THE JOURNAL OF VISUALIZATON AND COMPUTER ANIMATION, VOL. **8**: 131–146 (1997)

Metamorphosis of Cylinder-like Objects

FRANCIS LAZARUS AND ANNE VERROUST

INRIA-Rocquencourt, Domaine de Voluceau, B.P. 105, 78153 Le Chesnay, France
(e-mail: Francis.Lazarus@inria.fr, Anne.Verroust@inria.fr)

SUMMARY

A new technique is presented for computing continuous shape transformations between polyhedral objects. The correspondence and the interpolation problems are considered jointly as we construct and process on sampling polyhedral meshes of the objects. The approach is adapted to objects that are star-shaped around an axis. The process gives the animator a high level control over the shape transformation by providing natural specification and interaction.

KEY WORDS: computer animation; computer-aided geometric design; interpolation; deformation; shape transformation

INTRODUCTION

We are interested in automating the computation of animated deforming shapes. The way the deformation is specified is important: it determines both the flexibility and the usability of the deformation tool.

Many approaches propose a modelization of a time-dependent deformation process to apply on an initial shape.[1–5]

Another way is to use the metamorphosis metaphor. The shape transformation is then primarily specified by two shapes and the animation consists of the deformation of the initial shape into the final shape.

Our goal is to provide a general and intuitive tool that compute a metamorphosis between two objects. Such a tool is useful in both animation and design.

1. In animation, metamorphosis is well adapted to traditional techniques such as key-framing: the objects are given at each key-frame and the in-between shapes are computed by the shape transformation. Moreover, the visual effects of metamorphosis between two images or between two shapes are very popular. This effect is called 'morphing'[6, 7] and is often used in movies.
2. In design, metamorphosis is used to create new shapes by combining objects.[8–10]

Our concern is concentrated in the 3-D shape transformation problem; thus, we will not describe the 2-D approaches (the main approaches can be found in References 6, 8, 11 and 12.

The 3-D methods can be classified into two families according to the type of information used to compute the intermediate shapes.

CCC 1049–8907/97/030131–16 $17·50

Received 30 April 1995
Revised 7 June 1996

The volumetric approaches[7, 9]

Since the volumetric information is used to compute the shape transformation, there is no restriction on the topological correspondence between the initial and the final shapes. For example, a torus can be transformed into a sphere. Previous work using the volumetric approach includes the following:

1. Kaul and Rossignac[9] define the intermediate polyhedral objects using linear interpolation based on Minkowski sums of the two original polyhedra. The faces of the deformed shapes have a constant orientation and their vertices move on a straight line between a vertex of the initial shape and a vertex of the final shape. Moreover, when the two original shapes are convex, these faces form the boundary of the deformed polyhedron.

   A related work[10] uses the Bézier formulation with the Minkowski sum to compute each intermediate shape as a point in the space of polyhedra. This extension adds flexibility to the technique and is very intuitive for convex solids.
2. Hughes[7] proposes a method for sampled volumetric models. He defines an interpolation between the Fourier transforms of the two models, separating the processing of high frequencies and low frequencies. The most interesting point in this approach is the underlying idea of dissociating the general shape and the details of the shape during the transformation.

The boundary approaches[13–16]

The original solids are assumed to be polyhedral objects with equivalent surface topologies. For these methods, the shape transformation process is decomposed into two parts:

1. A correspondence process: where the correspondence between the two surfaces is established. Since the surfaces are polyhedral, the process can be performed by computing the correspondence between the topological structures of the two objects. All the boundary approaches build a vertex/edge/face network containing the two topological structures.
2. An interpolation process: once the correspondence is established, the intermediate polyhedra are computed. The computation is performed by either linearly interpolating or using a spline interpolating curve between the corresponding vertices (Hermite cubic path with end tangents set equal to the vertex normal in Reference 15).

All the boundary approaches consider the two processes separately and propose solutions to the correspondence problem. Previous work using the boundary approach includes the following:

1. Bethel and Uselton[13] approach the correspondence process as a graph correspondence problem. The geometry of the two polyhedra is not taken into account. For instance, the metric relationships between points are ignored. Thus two points that are close to one another on the first object may be associated with two points on the second object that are far apart. This can lead to a distorted transformation.
2. Kent, Parent and Carlson[14, 15] use the geometric information of the initial and the final shapes to build a common adjacency graph. Their approach is very intuitive and convincing for star-shaped objects: in this case, they use a spherical projection to solve the correspondence and the interpolation processes. More precisely, the correspondence process is decomposed into two steps:

(a) a projection of each adjacency graph is done onto the surface of the unit sphere,
(b) the common adjacency graph is computed by merging these projected graphs.

In Reference 15 they extend their projection method to other classes of polyhedra. In most of these cases, the geometry of the object and the mapping method are dissociated.

3. Parent[16] uses a recursive subdivision process to build the common adjacency graph. The user indicates explicitly the correspondence he wants to ensure between areas of the objects (areas composed of connected sets of faces). The common adjacency graph is then obtained by recursively subdividing these areas.

It is our opinion that the correspondence and the interpolation processes must be jointly solved in order to provide an intuitive control over the deformation of the shape during the interpolation. The shape transformation problem should be taken as a whole.

Our method can be classified as a boundary approach but it differs from the previous models at several points:

1. The correspondence and the interpolation calculations are closely related.
2. We do not build a common adjacency graph from the initial and the final adjacency graphs. A common parametrized polyhedral mesh is associated with the initial and final objects during a sampling process.
3. The user controls the 'global' evolution of the deformation by specifying two corresponding skeletal structures (here 3D curves or axes) from which the parametrizations of the sampling meshes are built.

In this paper we will focus our attention on objects that are *star-shaped around an axis*. For these objects there exists a 3D curve inside the object, namely its axis. This axis is such that each normal cross-section of the axis intersects the object in a star-shaped polygon and the ends of the object are star shaped with respect to the endpoints of the axis. These objects can naturally be split into three sheets:

(a) two hemispherical parts, corresponding to the endpoints of the axis
(b) one cylindrical part, corresponding to the points that are attached orthogonally to the axis.

Each of these parts admits a natural parametrization, respectively spherical and cylindrical. In this paper, we use a general cylindrical coordinate system, where the axis of the cylindrical parametrization is defined by the axis of the object, a 3D curve.

THE BASIC SCHEME

In order to perform the transformation of one object into another, we construct a new polyhedral mesh (a discretization) for each object. Actually, this discretization is one of the main features of our method. We use the fact that a three-dimensional object can be defined by any polygonal mesh that captures its shape and its features. We construct a mesh using the parametrization. This parametrization is used both during the correspondence and the interpolation processes. This is how we jointly solve the correspondence and the interpolation problems.

To define the discretization, the user must specify two 3D axes, one for each object, such that the objects are star-shaped around them. The curves are discretized and a 3D frame is associated with each discretization point. As in References 17 and 18, we have chosen an orthogonal frame along each 3D curve which minimizes rotation.[19, 20] Each axis is used to define an appropriate parametrization of the associated object, such as described earlier.

The interpolation of the parametrization defines the shape transformation. Our algorithm is composed of two main steps:

1. Using the axes, a sampling of each object is computed using the *natural* parametrization. The sampling process is followed by an adjustment of the sampling vertices/edges/faces to the *singular features* of each object. For polyhedral objects these singular features are sharp edges or vertices. This adjustment is very important in order to obtain a good approximation of the shapes. The correspondence is deduced from the two samplings. Two sampling points are associated whenever they have the same parameters.
   This process will be detailed in the next section.
2. Next, we simultaneously interpolate between the two cylindrical axes and for each sampling point the local radius. This will be described in the following section.

At this stage it is useful to mention that the correspondence problem is automatically solved by taking the same *discretization* in the parameter space of two corresponding parametrizations of the objects.

SAMPLING AND ADJUSTMENT

Sampling

Given an object and its axis, we sample the object by computing the intersections of its boundary with a set of half-lines attached to the axis. These half-lines are built from the axis in one of two ways.

1. For the spherical parametrization, the half-lines will radiate from one of the two axis endpoints, in the half-space determined by the normal cross-section of the axis at the endpoint. These half-lines will intersect the object at two hemispherical sheets. Given one extremity, each half-line is defined by two angular parameters, as shown in Figure 1.
2. For the cylindrical parametrization, the half-lines will belong to one normal cross-section of the axis. The collection of intersection points will create the cylindrical sheet of the object. In this case, each half-line is defined by one angular parameter and an abscissa along the axis (see Figure 1).

We will call these half-lines *sampling half-lines* in the following.

In both cases, the parameters are varied in a rectangular domain. The sampling is obtained by mapping rectangular lattices in the parameter spaces onto the object. Each of the vertices in the lattices corresponds to a sampling parameter pair. They are mapped to sampling points on the object. By connecting the sampling points according to the rectangular grids associated with the rectangular lattices, we define a mesh subdivided in three sheets. The mesh is composed of rectangular faces except for the poles of the two hemispheres where triangular faces are joined at the respective pole (see Figure 1). Note that for the hemisphere, one edge of the associated square domain $[0, 2\pi (\times [0, \pi/2]$ is mapped on a single point.

The sampling of an object is performed as follows.

1. Each vertex of the original object is examined and localized with respect to the parametrization induced by the axis. In Reference 17 this was done by finding the closest point on the axis. Here, we find an extremal point of the mapping which computes the distance to the vertex for each point of the axis. When several axis points are candidates, we take into account the adjacency, making a breadth-first search[21] on the adjacency graph of the vertices of the object.

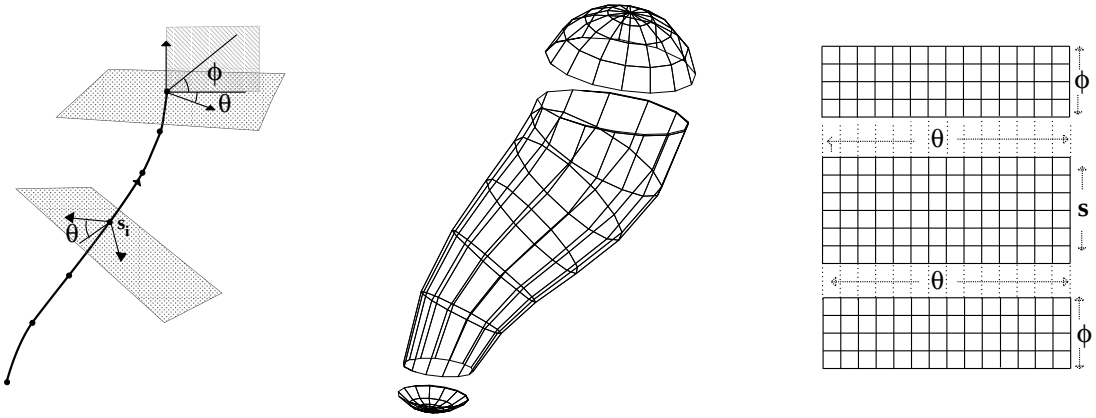

Figure 1. The three sheets

2. For each face $\mathcal{F}$ of the object, we compute the intersections with the sampling half-lines and $\mathcal{F}$. More precisely:

   (a) If necessary the face $\mathcal{F}$ is cut into three parts with respect to the three sheets (one or two may be empty),
   (b) Inside each sheet, $\mathcal{F}$ is projected in the parameter space, using the parameters of its vertices previously evaluated. Then, the sampling parameter pairs appearing in the interior of the projected face are located and the corresponding sampling half-lines are intersected with $\mathcal{F}$. For the cylindrical part, this is performed by a sweep along the cylindrical axis. We cut $\mathcal{F}$ with every intermediate cross-section and find the sampling parameter pairs for each intersecting segment (see Figure 2). The length of the sweep is determined by the the extremal parameters of the vertices of $\mathcal{F}$. For the hemispherical parts, the process is similar but the intermediate cross sections are the planes determined by the angular parameters θ (see Figure 1).

Adjustment

If we apply this algorithm to a cube with a straight axis as in Figure 3, the resulting object will look very different from the original one: the sharp edges of the cube do not appear on the sampling model. Notice that the salient features (i.e. sharp edges and vertices) determine the appearance of an object, since they mark the discontinuities of the light reflectance function.

Hence, if we want to obtain a good reproduction of the shapes, we have to take into account the sharp edges and vertices of the original model during the sampling process.

In order to minimize the topological change in the sampling mesh, we adjust the mesh by using local deformations, moving vertices and adding some diagonal edges to the rectangular faces. We proceed by:

(a) finding the sharp edges and vertices of the original object, and,

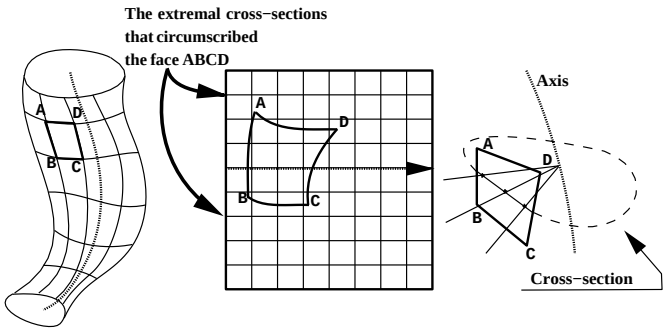

Figure 2. Image of a face in the parameter space

(b) then locally deforming the sampling mesh. For each of these features, we locate the vertices to be moved, compute their displacement and, if necessary, insert diagonal edges in the sampling mesh.

Salient features

A simple angle criterion between the adjacent faces to an edge allows us to select the sharp edges. A similar criterion can be applied to select the sharp vertices. A sharp vertex is selected by looking at the angle of the smallest cone containing the adjacent face normals. In either cases, the angle is defined by the user. The user may also interactively edit the list of sharp edges and the list of conic points.

Deforming the sampling mesh

The sampling mesh is deformed by fitting the vertices and the edges of the sampling mesh to the sharp edges and vertices. This is achieved by adding diagonal edges in the sampling mesh in such a way that, at the end:

(a) each sharp edge is described by a set of disjoint edges of the sampling mesh, and,
(b) the sharp edges belong to the vertices of the sampling mesh.

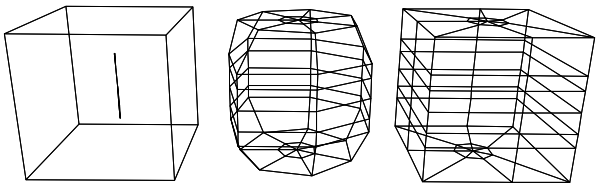

Figure 3. The sampling of a cube (centre) and the same sampling after adjustment (right)

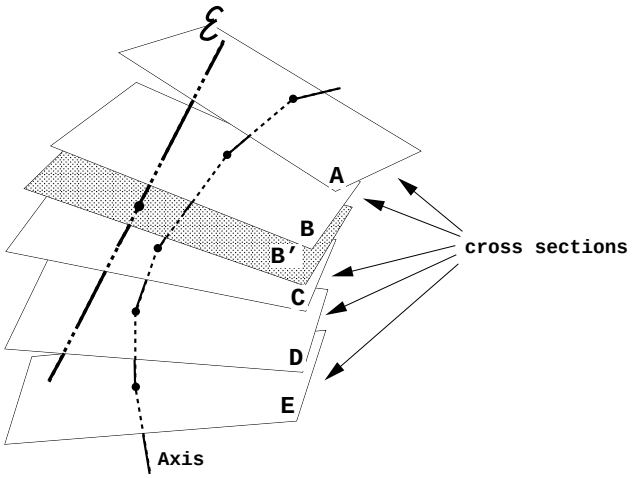

Figure 4. $\mathcal{E}$ is cut by the intermediate cross-section B′

The displacements of the vertices of the sampling mesh are local and bounded with respect to the parametrization. That is, a covering of the parameter space in 'deformation zones' associated with the vertices of the sampling mesh is defined. These zones are rectangular neighbourhoods associated with a parallel, intermediate mesh constructed using the midpoints of the sampling mesh. Figures 4 and 5 depict the construction process. In Figure 5, the deformation zones correspond to the rectangles filled in grey surrounding the selected vertices.

This parallel mesh is used to adjust the vertices as follows.

1. For each sharp edge $\mathcal{E}$, a sweep is performed (similar to the sweep previously used to sample a face). $\mathcal{E}$ is cut with every intermediate cross section (see Figure 4) and the parameters of the intersection points are used to determine the vertices to be moved. Note that the number of selected vertices is not fixed.

Figure 5. The selected vertices are marked in grey and the intersection points in black. The black curve is the projection of $\mathcal{E}$ in the parameter space. On the left, the vertices have been selected and, on the right, the sampling mesh has been deformed to fit $\mathcal{E}$

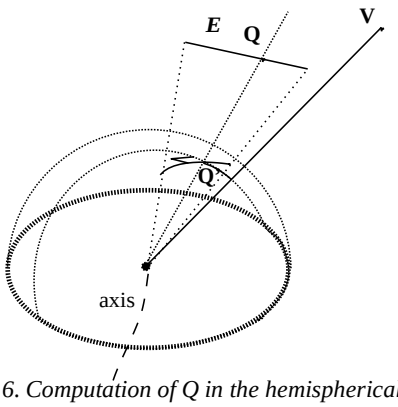

Figure 6. Computation of Q in the hemispherical case

In Figure 5, two vertices have been selected on section D (between the intersections of $\mathcal{E}$ with sections C′ and D′) whereas only one has been selected on section B (between the intersections of $\mathcal{E}$ with sections A′ and B′).

2. When the set of vertices to be moved has been selected, each vertex *V* is displaced at a point *P* belonging to $\mathcal{E}$ as follows:

   (a) for the cylindrical sheet, we compute the nearest point *Q* belonging to $\mathcal{E}$ of the sampling half-line corresponding to *V*. If the parameters of *Q* belong to the deformation zone associated to *V* then *P*=*Q*. In the other case, *P* is determined by the intersections of $\mathcal{E}$ and the boundaries of the deformation zone associated to *V*.
   (b) in the hemispherical cases, we use spherical projection. We compute *Q* belonging to *E* such that its projection *Q*′ is the nearest point from the projection of *V* on a sphere centred at the extremity (see Figure 6). We take *P*=*Q* when the parameters of *Q* are inside the deformation zone associated with *V*. If it is not the case, we proceed as in the cylindrical case, using the intersections of $\mathcal{E}$ and the boundaries of the deformation zone associated with *V*.
3. The cases of the conic points and of the endpoints of the sharp edges are treated in a similar fashion. For each of them, we force the nearest sampling vertex to coincide with it.

Once the vertices of the sampling mesh have been moved on the sharp edges, the deformed mesh may not totally cover $\mathcal{E}$. This is the case when the sequence of selected vertices does not form a sub-path of the sampling mesh. In this case, a diagonal edge must be added each time there is a 'hole' in the sequence of vertices displaced on $\mathcal{E}$. For example, in Figure 5, four diagonal edges have been added, one for each jump between two cross-sections, i.e., between A and B, B and C, C and D and between D and E.

If the fineness of the sampling is not sufficient, two sharp edges may deform the same vertex of the sampling mesh. As taking into account the two edges to compute the final position of the vertex makes no sense, we move the vertex even though it has been displaced before. Since the positions of the conic points are of prime importance, we choose to examine first the sharp

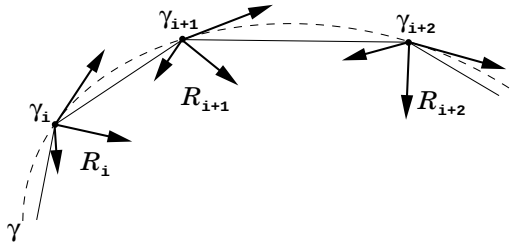

Figure 7. A discretized axis with its frames

edges (with respect to their order of appearance in the list of sharp edges) and then the conic points.

The order of the edges in the list of sharp edges determines the relative importance of the salient features. The ordering can be changed via an interactive tool.

Anyway, the co-occurrence of several constraints over one vertex of the sampling mesh can be a criterion to refine the mesh.

INTERPOLATION

Before we begin the interpolation process, we must ensure that the two sampled meshes have the same topology. This was true before we adjusted these meshes on the sharp edges. However, in the previous section we saw that diagonals could be added to the meshes. Thus, in order to maintain the same topology, we must add some additional diagonals. This is done by splitting the rectangular faces where diagonal edges have been added in two or four triangular faces:

(a) if only one diagonal appears in one of the two sampling meshes, then it cuts the rectangular face in two,
(b) if two diagonal edges have been added (one in each sampling mesh), we have to insert a new vertex and split the face into four triangular faces.

If desired, the remaining rectangular faces may be triangulated to ensure face planarity during the interpolation.

Axes interpolation

We treat the discretized axis as a polygonal line rather than a differentiable entity. We discretize the two curves with the same number of points since they correspond to sampled values in the parameter space. We need to interpolate between the two polygonal lines. Henceforth, the terms axis, curve, and polygonal line will be used interchangeably.

As observed in Reference 12, a linear interpolation of the two polygonal lines is inadequate. For instance, the interpolation of two parallel segments oriented in opposite directions collapses for some interpolation value; such an interpolation process is not acceptable.

We associate with each axis a moving frame (see Figure 7). A moving frame associated with a curve γ is a pair $(\gamma, \mathcal{R})$, where $\mathcal{R}$ is an orthonormal parametrized frame (e_1, e_2, e_3)

such that $e_1(s)$ has the same direction as the tangent $\gamma'(s)$ at the point $\gamma(s)$. We interpolate the relative coordinates and the relative rotation between two consecutive points and frames of the polygonal line. Let us denote $(\gamma_1, ..., \gamma_n)$ (resp. $(\mathcal{R}_1, ..., \mathcal{R}_n)$) the digitized points (resp. frames) of the curve γ, then we have:

$$\gamma_{i+1} = \gamma_i + \mathcal{R}_i X_{i+1} \tag{1}$$

$$\mathcal{R}_{i+1} = r_{i+1}\mathcal{R}_i \tag{2}$$

where X_{i+1} are the local co-ordinates of γ_{i+1} in $(\gamma_i, \mathcal{R}_i)$, and r_{i+1} is the rotation between $\mathcal{R}_i$ and $\mathcal{R}_{i+1}$. We construct a polygonal line representing the interpolated axis by interpolating the X_i and the r_i of the two axes. First we linearly interpolate between the two axis endpoints to calculate a starting point for the interpolated axis. Using the starting point, we recursively compute all the axis points using equations (1) and (2) where X_{i+1} and r_{i+1} have been replaced by the interpolated values.

We use linear interpolation for the coordinates X_i and transform the rotations r_i into quaternions to use spherical linear interpolation (slerping) as proposed in Reference 22. We follow the notations of Reference 22 in equation (4). Hence we have:

$$X_i(t) = (1-t)X_{i,1} + tX_{i,2} \tag{3}$$

$$q_i(t) = Slerp(q_{i,1}, q_{i,2}, t) = q_{i,1}(q_{i,1}^{-1}q_{i,2})^t \tag{4}$$

where subscripts 1 and 2 account for the initial and the final curves.

A formal study of this interpolation method can be found in Reference 18. It is shown there that if the process is applied to digitized curves, with a discretization step approaching zero, the result will be a continuous interpolated curve with controlled curvature and torsion. The orientation of the moving frame $\mathcal{R}$ is also discussed in detail in this reference.

As one can see by examining Figure 11, this curve interpolation scheme leads to good visual results. Moreover the computation is simple and fast. The whole interpolation process is fully interactive on a Silicon Graphics INDIGO 2 workstation.

Vertex interpolation

A linear interpolation of the local coordinates is performed between two corresponding vertices of each sampled object. Hence, in the local frame, the tangent to the trajectory of a vertex is radial to the axis at the beginning and the end of the animation. In order to preserve this property for the adjusted vertices, we compute the cubic spline defined by the vertices and their associated radial directions.

The vertex path is obtained by composing this movement and the movement of the associated 3D affine frame. This results in a complex movement which could not have been simply designed.

As one can see here, interpolation and correspondence (a consequence of the sampling) are strongly related, in both cases, the main role is played by the sampling half-lines attached to the axis.

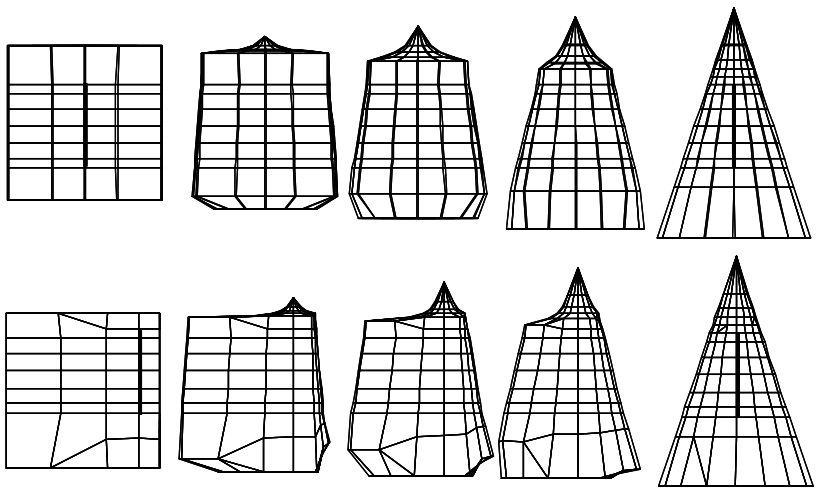

Figure 8. The same axis is used to sample the two original objects

ANIMATION EXAMPLES

In this section we demonstrate the algorithm by applying it to several examples.

1. Figures 8, 9 and 10 illustrate the role played by the axes in controlling the shape transformation. The original objects are a cube and a cone. The axes are discretized using seven frames and, for each planar section, sixteen angular parameters are used to

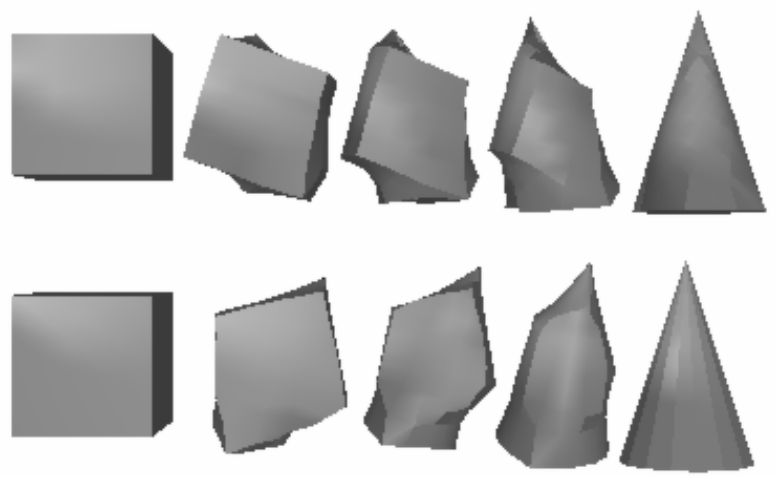

Figure 9. The influence of the axis interpolation in the shape transformations

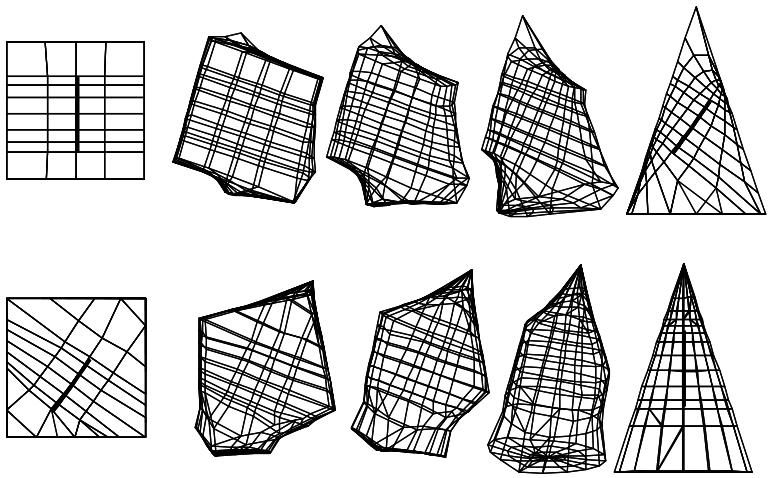

Figure 10. The same shape transformations in wire-frame

sample the objects. The axes are drawn in bold for the initial and final objects.

In Figure 8, the same axis, a vertical line segment, is used to sample the initial and the final objects. In the second shape transformation the axis has been translated inside the cube. This is similar to what Kent, Parent and Carlson[14] showed by moving the centre of projection inside one of the star-shaped objects.

When we use two different axes to sample the initial and the final objects, the axis interpolation strongly influences the shape transformation. In both Figures 9 and 10 two shape transformations are shown; on the top, the axis rotation is clockwise from the left to the right whereas on the bottom the rotation is counterclockwise. These rotations lead to different movements of the salient features with different intermediate shapes.

2. Figure 11 shows two different views of the same shape transformation process. Here, the two original objects are generalized cylinders around two different curves. This example illustrates the visual effect produced by our axis interpolation method.
3. In Figure 12, the fish has been deformed into a sickle. The axis of the fish is a straight line whereas the axis associated with the sickle is bent on the sharp part of the instrument. One can see the variety of intermediate shapes and the appearance/vanishing of the salient features in the in-between shapes.
4. Plates 1 and 2 show several metamorphoses of a woman's head. The images are taken from the film Metadata.[23] The polyhedral mesh associated with each object is composed of 7200 facets. The axes are approximated by 50 points, each cylindrical section is sampled into 80 angular sectors and each spherical sheet is composed of 1600 facets. The two heads are sampled with respect to the same axis which is nearly straight, the axis of the piano is nearly a vertical line segment whereas the axis of the swan follows the global bending of its shape.

The four transformations illustrate visual effects characteristic of our method.

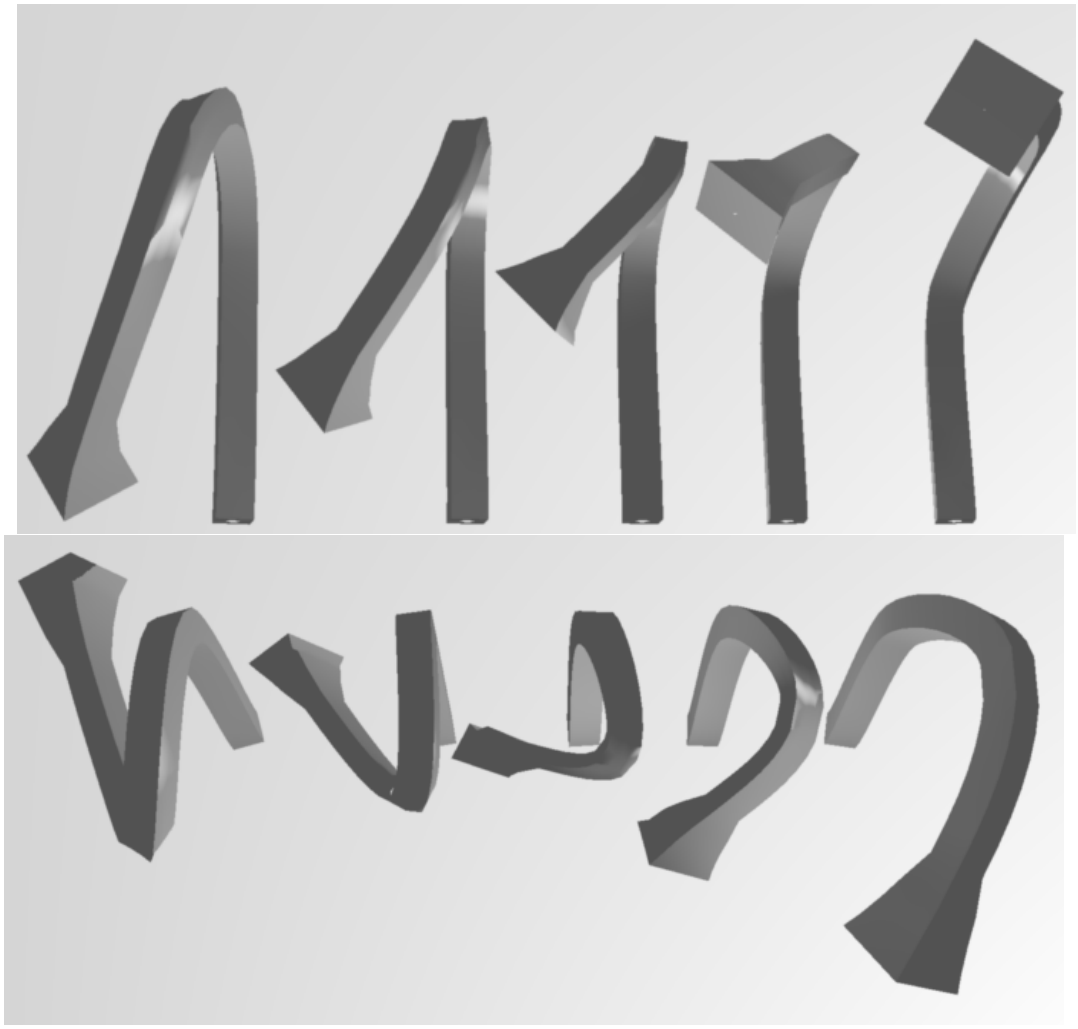

Figure 11. Axis interpolation: front-view (bottom) and top view (top)

(a) The role of the salient features in the intermediary shapes is shown when the face is transformed into a piano.
(b) The role of the axes interpolation is shown in the metamorphosis of the face into a swan.
(c) The appearence of recognizable features in the interpolated faces can be observed in the metamorphosis of the two heads.
(d) The speed at which the transformation occur can be varied along the axis. This is shown in the example where the woman's head is transformed into a cube.

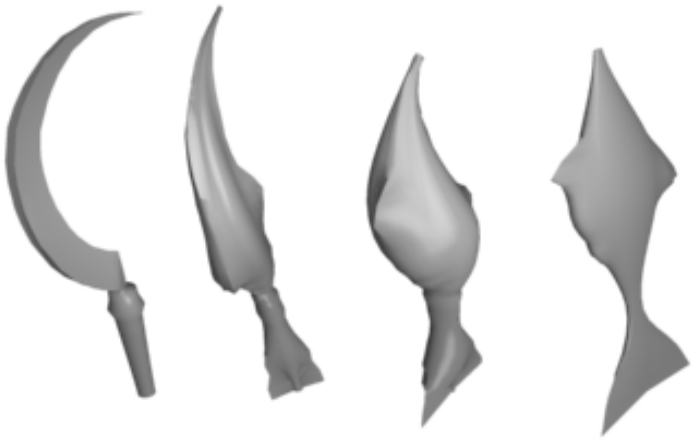

Figure 12. Shape transformation between a sickle and a fish

CONCLUSION

We have proposed a method for computing a continuous deformation between two polyhedral shapes. The proposed method has the following features.

1. The method uses an intuitive technique to jointly solve the correspondence and the interpolation problem.
2. It features interactive control of the shape transformation. The use of axes to specify the transformation is intuitive and can be viewed as the specification via a 'skeleton' of the global deformation of the shapes. It is a natural extension of the first approach of Kent, Parent and Carlson.[14]
3. Good visual effects are obtained during the transformation, as shown in the examples of the previous section.
4. The method is efficient. The use of an intermediate polyhedral mesh reduces the computational time. The user may obtain approximate sketches of the deformation process using a rough discretization during the sampling.

This algorithm has been integrated into ACTION3D, a general interactive modelling system developed jointly by SOGITEC and INRIA. The whole process (sampling and interpolation) is performed at interactive rates on a Silicon Graphics INDIGO 2 workstation.

The approach can be extended to compute transformations for a sequence of shapes. In this case, all the objects have to be sampled with the same discretization for the axes and the angular parameter. The axis interpolation and the local interpolation have to be extended into spline interpolation.

The method can be applied to any objects that are star-shaped around an axis. In particular, our method can be simply extended to implicit surfaces.[24, 25] For it is sufficient to be able to compute the intersection of a sampling half-line with the surface of the object, which is an easy task in this case.

Work in progress includes attempts to automatize the construction of a 3D curve which can be the axis associated with an object.

ACKNOWLEDGEMENTS

We wish to thank Pierre Jancène for his help during the implementation and Arghyro Paouri for the realization of the film Metadata. Special thanks to Bill Horn for his valuable comments on this paper.

REFERENCES

1. A.H. Barr, 'Global and local deformations of solid primitives', *Computer graphics (SIGGRAPH '84 Proceedings)*, **18**, 21–30 (1984).
2. D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, 'Elastically deformable models', *Computer Graphics (SIGGRAPH '87)*, **21**, 205–214 (1987).
3. A. Pentland, I. Essa, M. Friedmann, B. Horowitz and S. Sclaroff, 'The thingworld modelling system: virtual sculpting by modal forces', *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, **24**(2), 143–144 (1990).
4. S. Coquillart and P. Jancène, 'Animated free-form deformation: an interactive animation technique', *Computer Graphics (SIGGRAPH '91)*, **25**, 23–26 (1991).
5. D. Bechmann and N. Dubreuil, 'Animation through space and time based on a space deformation model', *The Journal of Visualization and Computer Animation*, **4**(3), 165–184 (1993).
6. T. Beier and S. Neely, 'Feature-based image metamorphosis', *Computer Graphics (SIGGRAPH '92)*, **26**(2), 35–42 (1992).
7. J.F. Hughes, 'Scheduled Fourier volume morphing', *Computer Graphics (SIGGRAPH '92)*, **26**(2), 43–46 (1992).
8. S.E. Chen and R.E. Parent, 'Shape averaging and its applications to industrial design', *IEEE Computer Graphics and Applications*, **9**(11), 47–54 (1989).
9. A. Kaul and J. Rossignac, 'Solid-interpolating deformations: construction and animation of PIPS', *Eurographics '91*, Vienna, September 1991, pp. 493–505.
10. J. Rossignac and A. Kaul, 'AGRELS and BIBs: metamorphosis as a Bézier curve in the space of polyhedra', *Eurographics '94*, Oslo, September 1994, pp. C-179–C-184.
11. T.W. Sederberg and E. Greenwood, 'A physically based approach to 2-D shape blending', *Computer Graphics (SIGGRAPH'92)*, **26**(2), 25–34 (1992).
12. T.W. Sederberg, P. Gao, G. Wang and H. Mu, '2D shape blending: an intrinsic solution to the vertex path problem', *Computer Graphics (SIGGRAPH '93)*, **27**(2), 15–18 (1993).
13. E.W. Bethel and S.P. Uselton, 'Shape distortion in computer-assisted keyframe animation', in *Computer Animation'89, State-of-the-art in Computer Animation*, Computer Graphics International, Springer-Verlag, 1989, pp. 215–224.
14. J.R. Kent, R.E. Parent and W.E. Carlson, 'Establishing correspondences by topological merging: a new approach to 3-D shape transformation', *Graphic Interface'91*, Calgary, June 1991, pp. 271–278.
15. J.R. Kent, R.E. Parent and W.E. Carlson, 'Shape transformation for polyhedral objects', *Computer Graphics (SIGGRAPH'92)*, **26**(2), 47–54 (1992).
16. R.E. Parent, 'Shape transformation by boundary representation interpolation: a recursive approach to establishing face correspondences', *The Journal of Visualization and Computer Animation*, **3**, 219–239 (1992).
17. F. Lazarus, S. Coquillart and P. Jancène, 'Axial deformations: an intuitive deformation technique', *Computer Aided Design*, **26**(8), 607–613 (1994).
18. F. Lazarus, 'Smooth interpolation between two polylines in space', *Computer Aided Design*, to appear.
19. R.L. Bishop, 'There is more than a way to frame a curve', *American Mathematical Monthly*, **82**, 246–251 (1975).
20. F. Klok, 'Two moving coordinate frames for sweeping along a 3D trajectory', *Computer Aided Geometric Design*, **3**(3), 217–229 (1986).
21. T. Cormen, C. Leiserson and R. Rivest, *Introduction to Algorithms*, MIT Press, 1990.
22. K. Shoemake, 'Quaternion calculus and fast animation, computer animation: 3-D motion specification and control', *SIGGRAPH'92*, 1987 Tutorial, 1987, pp. 101–121.

23. A. Paouri, F. Lazarus, A. Verroust and C. Blonz. *Metadata*, videotape, INRIA UCIS Audiovisuel, France, 1994.
24. B. Wyvill, J. Bloomenthal, T. Beier, J. Blinn, A. Rockwood and G. Wyvill, 'Modelling and animating with implicit surfaces', *Siggraph Course Notes*, Vol. 23, 1990.
25. B. Wyvill, 'Metamorphosis of implicit surfaces', *Siggraph Course Notes*, Vol. 25, 1993.