* of *S* and *D*, respectively. We employ the method proposed by Cohen-or [3] to construct distance field volumes. The second task is to specify the corresponding features of *S* and *D* by cylinder fields, and then organize the cylinder fields into hierarchies E_S and E_T, respectively. We shall describe the ideas of cylinder fields and feature-element hierarchies separately in Section 3 and Section 4.

After the preprocessing stage, the morphing stage performs hierarchical feature-based warping (described in Section 3 and Section 4) and DFI [3] to produce intermediate volumes. To render the intermediate volumes, we can either extract the isosurfaces by marching cubes [10] or use direct volume rendering approaches [14].

3 Cylinder Field

In feature-based volume morphing, source *S* and destination *D* volumes are associated with pairs of feature elements, which specify the correspondence of key features between *S and D*. The warping process to generate intermediate volumes is also controlled and influenced by feature elements.

Many kinds of feature elements are proposed in the literature [1][2][3][8], including points, rectangles, box, and disks. These feature elements have their pro and con. If users are given too many kinds of feature elements, they will get confused when they attempt to choose a feature element to fit an object feature. The characteristics of feature elements are crucial in approximating object features and have strong impacts on the warping process. The issues of simplification, intuitiveness, and fitness should be considered while a new kind of feature element is designed. Instead of using these feature elements proposed in the literature, we develop a new kind of feature element in this paper–***cylinder fields***.

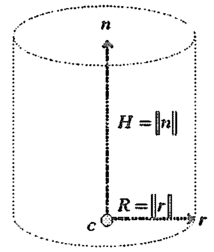


Figure 3 Spatial configuration of a cylinder

3.1 Spatial Configuration of a Cylinder Field

The spatial configuration of a cylinder field is shown in Figure 3. A cylinder field can be easily specified by its radial vector r, normal vector n, and the world coordinate position c, which represents the center of the base circle. The length of the radial vector, $\|r\|$, is the radius of the cylinder field, and the length of the normal vector, $\|n\|$, is the height of the cylinder field. Let the normalized radial vector ($\frac{r}{\|r\|}$) be x-axis of the local coordinate system of a cylinder field, and the normalized normal vector ($\frac{n}{\|n\|}$) be z-axis. Then, the cross product of z-axis and x-axis is y-axis. In other words, the local coordinate system of a cylinder field can be constructed and represented as $E(c,x,y,z)$, where $x=\frac{r}{\|r\|}$, $z=\frac{n}{\|n\|}$, and $y=z\otimes x$. In general, it can be expressed by a 4 by 4 matrix $M = \begin{matrix} x & y & z & c \\ 0 & 0 & 0 & 1 \end{matrix}$

Transformations of the cylinder field can be computed by manipulating *M*. Through transformations, cylinder fields can be used to approximate other kinds of feature elements. For example, a cylinder can simulate a line segment by shrinking its x-axis and y-axis. Therefore, cylinder fields offer a single choice with many variations. In addition, the task of feature specification is just to use cylinder fields to enclose object features. It is easy, intuitive, and clear.

3.2 Warping with a Single Pair of Cylinder Fields

Figure 4 illustrates the warping process with a single pair of cylinder fields. Given *S* and *D*, a pair of cylinder fields

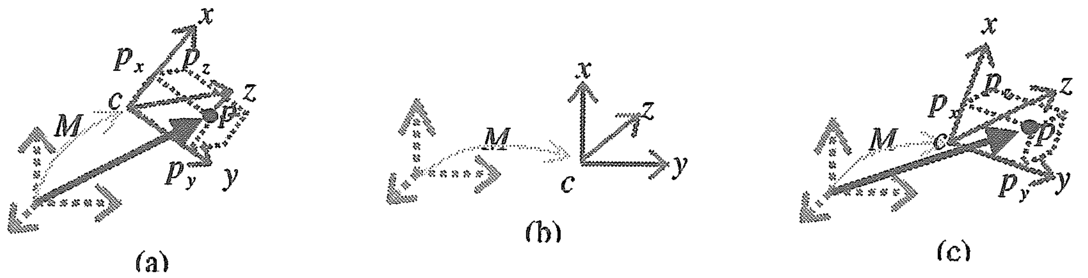


Figure 4 Warping with a single pair of cylinder fields. (a) $E(c,x,y,z)$ in the source volume. (b) $E(c,x,y,z)$ in the destination volume. (c) $E(c,x,y,z)$ in an intermediate

defines a mapping from one volume to the other. This pair of cylinder fields corresponds to features of S and D, and their coordinate systems are $E(c,x,y,z)$ and $E(c,x,y,z)$, respectively. Figure 4 (a) shows the coordinate system of the cylinder field in S, and Figure 4 (b) shows its counterpart in D. To warp S (or D), we first interpolate $E(c,x,y,z)$ and $E(c,x,y,z)$, according to a parameter t in the range of $[0..1]$, to obtain $E(c,x,y,z)$, as shown in Figure 4 (c). $E(c,x,y,z)$ represents a feature in an intermediate volume. Then we compute the value of each point in the intermediate volume form S (or D) by reverse mapping [1].

The reverse mapping comprises two steps:

(1) For a point p in the intermediate volume, transform it to $E(c,x,y,z)$ and obtain its local coordinate (p_x, p_y, p_z), as shown in (Eq. 1).

$$[p_x, p_y, p_z]^T = (M)^{-1} * p \quad \text{.........(1)}$$

(2) Apply (p_x, p_y, p_z) to the $E(c,x,y,z)$ (or $E(c,x,y,z)$) coordinate system and get a point p in S (or D), as shown in (Eq. 2)

$$p = M * [p_x, p_y, p_z]^T \quad \text{..................(2)}$$

$$(\text{or } \ p = M * [p_x, p_y, p_z]^T)$$

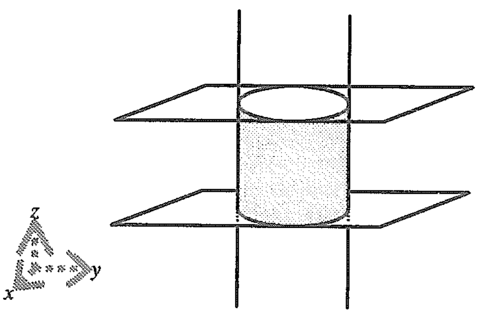


Figure 5 Spatial partition with respect to a cylinder field.

Finally, the value of p is equal to the value of p.

3.3 Warping with Multiple Pairs of Cylinder Fields

Given two sets of cylinder fields :

$$C_s = \{c_{s,1}, c_{s,2}, \ldots, c_{s,m}\} \quad \text{and}$$

$$C_t = \{C_{t,1}, C_{t,2}, \ldots, C_{t,m}\}$$

associated with two volumes S and T, each point p in T can be mapped onto a set of points $\{p_{s,1}, p_{s,2}, \ldots, p_{s,m}\}$ in S by the reverse mapping described in Section 3.2. A point p_s is then obtained as a weighted average of $\{p_{s,1}, p_{s,2}, \ldots, p_{s,m}\}$ and the value of p is equal to the value of p_s. We use (Eq. 3), which is a modification of the weighted function proposed by Beier and Neely [1], for calculating the weight of a cylinder field imposed on a

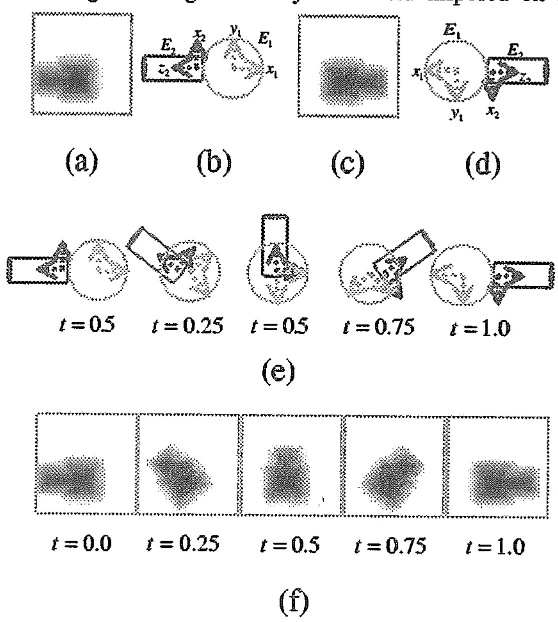


Figure 6 Intersection of cylinder fields during interpolation. (a) The source object. (b) The cylinder fields of the source object. (c) The destination object. (d) The cylinder fields of the destination object. (e) The interpolation process of the cylinder fields. (f) The warping result guided by (e).

$$Influence\ distance = \quad \text{(Eq. 4)}$$

$$\begin{cases} \left(\dfrac{\|ProjectXY(p)\|}{\|r\|}\right)^{\alpha} & ;p \text{ in area } 1 \\ \left(\|ProjectZ(p)\| - \|n\| + 1.0\right)^{\beta} & ;p \text{ in area } 2 \\ \left(\|ProjectXY(p)\| - \|r\| + 1.0\right)^{\beta} & ;p \text{ in area } 3 \\ \left(\|ProjectXY(p) * (\|ProjectXY(p)\| - \|r\|) + ProjectZ(p) - n\| + 1.0\right)^{\beta} & ;p \text{ in area } 4 \\ \left(\|ProjectZ(p)\| + 1.0\right)^{\beta} & ;p \text{ in area } 5 \\ \left(ProjectXY(p) * (\|ProjectXY(p)\| - \|r\|) + ProjectZ(p) + 1.0\right)^{\beta} & ;p \text{ in area } 6 \end{cases}$$

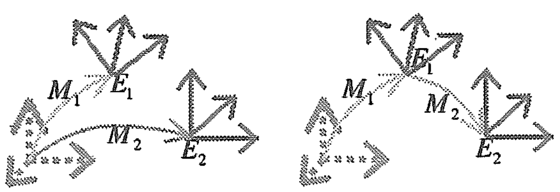
point *p*.

$$weight = \left(\frac{(n \text{ or } r)^{c}}{a + b * (influence\ distance)}\right)^{d} \quad ....(3)$$

According to the location of a point *p* with respect to a cylinder field (see Figure 5), the influence distance of *p* is defined in (Eq. 4). In (Eq. 4), *ProjectXY(p)* projects *p* onto the xy-plane, and *ProjectZ(p)* projects *p* onto the z-axis. n and r are defined in Figure 3.

4 Hierarchical Feature Elements

Traditionally, feature elements in feature-based volume morphing are operated independently. From our study, we find independent feature elements may produce dissatisfactory morphing. In this section, we discuss this problem and propose a method, ***hierarchical cylinder fields***, to conquer the problem. We demonstrate the problem by an example shown in Figure 6. Figure 6 illustrates the intersection of feature elements during interpolation, caused by independently interpolating multiple pairs of cylinder fields. The source and destination objects are shown in Figure 6 (a) and (c). The cylinder fields of the source, (E_1, E_2), and destination, (E_1, E_2), are shown in (b) and (d), respectively. Figure 6 (e) is the interpolation process from (E_1, E_2) to (E_1, E_2). It is obvious that we can rotate E_1 180°



(a) (b)

Figure 7 A comparison between (a) non-hierarchical cylinder fields and (b) hierarchical cylinder fields.

counterclockwise about z-axis to get E_1, and move E_2 to the right side of the first cylinder field and then rotate 180° clockwise to get E_2. Guided by the process shown in Figure 6 (e), a warping from the source to the destination is given in (f). From Figure 6 (e) and (f), We can see the object associated with E_2 is intruded into the object associated with E_1 during warping. But in fact, we can obtain the destination object simply by rotating the source object 180° counterclockwise about z-axis. To avoid the above problem and get better morphing results, we propose in this section the idea of ***feature-element hierarchy*** to organize cylinder fields.

We introduce the idea of hierarchical feature elements in the following. After adding corresponding feature elements in the source and destination volumes, we group these feature elements into a hierarchy. A hierarchy is represented by a tree structure, and the two trees of the source and destination volumes are isomorphic. Each node of a tree is associated with a feature element. A child node inherits the transformation effect of all its ancestors.

A comparison between non-hierarchical and hierarchical cylinder fields is shown in Figure 7. Figure 7 (a) shows two independent cylinder fields. They have independent transformation from their own local coordinate systems to the world coordinate system, M_1 and M_2, respectively. Figure 7 (b) shows the same cylinder fields but now they are organized into a hierarchical structure. E_1 is constructed as the father node of E_2. E_1 has a transformation from its local coordinate system to the world coordinate system, M_1. By contrast, the transformation of E_2, M_2, is just a transformation from its local coordinate system to the local coordinate system of E_1. If we want to obtain the transformation from E_2 to the world coordinate system, we must transform E_2 to

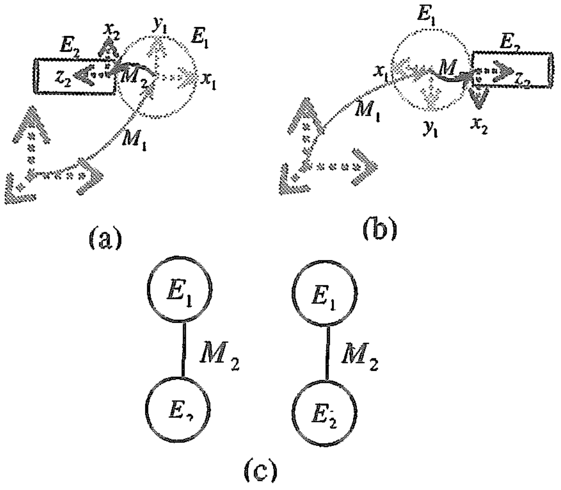


Figure 8 Apply a hierarchical structure to the cylinder fields in Figure 6. (a) The cylinder fields in the source volume. (b) The cylinder fields in the destination volume. (c) The hierarchical feature-element

the local coordinate system of E_1 first and then transform to the world coordinate system, i.e. $M_2 = M_1 * M_2$.

Now, we apply idea of feature-element hierarchy to Figure 6. The cylinder fields and their transformation of the source and destination objects are shown in Figure 8 (a) and (b), respectively. The tree representation of the feature-element hierarchy is shown in Figure 8 (c). Relative to the first cylinder fields, the transformation M_2 in Figure 8 (a) is identical to M_2 in Figure 8 (b). During interpolation, we interpolate the father node pair first, i.e. M_1 to M_1, and then interpolate the child node pair, i.e. M_2 to M_2. As Figure 6 (e), the first cylinder field is rotated 180° counterclockwise about z-axis. The second cylinder field does not perform any interpolation. It is just guided by the interpolation of E_1 and E_1 and circles around the first cylinder field. The interpolation process of the two pairs of cylinder fields with a hierarchical structure is shown in Figure 9 (a), and (b) is the warping result guided by (a). Comparing Figure 6 (f) and Figure 9 (b), the result in Figure 9 (b) is more nature.

5 Implementation and Results

To implement our ideas presented in this paper, we exploit a popular modeling tool, 3D Studio Max, to specify and organize hierarchical cylinder primitives, and use the VRML file format to store cylinder fields. Except the above tasks, all other procedures described in this paper are implemented by the C++ language with OpenGL library in Visual C++ environment, and run on Pentium II 450MHz PC. After intermediate volumes are produced, we use marching cubes [10] to reconstruct object shapes.

Figure 10 illustrates a morphing from F14 to X29. The

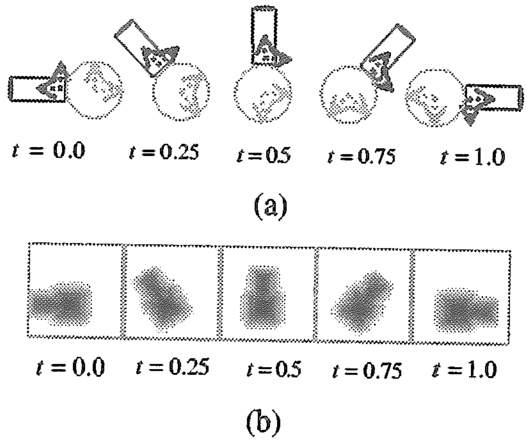


Figure 9 The warping process by means of hierarchical feature elements. (a) The interpolation process of the cylinder fields. (b) The warping result guided by (a).

resolution of the two volumes is 320x240x51. Figure 10 (a) and (c) show F14 and X29, and it takes 44.534 and 44.494 seconds (CPU time) to construct their distance field volumes, respectively. We use 13 pairs of cylinder fields to specify key features, which are shown in Figure 10 (b) and (d). Table 1 shows the total computing time when we construct 49 frames of morphing. Figure 10 (e) shows the transition of the cylinder fields, and (f) shows a few frames of the morphing result. Because the transitions of features are moderate, the morphing results with and without feature hierarchies are similar.

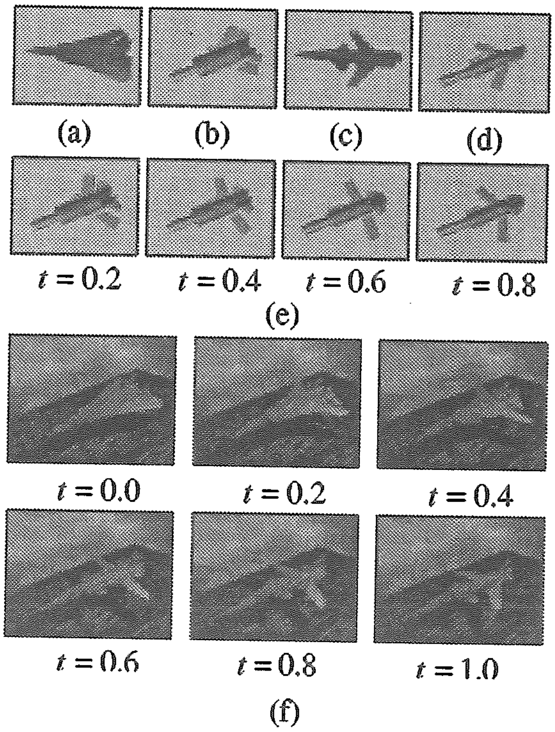


Figure 10 A morphing from F14 to X29. (a) F14 (b) The cylinder fields of F14. (c) X19 (d) The cylinder fields of X29. (e) The transitions of the cylinder fields. (f) The morphing result.

Hierarchical Feature-Based Warping		DFI	Marching Cubes
Source to Destination	Destination to Source		
11110.476 sec	11098.359 sec	103.199 sec	85.362 sec

Table 1 Total computing time of 49 frames for morphing from F14 to X29.

Hierarchical Feature-Based Warping		DFI	Marching Cubes
Source to Destination	Destination to Source		
2197.042 sec	2197.590 sec	103.199 sec	64.147 sec

Table 2 Total computing time to construct 49 morphing frames from a teapot to a few geometric objects.

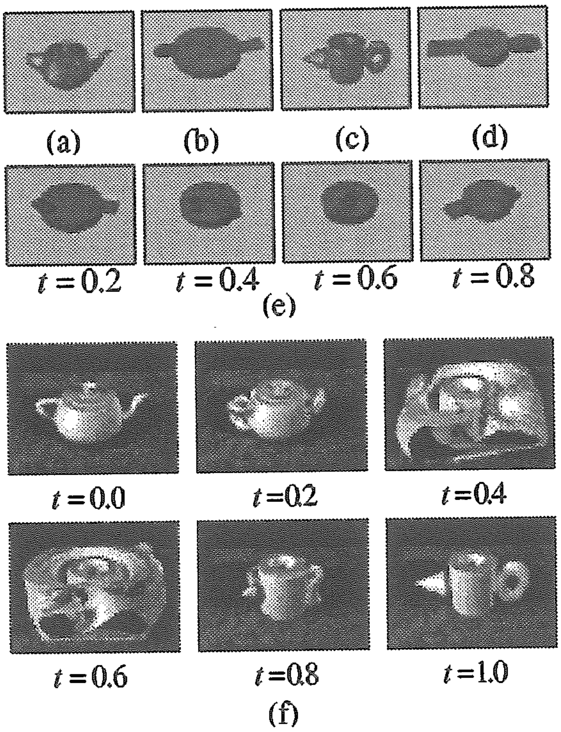


Figure 11 A morphing from a teapot to a few geometric objects without feature hierarchy. (a) The source object. (b) The cylinder fields of the source object. (c) The destination object. (d) The cylinder fields of the destination object. (e) The transitions of the cylinder fields. (f) The morphing result.

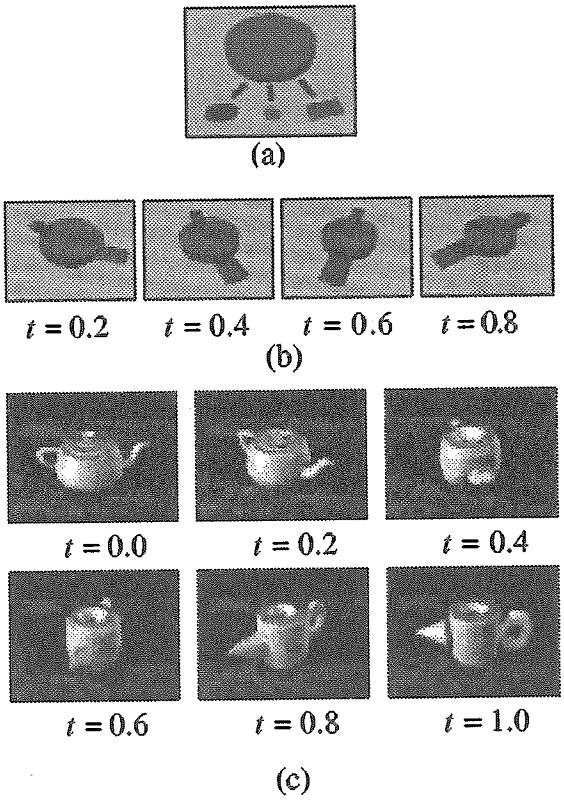


Figure 12 A morphing from a teapot to a few geometric objects with feature hierarchy. (a) The hierarchy of cylinder fields. (b) The transitions of the cylinder fields. (f) The morphing result.

Figure 11 and Figure 12 illustrate morphing results from a teapot to a few geometric objects. The resolution of the two volumes is 320x240x31. In Figure 11, we operate cylinder fields independently, but in Figure 12 we organize cylinder fields into a hierarchy. The source and destination objects are shown in Figure 11 (a) and (c), and it takes 27.51 and 27.529 seconds (CPU time) to construct their distance field volumes, respectively. In this case, we use 4 pairs of cylinder fields to specify key features, which are shown in Figure 11 (b) and (d), respectively. Figure 11 (e) shows the transitions of the cylinder fields without feature hierarchy. The handhold and beak of the source are corresponding to the cone and torus of the destination. In Figure 11 (e), we can see that the purple and blue features impenetrate the green feature and intersect each other. Figure 11 (f) shows the morphing result guided by (e).

Figure 12 uses the idea of feature hierarchy to guide morphing. Figure 12 (a) shows the feature hierarchy, and (b) shows the transition of the cylinder fields with hierarchy. We can see that the purple and the blue features just circle around the green feature. The morphing result is shown in Figure 12 (c), and it depicts a smooth transition from source to destination. The computing time to produce 49 morphing frames is given in Table 2.

Comparing Figure 11 with Figure 12, it is evident that the idea of feature hierarchy improves feature-based volume morphing significantly.

6 Conclusions

In this paper, we propose an algorithm that integrates the ideas of distance field interpolation (DFI), feature-based warping, and feature-element hierarchy to accomplish volume morphing. For feature-based warping, we propose a new kind of feature element, the ***cylinder field***. The cylinder field can enable users to intuitively define object features and thus achieve warping easily. In addition, the cylinder field can approximate other kinds of feature elements proposed in the literature. In this paper, we also show that grouping cylinder fields into hierarchies can solve the problem of unexpected intersection of features during feature-based warping, and thus can improve feature-based volume morphing significantly.

NSC89-2213-E-009-018

Reference

[1] T. Beier and S. Neely, "**Feature-based image metamorphosis**", *Proceedings of SIGGRAPH'92*, Vol. 26, No.2, ACM, New York, pp.35-42, 1992.

[2] M. Chen, M. W. Jones and P. Townsend, "**Volume Distortion and Morphing using Disk Fields**", *Computer and Graphics*, Vol. 20, No. 4, pp. 567-575, 1996.

[3] D. Cohen-or, D. Levin and A. Solomovici, "**Three-Dimensional Distance Field Metamorphosis**", *ACM Transactions on Graphics*, Vol. 17, No. 2, pp. 116-141, 1998.

[4] E. Galin and S. Akkouche, "**Blob Metamorphosis based on Minkowski Sums**", *Conputer Graphics Forum (Eurographics'96)*, vol. 15, pp. 143-153, 1996.

[5] T. Kanai, H. Suzuki and F. Kimura, "**3D Geometric Metamorphosis based on Harmonic Maps**", *Pacific Graphics*, pp. 97-104, 1997.

[6] A. Kaufman, D. Cohen-or and R. Yagel, "**Volume graphics**", *IEEE Computer*, Vol. 26, No. 7, pp. 51-64, 1993.

[7] S. Lee, G. Wolberg, K. Y. Chwa and S. Y. Shin, "**Image Metamorphosis with Scattered Feature Constraints**", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 2, No. 4, pp. 337-354, 1996.

[8] A. Lerios, C. D. Garfinkle and M. Levoy, "**Feature-based volume metamorphosis**", *Proceedings of SIGGRAPH'95*, ACM, New York, pp. 449-456, 1995.

[9] D. Levin, "**Multidimensional reconstruction by set-valued approximation**", *IMA J. Numer. Anal.*, Vol. 6, pp. 173-184, 1986.

[10] W. E. Lorensen and H. E. Cline, "**Marching cubes: A high resolution 3-D surface construction algorithm**", *Computer Graphics*, Vol. 21, No. 4, pp. 163-169, 1987.

[11] D. Ruprecht and H. Muller, "**Image warping with scattered data interpolation**", *IEEE Comput. Graph. Appl.*, (March), pp.37-43, 1995.

[12] G. Wolberg, "**Digital Image Warping**", IEEE Computer Society Press. 1990.

[13] W. Wyvill, "**Metamorphosis of Implicit Surfaces**", SIGGRAPH'90 Course 23 – Modeling and Animation with Implicit Surfaces, 1990.

[14] M. Levoy, "**Display of surfaces from volume data**", *IEEE Comput. Graph. Appl.*, Vol. 8, No. 3, pp. 29-37, 1988.