Hierarchical Feature-Based Volume Morphing

HAO-REN KE*, YU-PAO TSAI*+ AND ZEN-CHUN SHIH*

*Department of Computer and Information Science
National Chiao Tung University
Hsinchu, 300 Taiwan

+Institute of Information Science
Academia Sinica
Taipei, 115 Taiwan

Morphing techniques have been applied to many fields to create astounding visual effects. In this paper, we present a hierarchical feature-based volume morphing algorithm, which leverages the idea of feature-element hierarchy to significantly improve morphing effects. A new type of feature element, the cylinder field, is also given to facilitate the task of feature specification and correspondence. Additionally, with a slight modification, the proposed algorithm can be applied to volume animation.

Keywords: feature-based volume morphing, feature-element hierarchy, cylinder fields, distance field interpolation (DFI), volume animation

1. INTRODUCTION

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

There has been much research on image morphing [1-4]. Image morphing produces a series of intermediate images, which represent a gradual transformation from a source image to a destination image.

If the source and destination images in image morphing are rendered from 3D models, image morphing may produce some errors. First, if a feature of the original 3D model is not visible in the source or destination images, 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 sequence of intermediate 3D models, and then 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) [5-7] or volumetric data (volume morphing) [8-10]. With the rapid development of volume graphics [11], research on volume morphing has attracted more and more attention in recent years.

One popular method for volume morphing is feature-based volume morphing, in which 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 $N$ intermediate volumes $V_1, V_2, \ldots, V_N$ is then generated as follows (we assume $V_0 = S$ and $V_{N+1} = D$). For the $i$th intermediate volume $V_i$, warp $S$ and $D$ to create two warped volumes $S_i$ and $D_i$ respectively. The warping process is controlled by the feature elements associated with $S$ and $D$. Then, $V_i$ is constructed by interpolating $S_i$ and $D_i$. Two popular interpolation approaches are Gray-Level Interpolation and Distance Field Interpolation (DFI) [9, 12]. DFI achieves better effects for object shapes by interpolating distance values rather than gray values.

The primary goal of this paper is to reduce artifacts in feature-based volume morphing by organizing feature elements into hierarchies. A new kind of feature element, the cylinder field, is also proposed to simplify the process of feature specification and correspondence. 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 hierarchy to improve feature-based volume morphing. With a small modification, our hierarchical feature-based volume morphing algorithm can be leveraged to achieve volume animation, which is detailed in Section 5. Section 6 shows our experimental results. Finally, conclusions and future directions are described in Section 7.

2. METHOD OVERVIEW

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

Beier and Neely [1] proposed feature-based image metamorphosis, and Lerios et al. [10] extended their method to three dimensions. Our algorithm is based on these two approaches with the following modification:

(1) Instead of using the feature elements as proposed by the two approaches, we design a new kind of feature, the cylinder field.
(2) The two approaches employ gray-level interpolation; however, we use distance field interpolation.
(3) The two approaches operate on feature elements independently, which may produce unsatisfactory morphing effects. We propose a feature-element hierarchy for organizing feature elements, so that morphing effects can be enhanced.

The entire morphing process of our algorithm is divided into a preprocessing stage and a morphing stage. Flow charts of the two stages are illustrated in Fig. 1 and Fig. 2, respectively.

There are two tasks in the preprocessing stage. Given source volume $S$ and destination volume $D$, 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 et al. [9] to construct $S$-DF and $D$-DF. The second task is to specify the corresponding features of $S$ and $D$ by pairs of cylinder fields, and then organize the cylinder fields into hierarchies $E_S$ and $E_D$, respectively. We shall describe cylinder fields and feature-element hierarchies separately in Section 3 and Section 4.
After the preprocessing stage is completed, the morphing stage performs hierarchical feature-based warping (described in Section 3 and Section 4) and DFI [9, 12] to produce intermediate volumes. To render the intermediate volumes, we can either extract isosurfaces by marching cubes [13] or use direct volume rendering approaches [14].
3. CYLINDER FIELDS

In feature-based volume morphing, source $S$ and destination $D$ volumes are associated with pairs of feature elements to specify the correspondence of key features. Then during morphing, $S$ and $D$ are warped under the control of the feature elements.

Many kinds of feature elements have been proposed in the literature [1, 2, 9, 10], including points, rectangles, boxes, and disks. These feature elements have their pros and cons. If users are given too many kinds of feature elements, they will get confused when attempting to select a feature element to fit an object feature. The characteristics of feature elements are crucial in approximating object features and strongly affect the warping process. The issues of simplification, intuitiveness, and fitness should be considered when a new kind of feature element is designed. Instead of using the feature elements proposed in the previous works, we develop a new kind of feature element in this paper—the **cylinder field**.

3.1 Spatial Configuration of a Cylinder Field

The spatial configuration of a cylinder field is depicted in Fig. 3. A cylinder field can be specified by its radius vector, $\vec{r}$, normal vector, $\vec{n}$, and the world coordinate position, $c$, which represents the center of the base circle. The length of the radius vector, $\|\vec{r}\|$, is the radius of the cylinder field, and the length of the normal vector, $\|\vec{n}\|$, is the height of the cylinder field.

![Fig. 3. Spatial configuration of a cylinder field.](image)

Let the normalized radius vector ($\frac{\vec{r}}{\|\vec{r}\|}$) be $x$-axis of the local coordinate system of a cylinder field, and the normalized normal vector ($\frac{\vec{n}}{\|\vec{n}\|}$) be $z$-axis. Then, the cross product of the $z$-axis and the $x$-axis is the $y$-axis. In other words, the local coordinate system of a cylinder field can be constructed and represented as $E(c, \hat{x}, \hat{y}, \hat{z})$, where $\hat{x} = \frac{\vec{r}}{\|\vec{r}\|}$, $\hat{z} = \frac{\vec{n}}{\|\vec{n}\|}$, and $\hat{y} = \hat{z} \times \hat{x}$. In general, $E(c, \hat{x}, \hat{y}, \hat{z})$ can be expressed by a 4 by 4
matrix \( M = \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \). 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 \( x \)- and \( y \)-axes. Therefore, cylinder fields offer a single choice with many variations. For users, the task of specifying features is accomplished by using cylinder fields to enclose object features. This is easy, intuitive, and clear.

### 3.2 Warping With a Single Pair of Cylinder Fields

As mentioned in Section 1, the first step to produce the \( i \)th intermediate volume \( V_i \) is to warp \( S \) and \( D \) to create their warped volumes. The purpose of warping lies in transforming (by interpolation) the corresponding features in \( S \) and \( D \) to one another during the morphing. The transformation is performed with respect to their positions, orientations, and sizes. Therefore, for each intermediate volume, the warping has to generate warped volumes of \( S \) and \( D \). The features of the intermediate volumes should possess intermediate positions, orientations, and sizes of the features of \( S \) and \( D \) for the aforementioned transformations [1, 10].

Fig. 4 illustrates the warping process when only a single pair of cylinder fields is associated with \( S \) and \( D \). Fig. 4 (a) shows the coordinate system of the cylinder field in \( S \), \( E(c, \bar{x}, \bar{y}, \bar{z}) \), and Fig. 4 (b) shows its counterpart in \( D \), \( E(c', \bar{x'}, \bar{y'}, \bar{z'}) \). To warp \( S \) (or \( D \)) for creating its warped volume, we first interpolate between \( E(c, \bar{x}, \bar{y}, \bar{z}) \) and \( E(c', \bar{x'}, \bar{y'}, \bar{z'}) \) to obtain \( E(c^*, \bar{x}^*, \bar{y}^*, \bar{z}^*) \), as indicated in Fig. 4 (c). \( E(c^*, \bar{x}^*, \bar{y}^*, \bar{z}^*) \) represents the coordinate system of the single feature in the warped volume of \( S \) (or \( D \)). Then we compute the value of each point in the warped volume from \( S \) (or \( D \)) by the reverse mapping given in [1].

![Fig. 4](image-url)
The reverse mapping comprises two steps:

1. For a point \( p' \) in the warped volume, transform it to \( E(c', \bar{x}', \bar{y}', \bar{z}') \) and obtain its local coordinate \((\bar{p}'_x, \bar{p}'_y, \bar{p}'_z)\), as shown in Eq. (1).

\[
[p'_x, p'_y, p'_z] = (M')^{-1} \ast p'
\]

2. Apply \((\bar{p}'_x, \bar{p}'_y, \bar{p}'_z)\) to \( E(c, \bar{x}, \bar{y}, \bar{z}) \) (or \( E(c', \bar{x}', \bar{y}', \bar{z}') \)) and get a point \( p \) in \( S \) (or \( D \)), as shown in Eq. (2).

\[
p = M \ast [p'_x, p'_y, p'_z]^T \quad \text{(or \( p = M' \ast [p'_x, p'_y, p'_z]^T \))}
\]

Finally, we assign the value of \( p' \) as the value of \( p \).

### 3.3 Warping With Multiple Pairs of Cylinder Fields

The previous section describes the process of warping a single pair of cylinder fields associated with \( S \) and \( D \). In real cases a volume often contains many feature elements, and all of them affect each point of the volume. In this section, we show how to compromise the influence imposed by multiple cylinder fields on a single point during warping.

Given two sets of cylinder fields \( C_i = \{C_{i,1}, C_{i,2}, \ldots, C_{i,m}\} \) and \( C_i = \{C_{i,1}, C_{i,2}, \ldots, C_{i,m}\} \) associated with \( S \) 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 \( p_i \) is then obtained as a weighted average of \( \{p_{s,1}, p_{s,2}, \ldots, p_{s,m}\} \), and the value of \( p \) is assigned the value of \( p_i \) [1, 2, 10]. 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 \( p \):

\[
\text{weight} = \left( \frac{\bar{n} \ast \bar{r}}{a + b \ast \text{distance}} \right)^d
\]

where “\( \ast \)” denotes the inner-product operation; we use \( \bar{n} \ast \bar{r} \) to approximate the size of a cylinder field. According to the location of \( p \) with respect to a cylinder field (see Fig. 5), distance in Eq. (3) is defined in Eq. (4). In Eq. (4), \( \text{ProjectXY}(p) \) projects \( p \) onto the xy-plane, and \( \text{ProjectZ}(p) \) projects \( p \) onto the z-axis.

---

![Fig. 5. Spatial partition with respect to a cylinder field.](image-url)
The four constants in Eq. (3), \(a\), \(b\), \(c\), and \(d\), are assigned by users to control the effect of a cylinder field, and are in the ranges of \((0, \infty)\), \([0, \infty]\), \([0, 1]\) and \((0, \infty)\), respectively. \(a\) is used to keep the denominator larger than zero. \(b\) tunes the importance of distance, \(c\) adjusts the importance of the size of a cylinder field, and \(d\) controls the relative influence that each cylinder field imposes upon a point.

As we mentioned previously, cylinder fields can simulate other kinds of features, such as disks, points, lines, rectangles, and boxes. For example, the shape of a cylinder field will approximate a rectangle feature when we shrink its \(x\)-axis or \(y\)-axis. Furthermore, we can give large \(\alpha\) to points within a cylinder field to simulate the influence of a rectangle feature. Hence, both the shape and influence of a cylinder field can be manipulated to be similar to a rectangle feature element as presented in [10].

\[\text{Distance} = \begin{cases} \left(\frac{\|\text{ProjectXY}(p)\|}{\|p\|}\right)^a & ; \text{p in area 1} \\ (\|\text{ProjectZ}(p)\| - ||\bar{e}|| + 1.0)^b & ; \text{p in area 2} \\ (\|\text{ProjectXY}(p)\| - ||\bar{e}|| + 1.0)^b & ; \text{p in area 3} \\ (\|\text{ProjectXY}(p)\|^2 + ||\bar{e}|| + 1.0)^b & ; \text{p in area 4} \\ (\|\text{ProjectZ}(p)\|^2 + 1.0)^b & ; \text{p in area 5} \\ (\|\text{ProjectXY}(p)\|^2 + ||\bar{e}|| + 1.0)^b & ; \text{p in area 6} \end{cases}\]

4. HIERARCHICAL FEATURE ELEMENTS

Traditionally, feature elements in feature-based volume morphing are used independently. From our study, we find that independent feature elements may produce unsatisfactory morphing. In this section, we discuss this problem and propose a method, **hierarchical feature elements**, to overcome it. We demonstrate the problem by the example shown in Fig. 6 which illustrates the intersection of feature elements during warping caused by independently interpolating multiple pairs of cylinder fields. The source and destination objects are shown in Fig. 6 (a) and (c). The cylinder fields \((E_1, E_2)\) and \((E'_1, E'_2)\) of the source and destination are shown in (b) and (d), respectively. Fig. 6 (e) is the interpolation process from \((E_1, E_2)\) to \((E'_1, E'_2)\). Guided by the process shown in Fig. 6 (e), a morphing from the source to the destination is shown in (f). From Fig. 6 (e) and (f), we can see the feature associated with \(E_2\) intrudes into the feature associated with \(E_1\) during morphing.
To avoid the above problem and get better morphing effects, we propose a new concept, using **feature-element hierarchy** to organize key features. After adding corresponding feature elements in the source and destination volumes, we group the 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 transformations of all its ancestors.

A comparison of non-hierarchical and hierarchical cylinder fields is shown in Fig. 7 (a) shows two independent cylinder fields. They have independent transformations, $M_1$ and $M_2$, from their own local coordinate systems to the world coordinate system. Fig. 7 (b) shows the same cylinder fields, but now they are organized into a hierarchical structure. $E_1$ is constructed as the parent node of $E_2$. $E_1$ has a transformation, $M_1$, from its local coordinate system to the world coordinate system. By contrast, the transformation of $E_2$, $M_2'$, is just a transformation from its local coordinate system to the local coordinate system.
nate 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 the local coordinate system of $E_1$ first, and then transform to the world coordinate system, i.e. $M_2 = M_1 * M'_2$.

![Diagram](image)

**Fig. 7.** A comparison between (a) non-hierarchical cylinder fields and (b) hierarchical cylinder fields. $M_1$ and $M_2$ represent the transformations from local to world coordinate systems. $M'_2$ represents the transformations from the local coordinate system of $E_2$ to the local coordinate system of $E_1$.

![Diagram](image)

**Fig. 8.** Apply a hierarchical structure to the cylinder fields in Fig. 6. (a) The cylinder fields in the source object. (b) The cylinder fields in the destination object. (c) The feature-element hierarchy.

Now, we apply the concept of feature-element hierarchy to Fig. 6. The cylinder fields are shown in Fig. 8 (a) and (b). The tree representation of the feature-element hierarchy is shown in Fig. 8. (c). Relative to the first pair of cylinder fields ($E_1$ and $E'_1$), the transformations ($M_2$ and $M'_2$) of the second pair of cylinder fields ($E_2$ and $E'_2$) are equal. During morphing, we interpolate the parent 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 shown in Fig. 6 (e), the first cyl-
nder field is rotated 180° counterclockwise about the z-axis. The second cylinder field does not perform any interpolation. It is just guided by the interpolation between \( E_i \) and \( E'_i \), and encircles around the first cylinder field. The interpolation process of the two pairs of cylinder fields using a hierarchical structure is shown in Fig. 9 (a), while (b) is the result of using (a) to guide morphing. Comparing Fig. 6 (f) and Fig. 9 (b), the result in Fig. 9 (b) is more natural.

We must point out that the idea of feature-element hierarchy can be applied to feature-based volume morphing using any kind of feature element, not only cylinder fields.

![Diagram](image)

Fig. 9. Warping by means of hierarchical feature elements. (a) The interpolation process of the cylinder fields for producing \( V_i \). (b) The result of using (a) to guide warping.

### 5. VOLUME ANIMATION

Volume animation deforms an object contained in a volume according to an animation path and produces a sequence of intermediate results. Gagvani et al. [15] used skeleton trees of distance field volumes to achieve volume animation. They employ spheres to approximate and reconstruct object shapes; however, the reconstructed object shapes look like ellipsoidal sets and the details of object shapes are lost.

In principle, warping and animation are similar in that they both distort objects and produce a series of intermediate results; therefore, we may leverage some ideas of warping to accomplish animation. In this section, we propose a volume animation algorithm based on a variation of our hierarchical feature-based warping algorithm.

Our volume animation algorithm comprises a **preprocessing** stage and an **animation** stage. The flow chart of the preprocessing stage is shown in Fig. 10, and that of the animation stage is shown in Fig. 12. In addition to constructing the distance field volume and feature hierarchy of the object to be animated, the preprocessing stage also defines
the keyframes of an animation path. We use hierarchical feature elements to facilitate the animation of an object. We adjust the hierarchical cylinder field $E_i$ to obtain the keyframes of an animation path. $\hat{E}_i$ denotes the transformed hierarchical cylinder fields of $E_i$ at the $i$th keyframe, i.e., $t = t_i$. We show three cylinder fields $E_1$, $E_2$, and $E_3$ in Fig. 11 (a), and the corresponding hierarchy in (b). We also depict an animation path in Fig. 11 (c).

After keyframes are defined, the animation stage can proceed according to the flow chart presented in Fig. 12. The variable $i$ denotes the index of the current keyframe, and $t$ is the time counter. The whole animation consists of several loops of warping. In each loop, we deal with a portion of an animation path defined by two keyframes of $t = t_{i-1}$ and $t = t_i$. $\hat{E}_{i-1}$ and $\hat{E}_{i}$. In the first loop, we input $S-DF$, $\hat{E}_s = \hat{E}_{s,0}$ and $\hat{E}_{s,1}$ to our feature-based warping algorithm and produce a series of intermediate volumes $V_t$. In the second loop, we input $V_{t_1}$ (the intermediate volume produced at $t = t_1$), $\hat{E}_{s,1}$ and $\hat{E}_{s,2}$ to the warping process. The following loops proceed in the same manner. Finally, the result of the entire volume animation is the collection of all intermediate volumes.
6. IMPLEMENTATION AND RESULTS

To implement the ideas proposed 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. We also use 3D Studio Max to define keyframes of volume animation. Except for the above tasks, all other procedures described in this paper are implemented in 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 [13] to reconstruct object shapes.

Fig. 13 illustrates a morphing from an F14 jetfighter to an X29. The resolution of the two volumes is 320x240x51. Fig. 13 (a) and (c) show the F14 and X29. It takes the same CPU time, 44 seconds, to construct their distance field volumes. To specify key features, we use 13 pairs of cylinder fields, which are shown in Fig. 13 (b) and (d). Fig. 13 (e) shows the transition of the cylinder fields, and (f) shows a few frames of the
Fig. 12. The animation stage of volume animation. Volume $V_t$ is the frame produced at time $t$.

morphing result. Table 1 shows the computing time required to construct 49 frames of morphing. Because the transitions of features are moderate, the results of morphing with and without feature hierarchies are similar.

Table 1. Computing time to construct 49 morphing frames from F14 to X29.

<table>
<thead>
<tr>
<th>Hierarchical Feature-Based Warping</th>
<th>DFI</th>
<th>Marching Cubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source to Destination</td>
<td>Destination to Source</td>
<td>103 sec</td>
</tr>
<tr>
<td>11,110 sec</td>
<td>11,098 sec</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 13. Morphing from an F14 into an X29. (a) F14, (b) The cylinder fields of F14. (c) X29, (d) The cylinder fields of X29. (e) The transitions of the cylinder fields for producing $V_i$ (the total number of intermediate volumes is four). (f) The morphing result.

Fig. 14 and Fig. 15 illustrate morphing results from a teapot into a few geometric objects. The resolution of the two volumes is 320x240x31. In Fig. 14, we operate on cylinder fields independently, while in Fig. 15, we organize cylinder fields into a hierarchy. The source and destination objects are shown in Fig. 14 (a) and (c), and it takes the same CPU time, 28 seconds, to construct their distance field volumes. In this case, we
use four pairs of cylinder fields to specify key features, which are shown in Fig. 14 (b) and (d), respectively. The handle and spout of the source corresponds to the torus and cone of the destination. Fig. 14 (e) shows the transitions of the cylinder fields without feature hierarchy, from which we can see that the purple and blue features penetrate the green feature and also intersect each other. Fig. 14 (f) shows the morphing result, as guided by (e).

Fig. 15 uses the idea of feature hierarchy to guide morphing. Fig. 15 (a) shows the feature hierarchy, and (b) shows the transition of the cylinder fields using feature hierarchy. We can see that the purple and the blue features just encircle around the green feature. The morphing result is shown in Fig. 15 (c), and it depicts a smooth transition from source to destination. The time to compute 49 morphing frames is given in Table 2.

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

<table>
<thead>
<tr>
<th></th>
<th>Hierarchical Feature-Based Warping</th>
<th>DFI</th>
<th>Marching Cubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source to Destination</td>
<td>2,197 sec</td>
<td>103 sec</td>
<td>64 sec</td>
</tr>
<tr>
<td>Destination to Source</td>
<td>2,197 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We also create an example of morphing from a candlestick into a sword. The volume size is 320x240x41, and it takes 36 and 38 seconds of CPU time to construct their distance field volumes. The morphing results without and with feature hierarchy are illustrated in Fig. 16 and Fig. 17. From Fig. 16, we see that during morphing, candles are detached from the candlestick. On the other hand, Fig. 17 produces a better morphing. In this case, we use 15 cylinder fields to specify object features; the time to compute 49 morphing frames is given in Table 3.

Table 3. Computing time to construct 49 morphing frames from a candlestick to a sword.

<table>
<thead>
<tr>
<th></th>
<th>Hierarchical Feature-Based Warping</th>
<th>DFI</th>
<th>Marching Cubes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source to Destination</td>
<td>10,596 sec</td>
<td>82 sec</td>
<td>51 sec</td>
</tr>
<tr>
<td>Destination to Source</td>
<td>10,499 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It can be seen from our experimental results that the use of feature hierarchy improves feature-based volume morphing remarkably. There are two factors that make our algorithm outperform the traditional one: (1) the objects to be morphed are composed of hierarchical parts (which is generally true in real cases), and (2) the orientations of corresponding features in source and destination volumes differ significantly.

Fig. 18 demonstrates an example of volume animation for a razor. Fig. 18 (a) shows the keyframes of an animation path, and (b) shows the animation result. The keyframes 0, 1, 2 and 3 of Fig. 18 (a) correspond to frames 0, 4, 6 and 8 in Fig. 18 (b).
Fig. 14. Morphing from a teapot into 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 for producing $V_i$ (the total number of intermediate volumes is four). (f) The morphing result.
Fig. 15. Morphing from a teapot into a few geometric objects using feature hierarchy. (a) The hierarchy of cylinder fields. (b) The transitions of the cylinder fields for producing $V_i$ (the total number of intermediate volumes is four). (c) The morphing result.
7. CONCLUSIONS

In this paper, we propose an algorithm that integrates the concepts of distance field interpolation (DFI), feature-based warping, and feature-element hierarchy to accomplish volume morphing. The primary contribution of this paper is the idea of grouping feature elements into hierarchies, which solves the problem of unexpected intersection of
features during warping, and, thus, can improve feature-based volume morphing significantly. We emphasize that the idea of hierarchical feature elements can be applied to any kind of feature. For feature-based warping, we propose a new kind of feature element, the cylinder field. The cylinder field enables users to intuitively define object features and, thus, achieve warping easily. Furthermore, the cylinder field can approximate other kinds of feature elements proposed in the literature. After a few modifications are made, our algorithm can also be applied to volume animation.

Although users can employ cylinder fields to easily specify object features, it would be better if we can exploit computers to automatically suggest features for users during the process of feature identification and correspondence. In the future, we will develop a computer-aided method to identify and map object features so that the performance of volume morphing and volume animation can be enhanced.

The feature trees of the source and destination volumes in our current implementation are required to be isomorphic. We will conduct further study to overcome this constraint.
The hierarchical structure of feature elements proposed in this paper is useful for volume animation. Currently, we only make a slight modification to our hierarchical feature-based volume morphing algorithm to achieve volume animation. Though it is far from mature, we are continuing our efforts to improve the quality of volume animation.
ACKNOWLEDGEMENT

This research is granted by National Science Council, Taiwan, R.O.C., under Project No. NSC-89-2213-E-009-018.

REFERENCES

Hao-Ren Ke (柯皓仁) was born on 29 June 1967 in Taipei, Taiwan, Republic of China. He received his B. S. degree in 1989 and his Ph. D degree in 1993, both in Computer and Information Science from National Chiao-Tung University. Now he is an associate professor and the Head of Digital Library Division in Library, National Chiao-Tung University. His main interests are in virtual reality, digital library, information retrieval, and knowledge discovery.

Yu-Pao Tsai (蔡玉寶) received his B.S. degree in Computer Science from National Cheng-Chi University in 1997 and his M.S. degree in Computer and Information Science from National Chiao-Tung University in 1999. Currently, he is a research assistant of the Institute of Information Science at Academia Sinica in Taipei, Taiwan, ROC. His research interests include virtual reality and image-based rendering.

Zen-Chung Shih (施仁忠) was born on 10th February 1959, in Taipei, Taiwan, ROC. He received his B.S. degree in Computer Science from Chung-Yuan Christian University in 1980, M.S. degree in 1982 and Ph.D. degree in 1985 in Computer Science from the National Tsing-Hua University. Currently, he is a professor in the Department of Computer and Information Science at the National Chiao-Tung University in Hsinchu.